



SCICAST ANNUAL REPORT (2015)

25-May-2015, Year 4

• 1 of 143

This research was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior (DoI) contract number D11PC20062. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI, or the U.S. Government.

This report has been approved for unlimited public release.

SciCast Annual Report (2015)

25-May-2015, Year 4

Performer

George Mason University

Project Title

Decomposition-Based Elicitation & Aggregation

Award Instrument Number

IARPA-BAA-10-05

Period of Performance

Y4: 26 May 2014 – 25 May 2015

Total: 26 May 2011 – 25 May 2015

Work Performed Under U.S. Government Contract Number

DIIPC20062

Table of Contents

Table of Contents	3
Executive Summary	8
Y4 Research and Development	8
Registration and Activity	11
Participant Profile	12
Forecasting Method	14
Accuracy	15
Outreach	17
Publications and Presentations	18
Y4	18
Publications Y4	18
Conference Proceedings Y4	18
Presentations Y4	19
Previous Years Y1 – Y3	20
Publications Y1 – Y3	20
Conference Proceedings Y1 – Y3	21
Presentations Y1 – Y3	21

This research was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior (DoI) contract number D11PC20062. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI, or the U.S. Government.

This report has been approved for unlimited public release.

Experiments and Studies	23
Overview	23
Ordered Questions.....	24
Introduction	24
Lone Value Trees.....	24
Value Tree As Bayes Net – Factorized Representation.....	25
Performance	27
Target Journals and Conference.....	27
Correlates of Good Forecasting.....	28
Background	28
Objectives	29
Surveys & Survey Scoring.....	29
Forecast Quality.....	34
Measures of Accuracy	35
Surveys as Predictors of Good Forecasting	39
Super-Users.....	42
Data Mining Analysis.....	45
Incentives for Crowdsourcing Forecasts	48
Background	48
First Two Experiments: Activity & Accuracy Incentives	48
Third Experiment: Four-month Accuracy	49
Target Journals and Conference.....	54
Incentives for Combinatorial Edits.....	55
Asset Management.....	57



Background	57
Objectives	57
Asset Models.....	58
Experiment Design.....	61
Analysis and Results.....	63
Discussion	68
Target Journals and Conference.....	68
Recommender Testing.....	69
Background	69
Objectives	69
Experiment Design.....	69
Results and Analysis.....	71
Forecaster Survey	73
Fraud Detection & Analysis.....	75
Background	75
Market Manipulation.....	75
Results.....	83
Lessons Learned	84
Question Management	86
Streamlining Question Generation.....	86
Question Invalidation.....	86
General Guidelines for Writing Questions.....	87
New Question Type: Scaled Continuous.....	87
Formatting Guide and Fine Points	88

Lessons Learned	88
FUSE Questions	90
Overview	90
Activity	90
BAE ARBITER System.....	92
Summary Statistics.....	93
Lessons Learned	93
SRI Copernicus System	94
Summary Statistics.....	94
Lessons Learned	94
User Experience (UX) Design.....	96
Conditional Edits	96
Conditional Edit Re-Ordering	96
Choice of Conditional Edits.....	97
User Added Links	98
Network Visualization	99
New Look and Feel for Transition Sites.....	101
Executive Dashboard.....	101
Software Tools and Data Resources	106
Predict.....	107
3 rd party modules used.....	111
System modules.....	111
Javascript modules.....	111
Python modules	111



DataMart	112
ETL (Extract, Transport, Load).....	112
Spark	113
SciCast/iOS.....	113
Recommender.....	116
Technical Approach.....	116
Tuuyi Inference Engine	118
Objective.....	118
Technical Approach.....	119
UnBBayes Inference Engine.....	119
Features developed prior to Year 4	121
Features developed in Year 4	126
UnBBayes project architecture	130
Recruiting, Outreach, and User Engagement	133
Advertising	133
Social Media.....	135
Blog.scicast.org	135
Facebook: https://facebook.com/scicast	135
Twitter.....	137
Other Outreach Activities	139
SciCast in the Media	139
References	140

Executive Summary

Expert forecasts reliably underperform simple statistical models like “no change” or linear models with equal weights or even randomly-chosen weights (Dawes, Faust, & Meehl, 1989; Grove, Zald, Lebow, Snitz, & Nelson, 2000; Marchese, 1992; Meehl, 1954; Silver, 2012; Tetlock, 2005). Yet human judgment is essential for tasks like intelligence analysis, and it is increasingly clear that when we want accurate forecasts, we do better by aggregating the judgments of many experts (Mellers et al., 2014), either because individuals reasoning on their own are not very efficient (Mercier & Sperber, 2010), because errors cancel in the “wisdom of crowds” (Surowiecki, 2005), or because expertise is often not where we expect it but can be revealed with proper incentives (Hanson, 2002). Often experts can be used to build computer models that later outperform the same experts (Shwe et al., 1991).

We report on the fourth and final year of a large project at George Mason University developing and testing combinatorial prediction markets for aggregating expertise. For the first two years, we developed and ran the DAGGRE project on geopolitical forecasting. On May 26, 2013, renamed ourselves SciCast, engaged Inkling Markets to redesign our website front-end and handle both outreach and question management, re-engineered the system architecture and refactored key methods to scale up by 10x – 100x, engaged Tuuyi to develop a recommender service to guide people through the large number of questions, and pursued several engineering and algorithm improvements including smaller and faster asset data structures, backup approximate inference, and an arc-pricing model and dynamic junction-tree recompilation that allowed users to create their own arcs. Inkling built a crowdsourced question writing platform called Spark. The SciCast public site (scicast.org) launched on November 30, 2013, and began substantial recruiting in early January, 2014.

As of May 22, 2015, SciCast has published 1,275 valid questions and created 494 links among 655 questions. Of these, 624 questions are open now, of which 344 are linked (see Figure 1). SciCast has an average Brier score of 0.267 overall (0.240 on binary questions), beating the uniform distribution 85% of the time, by about 48%. It is also 18-23% more accurate than the available baseline: an unweighted average of its own “Safe Mode” estimates, even though those estimates are informed by the market. It beats that ULinOP about 7/10 times.

Y4 Research and Development

One of the main contributions in Year 4 (Y4, or contract Y4) was a set of incentives studies. The first pair of studies was planned at the end of Y3 and ran at the beginning of Y4. In the first randomized controlled 4-week trial, it showed that activity incentives strongly affect activity, without hurting accuracy. The second 4-week trial was modified to also test for accuracy incentives, but was too small and too weak to detect an effect. Therefore we designed a four-month matched-question randomized controlled trial using over 300 questions and over 100 actual final resolutions. This study showed that within the experiment, questions in their award-eligible state (e.g. Dec and Feb) were three times as active as those in their award-ineligible state (e.g. Jan and Mar), with fifteen times as much information per edit, for an average gain from incentives of forty times the information. Furthermore, the customary mean-of-mean daily Brier scores (MMDB) for experiment questions

was better than for non-experiment questions and better than averages from before the experiment. The out-of-experiment questions were all longer-term questions, and did not get much traffic. But when they did, it tended to be very informative.) The final study examined the effect of incentivizing conditional forecasts while paying for (expected) accuracy: it created more conditional edits, more links, and more mutual information in the joint distribution.

We also developed a novel approach to approximate inference that is well-suited to market demands, and analyzed but did not fully test the tree-based data-structure we developed last year for ordered questions. As noted, we deployed user-added arcs, and saw as many as 60 new arcs in a single day. Prior to that, we reorganized the combo interface so that assumptions are always salient. We have added the ability to remove links, to measure link strength via mutual information, and to find the exact network complexity a new arc will require (or will release if removed). Many of these features are enabled by this year's dynamic junction-tree compilation algorithm (Figure 3), which dramatically speeds queries requiring the addition or reduction of a single arc.

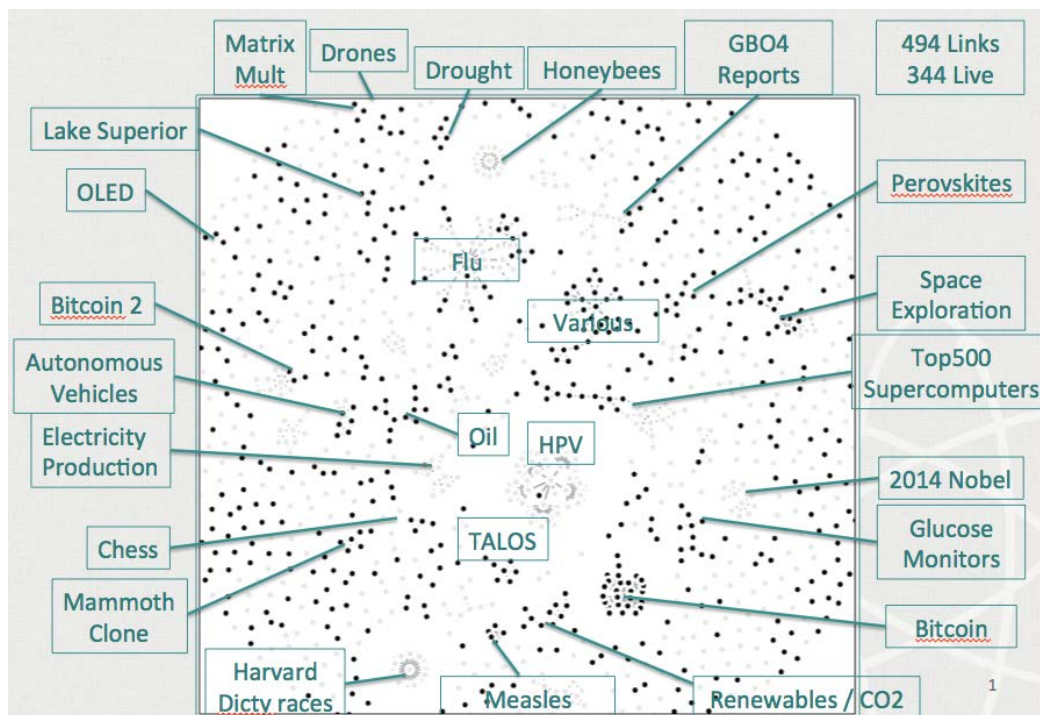


Figure 1: SciCast link structure as of May 2015, with select clusters labeled.

We also developed an executive dashboard (Figure 2) that allows market administrators and clients to see market activity, accuracy, and calibration on one screen, and filter or “drill down” interactively.

SciCast Annual Report (2015)

...

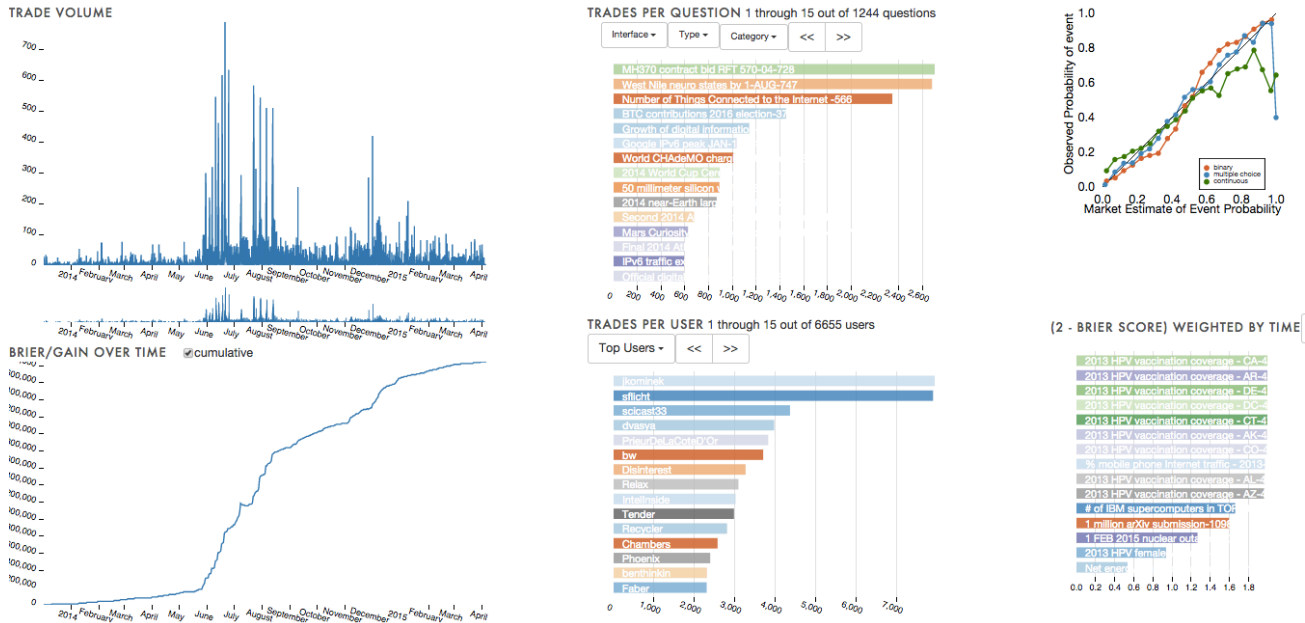


Figure 2: Executive dashboard after loading the full question history. Panels clockwise from top left: Trade volume over time prominently showing the participation incentives, most active questions, calibration curves, most accurate questions (reverse Brier score), most active forecasters, and Brier gain over time. Panels can be filtered by time window, category, question, or forecaster.

Compilation time

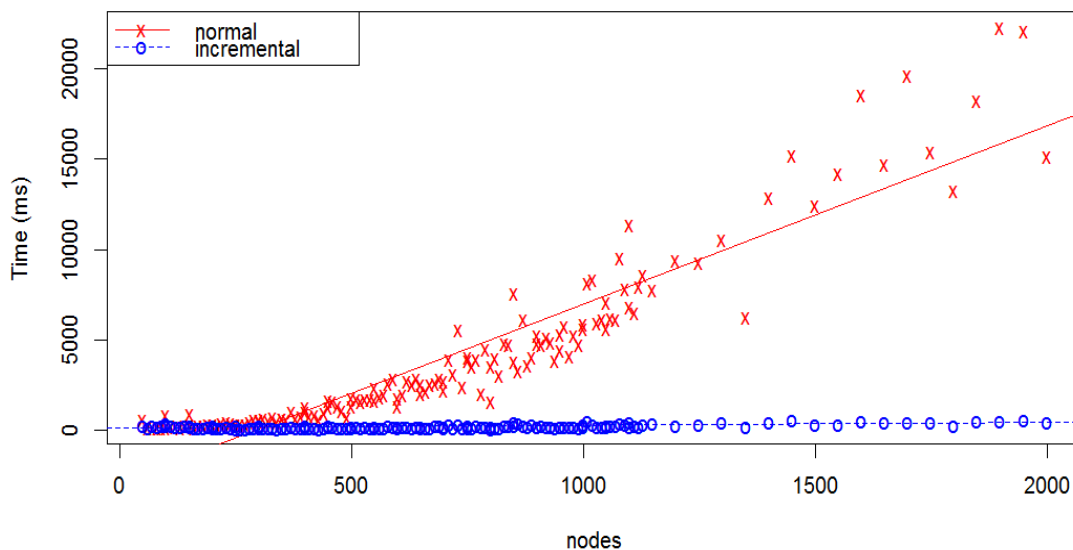


Figure 3: Junction tree recompilation time for normal vs. incremental algorithm

Registration and Activity

SciCast has seen over 11,000 registrations, and over 129,000 forecasts. Google Analytics reports over 76K unique IP addresses (suggesting 8 per registered user), and 1.3M pageviews. The average session duration was 5 minutes.

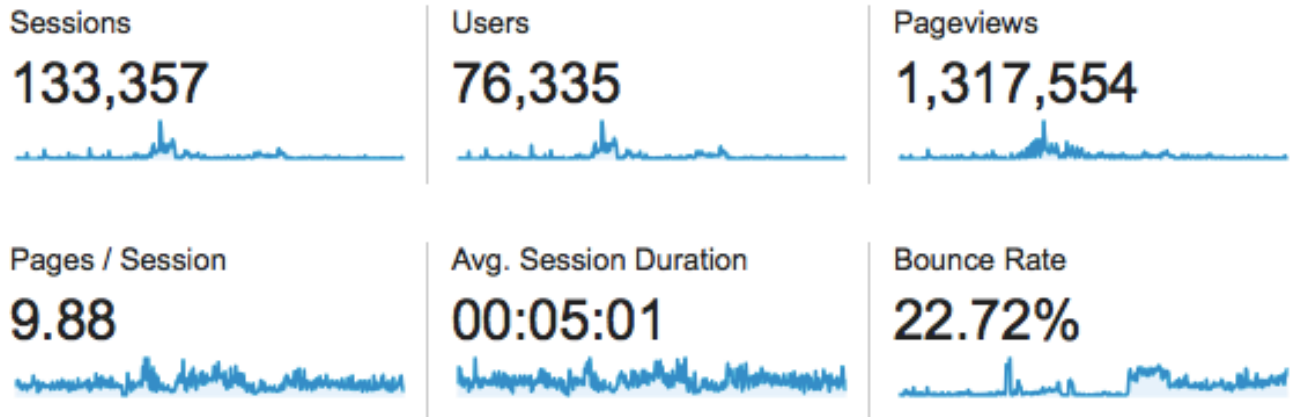


Figure 4: SciCast activity according to Google Analytics

The following figures show the activity over time from several angles, starting with total forecasts (a.k.a. trades or edits) per day, then forecasts per user per day. Activity was strongly influenced by incentive programs, and indeed it is possible to pick out some of the incentive programs just by looking at the activity graphs. The following figure shows Trades/Day, which is probably our key metric. We wanted it to be above 150 trades/day, and often were, but could not maintain it as a baseline.

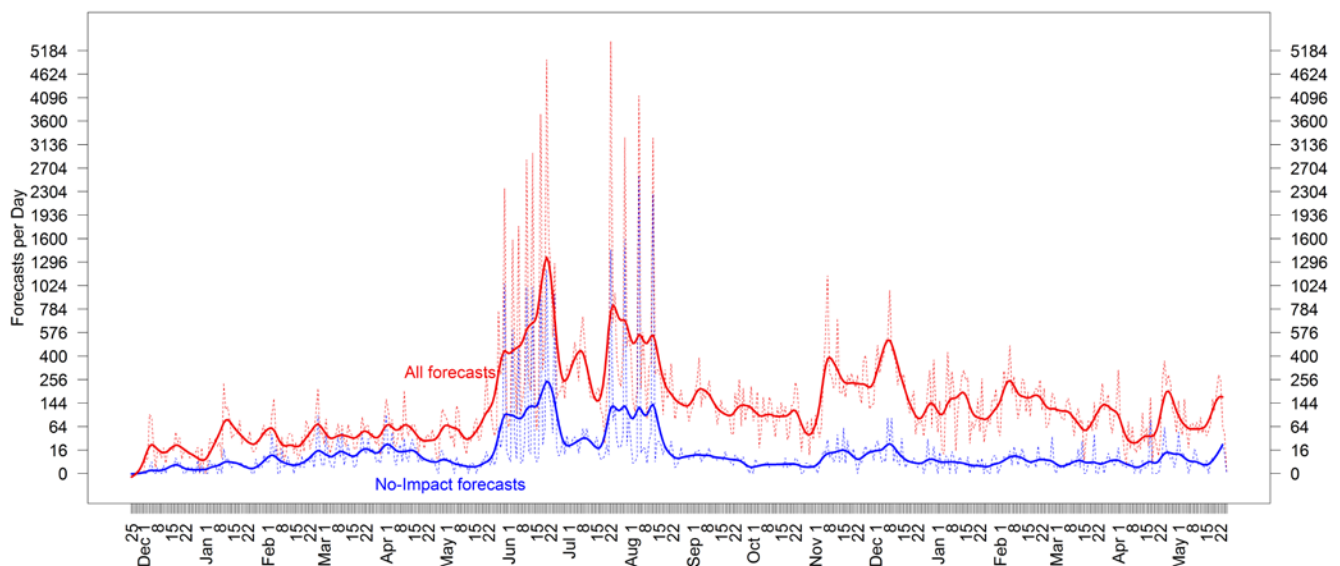


Figure 5: Trades / Day showing paired activity/accuracy study around July 2014, and four-month accuracy study November 2014 to March 2015. “No-impact” forecasts are those which simply “agree” with the current estimate (Safe Mode).

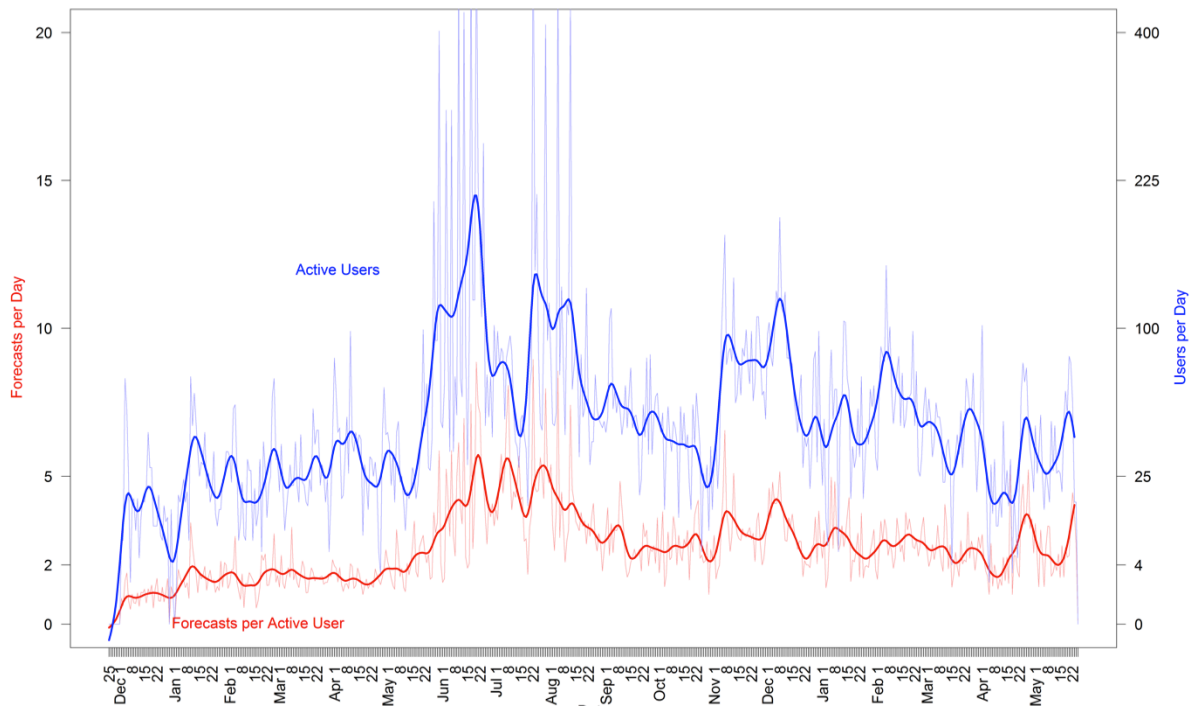


Figure 6: Trades/Person/Day showing both #active users (right axis) and #forecasts (left axis).

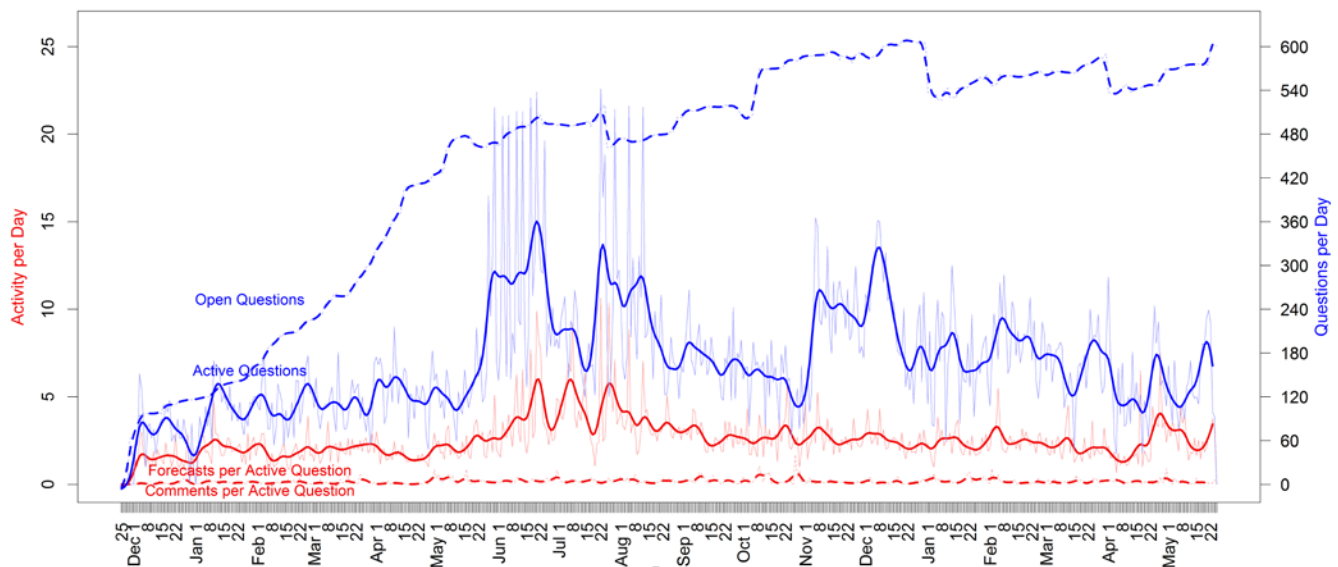


Figure 7: Forecasts/Question/Day also showing #open questions, #active questions, and #comments/question.

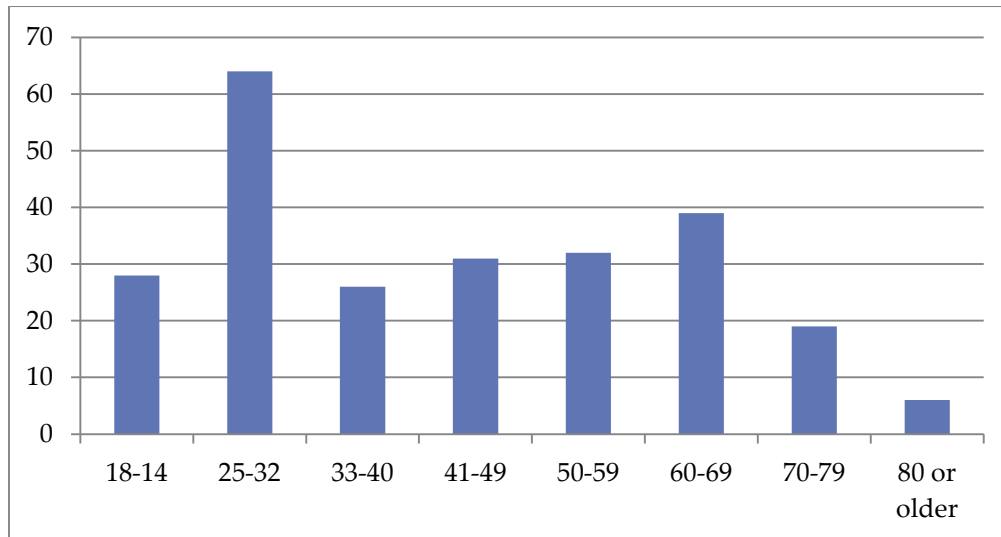
Participant Profile

Similar to previous years, our population is overwhelmingly male (80%), highly educated (1/4 of respondents have a Ph.D. or equivalent), and American (80% US alone).

Table 1: SciCast Participant Profile by Survey

Variable	Values	N
Gender	80% male	244
Age	25-32: 26% 60-69: 16%	245
Education	Assoc: 49% Ph.D.: 26%	245
Location	US: 80% N.Am: 84% EUR: 10% Asia: 4%	133K Google Analytics

The following figure shows the age demographics. With an extra ~70 surveys completed, the 25-32 category has eclipsed 60-79 as the mode, but we still show a distinctly bimodal distribution. Combined with the other demographics, we summarize our population as high-educated American males with college degrees and spare time.


Figure 8: Age demographics, based on survey responses

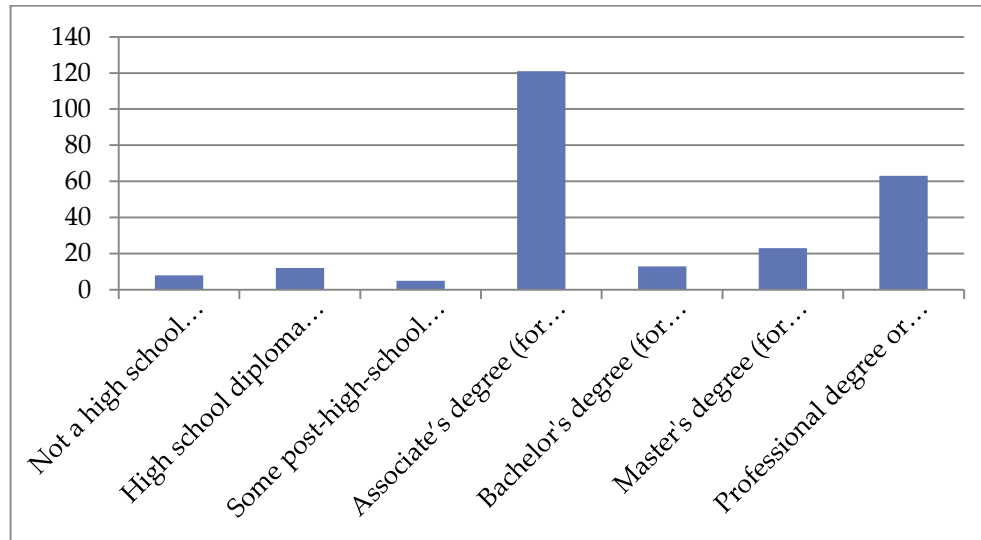


Figure 9: Education demographics, based on survey responses.

Forecasting Method

SciCast is a combinatorial prediction market using the Logarithmic Market Scoring Rule (LMSR) and an “edit-based” interface where participants see and directly set the probability, without the abstraction of buying and selling shares. Combinatorial means that questions can be linked to each other, so that in theory, and forecasters can bet on (edit) any marginal, conditional, or joint probability in the entire joint probability space. We impose three limits, one necessary and two contingent.

First, it is necessarily impossible to represent the full joint state among a large number of variables. 20 binary questions already have over 1M combinations. SciCast has maintained about 600 questions live at any given time, and is designed to handle 1,000 or so. As of May 22, 2015, SciCast has published 1,275 valid questions and created 494 links among 655 questions. Of these, 624 questions are open now, of which 344 are linked (see Figure 1).

Fortunately for computation tractability, most combinations are meaningless: either they do not even correlate, or there is no reason to think the correlation is predictive. Most any joint probability distribution will factor – this is the key insight in causal modeling and the foundation of Pearl’s approach to causation and causal inference. (Galles & Pearl, 1995; Neapolitan, 1990; J. Pearl, 2000; Judea Pearl, 1988; Woodward, 2005) The advent of Bayesian networks in the late 1980s revived probabilistic methods in Artificial Intelligence, dismissed as intractable in the 1960s and famously declared irrelevant by AAAI in 1984. Probabilistic methods have since come to dominate AI, from basic vision to causal inference.

A major advance in Y4 was the introduction of user-added links. This required conceptual, UX, algorithm, and engineering work to allow the underlying system to look ahead to see the compiled structure (junction tree) that would result from the new link, and set a price (minimum edit size) as a function of the computational

complexity. That in turn required faster compilations, for which we adapted dynamic junction tree algorithms (see Figure 3).

The current SciCast UX supports only conditional probabilities, not joint probabilities. Furthermore, right now it restricts those to pairwise conditions, simply as a matter of UX simplification to try to open up combo edits to a wider crowd. Conditional typically comprised <5% of total edits, but the combo edits contest successfully created a huge spike in conditional edits at the end of Y4, which spike was reflected in number of arcs and mutual information (see Figure 10).

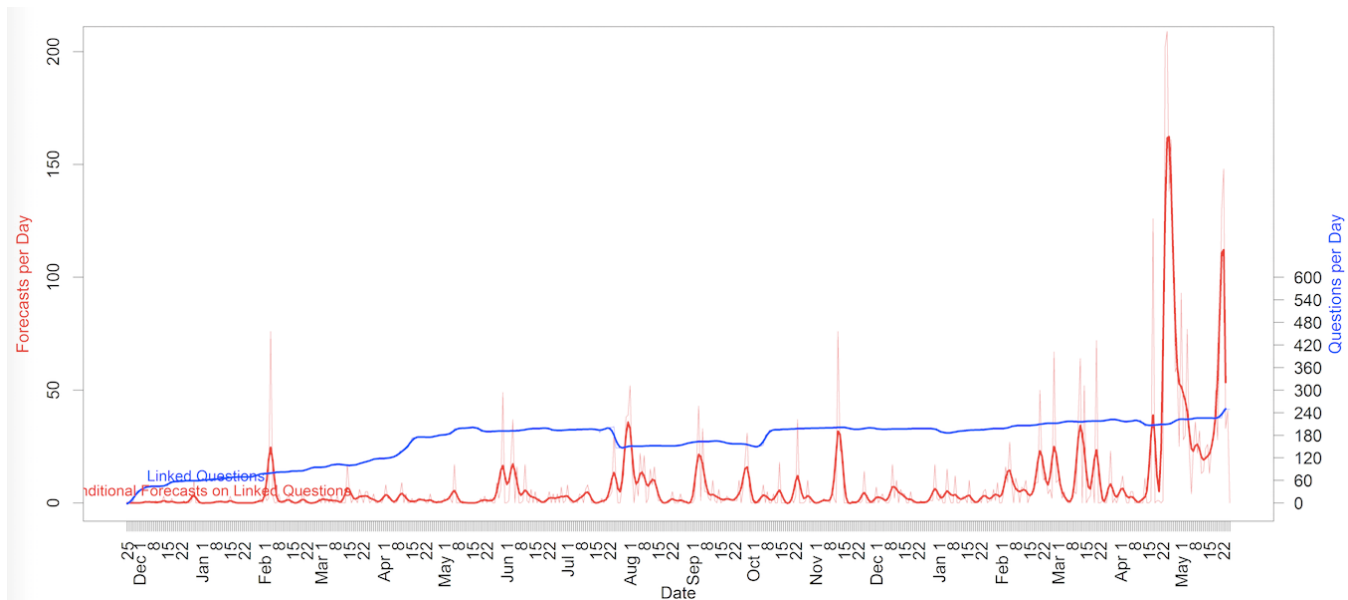


Figure 10: Conditional edits per day, and #questions with at least one link.

Accuracy

SciCast has an average Brier score of 0.267 overall (0.240 on binary questions), beating the uniform distribution 85% of the time, by about 48%. It is also 18-23% more accurate than the available baseline: an unweighted average of its own “Safe Mode” estimates, even though those estimates are informed by the market. It beats that ULinOP about 7/10 times. Figure 11 shows the overall distribution of mean daily Brier scores (by question) for SciCast and for the uniform distribution. Figure 12 separates the histogram for questions with ≤ 4 choices, and those with >4 choices (as many as 35). As expected, SciCast fares better with fewer choices, because it is much easier to put 90+% probability on the correct answer when there is only *one* alternative, as opposed to 34 with 2 near neighbors.

SciCast Annual Report (2015)

• • •

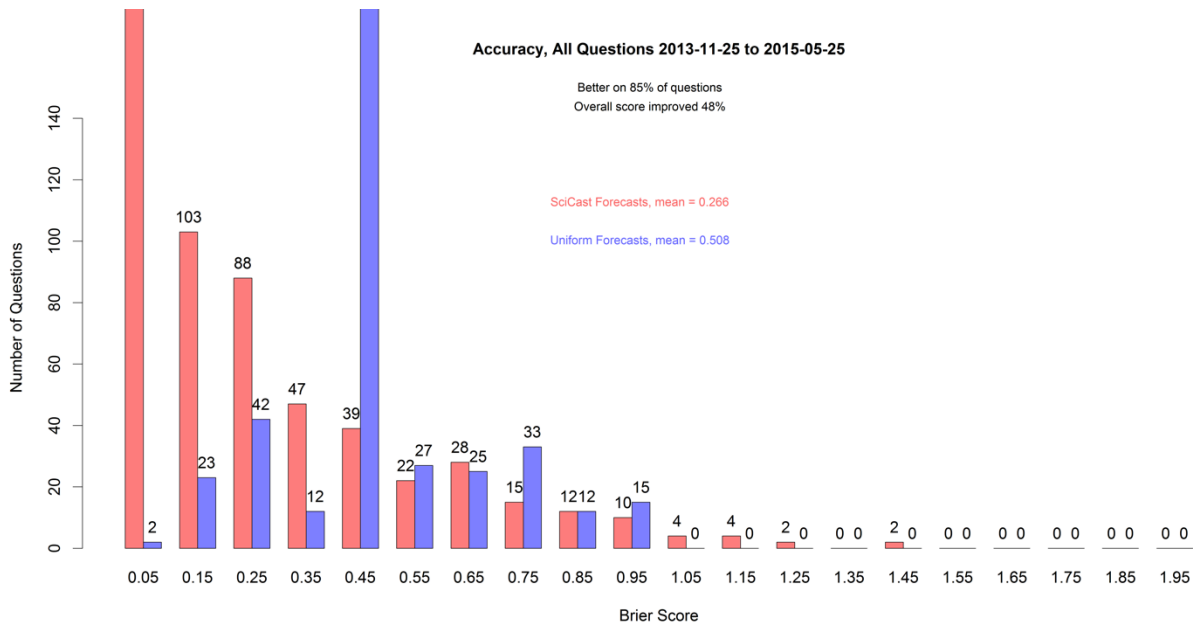


Figure 11: SciCast accuracy histogram vs. Uniform -- all Questions

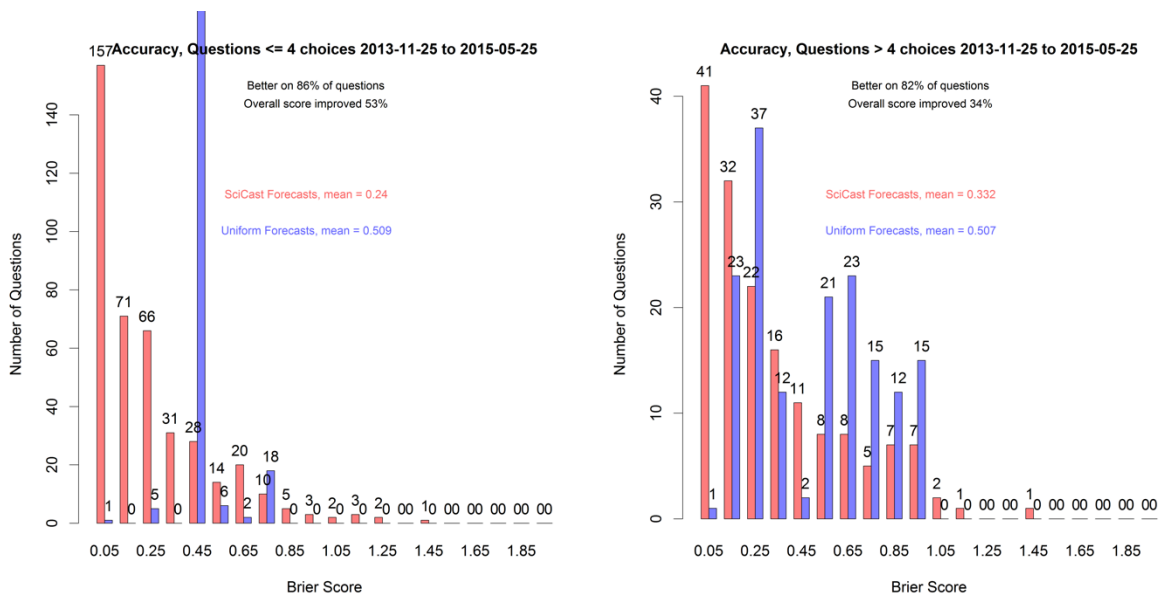


Figure 12: Separate accuracy histograms for questions with <=4 choices, and >4 choices.

Outreach

In addition to ad campaigns with professional banner ads like Figure 13, SciCast had over 25 media mentions including an ACS webinar and a Reddit “AMA” session with Drs. Twardy and Hanson. We continued to add affiliates so our list includes: AAAS, IEEE, ISACA, AMIA, ACS, the Cooper Extension Services, ICE, and more recently the Policy Design Lab and Shaping Tomorrow. We had particularly strong participation this year from IEEE and ISACA, who contributed over 30 questions between them, and reached out to their membership. IEEE provided excellent coverage online and in IEEE Spectrum.



Figure 13: SciCast banner ad (one of several)

Publications and Presentations

Organized alphabetically within each section. Some conference presentations also required full papers for review; we have attempted to list these under “Conference Proceedings” as well as “Presentations”.

Y4

See also SciCast Transition Briefings, in SciCast Technology Transition, p.**Error! Bookmark not defined.**, below.

Publications Y4

Laskey, Kathryn B., Robin Hanson, Wei Sun, Charles Twardy, and Shou Matsumoto. “Managing Assets and Probabilities in Combinatorial Prediction Markets, ” in preparation for *IEEE Transactions on Human-Machine Systems*.

Twardy, Charles and Kathryn Laskey, Robin Hanson, Walter Powell, Kenneth Olson. “Combinatorial Prediction Markets for Aggregating Expert Forecasts.” submitted to the *Int’l. J. Forecasting*, Special issue on elicitation and aggregation, 2015.

Hanson, Robin. “What Will It Be Like To Be an Emulation?” In *Intelligence Unbound: The Future of Uploaded and Machine Minds*, 298–309. Wiley, 2014.

Conference Proceedings Y4

Laskey, Kathryn. “Combinatorial Prediction Markets for Fusing Information From Distributed Experts and Models.” In *Proceedings of the 17th International Conference on Information Fusion*. Washington, D.C.: International Society on Information Fusion and/or IEEE, 2015. <http://fusion2015.org/>.

Olson, Kenneth, Charles Twardy, Kathryn Laskey, and M. Burgman. “Interval Elicitation of Forecasts in a Prediction Market Reveals Lack of Anchoring ‘Bias.’” Collective Intelligence Conference, Cambridge, MA, June 2014.

Sun, Wei, Kathryn B. Laskey, Charles R. Twardy, Robin D. Hanson, and Brandon R. Goldfedder. “Trade-Based Asset Model Using Dynamic Junction Tree for Combinatorial Prediction Markets.” Collective Intelligence 2014, Cambridge, MA, 2014.

<http://humancomputation.com/ci2014/papers/Active%20Papers%5CPaper%20126.pdf>.

Sun, Wei, Shou Matsumoto, Robin Hanson, Kathryn Laskey, and Charles Twardy. “Trade-Based Asset Models for Combinatorial Prediction Markets.” In *Proceedings of the Eleventh UAI Bayesian Modeling Applications Workshop (BMAW 2014)*, edited by Kathryn Laskey, J. Jones, and R. Almond, pp. 99–100, 2014. http://ceur-ws.org/Vol-1218/bmaw2014_abstract_1.pdf.

Twardy, Charles, Robin Hanson, Kathryn Laskey, Tod Levitt, Brandon Goldfedder, Adam Siegel, Bruce D’Ambrosio, and Daniel Maxwell. “SciCast: Collective Forecasting of Innovation.” Collective Intelligence 2014, Cambridge MA., 2014. http://blog.scicast.org/wp-content/uploads/2014/02/Twardy_etal_SciCast_Overview_CI2014.docx.

Presentations Y4

- Hanson, Robin. "Factoring Geopolitical Risk into Decision-Making." presented at the Global ICON Conference, Boston, MA, July 2014.
- Hanson, Robin. "Giving the Truth Orientation of Engineers to Everyone Else." presented at the USENIX, Vail, CO, June 2014.
- Hanson, Robin. "On Panel Bitcoin and the Future: Projections and Predictions." presented at the Future of Bitcoin & the Blockchain O'Reilly Radar Summit, San Francisco, CA, January 2015.
- Hanson, Robin. "Prediction Markets." presented at the Game Design For Citizen Science, New York University Game Center, New York, NY, February 2015.
- Hanson, Robin. "Prediction Markets & Related Topics." presented at the Neural Information Processing Systems conference, Montreal, Canada, December 2014.
- Hanson, Robin. "Shall We Vote On Values, But Bet On Beliefs?" presented at the Dept. of Economics Seminar, Northeastern University, Boston MA, July 2014.
<http://onlinelibrary.wiley.com/journal/10.1111/%28ISSN%291467-9760>.
- Hanson, Robin. "The Promise of Prediction Markets." presented at the Cognitive Computing Forum, San Jose, CA, August 2014.
- Laskey, Kathryn. "SciCast: A Combinatorial Prediction Market for Science and Technology Forecasting." presented at the Fifth Workshop on Game-Theoretic Probability, Centro de Investigación en Matemáticas (CIMAT), Valenciana, Mexico, November 2014.
- Laskey, Kathryn, and Shou Matsumoto. "Trade-Based Asset Models for Combinatorial Prediction Markets." presented at the Eleventh UAI Bayesian Modeling Applications Workshop (BMAW 2014), Quebec, Canada, 2014.
- Maxwell, Daniel. "Crowd Sourced Technology Forecasting Using Prediction Markets." presented at the Military Operations Research Society Workshop on Risk Assessment, Alexandria, VA, October 2014.
- Olson, Kenneth, Robin Hanson, Charles Twardy, and Kathryn Laskey. "Incentives for Crowdsourced Forecasts: Quality and Quantity." presented at the Society for Judgment and Decision Making Annual Meeting, Long Beach, CA, November 2014.
- Olson, Kenneth, Charles Twardy, Kathryn Laskey, and M. Burgman. "Interval Elicitation of Forecasts in a Prediction Market Reveals Lack of Anchoring 'Bias.'" presented at the MIT Collective Intelligence Conference, Boston, MA, June 2014.
- Sun, Wei, Kathryn Laskey, Charles Twardy, Robin Hanson, and B. Goldfedder. "Trade-Based Asset Model Using Dynamic Junction Tree for Combinatorial Prediction Markets." presented at the MIT Collective Intelligence Conference, Boston, MA, June 2014.
- Twardy, Charles. "Crowdsourcing Cybersecurity?" presented at the 5eyes Analytic Training Workshop, University of Mississippi, Oxford, MS, March 2015.
- Twardy, Charles. "Now How Much Would You Pay? The SciCast S&T Prediction Market." presented at the 5eyes Analytic Training Workshop, Pentagon City, VA, November 2014.

Twardy, Charles. "Using the SciCast Prediction Market to Complement Cyber Early Warning." presented at the FS-ISAC Meeting, New York, December 2014.

Twardy, Charles, and Robin Hanson. "Ask Me Anything Session (AMA)." presented at the ACS Science Reddit, August 2014.

Twardy, Charles, and T. Sanders. "Forecasting Chemistry." presented at the ACS Webinar, August 2014.

Previous Years Y1 – Y3

Publications Y1 – Y3

Berea, A. "Adaptive Agents in Combinatorial Prediction Markets." In *Handbook of Human Computation* Ed. Pietro Michelucci, 367–76, 2013.

Berea, A., Charles Twardy, and Daniel Maxwell. "Forecasting the Failed States Index with an Automated Trader in a Combinatorial Market." *Journal of Strategic Security* 6, no. 3 Supplement (2013): 38–51.

Hanson, Robin. "Shall We Vote on Values, But Bet on Beliefs?" *Journal of Political Philosophy, Published Online* 21, no. 2 (February 2013): 151–78. doi:10.1111/jopp.12008.

Hanson, Robin, and E. Yudkowsky. *The Hanson-Yudkowsky AI-Foom Debate eBook*. Machine Intelligence Research Institute, 2013.

Karvetski, C.W., Kenneth Olson, D.T. Gantz, and G.A. Cross. "Structuring and Analyzing Competing Hypotheses with Bayesian Networks for Intelligence Analysis." *EURO Journal on Decision Processes, Special Issue on Risk Management*, 1, no. 3–4 (2013): 205–31.

Karvetski, C.W., Kenneth Olson, D.R. Mandel, and Charles Twardy. "Probabilistic Coherence Weighting for Optimizing Expert Forecasts." *Decision Analysis* 10, no. 4 (December 2013): 305–26. doi:10.1287/deca.2013.0279.

Lyon, A., and E. Pacuit. "The Wisdom of Crowds: Methods of Human Judgment Aggregation." In *Handbook of Human Computation* Ed. Pietro Michelucci, 599–614, 2013.

Powell, Walter A., Robin Hanson, Kathryn B. Laskey, and Charles Twardy. "Combinatorial Prediction Markets: An Experimental Study." In *Proceedings of the Seventh International Conference on Scalable Uncertainty Management (SUM 2013)*, edited by Weiru Liu, V.S. Subrahmanian, and Jef Wijsen, 8078:283–96. Lecture Notes in Computer Science. Alexandria, VA: Springer Berlin Heidelberg, 2013. http://dx.doi.org/10.1007/978-3-642-40381-1_22.

Twardy, Charles, and Kathryn Blackmond Laskey, eds. "DAGGRE Annual Report 2012," May 25, 2012.

Twardy, Charles, and Kathryn Blackmond Laskey, eds. "DAGGRE Annual Report 2013," May 25, 2013.

Twardy, Charles, and Kathryn Blackmond Laskey, eds. "SciCast Annual Report 2014," May 25, 2014.

Wintle, B., S. Mascaro, F. Fidler, M. McBride, M. Burgman, L. Flander, G. Saw, C. Twardy, A. Lyon, and B. Manning. "The Intelligence Game: Assessing Delphi Groups and Structured Question Formats." Perth, 2012.

Conference Proceedings Y1 – Y3

- Berea, Anamaria, Daniel Maxwell, and Charles Twardy. "Improving Forecasting Accuracy Using Bayesian Network Decomposition in Prediction Markets." Alexandria, VA, 2012.
- Olson, Kenneth, and C.W. Karvetski. "Improving Expert Judgment with Coherence Weighting." In *Proceedings of IEEE Intelligence and Security Informatics*. Seattle Washington, USA, 2013.
- Sun, Wei, Robin Hanson, Kathryn Laskey, and Charles Twardy. "Learning Parameters by Prediction Markets and Kelly Rule for Graphical Models," 39–48, 2013. <http://ceur-ws.org/Vol-1024/paper-05.pdf>.
- Sun, Wei, Robin Hanson, Kathryn Laskey, and Charles Twardy. "Probability and Asset Updating Using Bayesian Networks for Combinatorial Prediction Markets." In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI-12)*. Catalina Island, USA, 2012. <http://arxiv.org/abs/1210.4900>.
- Twardy, Charles. "What Is the Use of a Conditional Forecast?" Linthicum, MD, 2012.

Presentations Y1 – Y3

- Berea, A. "Automated Trading in Prediction Markets - The Case of DAGGRE Autotradars." presented at the IAFIE Annual Meeting, El Paso, TX, 2013.
- Berea, A. "Frameworks for Forecasting Science and Technology Innovation." presented at the 5th Annual Conference of the Center for Complexity in Business, University of Maryland, College Park, MD, 2013.
- Berea, A., Daniel Maxwell, and Charles Twardy. "Automated Trading in Prediction Markets." presented at the SBP'13, Washington, D.C, 2013.
- Berea, A. "Automated Trading in Prediction Markets." presented at the SBP'13, Washington, D.C, 2013.
- Berea, A. "Forecasting the Failed States Index." presented at the IAFIE Annual Meeting, El Paso, TX, 2013.
- Berea, A., and Charles Twardy. "Automated Trading in Prediction Markets." presented at the International Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction, Washington, D.C., 2013.
- Hanson, Robin. "A New Form of Government: Futarchy." presented at the Public Choice Outreach Conference, George Mason University, Fairfax, VA, August 2013.
- Hanson, Robin. "Bayes Net Based Combinatorial Prediction Markets." presented at the Prediction engines panel. Microsoft Research Faculty Summit, Redmond, WA, July 2013.
- Hanson, Robin. "Bayes Net Based Combinatorial Prediction Markets." presented at the Microsoft Research, New York, July 2013.
- Hanson, Robin. "Em Econ 101: Sketching A Society of Emulated Minds." presented at the Digital Seminar, Harvard Business School, Boston, MA, April 2013.
- Hanson, Robin. "Prediction Market Forecasts – What Gets Used and Why?" presented at the MITRE Technology Forecasting Perspectives workshop, McLean, VA, June 2013.
- Hanson, Robin. "Prediction Markets for IT Finance Management." presented at the The Research Board (of CIOs), Houston, TX, January 2013.

- Hanson, Robin. "Presentation and Discussion of Prediction Markets and SciCast." presented at the Potomac Institute CREST Seminar, Arlington, VA, January 2014.
- Hanson, Robin. "Shall We Vote On Values, But Bet on Beliefs?" presented at the Philosophy, Politics & Economics Seminar, Duke University, Durham, NC, March 2014.
<http://onlinelibrary.wiley.com/journal/10.1111/%28ISSN%291467-9760>.
- Hanson, Robin. "The Age of Em: Imagining A Future of Emulated Minds." presented at the NYU Abu Dhabi Economics Seminar, Abu Dhabi, UAE, December 2013.
- Karvetski, C.W. "A Comparison of Statistical Smoothing Models." presented at the IARPA ACE PI meeting, Tysons Corner, VA, May 2012.
- Karvetski, C.W. "Improving Prediction Market Forecasts." presented at the 50th Annual Edwards Bayesian Research Conference, Fullerton, CA, January 2012.
- Karvetski, C.W. "Statistically Improving Prediction Market Forecasts." presented at the INFORMS annual meeting, Phoenix, AZ, October 2012.
- Karvetski, C.W. "Statistically Improving Prediction Market Forecasts." presented at the The Annual Meeting of the Society for Risk Analysis, Baltimore, MD, 2013.
- Karvetski, C.W., and Kenneth Olson. "Optimizing Expert Forecasts with Probabilistic Coherence." presented at the Intelligence Community Postdoctoral Research Conference, Washington, D.C., 2013.
- Olson, Kenneth. "Best Practice for Eliciting and Weighting Incoherent Judgments." presented at the Society for Judgment and Decision Making, Toronto, ON, Canada, 2013.
- Olson, Kenneth. "Improving Expert Judgment by Coherence Weighting." presented at the The Annual Meeting of the Society for Risk Analysis, Baltimore, MD, 2013.
- Olson, Kenneth, and C.W. Karvetski. "Analyzing Competing Intelligence Hypotheses with Belief Network Sharing." presented at the Intelligence Community Postdoctoral Research Conference, Washington, D.C., 2013.
- Sun, Wei, Robin Hanson, Kathryn Laskey, and Charles Twardy. "Learning Parameters by Prediction Markets and Kelly Rule for Graphical Models." presented at the Uncertainty in Artificial Intelligence(UAI), Bayesian Modeling Application Workshop, UAI, Bellevue, WA, July 2013.
- Twardy, Charles. "Overview of ACE & DAGGRE Lessons Learned." presented at the Teaching & Researching Intelligence panel. Global Intelligence Forum, Dungarvan, Ireland, July 2013.
- Twardy, Charles. "SciCast." presented at the FEDLINK Analytical Methods for Technology Forecasting Workshop, Washington, D.C., March 2014.
- Twardy, Charles. "SciCast: Collective Forecasting of Science and Technology." presented at the 5eyes Analytic Workshop, Oxford, MS, March 2014.

Experiments and Studies

Overview

This chapter reports on the results from the following key studies and experiments:

1. Combinatorial Representations of Ordered Questions
2. Incentives for Crowdsourcing
3. Correlates of Good Forecasting
4. Algorithms for Asset Management
5. Recommender Effectiveness
6. Fraud Detection and Analysis

“Ordered Questions” discusses an algorithm for representing distributions on dates and other ordered questions with large state spaces. Scaled questions can represent expected value, but not distributions. Regular multiple choice questions are fine for coarse bins, but often writers want both range and precision. For example, the flu question authors insisted that forecasters be able to specify any of the ~30 weeks of the flu seasons. That was awkward enough as a single question, and nearly impossible in combination. “Ordered Questions” describes an approach allowing progressive abstraction.

“Incentives” reports the results of our 2014 activity experiment and accuracy pilot, as well as the large 2014-2015 accuracy study, the 2015 combo edits contest, and some non-forecasting incentives used in Y4. Briefly: (1) paying for activity strongly increased activity without affecting overall accuracy; (2) paying for accuracy increased both activity and accuracy, but the details were complicated; (3) paying for combo edits increased combinatorial edits, number of links, and information on the links; and (4) paying for surveys attracted participation mainly from our already most active forecasters.

“Correlates” reports an exploratory data analysis of the correlates of good forecasting, particularly among forecasters who responded to our demographic and psychometric surveys. Our dependent measure was the information gain per edit. R^2 values were necessarily low ($<.1$) as there is a great deal of variation in an individual’s forecasts and between individuals. Nevertheless, the Berlin numeracy test was selected as the most diagnostic survey both when considering each trade individually (exceptionally low R^2) and when considering each forecaster’s average info gain. The best hierarchical model also included the Cognitive Reflectance Test (CRT), Forecasting Motivation (negative effect), S&T Accuracy, and Extraversion (more is better). We also found that the total Survey Score for the Top5 and Top10 was substantially above the rest, but the Top20 average is similar to the rest.

“Algorithms” compares the DAGGRE and SciCast asset management systems, with a detailed apples-to-apples space and time analysis. The SciCast system provides a huge space savings that in practice created a huge time savings due to reduced garbage collection and memory management. We describe an enhancement to the currently implemented SciCast asset management system that avoids the need to resort to



approximation when there are many linked questions and is much faster at cash and expected value calculations. However, we also note that probability calculations cost 4x as much as asset calculations, which limits the overall advantage of the enhanced asset calculations.

“Recommender Effectiveness” reports on A/B trials comparing the actual recommender to random recommendations. There is no significant difference with respect to actual edits, possibly because forecasters do not use the question carousel to drive their forecasting behavior.

Finally, “Fraud Detection” reports lessons learned from our efforts to detect and deter fraud in the large 2014-2015 accuracy incentives study. The most common – or detectable – type of fraud was an attempt to dump points from a “skill” account into a “real” account, via coordinated trades.

Ordered Questions

Introduction

Most algorithms and data structures designed for Bayesian/Markov networks focus on the case of variables with a small number of values, usually less than a dozen. However, many variables that are interesting to forecast have thousands or millions of possible values. Examples include dates, geographic locations, real numbers, or positions in large category hierarchies. Naively using standard Bayesian/Markov network techniques on such variables would be extremely inefficient.

We have begun to explore algorithms and data structures more appropriate to the case of variables with a great many values. We have initially focused on the case of ordered variables, represented via an abstraction tree. For example, a time variable has abstractions such as days, weeks, months, and years. We call this representation a “value tree.”

In previous years we developed efficient data structures and algorithms for the case of a single stand-alone value tree. This year we developed ways to integrate such variables into larger Bayesian/Markov networks.

Lone Value Trees

As discussed in last year’s report, our first task was to design algorithms and data structures for managing combinatorial prediction markets in the case of a single value tree. The data structure is a tree, and most of this tree can be virtual, with real data structures only created when there is a deviation from default local values. Each node in this tree holds a fraction that is the conditional probability of that node given its parent, and for each user it holds one local min asset number. Each node also stores dates when these numbers were last updated.

To read the probability of a node, one need only sweep down from the root multiplying together the fractions found. To read the min asset of a node, one just sweeps down from the root adding up the local min assets found. However, to read the expected assets of a node, one must also sweep below that node to update

all child etc. nodes that need updating, computing the expected asset of a node in terms of its child fractions and assets. At the leaves of the value tree the expected asset is the same as the local min asset.

An edit changes the probability and assets of a set of nodes that exactly and minimally covers the value range for the edit. To implement an edit, one first sees how the local fractions would change in this node set, finds the matching asset changes, and checks that the new asset values won't make any min asset value negative. Then one changes the local fractions and min assets, and sweeps back toward the root touching all the siblings along the way, to ensure that all the fractions of children of a node add to one, and finding new local min values.

For most of these operations the time cost is proportional to a node's depth times the tree's branching ratio, which is very efficient. Only the expected asset calculation takes longer. The space cost can also be very efficient when only nodes near the tree root are actually edited; in that case the rest of the tree is never explicitly created.

Value Tree As Bayes Net – Factorized Representation

Graphical models have revolutionized the use of probabilistic modeling over simple discrete variables in many domains. This success has resulted from their ability to compactly represent context-free conditional independence relationships over a set of such variables, together with algorithms for exploiting this independence. Limited success in exploiting context-specific independence and in extending results to relational and first-order logics has further extended the range of applicability. Little has been done, however, in exploring the structure of the domain variables themselves. Questions of interest often have highly structured domains, such as dates, geographies, research areas, or organizations. For example, given an event such as "date of the detection of the first Ebola transmission within the US" we may have knowledge or wish to query on the day, week, month, year, or other term within a semi-lattice of calendar terms. Similarly, for a question such as "winner(s) of the Nobel prize in Physics on 2015" relevant terms might include individuals, research groups, organizations, countries, or a variety of other such groupings of the base terms (individuals). Note also in this latter case the base terms may not be mutually exclusive (there can be more than one winner), although there may be constraints on the number of possible awardees. While such structure could in most cases be flattened, for example by simply representing each day in a calendar variable as a separate propositional variable, such an approach conflates levels of problem structure, and risks losing the computational efficiencies gained by separating these levels. For example, anyone who has tried has quickly discovered that it is a bad idea to represent a simple multi-valued discrete domain explicitly as a set of mutually exclusive and exhaustive binary variables.

The ValueTree algorithm discussed above can be embedded in a graphical model to capture its attractive inference properties and extend them to combinatorial trading models. We have adapted the intuitions from the ValueTree for one specific domain structure: a hierarchy. We have shown how much of this efficiency can be captured in a general graphical-model setting through the use of a sub-graph for each such

variable with multiplicatively-factored dependencies within the subgraph. Experimental results have demonstrated the efficiencies gained over simpler representations of calendar events and other similar hierarchical structures.

A robust way to embed a value tree in a larger Bayes net structure is to represent the value tree itself as a Bayes net. We have found a way to do this via a clever choice of possible values for intermediate nodes in the Bayes net.

This Bayes net representation reverses the parent/child relations in the value tree. All leaves of the value tree are roots of the net, and all other nodes have only one child in the net, except for the root of the value tree, which has none.

The leaves of the value tree are all described via binary variables in the net. Each of these binary root nodes is associated with a base term in the hierarchically structured domain, and the true/false states of the binary variable indicate whether the associated base term is true. . All other nodes in the Bayes net are described via variables with N+2 values, where N is the branching factor of the value tree (each tree node has N children). The N values each specify that a particular net-parent is true, and all the other net-parents are false. The other two values specify (NONE) that none of those net-parents are true, or (MANY) that more than one of them are true.

A simple truth-table specifies the value of a non-root node given all possible values of its parents in the Bayes net. For example, the following table gives the value of the node **A or B** given the values of the nodes **A** and of **B**:

Table 2: Truth table for factored value tree node

A	B	A OR B
NO	NO	NONE
YES	NO	YES
NO	YES	NO
YES	YES	MANY
NO	NONE	NONE
YES	NONE	YES
NO	MANY	MANY
YES	MANY	MANY
NONE	NO	NONE
MANY	NO	MANY
NONE	YES	NO
MANY	YES	MANY
NONE	NONE	NONE



A	B	A OR B
MANY	NONE	MANY
NONE	MANY	MANY
MANY	MANY	MANY

Given this construction of Bayes net nodes, variables, and relations between variables, we simply give this network evidence saying that these two extra values, none-true and more-than-one true, are false for the root of the value tree. Bayesian inference then ensures that any other evidence will propagate correctly to update the probabilities of all other nodes in the Bayesian network.

With this construction of a value tree as a Bayes net, it becomes in principle straightforward to allow users to make edits of one value tree variable conditional on another such variable. One need only add links in the total Bayes net connecting the different nets corresponding to the different value trees. For a small number of such connections, the entire Bayes net might remain singly-connected, and thus easy to compute. However, once more connections are made the network will no longer remain singly connected, and so more advanced methods must be used to manage the total network.

Performance

We stopped development of the value tree model just prior to performing empirical performance tests, when it became clear that we would not be able to implement, test, and deploy the feature during Y4. The key question is the computational performance when conditioning one value tree upon another. The key benefit is that the conditioning can be done at a higher level of abstraction than the leaf nodes. Therefore, although inference is fundamentally exponential in the product of the state space, value tree can greatly reduce the state space, for example from an intractable 360 (days) to 12 (months) or 4 (quarters).

However, the current representation lacks the ability to express arithmetic dependency, and cannot for example, compactly express that the delivery date is expected to be 3 days after the ship date, at least not if the ship date itself is highly uncertain.

Target Journals and Conference

It is not clear that this representation is sufficient by itself for journal publication, but possibilities include:

- International Journal of Approximate Reasoning
- Journal of Artificial Intelligence Research

Correlates of Good Forecasting

Background

Last year's participant profile was based on the roughly 150 forecasters who had completed the voluntary user surveys. Based on self-reported answers and online psychometric tests, we reported in 2014:

- 1 in 3 of our forecasters were over age 60, mostly 60-70
- 1 in 4 were 30-40 years old.
- 1 in 4 had a Ph.D.
- Top occupations were Sciences, Computers & Mathematics, and Education & Library
- Market score had a 0.57 correlation with #edits
- Accuracy (2-Brier) correlates moderately well (0.32) with Actively Open-minded Thinking (AOT)

In summary, the strongest predictor of accurate forecasting was user performance on the Actively Open-Minded Thinking task, which had a -0.32 correlation with Brier, followed by declaration of being active in research (-0.13) and score on the cognitive reflectance test (-0.12). Table 3 shows the correlation of Brier score with various predictor variables, based on voluntary survey responses obtained through December 2014. Results of additional incentivized surveys, including a fifth survey suggested by IARPA, are presented later in this chapter.

Table 3: Correlates of good forecasting as of December 2014 (negative is more accurate).

Variable	Average	N	Corr. w/ Brier
Gender	81% male	175	+0.03
Age	47	177	-0.01
Citizenship	U.S.	173	NA
Education	Bachelor's	176	-0.08
Research	42% active	177	-0.13
Publication	56%	176	-0.10
Patent	12%	177	-0.07
AOT	21 /31	171	-0.32
CRT	2.35 /3	156	-0.12
#Edits	24.21	4152	-0.04
Points	5019	4152	-0.06

Note that while Brier score measures the (in)accuracy of each forecast, the primary coin in a prediction market is the market score, which measures the total amount of information provided by the forecaster. Brier score is at best weakly correlated with the number of edits (-0.04), but market score, as may be expected, is correlated



with the number of edits (0.57). Brier score and market score are only weakly related (-0.06) if at all. This may at first be surprising, but is simply a consequence of rewarding different activities. A strict Brier optimizer would make a few forecasts on select questions, and only update when their beliefs changed, consequently having too little activity to amass huge market resources. A strict point optimizer would make as many forecasts as possible, concentrating on efficiently correcting where the market is “most wrong”, to the likely detriment of his or her Brier score. Most people are somewhere in between, opportunistically taking the easy points (making big corrections) wherever possible, but spending fair resources calibrating, improving, and extremizing stable estimates. The more resources a forecaster has, the more they are likely to have left over to extremize. Extremizing is an expensive operation: moving a 98% to a 99% ties up 100 points for a possible gain of about 1.5.

Objectives

The power-law distribution of the leaderboard shows that some traders are much, much better than others at score optimizing. There are superforecasters in the market. It is natural to ask what makes them so. Certainly they tend to be more active, but even in the TopN, the correlation is moderate at best – some traders make fewer but more lucrative trades, while others accumulate less per trade but are much more active. Can we find behaviors or predisposing factors of superforecasters? Can we help train forecasters to become better? If so, we could award more starting points for completing key training, or scoring well on a pretest. Alternatively, are there pure subject matter experts in the market who make a few well-considered safe-mode estimates and then move on? They might accumulate few market points but do exceptionally well on per-trade accuracy and calibration.

To facilitate investigating these questions, SciCast traders were asked to participate in a series of surveys. These surveys covered demographic, analytic, and psychometric domains. Late in Y4, there were offered a chance at \$100 Amazon Gift Cards based on how many of the surveys they had completed, raising the total completed from ~170 to ~245. This section examines the relationship between the survey responses and a forecaster’s ability to make good forecasts.

The next section details the surveys and scores, including five possible measures of “goodness”: Trade Brier Score (TBS), Question Brier Score (QBS), Brier Difference, Info Gain (gain), and Normalized Gain (nGain), which is related to market score. Most of our results used Info Gain, or a transformation thereof.

Surveys & Survey Scoring

SciCast has always offered its forecasters four demographic and psychometric surveys comparable to those used by the Good Judgment Project (Baron, Mellers, Tetlock, Stone, & Ungar, 2014).. Periodically, we ran banner ads to ask participants to complete them. Late in Y4, we added a fifth survey and offered chances at gift cards proportional to the number of sensibly completed surveys (out of the five possible surveys). The surveys resided on SurveyMonkey. Forecasters were asked:

Help us learn more about you so that

Experiments and Studies • 29 of 143

This report is approved for unlimited public release, and is subject to the disclosure on the title page.



1. We can recommend questions that match your interests, expertise, etc.
2. We can contribute to research on correlates of good forecasting. (SciCast Auxiliary Question and Forecaster Metadata, p. 1)

The survey questions have been grouped into the 9 survey sections listed below:

Table 4: Survey Sections, Names, and Lengths

Survey Section Name	# Questions in the Survey Section
Science & Technology Accuracy	13
Science & Technology Calibration	13
Forecasting Motivation	9
Extraversion	8
Actively Open-minded Thinking	9
Hedgehog-Fox	1
Berlin Numeracy	4
Expanded Cognitive Reflection Test	15
Quantitative Analysis Methods ¹	9

Where possible, survey question responses are scored as a number between 0 and 1. This makes statistical analysis of these questions easier to interpret. Survey Questions were classified into the following types:

1. Yes/No
2. Boolean
3. Percent
4. Open-Ended Response
5. Ordered
6. Reversed

¹ This section was added in 2015 in response to a Y4 IARPA request.

Yes/No Survey Question Scoring

These questions were scored using the following table:

Response	Score
No	0
Yes	1

Boolean Survey Question Scoring

These questions were scored using the following table:

Response	Score
False	0
True	1

Percent Survey Question Scoring:

All of these responses were already scaled between 0% and 100%. Therefore these responses did not need any scaling. Here are two examples:

Response	Score
25%	0.25
100%	1

Open-Ended Response Survey Question Scoring:

Open-ended survey questions come in two varieties: questions with a known correct answer, and questions which have no correct answer. Responses to open-ended questions with no correct answer were all scored as 0.0. In our regression model they have no predictive value. Here is an example of a question with no correct answer:

“Where do you hold citizenship? (You can select more than one.)”

The survey participant was presented with a list of every country on Earth and could then select any combination of these countries. In the future, responses to questions of this type could be treated as a decomposed list of “Boolean” Questions, one for each country, but that was infeasible for this reporting period.

Responses to open-ended questions with a correct answer were scored using the following table:

Response	Score
Correct Answer	1
Incorrect Answer	0

Here is an example:

Question: “Imagine we are throwing a five-sided die 50 times. On average, out of these 50 throws how many times would this five-sided die show an odd number (1, 3, or 5)?”

Correct Answer: 30.

Sample Scores:

Response	Score
30	1
25	0

Ordered Survey Question Scoring

An ordered survey question measures something on a sliding scale. Here is an example from SciCast Auxiliary Question and Forecaster Metadata, p. 8:

“What is your highest level of education? [Priority 1]

- Not a high school graduate [0]
- High school diploma or equivalency (for example, GED) [1]
- Some post-high-school education or certification [2]
- Associate’s degree (for example, AA, AS) [3]
- Bachelor’s degree (for example, BA, AB, BS) [4]
- Master’s degree (for example, MS, MPH, MBA) [5]
- Professional degree or professional doctorate (for example, MD, DVM, JD, or PsyD) [6]
- Other Doctorate degree (for example, PhD or EdD) [7]”

In this example, the first response (“[0]”) denotes the lowest level of education and last response (“[7]”) denotes the highest level of education. There are 8 possible responses scored with the following formula:

$$score = \frac{response_index}{scale}$$

In our example, the scores are calculated as follows:

Response	Score
[0]	$score = \frac{0}{7} = 0$



[1]	$score = \frac{1}{7} = 0.143$
...	
[7]	$score = \frac{7}{7} = 1$

Reverse Survey Question Scoring

Reverse ordered (“Reverse”) question are like ordered questions with one critical difference, the first response indicates the “highest” level and the last response indicates the “lowest” level. The following “Reverse” question is measuring “open-mindedness”, and is clearly marked “[REVERSE]” to ensure that is reversed during coding:

Changing your mind is a sign of weakness. **[REVERSE]**

	1 (1)	2 (2)	3 (3)	4 (4)	5 (5)	6 (6)	7 (7)
Completely Disagree: Completely Agree (1)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(SciCast Auxiliary Question and Forecaster Metadata, p. 23).

The first response of 1 indicates complete disagreement with this statement. As such, it is the strongest indicator of open-mindedness. The last response of 7 indicates complete agreement with the statement and is the weakest indicator of open-mindedness. Reverse question are scored as follows:

$$score_{reverse} = 1 - score_{scaled} = 1 - \frac{response_index}{scale}$$

In our example the scores are computed as follows:

Response	Response Index	Score
1	0	$score = 1 - \frac{index}{scale} = 1 - \frac{0}{6} = 1$
2	1	$score = 1 - \frac{index}{scale} = 1 - \frac{1}{6} = 0.833$
...	...	
7	6	$score = 1 - \frac{index}{scale} = 1 - \frac{6}{6} = 0$

It can be seen that the first response gets a score of 1 and the last response gets a score of 0.

That covers all the different kinds of questions and how they are coded for regression analysis. We next consider how the answers are related to forecast quality or forecaster quality.

Forecast Quality

There are two ways to measure the quality of an individual forecast: **accuracy** and **information**. Accuracy measures closeness to the actual outcome. Information measures the reduction in uncertainty. Consider a binary question that will resolve True, and two edits that both increase the correct belief by 1 percentage point. The first edit changes the belief from 1% \rightarrow 2%. The second edit changes it from 97% \rightarrow 98%. Table 5 shows the difference between accuracy and information, where we measure accuracy using 2-Brier, so 0 is worst and 2 is best. The first edit has a terrible Accuracy score (nearly 0, the worst possible) but contributes a whole bit of information. The second edit has a great Accuracy score (nearly 2, the best possible), but only contributes 0.015 bits. We can also compare the *difference* in Accuracy, which measures how much the edit improves our Accuracy (or Brier score). Because Brier score is a quadratic loss, it will also favor correcting big errors over improving mostly correct forecasts. In this case, the first edit receives 39x the reward.

Table 5: Accuracy vs Information as measures of forecast quality. Higher is better. Accuracy is 2-Brier, so ranges from 0..2, with 2 best.

	p	p'	Actual	Acc (2-Br)	Info (bits)	Acc Diff
Edit 1	0.01	0.02	1	0.0792	1.000	0.039
Edit 2	0.97	0.98	1	1.9992	0.015	0.001

We want our *overall* market to be as accurate as possible. Each question is evaluated on its mean daily Brier score (or, in our own evaluation, the actual area under the Brier curve, accounting for all edits at their actual duration), and SciCast's overall performance is the mean over all question scores. Because Brier is a quadratic loss, we want to prioritize fixing big errors. The market score already does this: it pays 100 points per bit. Therefore, participants trying to maximize their market score will tend to seek corrections that help overall accuracy.

Nevertheless, there are at least two kinds of forecasting strategies: maximizing market score and maximizing accuracy as measured by Brier score. The stereotypical subject-matter expert has a high accuracy but a low information: she will make a few well-chosen forecasts that are highly accurate, or at least well-calibrated, and will not bother to reassert them unless she has changed her mind. She is closer to a value-trader, except that in the extreme case, she isn't really trading at all, but occasionally asserting an expert opinion and leaving the quotidian trading to others.

The stereotypical technical trader, on the other hand, will have exceedingly high information, but middling to low accuracy. This trader is very active, but faced with a choice of how to spend 100 points, will prefer to correct an egregious error rather than polish a mostly-correct forecast. Consider that it costs 100



points to move 98% → 99%, for a potential gain of only 1.5 points. Those same 100 points could be invested to correct a 1% → 51% for a gain of 566 points if correct. This choice will also yield a Brier/Accuracy improvement of 1.56 out of possible 2, compared with .001 for polishing the 98% to 99%.

The most accurate system is likely to mix the two kinds of trader: technical traders may actively seek to maintain the forecasts made by subject-matter experts. This may seem unfair to the experts. One solution is to create an automated system for reasserting expert beliefs, similar to a stock market limit order. In Y4 we experimented with such a system, but our implementation had a flaw making it gameable, so we disabled it. Instead, we opened up the API so forecasters could write their own automated agents. User @jkominek created an open-source bot well-suited to subject-matter experts, in that it takes a database of beliefs about questions, and makes well-informed edits on a repeating basis. (Kominek, 2014)

Measures of Accuracy

Five measures of accuracy are considered here: Trade Brier Score (TBS), Question Brier Score (QBS), Brier Difference (BD), and Information Gain (gain), and Normalized Gain (nGain).

Trade Brier Score (TBS) Calculations

Trade Brier scores are calculated using the following equation:

$$TBS = \sum_{i=1}^N (f_i - o_i)^2$$

Where:

- f_i is the forecast value for a question choice
- o_i is the actual outcome for a question choice
- i is the index for the question choice
- N is the number of choices for the question

Example

From SciCast DB, the raw data for trade 19 looks like this:

trade_id	old_value_list	new_value_list	assets_per_option	serialized_settled_values
19	0.20000008,0.20000008,0.20000008,0.20000008,0.20000008	0.3199998,0.16999991,0.14999993,0.14999993,0.20999986	[67.80713280347366, -23.44658307149238, -41.503807635674484, -41.503807635674484, 7.038875081349703]	[0, 0, 0, 0, 1]

The table below shows the same data in the normalized *historical_trade_choice* table. The table shows a probability for each of the possible answers to Question 19, normalized to sum to 1. Each of these choices has an associated *choice_brier_score*, which is the squared difference between the probability and the outcome for that choice

choice_trade_id	choice_id	choice_old_value	choice_new_value	choice_asset_resolution	choice_settled_value	choice_brier_score
19	1	0.20000008	0.3199998	67.8071328	0	0.102399872
19	2	0.20000008	0.16999991	-23.44658307	0	0.028899969
19	3	0.20000008	0.14999993	-41.50380764	0	0.022499979
19	4	0.20000008	0.14999993	-41.50380764	0	0.022499979
19	5	0.20000008	0.20999986	7.038875081	1	0.624100221

The *choice_brier_score* for trade 19 choice 1 is:

$$choice_brier_score = (0.3199998 - 0)^2 = 0.102399872$$

Trade Brier Score (TBS) for trade 19 is the sum of the values in the *choice_brier_score* column:

$$TBS = 0.102399872 + 0.028899969 + \dots + 0.624100221 = 0.800400021$$

Question Brier Score (QBS) Calculations

In order to calculate the Question Brier Score for a question we must aggregate the TBS for each trade made on that question. This aggregation is a weighted sum over the entire period the question was active, and is equivalent to the area under the Brier-score timeseries for the question. QBS is calculated using the following formula:

$$QBS = \sum_{t=1}^N (weight_t * TBS_t)$$

Where

$weight_t$ is the percent of time trade t was most recent

TBS_t is the TBS for trade t

t is the index of the trade

N is the total number of valid trades for the question

QBS weights are calculated using the following formula:

$$\frac{duration_{trade}}{duration_{question}}$$

Where

$duration_{trade}$ is the time between this trade and the next trade

$duration_{question}$ is the total time the question was traded

Here is some sample data:

<i>this_question_id</i>	<i>this_trade_id</i>	<i>this_trade_date</i>	<i>next_trade_created_at</i>	<i>question_resolved</i>
1	532	"2013-12-19 05:02:53.551341"	"2013-12-19 09:01:32.767487"	FALSE
1	533	"2013-12-19 09:01:32.767487"	"2014-01-10 03:21:07.642972"	FALSE
1	914	"2014-01-10 03:21:07.642972"	"2014-01-10 03:22:39.422948"	FALSE
1	915	"2014-01-10 03:22:39.422948"	"2014-03-13 14:09:20.48039"	TRUE

Notice that the *next_trade_created_at* for the first trade is equal to the *this_trade_date* for the second trade, and so on. Also notice that for the last trade the *question_resolved* value is TRUE. This is the final trade for this question and the *next_trade_created_at* is set to the question's *date_known*. Therefore:

$$duration_{trade} = next_trade_created_at_i - this_trade_date_i, \text{ and}$$

$$duration_{question} = next_trade_created_at_{last} - this_trade_date_{first}$$

Where

next_trade_created_at_i is the *next_trade_created_at* for trade i ;

this_trade_date_i is the *this_trade_date* for trade i ;



$next_trade_created_at_{last}$ is the $next_trade_created_at$ for the last trade made against the question;

$this_trade_date_{first}$ is the $this_trade_date$ for trade the first trade made against the question;

Calculating Brier Difference

Brier Difference (BD) is another measure of trade accuracy. BD represents the shift in QBS which can be attributed to any single trade. It is simply the Brier(this_trade) – Brier(prev_trade). Because Brier is an error measure, improvements have a negative BD.

Calculating Info Gain

Every trade constitutes a binding contract on how much will be won or lost for each possible resolution (actual outcome) of the question. Generically, these are called Gains, and are determined by the LMSR formula where, for each outcome, $gain = 100 * \log_2(p' / p)$, where p' is the new probability and p is the old probability. A natural measure of the information in a forecast is the actual gain, or the log Accuracy. Let p denote the probability vector such as [.8, .2] for a binary question, and let r denote the resolved probability such as [1,0] but including mixture resolutions. Then, the log accuracy of a single forecast is:

$$\log Acc = \sum_i r_i \log p_i$$

where we use \log_2 to get the result in bits. We can then define the information gained as the difference in logAcc between the old forecast p and the new forecast p' :

$$InfoGain = \log Diff = \log Acc(p') - \log Acc(p)$$

Calculating Normalized Gain

Although most SciCast questions are binary, the number of outcomes varies widely, and we want to compare them on equal footing – it is much harder to nail a 35-choice question than to nail a binary: more options mean that it requires more information to resolve the question.

Actual gain pays 100 points per bit of information. However, the entropy of the initial uniform distribution is $H = \sum_{i=1}^N p_i \log p_i = \sum_{i=1}^N (1/N) \log(1/N) = \log(1/N) = -\log N$, so the total amount of information in a forecast varies with the log of the number of outcomes. Therefore, we normalize it thus:

$$nGain = \frac{ActualGain}{\log(\# Choices)}$$

In the datamart, the *ActualGain* is recorded as the *asset_resolution* in the *historical_trades* table, and *choice_count* is the number of possible responses to the question.

Example:

Trade 80 in the SciCast DB was made against question 2. When question 2 resolved, the *asset_resolution* for trade 80 was set to 13.69 points.

Question 2: Which of the following changes will be reported about "trends in extent of selected biomes, ecosystems, and habitats" in the fourth edition of the Global Biodiversity Outlook report?

1. Positive changes
2. Negative changes
3. No clear global trend
4. Insufficient information available

The *nGain* is calculated as follows:

$$nGain = \frac{asset_resolution}{\log(choice_count)} = \frac{13.69}{\log(4)} = 22.73$$

Although SciCast pays in bits, for scoring purposes the base of the *nGain* log is not important, as it amounts to a constant scaling factor.

Surveys as Predictors of Good Forecasting

Because our approach was to recruit as widely as possible, we actively minimized any barriers to entry. Therefore, we did not compel our forecasters to answer our surveys. We made it as easy as possible to begin forecasting, and then later tried to entice them to answer surveys. However, that means of the 11,000 forecasters who have ever registered, we have only a few hundred completed surveys, and there is no reason to suspect they are a random sample of our forecasters. We performed exploratory data analysis, guided in part by previous results such as reported by the Good Judgment Project (Mellers et al., 2014; Satopää et al., 2014).

As noted above, an earlier analysis had found moderate correlations between a user's average Brier score and two survey sections: Actively Open-minded Thinking (AOT) and the enhanced Cognitive Reflectance Test (eCRT). With ~70 more surveys and thousands more edits, we did not find any correlation with basic Brier score. However, we did find correlations between a forecaster's average Info Gain and some survey sections -- notably the Berlin (objective) numeracy test. It should also be noted that we did not regress directly on Info Gain because the residuals were strongly non-normal, as discussed below. However, as shown in Figure 14

and discussed in the Incentives section below, a cube-root transform was effective at achieving approximate normality of the residuals.

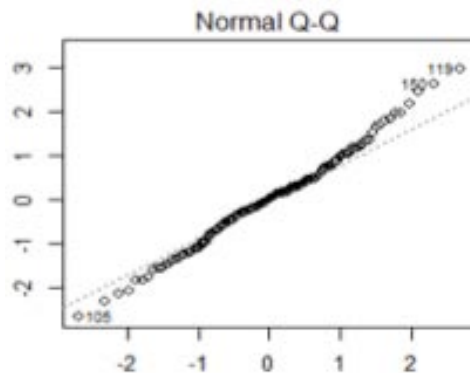


Figure 14: Q-Q Plot for cube-root info gain

We used a hierarchical analysis starting with total survey score (recall that each question was recoded to 0..1 with 1 being the presumed positive answer), breaking the overall score down by survey section to find the best sections, and finally examining the questions within a section. In this analysis, the survey sections are given in Table 6 below, omitting “4” which in our case was a recombination of two other sections:

Table 6: Participant Survey Section Numbers and Names

- 1 Expanded Cognitive Reflection Test
- 2 Berlin Numeracy
- 3 Forecasting Motivation
- 5 Science & Technology Accuracy
- 6 Science & Technology Calibration
- 7 Actively Open-minded Thinking
- 8 Extraversion
- 9 Hedgehog-Fox
- 10 Rely on Tools
- 11 Sources Consulted

Sections 1-9 were the same as used previously. Sections #10-11 were created late in Y4 by Artjay Javier (IARPA) and Ken Olson (then at GMU) to test advanced forecasters’ use of models or high-quality sources. It had far fewer responses (~70 instead of ~245).

Of these, the best single sections by R^2 were the 2, 1, 3, 5, and 9. Using a hierarchical approach, we added sections to Section 2 in attempts to improve R^2 until it stopped improving. The best model was Sections {2, 1, 3, 5, 8}. Adding more sections reduced the adjusted R^2 from its peak of about 0.06.

Within the best model, only Section 2 (Berlin numeracy) was statistically significant:

• • •

```

Call:
lm(formula = average_transformed_info_added ~ section_2 + section_1 +
    section_3 + section_5 + section_8, data = section_dataset)

Residuals:
    Min       1Q   Median       3Q      Max
-0.7801 -0.1783  0.0090  0.1521  0.8847

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.424025   0.254837  -1.664   0.0983 .
section_2    0.055280   0.027416   2.016   0.0457 *
section_1   -0.004888   0.016125  -0.303   0.7622
section_3   -0.025892   0.018634  -1.390   0.1668
section_5    0.032453   0.020530   1.581   0.1162
section_8    0.019598   0.016877   1.161   0.2475
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3023 on 142 degrees of freedom
(39 observations deleted due to missingness)
Multiple R-squared:  0.09434,    Adjusted R-squared:  0.06245
F-statistic: 2.958 on 5 and 142 DF,  p-value: 0.01429

> anova(avg_info_vs_survey_section)
Analysis of Variance Table

Response: average_transformed_info_added
      Df Sum Sq Mean Sq F value    Pr(>F)
section_2  1  0.8870  0.88698   9.7031 0.002226 **
section_1  1  0.0046  0.00456   0.0499 0.823545
section_3  1  0.0899  0.08987   0.9831 0.323117
section_5  1  0.2474  0.24741   2.7065 0.102150
section_8  1  0.1233  0.12326   1.3483 0.247517
Residuals 142 12.9806  0.09141
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figure 15: R output for linear regression of best section model on info gain.

Consider the number of questions in each section, and taking them in the order from the R output, a forecaster with the best possible survey score would have an estimated cube-root-gain of:

$$-0.424 + 0.055 \times 4 - 0.005 \times 0 - 0.026 \times 0 + 0.032 \times 13 + 0.0296 \times 8 \approx 0.2$$

This corresponds to an average gain of $0.2^3 = .008$ bits per edit for acing the survey questions, compared to a default (intercept) gain of $-0.424^3 = -0.076$ bits per edit for getting no points on the positive-coefficient survey sections, but at least losing no points on the negative-coefficient ones (motivation and, surprisingly, eCRT). As we see in the next section, the Top5 and Top10 Super-Users by market score have total survey scores considerably higher than the average, suggesting this regression may indeed help to pick out the best of the best.

In a similar analysis where each trade was a data point and the Brier Difference was used, the Berlin test likewise stood out, and even yielded a nice symmetric linear regression showing those scoring <2 had negative

gain and those >2 had positive gain (Figure 16). However, as the marginal boxplots suggest, that regression may have been overly sensitive to non-normal residuals.

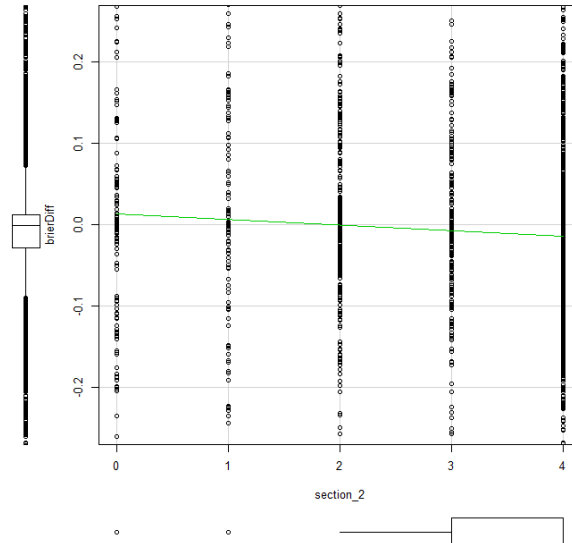


Figure 16: Dubious trade regression on Berlin numeracy test (Section 2) using untransformed Brier Difference on all trades individually (rather than one average per user). The main effect suggests that then average trades from forecasters with low Berlin numeracy hurts the Brier by 0.1, while 10 average trades from those with high Berlin numeracy helps by 0.1. This is a very strong effect, but is suspect for two reasons: (1) the distribution of trades heavily weights the contributions of our top forecasters to the regression, and (2) the marginal on the Y-axis shows very heavy tails, violating the assumed Gaussian error model. However, the Berlin test was also identified when the Y-axis was transformed to provide normal residuals.

Looking within the Berlin test, a hierarchical analysis was performed on each of the four questions. The best model included all of them, but in that model, Question 30 had the largest coefficients and best p values. The question is:

In a forest 20% of mushrooms are red, 50% brown and 30% white. A red mushroom is poisonous with a probability of 20%. A mushroom that is not red is poisonous with a probability of 5%. What is the probability that a poisonous mushroom in the forest is red? Please indicate the probability as a decimal between 0 and 1.

The correct answer is 0.5. The other questions are similar in nature.

Super-Users

It is also interesting to examine the total survey scores of our super-users. The obvious way to define “super-user” is to use the market leaderboard. Points are earned for giving information, and the leaders are the ones who have given us the most information. Their information makes the market accurate. Their strategy will be a mix of moving things to the “right” probability (when they have enough points to do so) and correcting what

is “most wrong” (which as noted above, also has the biggest bang per buck on our accuracy.) But how to define “Super”?

Note: the analyses in this section consider the 67K trades for those users active in the last 3 months, which is our default rule for showing up on the active leaderboard.

Evaluating them on bits/edit – how much on average their edits tended to contribute to correct answers, various regressions assign positive values (good) to the super-users and negative values to everyone else. The best R^2 and largest difference comes from setting the cutoff for “Supers” at the Top200 on the leaderboard. In that case, the average Super edit contributes +0.0084 bits (moving in the right direction), while the average non-super edit contributes -0.0006 bits/edit (moving in the wrong direction).

So, 200 is the best cut-point for overall R^2 and coefficient size. But for some purposes a more selective cut may be better. As noted above, the survey-score of the best super-users climbs substantially above background. Figure 17 shows how the Top5 and Top10 substantially exceed baseline, but more lenient thresholds remove the ability of survey score to discriminate a “super” from a non-super.

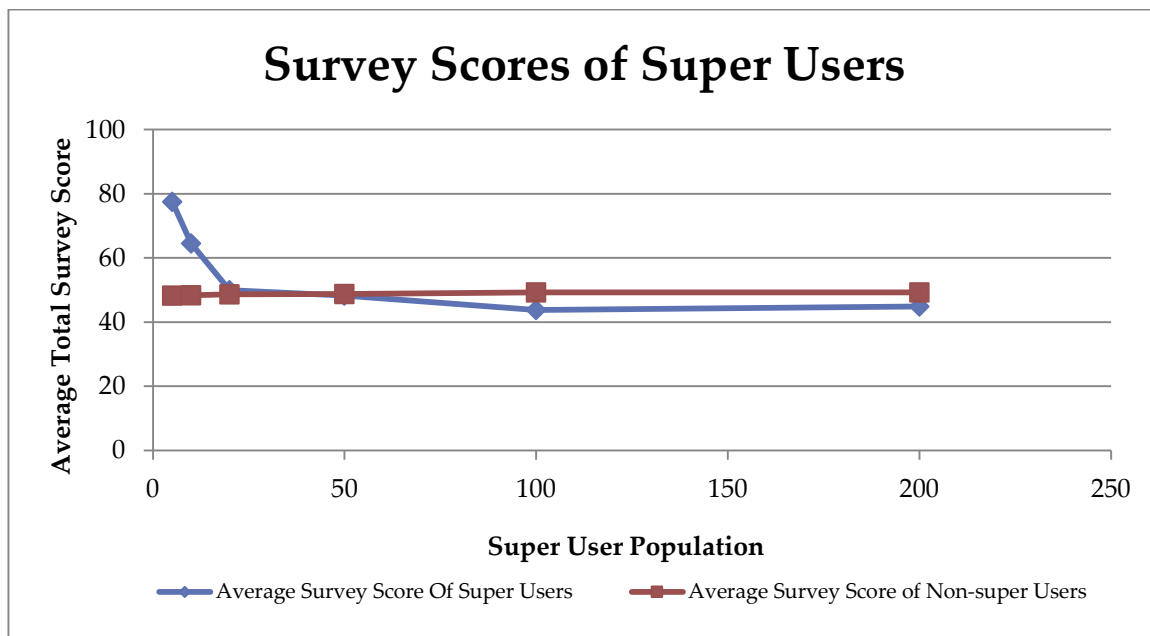


Figure 17: Survey scores of super-users

We can also select for raw accuracy of the edits, putting high probability on actual outcomes and low probability on non-outcomes. For this measure we have to remove people who only make 1-2 forecasts, because their accuracy measure is just a coin toss. We drop those with fewer than 5 edits.

There is a power-law distribution of activity and points, with most users making few edits, and a few making very many indeed. About 5,000 users have ever made an edit, and most of them have made only 1-2 and then disappeared. Restricting ourselves to those on the most recent leaderboard who have also made ≥ 10 edits

leaves about 300 forecasters. Figure 18 shows the function of average accuracy by rank, where $\text{Acc} = 2 - \text{Brier}$, and so ranges from 0..2 with 2 best and 1.5 chance for binary questions. Given that we required at least 10 edits, it is actually remarkable there are any forecasters so close to 2.0. These are either very lucky, or represent the rare forecaster playing a pure “subject matter expert” strategy of making a few good extreme edits. Of the 300 who qualify, about 275 are better than chance on this measure.

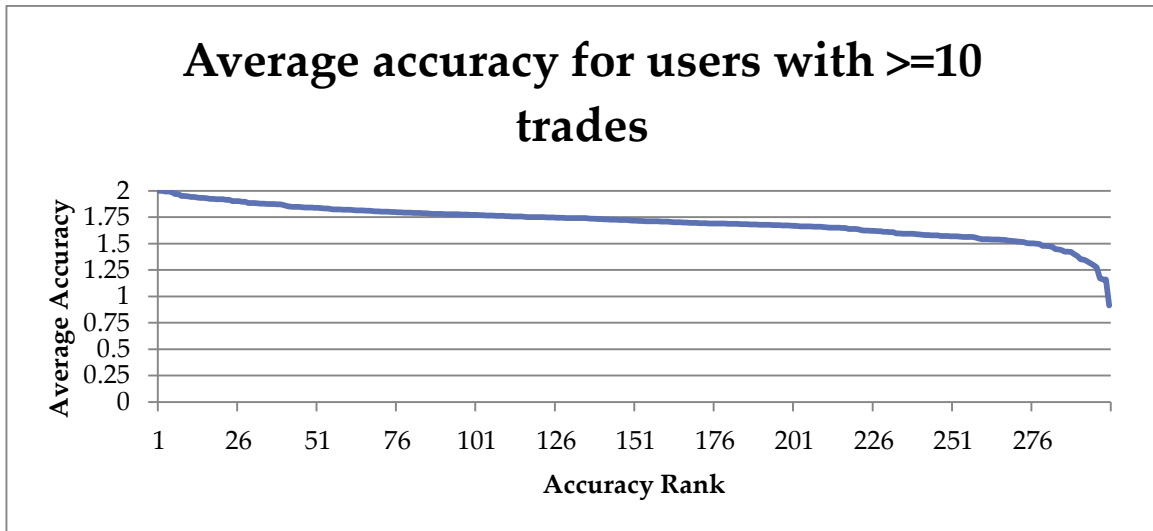


Figure 18: Raw Accuracy x Rank. "Accuracy" is just $(2 - \text{Brier})$, so 2 is perfect, 1.5 is binary chance, and 0 is the worst.

Data Mining Analysis

We have only begun to look at nonparametric analyses of the forecaster data. Note that here we are analyzing Brier gain, so negative parameters mean higher accuracy.

A regression tree of raw Brier difference on sections 1..9 splits on Section 1 (eCRT, 15 Qs), then Section 7 (AOMT, 9 Qs), and finally Section 2 (Berlin, 4 Qs). Figure 19 shows the tree. The R^2 for predicting an individual trade is, of course, poor. However, the average effects suggest:

1. eCRT > 13.5 (out of 15) yields average Brier gain of 0.15 per 10 trades
2. Else AOMT < 8.17 (out of 9) yields small loss of .008 per 10 trades
3. Else (very high AOMT) :
 - a. Berlin > 1.5 yields average loss of .34 per 10 trades
 - b. Berlin < 1.5 (very low) yields incredible loss of 0.24 per single trade

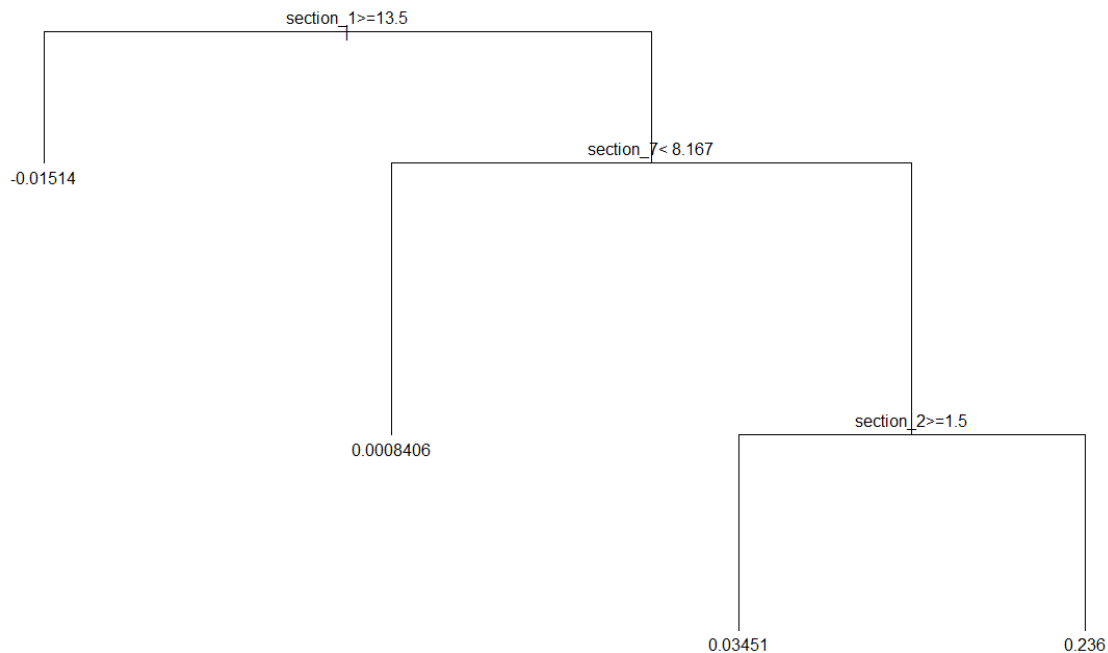


Figure 19: rpart regression tree for BD on survey sections 1..9. Section 1 is the eCRT (15 Qs), 7 is the AOMT (9 Qs), and 2 is the Berlin numeracy (4 Qs).

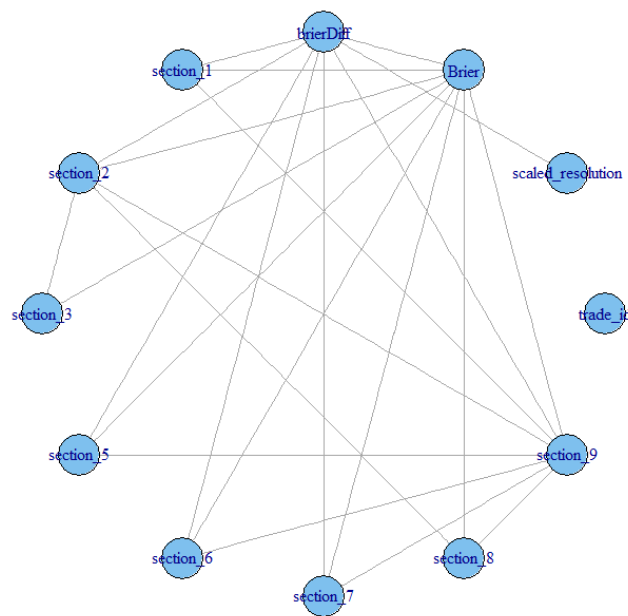
Using a decision tree on the survey questions themselves, rather than the sections, selects Q.30, splitting on “>0” meaning a correct response. Q.30 is the fourth (last) question in the Berlin numeracy test, and was selected above by the hierarchical regression as the best question in the best section. The decision tree suggests getting the question right corresponds to an average Brier gain of 0.016, while getting it wrong corresponds to an average loss of 0.004.



Figure 20: Decision tree chosen from all survey questions individually using the “ctree” procedure.

The following is a very preliminary search of the space of structural equation models (Gaussian Bayes nets) using the BDgraph library.

Graph with highest probability



Posterior probability = 0.4766

Figure 21: A single preliminary run of a structure search in the space of Gaussian graphical models.

The search linked Scaled Resolution and Brier Difference as expected, and likewise Brier Difference and Brier. Brier Difference is also linked to most sections (except 3 & 8). Brier is linked to all sections. The absolute posterior probability is not to be trusted, but this graph had twice the posterior probability of its nearest competitor. These results are at best suggestive, as a Gaussian graph model can be misled by heavy-tailed residuals at least as much as regular regression.

Incentives for Crowdsourcing Forecasts

Background

Four experiments have been run within SciCast this year to test the effects of various kinds of incentives on both activity and accuracy. The first two were designed in Y3 and run in the summer of 2014. The third ran from 7-NOV-2014 to 7-MAR-2015. The fourth ran April-May 2015.

First Two Experiments: Activity & Accuracy Incentives

The first experiment, planned in Y3, lasted for four weeks, May 26 to June 20, and was focused on incentives for activity, defined as either forecasts or comments. We announced different rewards for activity on different days of the week. On Tuesdays and Fridays, sixty valid activities that day were randomly selected to be given a \$25 Amazon Gift Card. On Wednesdays and Fridays, the same process was used to select users to gain a special badge visible on their profile. On Mondays and Thursdays, random activities were selected to receive a private “thank you” message. Users were not told that thank yous might be coming.

Four weeks after the end of the first experiment, a second four-week experiment began, lasting from July 20 to August 15. Eighty random activities on Tuesday were again selected for a \$25 card. Thursday trades were scored for accuracy against a later market price, and eighty cards were given randomly in proportional to trade accuracy. For twenty of these cards the market price the next day was used. For the other sixty cards, the market price at the end of the experiment was used. Using the same selection rules, Wednesday activities were awarded activity badges, and Friday trades were awarded accuracy badges. Random trades on Mondays were selected for private thank yous.

Figure 22 shows activity as a function of time during these first two experiments. The large spikes correspond to activity increases on the days when activity was rewarded with gift cards.

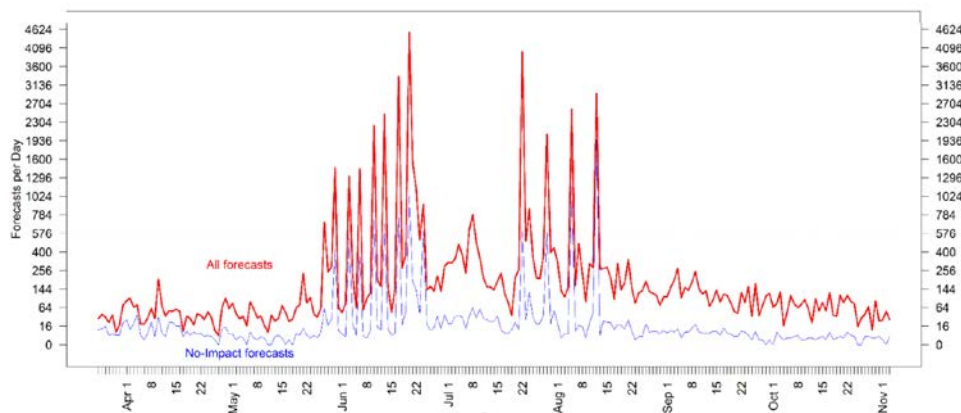


Figure 22: Activity as a function of time during the first experiment.

Table 7 shows the correlations between expected Brier score and activity or accuracy incentives, during the experiment. As noted, “Act” denotes activity incentives and “Acc” denotes accuracy incentives. “Cards1” and

“Cards2” denote having won gift cards in rounds 1 or 2 respectively. “Cards*: Cards Offer” denotes that a participant won cards in round *, but had not yet hit their \$575 limit, so they were eligible to receive still more cards. “Thanks” denotes thank-you notes. “Merits” denotes the badges and merits added to a user’s profile page (admittedly not very visible).

Table 7: December 2014 SciCast participant correlations with Brier score. Negative correlations mean more accuracy. See the text for definitions.

Variable	Coef.	Extra r^2	Variable	Coef.	P-val.	Extra r^2
Act. Cards1	0.04	0.026	Acc. Cards	-0.001	0.051	0.001
Act. Cards2	-0.24	0.003	Acc Cards: Cards Offer	0.001	0.485	0.000
Act. Cards1: Cards Offer	12.27	0.147	Acc. Thanks	0.019	0.027	0.002
Act. Cards2: Cards Offer	31.38	0.164	Acc. Merits	0.002	0.031	0.002
Act. Thanks	0.77	0.002	Act. Cards1	-0.003	0.202	0.001
Act. Merits	0.24	0.004	Act. Cards2	0.001	0.630	0.000
Acc. Cards	-0.03	0.000	Act. Cards2: Cards Offer	-0.002	0.272	0.000
Acc. Merits	0.16	0.000	Days Reg'd	-0.001	0.000	0.030
P<.004 in all rows because N=254K 12/17/2014			Act. = Activity; Acc. = Accuracy)			

Statistical (ordinary least squares) regression analysis on the first two experiments found that the best predictor of activity, by far, was an interaction effect: traders who had previously won a gift card were more active on days when activity incentives were offered. This effect explained 31% of the variance in daily activity.

The same sort of statistical analysis seeking predictors of trade *accuracy* in the first two experiments, however, found only rather weak effects. The strongest predictor was the number of days that a user had been registered. More experienced users were more accurate, but this variable explained less than 3% of the variance in trade accuracy. Other variables explained far less.

Due to (anticipated) time limits, in the second experiment, “accuracy” was estimated using the expected value of a question a few days to weeks after each trade. Because most questions take months to resolve, even well-informed users should expect that the accuracy of their trades as measured a few days later will be quite random. Because of this, these rewards for “accuracy” were arguably mostly rewards for activity.

Third Experiment: Four-month Accuracy

Given the weak results on accuracy in the second experiment, a third experiment was designed to give stronger accuracy incentives. This experiment ran four months, from November 7, 2014 to March 6, 2015. In this experiment, users were only rewarded for accuracy, and accuracy was only measured against prices at the end of this experiment. In addition, incentives were concentrated more among top users. The fifteen most



accurate users were each given \$2250 to spend at Amazon.com, and the next 135 most accurate users were each given \$225.

Out of the roughly 600 SciCast questions open near November 7th, 366 questions were selected for inclusion in this third experiment. A random half of these selected questions were designated Set A, with the other half being Set B. Over the four months of the experiment these sets alternated in whether they were eligible or not for the accuracy rewards. In the first month set A was eligible, while set B was not. In the second month, set B was eligible while set A was not. This alternation continued for all four months. Thus a user's accuracy score for the purpose of rewards in this experiment counted the accuracy of their trades in set A made during the first and third months, and their trades in set B during the second and fourth months.

Analysis of the data from this experiment was divided into analyses of activity and analyses of accuracy. In the analyses of activity, the dependent variable to predict is the number of edits (i.e., trades) per day per question. In the analysis of accuracy, the dependent variable to predict is the (Shannon) information gain per edit (i.e., trade). The total effects of the experimental treatment of increased rewards on information gained per question per day is then divided into effects in inducing more trades, and effects in changing the information per trade.

Because we use standard statistical techniques that assume normal (Gaussian) distributions, we sought to transform our variables of interest into versions that are more normally distributed. For example, there are great many day and question combinations where there are zero edits, and yet there is never a negative number of edits. We thus estimated the number of edits per question per day via a Poisson regression. Such a regression gives the probability of non-negative integers being chosen as a function of an expected value that is exponential in a linear combination of independent variables and coefficients. Such a regression thus in effect predicts the expected logarithm of the number of edits per question per day.

The independent variables for the accuracy and activity regressions are:

- `EligibleForRewards` – 1 indicates question was eligible for rewards on date of edit; 0 indicates question was not eligible for reward on date of edit.
- `DaysToResolution` – Number of days from date of edit until the day question resolved.
- `DaysSinceCreation` – Number of days from the date question went live on SciCast until edit was made.
- `Popularity` - Number of edits prior to the experiment.
- `OrderedMultiValue` – 1 indicates question has more than two possible answers and answers were ordered; 0 indicates other type of question.
- `UnorderedMultiValue` - 1 indicates question has more than two possible unordered answers; 0 indicates other type of question.
- `Scaled` - 1 indicates scaled continuous question; 0 indicates other type of question.
- `Shares` – 1 indicates a “shares” style question such as the market share of the Top500 by vendor or region; 0 indicates other type of question

- Month2 - 1 indicates edit was made during second month of 4-month experimental period; 0 indicates edit was made in first, third or fourth month.
- Month3 - 1 indicates edit was made during third month of 4-month experimental period; 0 indicates edit was made in first, second or fourth month.
- Month4 - 1 indicates edit was made during fourth month of 4-month experimental period; 0 indicates edit was made in first, second or third month.

The (Shannon) information gain per edit is the information about the final question answer contained in the system prices immediately after the edit, minus the information contained in the prices immediately before the edit. This distribution has many negative values, and is peaked near (but not necessarily exactly at) zero. In our data, this variable also happens to have very thick tails relative to a normal distribution.

Deviations from normality can be seen via the diagnostic tool of the Quantile-Quantile (Q-Q) plot, which for normally distributed data should look like a straight line. The following two Q-Q plots (Figure 23) are for the untransformed information gain distribution, and for these gains transformed by taking their cube roots:

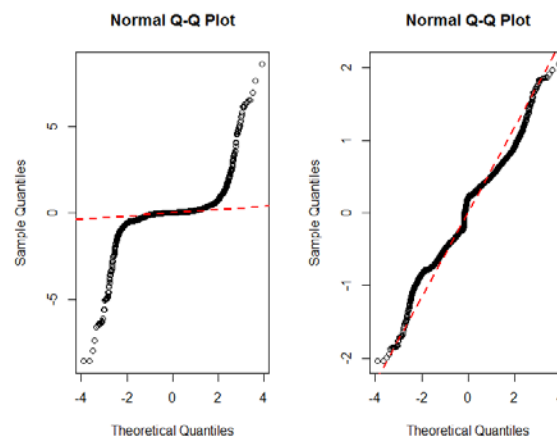


Figure 23: Q-Q plots for untransformed information gain (left) and cube-root information gain (right) during the Y4 accuracy incentives experiment.

A quick visual inspection makes it clear that the transformed version is much closer to a normal distribution. In fact, compared to other possible power-law transformations, taking the power of $1/3$ seems to bring our data distribution the closest to normality. Thus we did our accuracy statistical analysis with the cube-root of info gain per trade as the dependent variable.

Regarding activity, the following Poisson regression estimates the logarithm of the expected edits per day per question during the time period of the experiment, for the questions included in the experiment:

	Estimate	Std. Error	z value	Pr(> z)	
Intercept	-5.017e-01	2.303e-02	-21.782	< 2e-16	***
EligibleForRewards	1.061e+00	1.596e-02	66.505	< 2e-16	***
DaysToResolution	-1.624e-03	5.231e-05	-31.048	< 2e-16	***



DaysSinceCreation	-5.856e-04	7.916e-05	-7.398	1.39e-13	***
Popularity	1.014e-03	2.751e-05	36.878	< 2e-16	***
OrderedMultiValue	8.782e-02	1.665e-02	5.273	1.34e-07	***
UnorderedMultiValue	1.254e-01	2.948e-02	4.253	2.11e-05	***
Scaled	-4.733e-01	2.752e-02	-17.195	< 2e-16	***
Shares	3.607e-01	1.633e-01	2.209	0.027197	*
Month2	-1.911e-01	1.719e-02	-11.121	< 2e-16	***
Month3	-7.015e-02	1.983e-02	-3.537	0.000404	***
Month4	-3.767e-01	2.360e-02	-15.963	< 2e-16	***

Significance codes: *** = 0.001, ** = 0.01, * = 0.05, . = 0.1

(All regressions reported in this section will use these same significance codes.) As one can see, almost all of these variables are very strongly significant. Questions eligible for rewards get more edits. There are more edits soon after a question is created, and shortly before the question will resolve. More popular questions (judged before the experiment) have more edits during the experiment, and the questions with more possible values also get more edits. Finally, the rate of edits was highest in the first month, and declined in later months.

Regarding accuracy, the following regression estimates the cube-root of info gain per edit during the time period of the experiment, and within the questions that were included in the experiment:

	Estimate	Std. Error	t value	Pr(> t)	
Intercept	-1.517e-02	2.150e-02	-0.706	0.480488	
EligibleForRewards	4.078e-02	1.166e-02	3.496	0.000473	***
DaysToResolution	-2.043e-04	1.773e-04	-1.152	0.249320	
DaysSinceCreation	-8.909e-05	6.134e-05	-1.452	0.146432	
Popularity	4.303e-05	4.599e-05	0.936	0.349463	
OrderedMultiValue	-5.027e-02	1.340e-02	-3.752	0.000176	***
UnorderedMultiValue	-1.563e-01	2.031e-02	-7.700	1.48e-14	***
Scaled	-3.854e-02	2.743e-02	-1.405	0.160070	
Shares	-8.696e-02	8.650e-02	-1.005	0.314779	
SafeModeEdit	-1.371e-03	1.087e-02	-0.126	0.899625	
NumUserTrades	-5.040e-06	3.558e-06	-1.417	0.156630	
UserScoreAtStart	3.735e-06	6.936e-07	5.385	7.39e-08	***
Month2	4.000e-02	1.182e-02	3.384	0.000718	***
Month3	-2.683e-02	1.665e-02	-1.612	0.107036	
Month4	2.819e-02	2.067e-02	1.364	0.172665	

As one can see, edits that were eligible for rewards in the experiment added significantly more information. Also, users with higher initial scores (20,000 vs. 5,000 yields .075 vs. .019) made more informative edits, and edits of multiple valued questions were less informative. For reasons that are not clear, edits made during the second month were also more informative.

For comparison we ran a similar regression on the questions that were *not* included in the experiment, over the same time period:

	Estimate	Std. Error	t value	Pr(> t)
Intercept	3.977e-02	5.689e-02	0.699	0.4846
DaysToResolution	2.569e-04	5.388e-04	0.477	0.6335
DaysSinceCreation	-3.798e-04	3.262e-04	-1.164	0.2444
Popularity	1.458e-06	1.087e-04	0.013	0.9893
OrderedMultiValue	-9.782e-03	3.265e-02	-0.300	0.7645
UnorderedMultiValue	-1.140e-01	5.526e-02	-2.063	0.0392 *
Scaled	-4.743e-02	6.116e-02	-0.776	0.4381
SafeModeEdit	2.350e-02	2.834e-02	0.829	0.4072
NumUserTrades	-8.455e-06	9.272e-06	-0.912	0.3619
UserScoreAtStart	4.596e-06	2.002e-06	2.295	0.0218 *
Month2	-7.242e-03	4.048e-02	-0.179	0.8580
Month3	5.259e-02	4.408e-02	1.193	0.2330
Month4	1.014e-02	4.046e-02	0.251	0.8022

Here there are only two marginally significant effects, but they are consistent with the other results. This lack of significance is understandable as due to this set being much smaller, and due to this set differing in more unknown and uncontrolled ways from the included variables.

To see the overall information effect of our experimental treatment, we look at small regressions with only three variables, a unit constant and two dummy variables. These three variables cover the three mutually exclusive possibilities of questions that are out of the experiment, in the experiment but not currently eligible for rewards, and in the experiment and also currently eligible for rewards.

The following table (Table 8) takes the estimates from these small regressions and transforms their point estimates into direct estimates of edits per day, and of info gain per edit. We normalize these effects so that estimates for questions not in the experiment are set to have unit value. This gives us estimates of the relative effect on activity and accuracy that result from being in the experiment and eligible for rewards, and from being in the experiment and not eligible.

Table 8: Overall regression for Y4 accuracy incentives study

	Relative Edits per Day	Relative Info Gain Per Edit	Product	Info Gain Signif.	Edits Signif.
Not in Experiment	1	1	1	1.60E-11	<1e-16
In Exp, Not Eligible	2.93	0.0035	0.0103	1.30E-05	<1e-16

In Exp, Eligible	8.41	0.0518	0.4352	1.30E-04	<1e-16
---------------------	------	--------	--------	----------	--------

The table shows that these estimates are all strongly significant, that there is much more activity for questions in the experiment, and even more activity for questions that are eligible for rewards. On accuracy, we see that there is much more info gain for eligible questions, relative to ineligible questions, in the experiment. However, the per edit info gain is estimated to be even larger for questions outside of the experiment, and the net effect of the activity and accuracy effects is to give more total info gain per day for questions outside the experiment. Since the non-experimental questions are few in number, and differ in more unknown and uncontrolled ways, we place less trust in the comparison with questions out of the experiment. We do know they were out of the experiment because they were unlikely to resolve by 1-April. Therefore one plausible reading is that these far-term questions were only likely to be edited when there was a substantial news event, and that such edits were generally reliable.

Target Journals and Conference

Preliminary results from this experiment were included in our recent submission to *International Journal of Forecasting*, Special issue on Elicitation and Aggregation.

Incentives for Combinatorial Edits

From April 23 to May 22, 2015, we ran a short fourth experiment. \$16,000 was awarded to the users whose edits over this period were the most accurate overall. To be eligible, however, users had to have one quarter of their edits be combinatorial. Our hope was to analyze link creation behavior in this context. The following table (Table 9) gives some basic statistics comparing behavior in the thirty days prior to this experiment (during which time there was no experiment, and so no financial incentives).

Table 9: Activity statistics for combo edits contest

	Daily Count			Avg. Max Points at Risk on Day's Trades		
	Total Edits	Combo Edits	New Links	Total Edits	Combo Edits	New Links
30 Day Pre-Contest Ave	82.4	7	0.5	-46.2	-66.3	-84.7
Contest Daily Average	137.2	66.6	7.9	-42.2	-104.8	-74
% of Pre-Contest Ave	166%	952%	1573%	91%	158%	87%

The incentives of this experiment clearly increased overall activity, and also combinatorial edits, since such edits were required to be eligible for rewards. Notably, even though new links were not required for eligibility, and even though they require more work from users, the rate of new links increased even faster than did the rate of combinatorial edits. This offers weak evidence that the ability to add new links is actually helpful to users in making informative combinatorial edits.

Statistics on the average points at risk in related trades were collected in order to assess whether users were making very small combinatorial trades just to fill out their required one quarter of combinatorial trades. These statistics suggest that such a strategy was not common.

Figure 24 clearly shows the effect of the contest on the number of conditional edits, but it hides the number of new links because almost all new links through May 15 involved questions that already had another link. Table 10 shows the top ten link strengths (mutual information) on the market at the end of Y4. The strongest link is between accidental duplicates of the 9-outcome question on the DARPA grand challenge. There were 150 links with mutual information greater than 0.01, out of 340 total links of which 209 remain open.

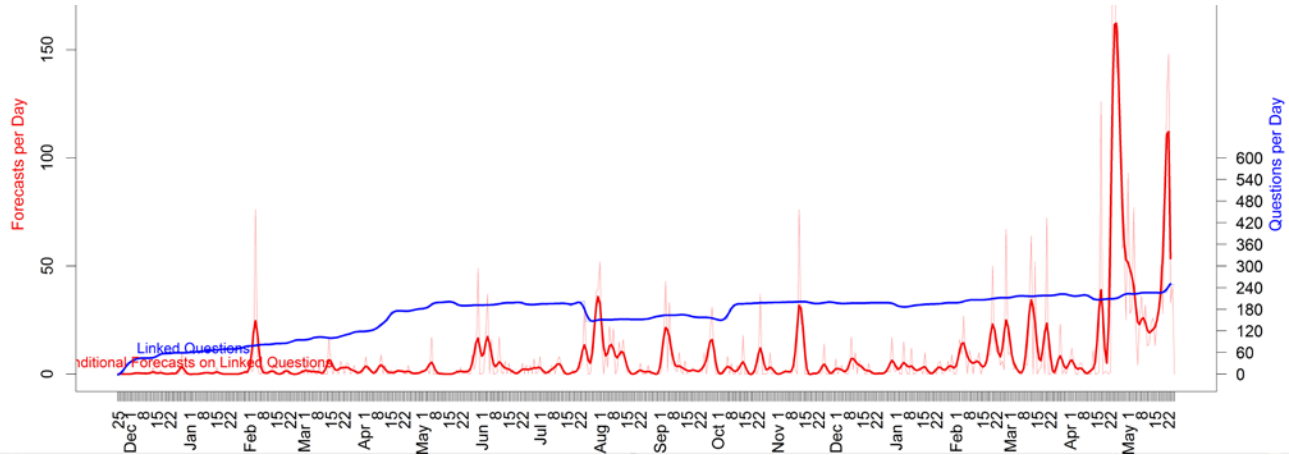


Figure 24: Conditional edits over time, showing the effect of the combinatorial edits contest. The contest also added over 70 links, but most of them between questions that already had links.

Table 10: Top-10 link strengths at the end of Y4. The strongest is for a duplicate question with 9 outcomes.

PARENT	CHILD	LINK STRENGTH
140	369	1.8610523
1258	1295	0.63389534
133	771	0.61535263
132	772	0.5443256
1224	1221	0.509771
721	629	0.49980426
28	71	0.48641908
1351	1352	0.43619958
1076	1077	0.42296323
1373	1372	0.42162958

Asset Management

Background

A combinatorial prediction market needs to maintain a single consensus joint probability distribution, and for each user, a joint asset distribution, where in both cases “joint” means the combined state space of all variables in all allowed combinations. For both DAGGRE and SciCast, the joint probability space is represented as a probabilistic graphical model, which is compiled into a junction tree for efficient inference. User edits are incorporated by entering soft evidence and updating beliefs via the junction tree algorithm. We showed (Sun, Hanson, Laskey, & Twardy, 2012) that assets factor according to the same decomposition as the probabilities, and asset updating could be performed by defining a parallel asset junction tree for each user and modifying the junction tree algorithm to find the user’s minimum and expected assets. This method, which we call Common Junction Tree (CJT), was implemented in the DAGGRE market.

Because most users will trade sparsely relative to the total number of questions, and even more sparsely compared to the whole joint probability space, the approach taken by DAGGRE is highly space inefficient. Further, it creates a synchronization problem, as any change to the probability structure (adding or resolving a question, adding a link) must be mirrored across thousands of users’ asset structures. In our DAGGRE implementation, changes required re-running the entire trade history. In addition, because each user’s asset management computations in the DAGGRE implementation are exponential in the size of the largest clique in the global junction tree whether or not the user trades in this largest clique, asset computations are highly inefficient when, as is typical, users trade sparsely relative to the entire question space. Finally, loading and unloading large junction trees for many users triggered frequent garbage collection so that the system runtime performance was limited more by system memory maintenance than by our inference algorithm.

SciCast’s tenfold increase in both questions and users made the DAGGRE approach infeasible. Substantial gains were achieved by modifying the asset representation to a user-specific asset structure constructed from each user’s trades. These trade-based asset models divorce the probability and asset structures. A basic asset unit called asset block containing only the traded questions is created for each user and updated after each trade. The approach, called the Dynamic Asset Cluster (DAC) model, dynamically builds a user-specific asset junction tree optimized to the user’s trades (Sun et al, 2014). At its launch in November 2103, SciCast featured a simplified version of DAC called Global Separator (GS), which groups all overlaps into a single cluster. This method is more efficient than DAC in the typical case in which asset blocks have few overlaps, but GS must resort to approximation as the overlap becomes larger. The general DAC model was implemented in an offline Matlab version.

Objectives

The aim of this study is to compare the dynamic asset cluster (DAC) model with the DAGGRE common junction tree (CJT) method and the current SciCast global separator (GS) method to analyze the space and time characteristics of each as a function of characteristics of the joint probability structure and of the distribution of trades for each user.

Asset Models

Consider a market-maker based combinatorial prediction market with base events of the form $V_k = v_k$, where $\mathbf{V} = (V_1, \dots, V_n)$ is an n -dimensional random variable with component $V_{n=k}$ having $n_k < \infty$ mutually exclusive and collectively exhaustive outcomes.² The market maker needs to manage a consistent consensus distribution $p_{\mathbf{v}} = p(\mathbf{V} = \mathbf{v})$ over joint states \mathbf{v} of \mathbf{V} . At any time, each participating user u has assets $a_{\mathbf{v}}^u \geq 0$ for each possible joint state \mathbf{v} . A user who disagrees with the current consensus distribution $p_{\mathbf{v}}$ changes it by making edits that can be covered by her current assets. For example, she should be able to set the conditional distribution $p(T | \mathbf{H} = \mathbf{h})$ to a new value $x(T | \mathbf{H} = \mathbf{h})$. In our logarithmic market scoring rule (LMSR) market, this edit results in changing u 's assets in state $\mathbf{v} = (t, \mathbf{h}, \mathbf{w})$ from $a_{\mathbf{v}}^u$ to

$$a_{\mathbf{v}}^u + b \ln \frac{x(t | \mathbf{H} = \mathbf{h})}{p(t | \mathbf{H} = \mathbf{h})}, \quad (\text{A-1})$$

where \mathbf{W} denotes the market variables other than T and \mathbf{H} .

Asset management requires data structures and algorithms to perform the following tasks:

- *Minimum assets.* Find the cash, or minimum assets, for user u : $a_{\min}^u = \min_{\mathbf{v}} \{a_{\mathbf{v}}^u\}$ for user u . No trade can be allowed if the user's post-trade assets can become negative in some state. The user's cash a_{\min}^u is the amount of assets available to u to invest in trades.
- *Edit limits.* Find upper and lower limits on $x(T | \mathbf{H} = \mathbf{h})$ such that an edit changing $p(T | \mathbf{H} = \mathbf{h})$ to $x(T | \mathbf{H} = \mathbf{h})$ will not make u 's assets $a_{\mathbf{v}}^u$ less than zero for any state \mathbf{v} . That is, for trades within the edit limits, u 's post-trade cash is greater than or equal to zero.
- *Expected assets.* Find the current expected user assets $\sum_{\mathbf{v}} p_{\mathbf{v}} a_{\mathbf{v}}^u$ for user u .

DAGGRE Asset Model

The DAGGRE asset model, which we call parallel junction tree (PJT), gives each user an asset junction tree of the same structure as junction tree representing the market joint probability distribution. Each user's asset junction tree is initialized to represent constant initial assets in every state \mathbf{v} . When user u makes an edit, u 's asset junction tree is updated to reflect the consequences of the edit. Specifically, an edit changing

² We use capital letters for random variables, lowercase letters for variable values, and bold letters for vectors.

$p(T | \mathbf{H} = \mathbf{h})$ to $x(T | \mathbf{H} = \mathbf{h})$ changes u 's assets according to (A-1). It is convenient to define a transformation $q^u(\mathbf{v})$ of the assets $\alpha_{\mathbf{v}}^u$, such that

$$\alpha_{\mathbf{v}}^u = b \ln(q^u(\mathbf{v})) \quad (\text{A-2})$$

We then have

$$\frac{q'^u(\mathbf{v})}{q^u(\mathbf{v})} = \frac{x(\mathbf{v})}{p(\mathbf{v})} \quad (\text{A-3})$$

where $q'^u(\mathbf{v})$ is the updated asset for joint state \mathbf{v} , corresponding to the probability change from $p(\mathbf{x})$ to $x(\mathbf{x})$.

Because q_c^u starts out independent of the state and changes in proportion to changes in p , we can decompose q^u in a similar manner to the factored decomposition of p . Specifically,

$$q^u(\mathbf{v}) = \frac{\prod_{c \in \mathbb{C}} q_c^u(\mathbf{v}_c)}{\prod_{s \in \mathbb{S}} q_s^u(\mathbf{v}_s)}, \quad (\text{A-4})$$

where q_c^u and q_s^u are local asset components defined on the clique and separator variables, respectively. Notice the similarity to the factored representation of p . Note also that there is a separate asset structure representing the assets of each user u , and this asset structure has size equal to that of the common junction tree.

All edits are required to be *structure preserving*. This means all questions involved in the edit must belong to the same clique in the junction tree. That is, for an edit changing $p(T | \mathbf{H} = \mathbf{h})$ to $x(T | \mathbf{H} = \mathbf{h})$, T and \mathbf{H} must all be contained in at least one clique. A structure-preserving probability edit can be implemented by applying soft evidence to a clique containing T and \mathbf{H} . For structure-preserving edits, the asset updating rule (A-3) preserves the asset factorization (A-4).

Note that (A-4) has an asset table for each clique, of size equal to the cross-product of the state spaces for the questions in the clique. If the user has not traded in a clique, the asset table for that clique will have all its entries equal to a constant value. Thus, a naïve implementation of the DAGGRE asset management method takes up storage for and performs computations on large arrays of constants. This is highly inefficient in both time and space. Below we describe a more compact and efficient representation for user assets.

Trade-Based Asset Model

Our solution to the efficiency challenge is to continue using a global junction tree for the consensus probability distribution (using approximate inference if computational load becomes too heavy), but to divorce asset management from probability management. Specifically, a data structure called an *asset block* groups the

user's trades on a set of questions and represents the gains and losses from those trades. A set of asset blocks is a compact representation of the user's gains or losses in any joint state. The user-specific asset representation can be exploited for efficient calculation of expected and conditional minimum assets. Because the asset representation includes only the questions the user has edited, there are large resource savings when trades are sparse. The basic asset management approach was reported previously (Twardy, 2014; Sun, et al., 2014). This report describes enhancements to the Matlab implementation and efficiency studies performed subsequently.

Global Separator (GS) Model

The current SciCast implementation uses a simplified asset model called *global separator*, which groups all overlapping variables in any asset block into a single cluster called the global separator. All asset blocks are conditionally independent given the global separator. The GS model stores a base value b plus a set of asset blocks, each of which represents gains or losses from a set of trades, where the trades for each asset block are distinct.

Asset computation methods for GS are as follows:

- *Minimum assets.* To find the user's minimum assets over all joint states, proceed as follows. Iterate over all joint states \mathbf{s} of the global separator variables \mathbf{S} . For each \mathbf{s} , find the conditional minimum value in each asset block, given that the variables in \mathbf{S} have values \mathbf{s} . Then calculate the conditional minimum asset value given \mathbf{s} by adding the conditional minimum for all asset blocks to the base b . The global minimum asset value α_{\min}^u is the minimum over \mathbf{s} of the conditional asset minima. A straightforward modification gives conditional minimum assets given that variables \mathbf{H} are in state \mathbf{h} .
- *Edit limits.* Suppose user u wishes to make an edit changing $p_i = p(T = t | \mathbf{H} = \mathbf{h})$ to a new value $p^\#$. Let $\mathbf{m}_i[\mathbf{m}_{-i}]$ denote u 's current minimum assets in states where $\mathbf{H}=\mathbf{h}$ and $T=t$ [$T \neq t$]. Then the allowable edit range is $p(T = t | \mathbf{H} = \mathbf{h}) / \mathbf{m}_i \leq p^\# \leq (1 - p(T = t | \mathbf{H} = \mathbf{h}) / \mathbf{m}_{-i})$ (Sun, et al., 2012). These limits are straightforward to calculate given the output \mathbf{m}_i and \mathbf{m}_{-i} of the conditional minimum asset computation described above.
- *Expected assets.* To compute expected assets, simply iterate over all asset blocks, computing the expected gain or loss from each, add these values together, and add the result to the base b .

The GS method is the current implementation in the production SciCast system. The method is simple to implement, results in very large space savings when trades are sparse, and is very fast if the global separator is small. When the global separator becomes large, approximation can be used to bound computation time of the cash calculation, at the cost of under-estimating minimum assets, and preventing the user from fully allocating all their points.

Dynamic Asset Cluster (DAC) Model

Like GS, the Dynamic Asset Cluster (DAC) model maintains a collection of asset blocks constructed from the user's trades (Sun, Laskey, Twardy, Hanson, & Goldfedder, 2014; Twardy & Laskey, 2014). To avoid the need for approximation in the cash computation, DAC constructs an asset junction tree from the user's trades and

performs min-propagation on the user-specific asset junction tree. When there are few overlaps, this is less efficient than GS due to the need to construct a junction tree, but when there are more overlaps, DAC is much more efficient.

The basic data structure for DAC is a collection of *asset blocks*. An asset block $B = (\mathbf{V}_B, \tau_B)$ consists of block variables \mathbf{V}_B and an asset table τ_B that specifies a real number $\tau_B(\mathbf{v}_B)$ for each joint state \mathbf{v}_B of the block variables \mathbf{V}_B . The asset block B represents aggregate gains and losses from a set of trades involving variables in the block variables \mathbf{V}_B . When the user makes an edit changing $p(T=t | \mathbf{H}=\mathbf{h})$ to a new value $x(T=t | \mathbf{H}=\mathbf{h})$, DAC creates a trade-specific asset block $B^{p \rightarrow x}$ with block variables (T, \mathbf{H}) and asset table computed according to the market scoring rule (A-1). This block may be merged into another asset block, creating a new combined asset block B^{new} , or may be added to the collection of asset blocks. In the DAC version described in (Twardy & Laskey, 2014), a trade-specific asset block was combined with an existing asset block only when one was a subset of the other; otherwise, a new asset block was added for the trade. We found that this approach led to inefficient expected value calculation because of the large number of probability marginalization operations (one for each asset block). A more efficient approach can be found by noting that because edits are structure preserving, the asset block for any single trade is a subset of the variables in some clique, called its *host clique*. Our new version of DAC combines an asset block with the other asset blocks in its host clique, creating a new asset block only if a trade involves a clique in the probability junction tree with no previous trades.

Asset computation methods for DAC are as follows:

- *Minimum assets.* To find the user's minimum assets over all joint states, first construct a junction tree from the set of asset blocks. Then perform min-propagation in the junction tree to find the minimum asset value α_{\min}^u . A straightforward modification finds the conditional minimum given $\mathbf{H}=\mathbf{h}$ – set evidence $\mathbf{H}=\mathbf{h}$ and perform conditional min-propagation.
- *Edit limits.* Suppose user u wishes to make an edit changing $p_t = P(T=t | \mathbf{H}=\mathbf{h})$ to a new value $p^\#$. Let \mathbf{m}_t [\mathbf{m}_{-t}] denote u 's current minimum assets in states where $\mathbf{H}=\mathbf{h}$ and $T=t$ [$T \neq t$]. Then the allowable edit range is $p(T=t | \mathbf{H}=\mathbf{h}) / \mathbf{m}_t \leq p^\# \leq (1 - p(T=t | \mathbf{H}=\mathbf{h}) / \mathbf{m}_{-t})$ (Sun et al., 2012). These limits are straightforward to calculate given the output \mathbf{m}_t and \mathbf{m}_{-t} of the conditional minimum asset computation described above.
- *Expected assets.* First compute expected gain or loss for the trades in each asset block, then sum over asset blocks, and add this amount to the base b . The expected gain or loss for an asset block is calculated by marginalizing its host clique down to the variables in the asset block, multiplying probability times assets for each joint state in the asset block, and then summing over states.

Experiment Design

We performed experiments to evaluate performance of DAC against the DAGGRE PJT method.

The experiment evaluated 200 randomly generated trade histories with the design parameters as shown in Table 11.

Table 11: Design for Asset Management Experiment

Design Variable	Distribution
N - number of variables in market	uniform 15:30
nos - number of states in each node	uniform 2:4
maxtw - allowable max treewidth of BN	uniform 3:6
nofu - number of users	uniform 10:30
usrcl - number of cliques traded by each user	uniform 2:5
usrnv - max number of variables user trades in any given clique	uniform 2:maxtw
ntrades - total number of trades	200

Each run of the experiment was conducted as follows:

1. Generate a random Bayesian network with N variables having nos states each, with treewidth less than or equal to maxtw, using the method of (Ide & Cozman, 2002). Compile the Bayesian network into a junction tree. Let m be the number of cliques in the junction tree.
2. For $u = 1, \dots, \text{nofu}$
 - a. Select the cliques on which user u makes trades by choosing $\max(\text{usrcl}, m)$ cliques at random without replacement.
 - b. For each clique c on which user u makes trades, let k_c be the number of variables in the clique. Choose the variables within the clique on which u trades by choosing $\min(\text{usrnv}, k_c)$ variables at random from c .
 - c. Make an initial trade by user u by calling function *makeTrade* while collecting timing statistics.
3. For $k=(\text{nofu}+1), \dots, \text{ntrades}$
 - a. Choose user u at random from users $1, \dots, \text{nofu}$
 - b. Make trade by user u by calling function *makeTrade* while collecting timing statistics

Function *makeTrade*

- c. Choose a clique c at random from u 's traded cliques.
- d. Choose a target variable T and a hypothesis variable H at random from c .
- e. Choose a random state t of T and a random state h of H .
 - i. Find the edit limits for changing $p(T=t \mid H=h)$.
 - ii. Choose a random value $x(T=t \mid H=h)$ within the edit limits.
 - iii. Change $p(T=t \mid H=h)$ to $x(T=t \mid H=h)$ by asserting soft evidence in the junction tree. Update probabilities. Collect timing data on this operation.
 - iv. Update PJT asset structure for user u and find user u 's cash. Collect timing data on this operation.
 - v. Update DAC asset structure for user u and find user u 's cash. Collect timing data on this operation.



- vi. Calculate expected assets for all users using PJT asset structure. Collect timing data on this operation.
- vii. Calculate expected assets for all users using DAC asset structure. Collect timing data on this operation.

Each run of the experiment therefore collected timing statistics for five operations:

- `edit` – Sum over all trades of time to implement the probability change and update probabilities in the junction tree.
- `PJT_cashup` – Sum over all trades of time to update the trading user’s asset structure and find trading user’s cash in PJT method.
- `DAC_cashup` – Sum over all trades of time to update the trading user’s asset structure and find trading user’s cash in DAC method.
- `PJT_scoreEV` – Sum over all trades of time to calculate expected score for all users in PJT method.
- `DAC_scoreEV` – Sum over all trades of time to calculate expected score for all users in PJT method.

Analysis and Results

Figure 25 shows a scatterplot matrix of total times for the five operations (`edit`, `cash/update` for DAC and PJT, `expected score` for DAC and PJT) for 200 runs of the experiment. Note the almost perfect correlation between time to implement a probability edit and time to calculate cash using the PJT method. This is as expected because both methods use the same junction tree. Note also the extreme skewness of the `edit` and `cash/asset update` time distributions. To address this, we do a logarithmic transformation of these variables; see Figure 26. When working with logarithms, we see a positive and roughly linear relationship between PJT `cash/update` and DAC `cash/update`, which was obscured by the extreme skew in the untransformed distributions.

Table 12 summarizes the log-transformed timing statistics. Note that the cash computation times for PJT are considerably smaller than the cash computation times for DAC, but both of these are smaller than the `edit` time, which includes belief propagation in the junction tree.

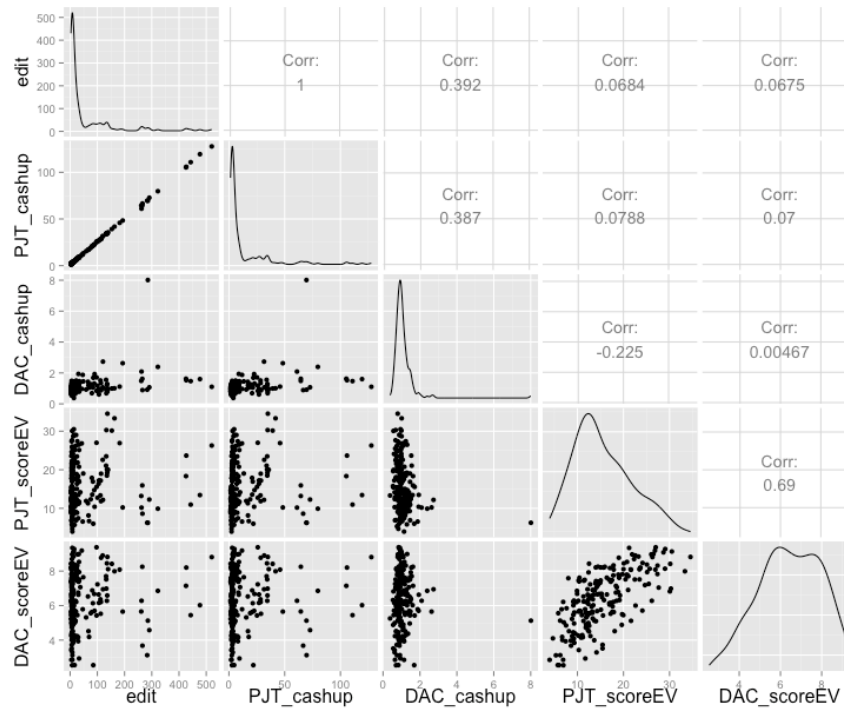


Figure 25: Scatterplot Matrix of Update Times for Edit-Related Operations

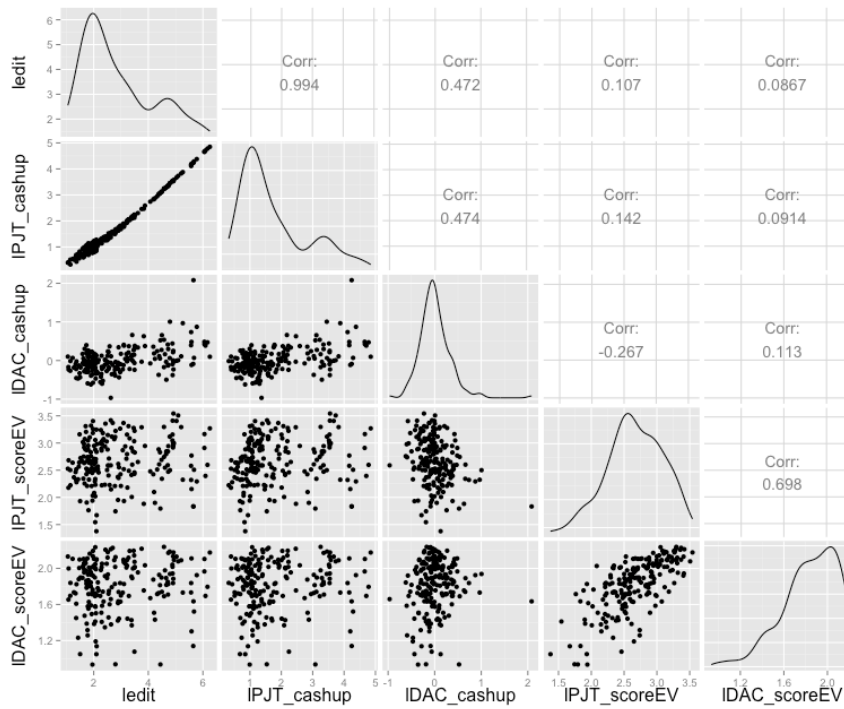


Figure 26: Scatterplot Matrix of Log Update Times for Edit-Related Operations

Table 12: Summary of Log-Transformed Timing Statistics

	tsuic	tsuic_cashup	tsuic_cashup	tsuic_scoreev	tsuic_scoreev
Min.	:1.054	Min. :0.3241	Min. : -0.967058	Min. :1.378	Min. :0.9338
1st Qu.:	1.896	1st Qu.:0.9912	1st Qu.: -0.182903	1st Qu.:2.410	1st Qu.:1.6859
Median	:2.482	Median :1.3718	Median :-0.023883	Median :2.650	Median :1.8479
Mean	:2.910	Mean :1.8122	Mean : 0.003584	Mean :2.666	Mean :1.8146
3rd Qu.:	3.653	3rd Qu.:2.3409	3rd Qu.: 0.165153	3rd Qu.:2.981	3rd Qu.:2.0311
Max.	:6.254	Max. :4.8498	Max. : 2.080816	Max. :3.542	Max. :2.2382

Time
to
Find

Cash

Figure 26 shows a positive and roughly linear relationship between PJT cash/update and DAC cash/update, which was obscured by the extreme skew in the untransformed distributions. The difference in log cash/update time between PJT and DAC depends most strongly on treewidth, number of nodes, number of states and number of variables edited within clique:

```
lm(formula = diffclu ~ usrnv + usrc1 + tw + nofu + N + nos, data = ts)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.90138	-0.34558	-0.06893	0.31279	1.38516

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-4.162016	0.289465	-14.378	< 2e-16 ***
usrnv	-0.126778	0.032485	-3.903	0.000131 ***
usrc1	-0.080739	0.032394	-2.492	0.013531 *
tw	0.629135	0.033619	18.714	< 2e-16 ***
nofu	0.012825	0.005812	2.206	0.028530 *
N	0.070131	0.007219	9.714	< 2e-16 ***
nos	0.667101	0.042941	15.535	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4739 on 193 degrees of freedom

Multiple R-squared: 0.7899, Adjusted R-squared: 0.7833

F-statistic: 120.9 on 6 and 193 DF, p-value: < 2.2e-16

An interaction plot is shown in Figure 27 below. The difference in cash update time generally increases with number of nodes in the Bayesian network, with larger differences for higher treewidths.

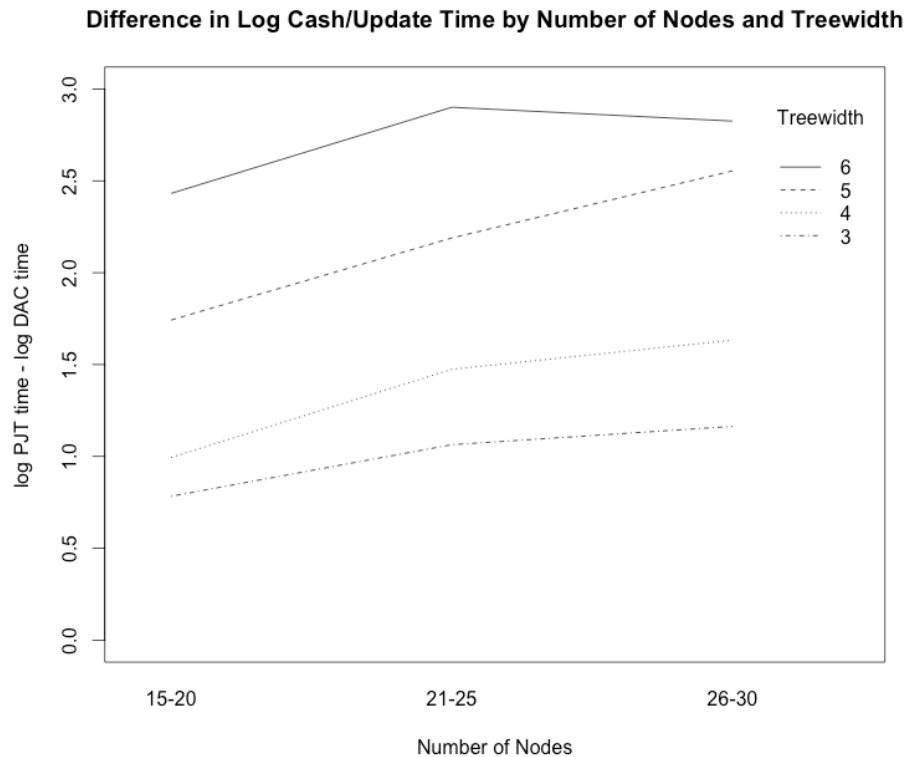


Figure 27: Interaction Plot of Difference in Log Cash Update

Time to Calculate Expected Score

A regression was performed of the difference in time between PJT and DAC to compute expected score. Statistically significant factors are the number of cliques edited, the number of users, and the number of nodes in the Bayesian network. The difference in computation time increases with number of users and number of nodes, and decreases with number of cliques edited (because this makes edits less sparse). Treewidth has no significant effect; nor does number of variables in a clique edited by the user. This regression used raw data not logarithms, because times to compute expected scores were not skewed.

```
lm(formula = scoreEVdiff ~ usrnv + usrcl + tw + nofu + N + nos,
   data = ts)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.7082	-1.0868	-0.1579	1.1009	4.6175

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-19.23682	0.97642	-19.701	< 2e-16 ***
usrnv	-0.12855	0.10958	-1.173	0.242
usrcl	-0.83121	0.10927	-7.607	1.20e-12 ***
tw	-0.11260	0.11340	-0.993	0.322



```

nofu      0.59844    0.01961  30.523 < 2e-16 ***
N         0.80855    0.02435  33.202 < 2e-16 ***
nos       0.65166    0.14485   4.499 1.18e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 1.599 on 193 degrees of freedom
Multiple R-squared: 0.9163, Adjusted R-squared: 0.9137
F-statistic: 352.2 on 6 and 193 DF, p-value: < 2.2e-16

An interaction plot is shown in below.

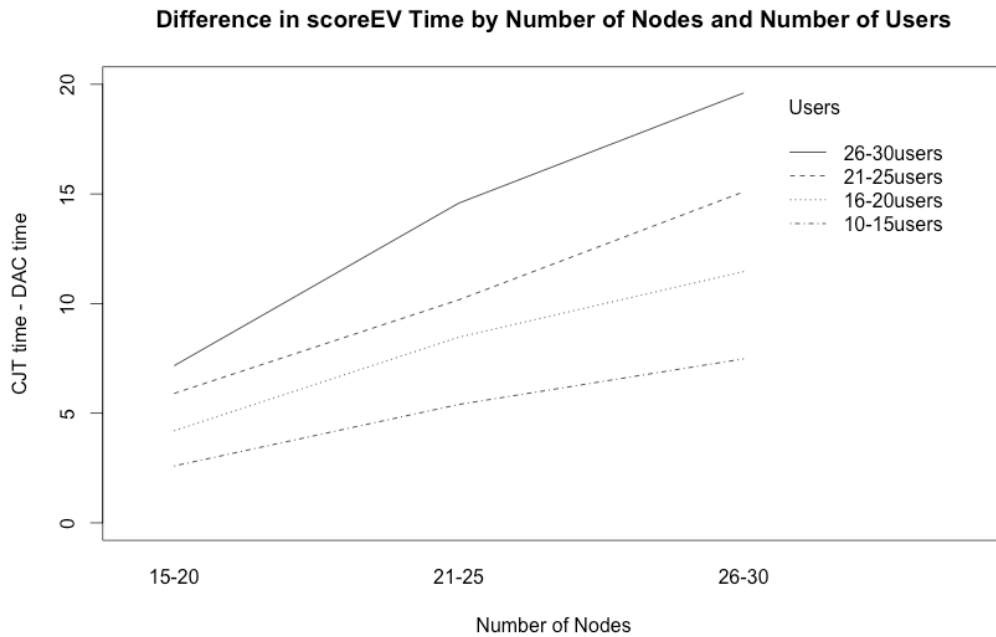


Figure 28: Interaction Plot of Difference in Time to Compute Expected Score

Total Savings in Computation Time

We computed the total savings in computation time from DAC by the formula:

$$\text{totSav} = \frac{\text{PJT_cashup} - \text{DAC_cashup} + \text{PJT_scoreEV} - \text{DAC_scoreEV}}{\text{edit} + \text{PJT_cashup} + \text{PJT_scoreEV}}$$

Because the edit time is the same for both algorithms, the numerator is the difference in cash/update times and expected score times; the denominator is the total computation time for PJT. The values of totSav ranged from a minimum of 7% savings to a maximum of 64% savings, with a mean of 33.6% and a median of 34.5%. An interaction plot in Figure 29 shows a strong relationship with number of nodes and the number of users.

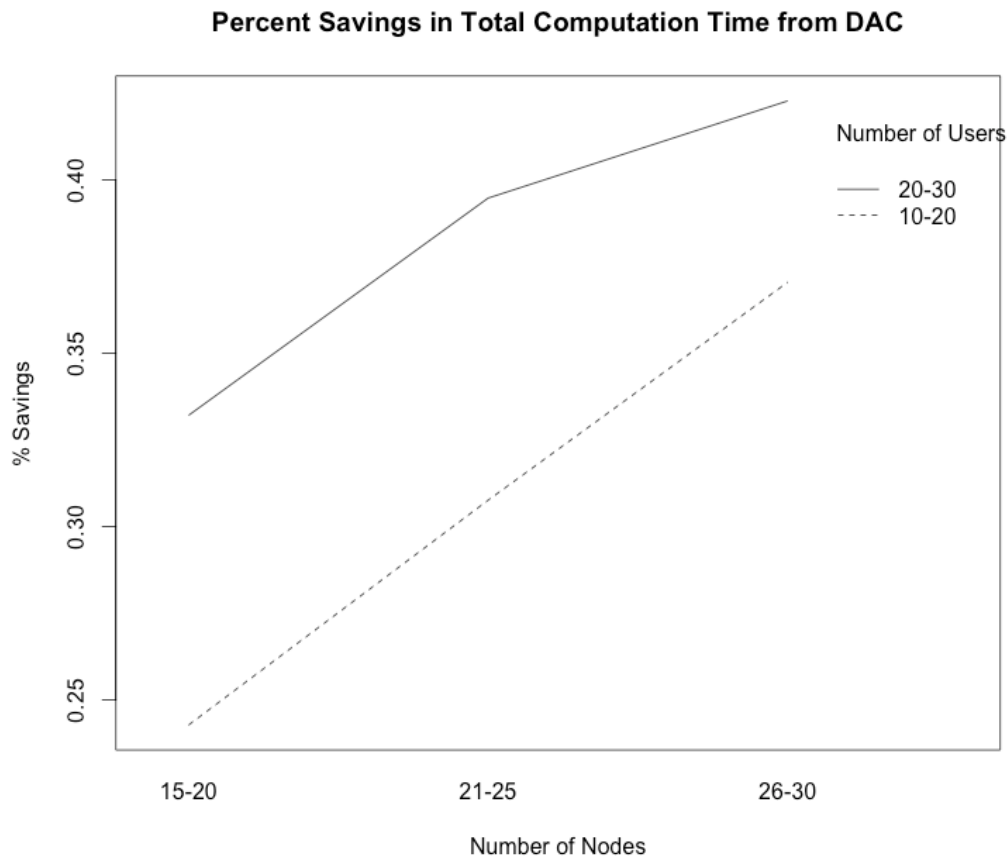


Figure 29: Savings in Total Computation Time

Discussion

Most of the results are in accordance with expectations: compared to PJT, the DAC algorithm incurs some extra overhead to compile the asset junction tree, which overhead pays dividends as the asset structure becomes larger or more complicated. Therefore we were surprised that treewidth had no effect on the expected value computation. PJT, by virtue of using the shared junction tree, always uses maximum treewidth, while DAC can achieve substantial savings where users did not trade on the nodes in the largest clique. However, in the parameter space we examined, that effect did not matter. We expect that eventually treewidth would matter, even holding all other factors constant.

Target Journals and Conference

To be submitted to *IEEE Transactions on Human-Machine Systems*.

Recommender Testing

Background

The Tuuyi Recommender recommends forecasting problems (FPs) thought to interest a particular forecaster, as described in “Recommender”, below. The most visible location is via the Carousel on the login landing page, also known as the Exchange. From April 2014 to December 10th 2014 the Carousel provided all participants with real-time recommendations based on participant FP editing history, participant demographic information, and other available data about the FPs. (Twardy & Laskey, 2014) However, from December 10th through March 24th 2015, we ran a randomized controlled trial where half the participants received random recommendations.

Objectives

The goal of this experiment was to provide an objective measure of the quality or value of the Tuuyi recommendation system. If good recommendations matter, they should generate more activity than random recommendations. Therefore from December 10th 2014 through March 24th 2015 forecasters were divided into two groups. One group consisted of users that viewed Tuuyi recommendations in the Carousel, while the alternate group consisted of users that viewed random recommendations in the Carousel. Participant click-through and trade statistics were collected for FPs shown by the recommender.

At minimum, we wish to reject the null hypothesis that there is no difference between forecast activity generated by the Tuuyi recommendations and random recommendations. In addition, we would like to measure the size of the effect (if any) on forecaster activity.

Experiment Design

From December 10th 2014 through March 24th 2015, all SciCast participants took part in a 15-week experiment to determine the effect of the Tuuyi Recommender system on forecast activity. Participants were divided into one of two evenly distributed groups based on userID. Users with an odd numbered userID were placed into the control group and received random recommendations. Those with an even numbered userID were placed into the treatment group and received Tuuyi recommendations.

The experiment design assumed: (1) there would be negligible difference in Carousel viewing between the two groups, (2) internal system usage does not have a significant effect on click-through or trade percentages, and (3) the same phenomena is generating random error in each of the groups.

A power calculation was performed on the percent difference between group click-through percentages to determine the sample size, or number of days, needed to achieve a power of 80% with a moderate difference, or an effect size of 0.5.

The following equations were used to determine effect size and sample size. The effect size denotes the confidence that can be placed on the calculated difference statistics. The larger the effect size the stronger the

confidence that there is a meaningful difference. The sample size is computed using a power analysis calculation based on Cohen's d (Cohen, 1988).

The necessary sample size n is given by:

$$n = \frac{2\sigma^2(Z_\beta + Z_{\alpha/2})^2}{d^2}$$

where Z_β is the desired power, $Z_{\alpha/2}$ is the two-tailed level of statistical significance, σ is either the standard deviation of the control group or the pooled standard deviation (see below), and d is the effect size defined as the absolute difference of means measured in standard deviations (Cohen's d):

$$d = \frac{|m_1 - m_2|}{s_p}$$

The pooled standard deviation s_p is given by:

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{(n_1 - 1) + (n_2 - 1)}$$

Within the given sample size equation, σ , or the population standard deviation of click-through percentage, can be estimated from (1) the standard deviation of the control group's click-through percentage or from (2) the pooled standard deviation, s_p . The control group's standard deviation is commonly used because it is not tainted by treatment, and thus thought to more closely reflect the population's standard deviation. The pooled standard deviation takes into account each group's variance, and is used when two groups have an assumption of homogeneous variance. As discussed below, we calculated both forms and used the larger estimate of required sample size. (The difference was about 7 days.)

Prior to the experiment, the site was not instrumented to measure click-through rates or their variance, but it was expected that there would be a noticeable difference between the Tuuyi recommender and random recommendations. However, based on initial click-through rates, it was clear the experiment would need to run for many weeks. After 12 weeks of data collection from Carousel clicks, there was no statistically significant difference in conversions (edits). Therefore, using the data to estimate standard deviations in conversion rate, we determined the sample size needed to achieve a power of $\beta=80\%$ with a medium effect size of $d=0.5$, and an α value of 0.05. A power of 80% is the generally accepted minimum level of power, corresponding to a Type II error rate of 20%, with α setting the Type I error rate at 5%. (Lakens, 2013). The effect size $d=0.5$ corresponds to half a standard deviation difference. For this experiment, an effect size equal to or greater than 0.5 suggests an obvious effect between the control group and the treatment group exists

(Sullivan & Feinn, 2012). Because the control condition was random recommendations, only obvious differences were acceptable.

Using the control group's 12 week data, σ was calculated to be 0.097. Using the 12 week data for the control and treatment groups, s_p was calculated to be 0.092. Utilizing the control group estimation of σ , the required sample size was calculated to be approximately 63 days. Likewise, utilizing the pooled standard deviation, s_p , the required sample size was calculated to be approximately 70 days. Both of these day ranges fell within the 12 weeks of sample data collection. Due to a calculation error conducted during week 12, the original sample size estimate produced called for a sample size of 15 weeks. Accordingly, the experiment was run for a 15 week period.

Results and Analysis

In total, the experiment was run for 15 weeks, or 105 days with an average effect of $d=-0.26$ (95% confidence interval $[-0.54, +0.01]$) where negative indicates the control outperforming the treatment. We did not detect an effect despite having a power of 92.5% to detect one of the desired magnitude $d=0.5$. In fact it is likely that the treatment group did worse than the control. Had we used the more lenient 90% CI ($d=[-0.49, -0.04]$), it is true that we would have rejected the null hypothesis $d=0$, but also any effect size $|d|\geq 0.5$. (See Figure 30.) It is highly likely that the treatment does not outperform the control, and likely that it does worse. The average effect size was robust. Estimates at 63, 70 and 105 days and were -0.25, -0.23, and -0.26, respectively. The difference between treatment and control groups is less than 30% of the standard deviation in either the control group or the pooled group data, and in the wrong direction.

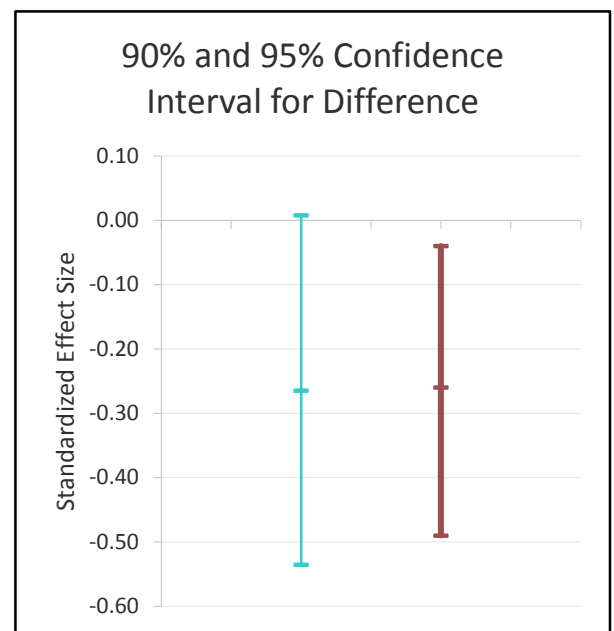


Figure 30: 90% and 95% confidence interval for standardized effect size at 15 weeks

The actual click-through rate of the random recommender is 0.126 (95% CI = $[0.105, 0.146]$), and that of the Tuuyi recommender is 0.100 (95% CI = $[0.083, 0.117]$), as shown in Figure 31. The measured difference of -0.026 is insubstantial, and again, in the wrong direction. As expected from the d analysis, this raw difference is (just barely) non-significant, as can be seen immediately by noting that the 95% CIs overlap by almost exactly half the length of one arm (hence $p\approx 0.5$, Cumming, 2009) or calculating the paired-difference two-tailed $p=0.0561$. The Tuuyi recommender generated no more click-throughs than random recommendation, and quite possibly fewer.

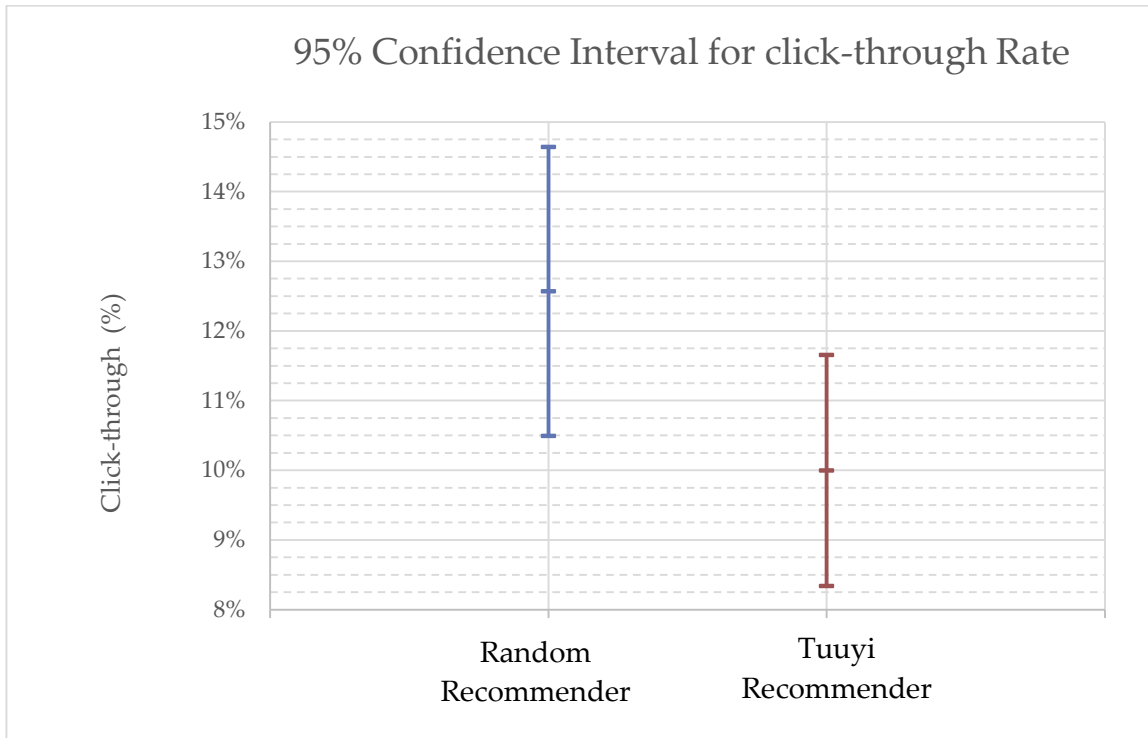


Figure 31: 95% Confidence Interval for click-through rate for the control (Random) and treatment (Tuuyi) groups

Figure 32 shows the weekly rates; the Random recommender had visibly higher click-through percentage 8/15 weeks, whereas the Tuuyi recommender was visibly higher only 3/15 weeks.

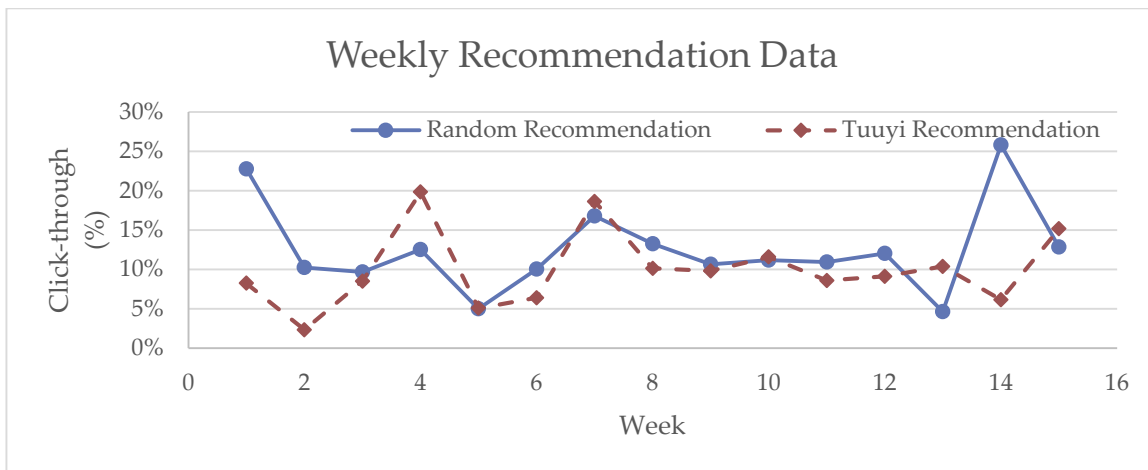


Figure 32: Weekly Click-through Percentages for the Control (Random) and Treatment (Tuuyi) Groups

Even if it had been in the right direction and statistically significant, the observed difference of 12% vs 10% click-through makes no practical difference, and does not justify using a more complicated recommender. One or more of the following may apply: (1) Tuuyi recommendations are in fact indistinguishable from random recommendations, (2) regardless of quality, recommendations in the Carousel may not affect site navigation,

(3) regardless of quality, recommendations in the Carousel have little effect on forecasting. It remains possible that the Tuuyi recommendations are in some sense better, but that their value is not to be found in click-through rates once on the site. To help get further insight, we asked forecasters what they thought of their recommendations.

Forecaster Survey

On February 16th, 2015, during week 10 of the experiment, we released a three question survey on the SciCast website in order to learn more about how forecasters used the Carousel. The questions asked on the survey were as follows:

- (1) How often do you view recommendations in the carousel?
- (2) How often do you click on a recommended question?
- (3) Please rate the recommendations using the scale provided:
 - a. N/A. I do not use the recommendations.
 - b. Random. (Recommended questions interest me about as much as often as if they were drawn randomly from the roughly 600 on the site.
 - c. Systematically Good. (Most suggested questions interest me)
 - d. Worse than Random. (Almost none of the suggested question interest me).

The survey gathered 37 respondents over the course of two weeks. According to survey responses, 57% of those surveyed occasionally use the Carousel recommender, 27% never use the recommender, and 16% use the recommender almost every time they login. Matching survey UserIDs to logins, 37% of those surveyed are within the top 3% of logins and therefore Carousel views.

Table 13 below shows the conditional probabilities of forecasters' perception of recommended questions given their assignment to the control group (random recommender), versus the treatment group (Tuuyi recommender), or on their identity (hence group assignment) being unknown.

	Group Association (N)		
Perceived Recommendation Quality	Control	Treatment	Unknown
Systematically Good	0.22 (4)	0.20 (3)	0.50 (2)
Random	0.67 (12)	0.53 (8)	0.50 (2)
Worse than Random	0.03 (1)	0.08 (3)	0 (0)
N/A	0.06 (1)	0.07(1)	0 (0)

Table 13: Conditional Probabilities: Recommendation Type Given Experimentation Group (N = Survey Count)

From the table above we can see that both the control and treatment groups have similar distributions related to the perceived recommendation type, with "random" being the most prevalent (50-70%) followed by "systematically good" (23%). If both "random" unknowns were in the control group and both "systematically good" unknowns were in the treatment group (a 1:4 chance), then we might begin to suspect a perceived



difference, otherwise there is not much evidence for subjective difference. The survey supports our click-through null result.

It is worth noting that the base rate for page views of the Carousel was *not* evenly distributed as assumed. For 14 out of the 15 weeks the random recommendation group (odd UserIDs) had more unique Carousel pageviews than the Tuuyi recommendation group. Ignoring the size of the difference, if the groups were truly equal, there is only a .05% chance that one group would be more active 14 or more of the 15 weeks. Users do not get to choose their UserID, so UserID should be as random as a coin flip. However, analyzing pre-contest activity for the 105 days prior to December 10th 2014 revealed 53.4% of logins were from odd UserIDs, while only 46.6% of logins were from even UserIDs. Nevertheless, we do not see that bias during the experiment (49.8% even vs. 50.2% odd), so the imbalance remains unexplained. It could be an actual difference in login behavior not well-represented by the log file, or to flaws in our mapping between Carousel pageviews and user logins, or finally, a real and unaccounted-for bias between even and odd userIDs.

It is not clear that having more absolute Carousel pageviews provided any material advantage in click-through rates conditional on the Carousel pageview, but it is an unexpected finding.

Fraud Detection & Analysis

Background

SciCast has conducted periodic contests in which the most successful point-winners over a specified period of time on specified contest questions are awarded significant cash prizes. To maintain fairness as well as accurate market predictions, contest rules specifically prohibit certain tactics, including:

- Market manipulation
- Dumping/shifting points from one user account to another
- Intentionally losing
- Operating multiple accounts by one user

To police these rules, trading activity is monitored and users judged to have broken contest rules are disqualified.

Note that it is virtually impossible to be certain that cheating has occurred. Trades that are counter-intuitive, very aggressive or apparently foolish, and that might be candidates for fraudulent activity can be the result of several possibilities:

- Naivete (about the workings of the prediction market or the specific question traded),
- True belief that the market probabilities are very wrong
- Malicious behavior intending to manipulate or disrupt the market.

SciCast management has limited ability to obtain additional information about users or the circumstances of their trades. Thus, our goal in fraud detection is to identify trades and traders that we believe beyond a reasonable doubt to be fraudulent. Actual cases of activity judged to be fraudulent are discussed below.

Market Manipulation

Market manipulation, point dumping and intentional losing are all present in the most commonly observed instances of cheating. The most frequent technique observed is for two users to collude, or for a single user to operate multiple accounts as if run by two colluding users. One of the accounts we designate the Loser account, the other the Winner account. The Loser account will typically make a large long-shot trade, which is followed very quickly thereafter by a reversing large trade by the Winner account, bringing the market probability back to approximately where it was before the Loser account trade. When the question has a high market probability of outcome in the direction of the Winner account's trade, the result is a high probability of large gain for the Winner account. When done repeatedly, this can produce large, contest-leading gains for the Winner account.

In some cases the manipulative trades are as blatant as described above (the larger the change in probability, the larger the gain, which makes such trades tempting to a cheating user). Users who are unaware that market manipulation is outlawed, or who are unaware that trades are monitored for cheating, may see no reason not to make large long-shot trades and quickly reverse them in another account, on a repeated basis on the same contest question. Generally, though, users seem to be aware of the rules and, if they are cheating, they are

more subtle. They make more modest trades, at less frequent intervals, using a variety of accounts and questions. The more subtle the cheating, of course, the more difficult it is to detect and to distinguish from legitimate trades.

Market Manipulation Detection Techniques

To detect market manipulation, there are several sources of information that can be synthesized to identify possible instances of cheating and help us judge the likelihood that cheating has occurred. Virtually all the information about trades and traders we obtain from the “Data Mart”, which is a web-service that provides queries allowing us to download data about the following:

- SciCast questions
 - Esp. resolution dates, possible outcomes
- User statistics
 - Esp. account creation date, user preferences, number of trades made
- Trading activity
 - Details of each trade
- Contest Leaderboard
- User comments

Most of the information used to detect cheating is obtained from a Data Mart Trade_History report. This report includes a record for each trade and consists of dozens of fields, the most useful of which for detection of cheating are the following:

- Question_ID #
- Choice Index – the ordinal number of the possible outcome on which the probability is being changed
- Old Value - market probability of outcome before trade
- New Value - market probability of outcome after trade
- Interface type – safe mode/power mode/API bot
- Asset resolution - points gained/lost on trade if already resolved
- Assets per option - points gained/lost by outcome for each possible outcome
- Trade date/time
- User_ID

The Trade_History report for the period of time under examination is downloaded to an Excel spreadsheet. The following fields are then obtained from other Data Mart reports and added to the spreadsheet:

- User_name – from Person query
- User Contest Rank – from Person_Leaderboard
- Contest Question Category – identifier of contest questions, from list of contest questions
- Question Resolution Date – from Question query

Finally, the following additional fields are calculated based on the fields already present:

- Potential Point Gain (if prediction direction is correct)



- Potential Point Loss (worst case outcome if prediction is incorrect)
- Expected Value of trade – dot product of gain/loss by outcome and market probability of outcome, based on “stable” probabilities prior to most recent unusual trades
- Minutes Till Next Trade on given Question, if less than selected threshold – to avoid cluttering the spreadsheet, minutes till next trade is only shown if it is less than 20 minutes; most instances of collusion consist of trades made within a few minutes of each other

The resulting spreadsheet is then sorted by trade date within question. The Potential Point Gain, Potential Point Loss and Resolved Point Gain/Loss fields are then filtered to highlight large trades that are out of character with the probability trends for the question, followed shortly thereafter by a reversing trade or trades. While high potential point gain questions are the highest priority for review, the total volume of trading activity in SciCast is low enough that an eyeball review of all trades can be made in a couple of hours per week.

A more subtle cheating approach than large-gain trades is to make a large number of small-gain Loser/Winner trades. As one way to detect this, counts can be made of the number of consecutive trades made by any pair of users. Those pairs that trade consecutively on a large number of questions merit further examination.

Note that successful Winner reversing trades must be made fairly quickly after the Loser trade, since one or more user bots is virtually always patrolling the market to reverse large trades for the benefit of the bot owner before other users can do so. Filtering for low Minutes Till Next Trade on given Question will generally yield most collusive trades. The more active the market bots, the shorter the time in which the collusive trades must be made in order to beat the bots to the reversing Winner trade.

Bots present an interesting complication for cheating analysis. On the one hand they help deter cheaters since they often will make the reversing trade of a Loser account trade before the Winner account can do so, discouraging the cheating duo from trying. Because the bot owner is likely not aware of the trades his bot makes, however, bot owners are difficult to implicate in cases of suspected cheating. After their bots capture Loser trade points, they can easily claim ignorance of all the trades involved. Cases for cheating against bot users are difficult to make, and may require ancillary evidence showing that the Loser account is controlled by or colluding with the Winner.

Once a suspicious series of trades is identified, the likelihood of cheating must be assessed. Indicators that increase the assumed likelihood of cheating include the following:

- **Very short time between offsetting trades** – a short time gap is necessary in order to beat the bots to the reversing trade(s), and it is unlikely that a random innocent user would notice the Loser trade and act on it quickly, before the bots. Thus, the short time gap is indicative of collusion.
- **Reversing trade is a power trade (i.e., not a bot)** – As noted above, bot trades are more difficult to successfully judge to be an act of cheating. Interestingly, safe mode trades are often used to make Loser trades but virtually never used to reverse Loser trades. Generally, safe mode is used by newer or less



sophisticated traders. When a Loser trade is made in safe mode, especially by a brand new user, it is more difficult to determine whether the trade was due to user naivete vs. sly camouflaging of a true Loser trade. Experienced traders (those who legitimately could be in contention for large contest prizes) virtually never use safe mode, so a reversing Winner trade made in safe mode would draw lots of suspicious attention. It also might require multiple time-consuming safemode reversing trades to offset the full amount of the Loser trade points.

- **Trade pattern is repeated on other questions for same pair/team of users-** Unless a huge point Winner trade is made, acquiring significant points by cheating requires repeated illicit trades. So, cheaters often repeat their crimes, either on the same question or on different questions. Repetition reduces the chances that the trades are isolated incidents, and increases the odds that suspicious activity is indeed cheating.
- **Winner trade exactly reverses the Loser trade** – A single reversing trade that exactly reverses a long-shot Loser trade draws attention to the Winner. To deflect apparent guilt, sometimes multiple colluding Winners will split the reversing trades (e.g., two or more reversing Winner trades will be made, each taking a portion of the points made available by the Loser trade, together moving the probability close to where it started). It is more difficult to identify the cheating Winner if he is only taking some of the points available and leaving the rest for other users. A variation on this theme is for the reversing trade to be split among several different outcomes for a multinomial question (i.e, a single Winner trading each of several outcomes in the opposite direction of the initial Loser trade).
- **Question is to be resolved in near future**-Until a question is resolved, a user's points are tied up in trades he makes (points equal to the largest possible point loss on the question are unable to be used for other trades to ensure that the user has sufficient points to make good on his bet if it loses). Thus, the sooner the question is resolved, the sooner the user's points are available for new trades. A second reason why soon-to-be-resolved questions are attractive is that the outcome of the question is more certain, and the Winner's trade is more likely to actually be a winning trade.
- **Question probabilities are unbalanced** – when one of the possible outcomes for a question is highly favored (greater than about 0.9), it is safer to make a Winner trade favoring that outcome, as opposed to a Winner trade on an outcome that is much less certain of actually being a winning trade.
- **Question is binary** – on a binary question the Winner is generally more certain that his reversing trade is in fact a winning trade.
- **Question is contest-eligible** - in the words of Willie Sutton, banks are robbed because that's where the money is. On the other hand, there are benefits to cheating on non-contest questions. First, it produces points that can be used to make more trades on contest questions. Second, it is less likely to draw attention from monitors and less likely to be judged cheating.
- **Traders have High/Low ranking** – The Loser half of a colluding pair is presumably not expecting to win a contest prize, so his contest ranking is likely to be low. The Winner half of the pair is attempting to win a prize and is likely to be ranked among, or just below, the prize-eligible group. Depending on how the contest prizes are structured, there will be incentives to reach certain contest ranks. Users for whom cheating can make a difference, based on their current rank, would appear to be the most likely cheaters. Note that suspicious trades involving pairs of highly-ranked users are judged unlikely to be fraudulent, since the motivation for intentional point-dumping by one of the pair would not appear to be present.



- **Traders have same/similar IP address, ISP, or ISP-reported geographic locations** – IP address analysis is not sufficiently scientific to be a primary determining factor in judging cheating, but Loser/Winner pairs with the same IP address or reported geographic location are more highly suspect. Dissimilar IP addresses are less diagnostic because they can be spoofed. However, it is often possible to detect the use of IP spoofing, which itself may be indicative of collusion.
- **Loser Trader is unknown/not previously successful** – Traders develop trading patterns over time that they tend to follow. Traders rarely seem to start cheating after having traded honestly for a lengthy period of time. Traders who are unsuccessful for many trades and suddenly enjoy success are suspect. Long-term successful traders who make long-shot bets are highly unlikely to be dumping points, so their long-shot bets are not suspicious.
- **First trader disappears after suspicious trades** – A clever cheater will use several Loser accounts to dump points to his Winner account (ignoring for the moment the fact that operating multiple accounts is in and of itself grounds for disqualification). By doing this, less attention is drawn to any individual Loser/Winner pair of users. Uncertainty is created in the eyes of the monitors because a Loser who disappears from trading after a losing spree might reasonably be a naïve new trader who made unsuccessful bets and either became discouraged or lost interest. Regardless of the number of Loser accounts a cheating Winner employs, unless the Winner can afford to split his Winner trades among multiple Winner accounts, at some point the most successful Winner accounts will draw scrutiny from market monitors.

All of the factors above are considered together in making a judgment about cheating. Additional sources of information that can be used in this process are the following:

- **User comments** – users are invited to make comments about their trades, about a specific market question, or about the SciCast market. Active traders are quite aware of market activity and often detect and report suspected cheating when they observe it. Comments on specific trades can also help explain motives behind unusual activity and exonerate suspicious trades.
- **Sort spreadsheet by trade date/time** – to obtain a better understanding of market trading activity at a given time, it is better to observe trades in chronological order rather than chronologically within question. Note that, in the spreadsheet, the Expected Value of trades must be converted from formulas to values before the spreadsheet is re-sorted in order to avoid misapplying the EV formulas.
- **Contest Leaderboard** – additional scrutiny is given to contest leaders to ensure that the large contest prizes are not awarded to cheaters. Also, examining the lowest ranked users is a rich source of potential Loser accounts. The worst losers are likely attempting to lose. The beneficiaries of the worst of their trades are often cheating suspects. While disqualifying the losers does not immediately affect the contest winner list, it does prevent them from further point-dumping.
- **Surveys** – A new user is required to disclose virtually no information about himself (an email address is optional). To facilitate academic studies of correlations between user demographic profiles and successful market trading, users are invited to complete surveys about their academic, professional and other background profiles. Users who complete the surveys would seem to be more legitimate market participants, and less likely to cheat, than those who do not complete the surveys and choose to remain anonymous (author's opinion). Also, the survey information can provide clues as to knowledge



domains in which a user has specific strengths, and therefore, might trigger apparently abnormal trading activity.

- **Nature of Question** – Some questions invite strong opposing viewpoints that can result in spirited back and forth trading that, in other circumstances, might appear to be point-dumping.

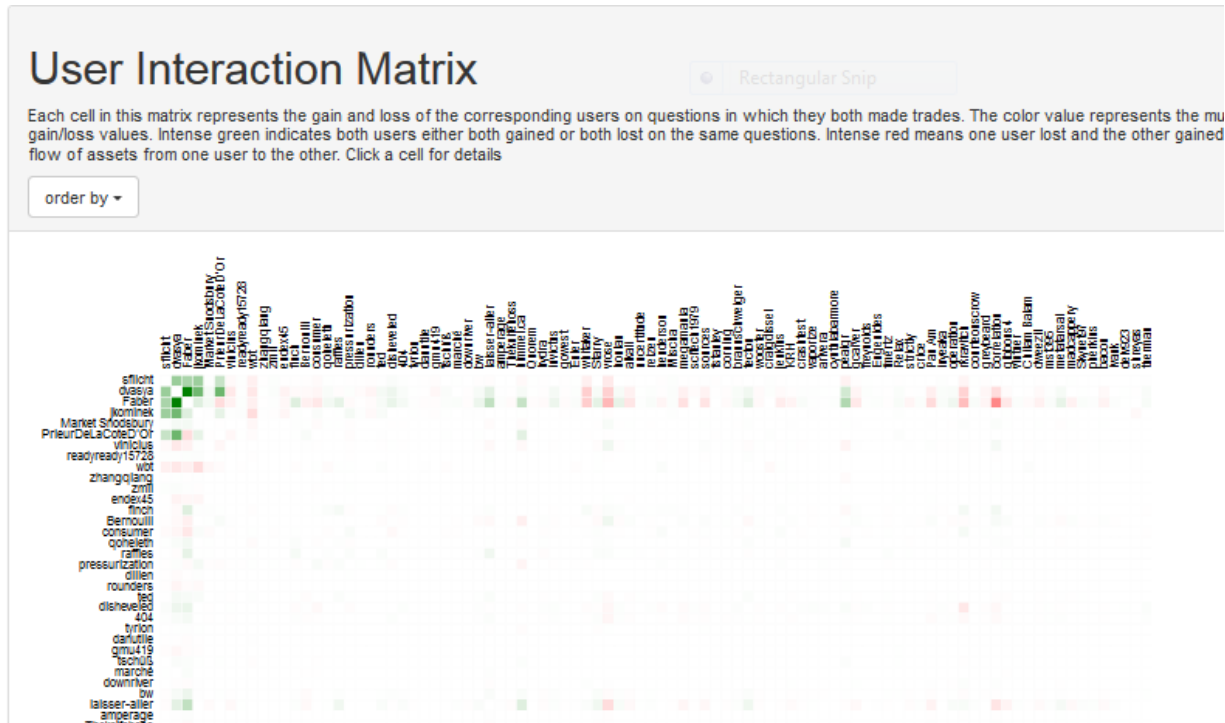
We have not yet attempted to create an automated cheater detector, but the aforementioned list could perhaps be used to create a Naïve Bayes model.

Collusion Visualization Tool

As the number of user accounts and trading volume grow, the task of identifying fraudulent activity becomes more time-consuming. In an effort to quickly highlight potential collusion between users, a collusion visualization tool (the User Interaction Matrix) was developed. (See also the Executive Dashboard section, below.)

In its initial formulation, the tool first calculates a collusion index for each pair of users. The formula for the collusion index for users A and B is the net gain or loss for A times the net gain or loss for B, on all questions on which both A and B have traded. The rationale for this formula is that, if A is dumping points to B, A will have a large negative net point total and B will have a large net positive point total, the product of which is a large negative number.

The tool consists of a 2 dimensional grid with users listed both across the top and down the left side of the grid. The cell at the intersection of user A's row and user B's column is colored green (if the collusion index is positive) or red (if negative). The intensity of the color is based on the magnitude of the collusion index. User pairs with the largest negative collusion indices will show bright red. These pairs become the leading candidates for further investigation into fraud. The users can be sorted in a variety of ways, including user ID, maximum total positive (or negative) point interaction with other users etc. A screenshot of the collusion tool is shown below.



For future generations of the tool, the collusion index can be refined to more accurately target likely fraud. Areas of improvement include the following;

- User sort - sorting the users on the grid in current contest rank order would permit easy incorporation of rank in fraud judgments
- Unresolved questions - the initial version of the tool includes only points won or lost on resolved questions. Unresolved questions can constitute a large number of a user's score, and are subject to short-term manipulation by forecasting.
- New collusion index formula - there are 2 shortcomings to the current collusion index formula. First, the index counts a user's total points earned. A user who earns many points on some questions and loses many points on other questions might have a small net point total and appear to be a minor trader. Second, the number of points "dumped" from trade A to trader B on question C cannot be greater than the lesser of the points lost by A or gained by B on question C. The current collusion index counts all points gained or lost by A and B on question C; unless the point values are equal, A and/or B must have gained or lost some points to other traders rather than to each other.

To address these shortcomings, an alternative collusion index formula for users A and B would sum over all questions the following formula:

$$\text{Min} [(\text{Min}(0, G/L_A(i), G/L_B(i)))^2, (\text{Max}(0, G/L_A(i), G/L_B(i)))^2]$$

where $G/L_A(i)$ is the gain/loss for user A on question i.



This alternative collusion index focuses on questions traded by both A and B and places more weight on those questions on which more points have been traded between the users.

- Time gap – as noted elsewhere in this chapter, point-dumping trades must be made within a narrow time frame in order to prevent bots from making the reversing Winner trade before the Winner does. Reversing trades made after a long time gap are considered much less likely to be fraudulent. Counting in the collusion index for A and B only trades made within a short time window would seem to better capture only fraudulent trades.
- Contest eligible questions – since most all fraud will occur on contest questions, counting only contest questions in the collusion index would seem to better focus the index on fraudulent trades.

Other tools

Fraud analysis would seem to be ripe for analysis using Bayesian networks. The various factors indicative of fraud, which are discussed above, could easily be used as network nodes. Unfortunately, as discussed above, we are unable to ever determine for certain whether a given trade or series of trades was fraudulent. So, we have no ground truth on which to train a Bayesian network.

Multiple Accounts

While market manipulation is the most serious form of trading fraud, use of multiple accounts is also prohibited, both to prevent “self-collusion” and to ensure that contest prizes are limited to one per person. A user might trade under multiple accounts for several reasons:

- He plans to operate one or more as Loser accounts that dump points to a Winner account
- He plans to operate multiple Winner accounts (depending on the contest prize structure, it may be more lucrative to have multiple mid-ranking winner accounts than a single high-ranking winner account). This approach is valid whether or not the user intends to feed the Winner(s) from Loser account.
- His initial account is scoring poorly and he wishes to start over with a “clean slate”.
- SciCast staff tend to have multiple accounts for various purposes, but these are known, and are not eligible to win contests.

Note that there is no cost to obtaining a SciCast account, and no identifying information is required to be disclosed in order to open an account, so there is no real downside to opening multiple accounts. If the user is fortunate enough to win multiple prizes he would need other persons to claim the prizes, but that does not seem overly difficult to arrange.

As with market manipulation, we are virtually never able to obtain 100% certainty that a user is trading with multiple accounts. If the user happens to be unaware that multiple accounts are not permitted he might provide identical or similar (optional) email addresses for his accounts, thus revealing the multiple accounts, but that is unlikely to occur.

To identify users who are trading with multiple accounts, the following techniques can be used:



- **IP Address/Browser** – if a user operating multiple accounts signs on to both accounts at the same time, he is likely to use the same ISP. He may or may not report the same IP Address and/or user agent, depending on which device(s) and browser(s) he is using to sign-on. At best we might observe the same IP address and user agent reported for log-ins at about the same time, which would be modest evidence of multiple account use.
- **Modus operandi** – users develop trading styles that can be learned by observation. Style features include user mode(s) employed, size of trades, frequency of trades/sessions, timing and duration of trading sessions, types of questions traded, user comments etc. Similarities in style between accounts might indicate a single user.
- **Point-dumping** – in any case in which point-dumping is identified, it seems likely that the Loser and Winner accounts are operated by the same user.

Clever users who are aware of monitoring and desire to hide their use of multiple accounts can likely evade detection by varying the trading behavior between the accounts, signing on at different times etc.

Results

Over the course of the SciCast contest that was monitored for fraud (covering trading from November 7, 2014 through March 7, 2015), **14 user accounts were judged to be fraudulent and were disqualified.**

When a user was judged to be cheating, his account was frozen and an email note was sent, informing him that he was suspected of cheating and requesting that he contact SciCast administrators if he believed disqualification was unjustified. Interestingly, only one of the disqualified users responded to these email notes. We took that as an indication that our fraud judgments were likely accurate, although a user might also realize that he has no real ammunition with which to argue his case other than a plea of innocence. If we were indeed correct in judging all of these cases to be fraudulent, it probably means that there were other cases of fraud that we did not detect.

The disqualified accounts included 4 pairs of users, 3 of whom executed 4 – 5 trades in the span of 1 -2 minutes. Each Loser trade pushed a very low binary question probability to the opposite end of the spectrum. The corresponding Winner trade followed virtually immediately, returning the probability close to where it was originally. Each trade netted several hundred points to the Winner.

The fourth pair of users followed this same m.o. except that close to an hour elapsed between the Loser and Winner trades. The trades occurred the evening before a major holiday, however, when bot activity was apparently low, thus permitting the Winner trades to be made. The time gap created a larger than normal doubt in the SciCast administrators' minds as to the guilt of the Winner. The Winner was sent an email offering to hear his case against collusion but has chosen not to reply.

In addition to the pairs of Winner/Loser traders there was also one set of a Loser apparently working with 2 different Winners. The Loser lost about 3600 points, dumped roughly equally to the two Winner accounts. As with the 3 pairs of Winner/Losers described above, the trades were executed in a very short timeframe.



A trio of users was disqualified for use of multiple accounts. The accounts were created on consecutive days and each user executed virtually identical trades, all using safemode, on the identical group of several dozen questions, each within a period of 15-20 minutes. The possibility that these trades were pure coincidence was judged too remote to allow for reasonable doubt, so the accounts were disqualified. Further evidence for a single user was the fact that all 3 users appear to use IP address scrambling, which is not generally employed by SciCast traders.

Interestingly, users posted comments pointing out several of these cases of fraud. Honest users want a fair contest and they want cheaters to be punished. SciCast is set up to allow users to easily observe trading activity, so fraud that occurs during active trading periods appears to stand a good chance of being detected by other users.

Also of interest, but not surprising, after the first pair of cheaters were publicly disqualified, easily detectable fraud disappeared.

There were other instances of suspicious activity that we were unable to determine beyond a reasonable doubt were cheating. A couple of these are described below:

- **Bots** – the use of bots to make trades is a double-edged sword for maintaining fair trading markets. On the one hand, alert bots can often reverse Loser trades faster than the Winner half of a colluding pair of accounts can make the reversing Winner trade. This tends to discourage colluding pairs from cheating. On the other hand, bot owners are deemed immune from cheating on trades their bots make. A dishonest colluding bot owner could make Loser trades in a non-bot account which he then programs his bot to reverse. The few bot accounts among SciCast users have done very well in the 4-month contest. A non-trivial amount of their points were earned from reversing Loser trades. If the bot owner happened to be colluding with the Loser account owner, SciCast administration would have no way of knowing that.
- **Multiple account users** – while it is illegal under SciCast rules for a user to maintain multiple accounts, it is extremely difficult to prove when it does occur. Evidence to indicate multiple accounts included instances of several user sign-ons to SciCast from the same IP address at virtually the same time.

Lessons Learned

The discussion above includes several ideas that could be incorporated in fraud analysis of prediction markets. The following additional conclusions might be considered:

- **Large cash prizes cause cheating** – contests were introduced in part as an experiment to encourage greater prediction accuracy. Whether or not greater accuracy resulted, the desire for cash prizes brought out the worst in some users. A possible alternative would be to offer added points for greater prediction accuracy rather than cash.
- **Self-policing can be a significant aid to fraud prevention** – encouraging user comments and maintaining an active discussion among users seems to create a sense of community that is quick to identify and ostracize cheaters.



- **Publicize policing** – after the first disqualifications of users was publicized, cheating seemed to virtually disappear, or at least become much more subtle. True miscreants might take such an announcement as a challenge to beat the system despite policing, but hopefully such individuals have more lucrative outlets for such behavior. Announcing that trading will be monitored did not have nearly the impact that actually disqualifying users had. Six of the disqualified users were among the contest leaders, so it was quite evident to users, who see the leaderboard prominently displayed on the SciCast website, that SciCast administrators were serious about rule enforcement.
- **Require greater user information** – in order to encourage and facilitate user sign-ups, SciCast requires no personal information other than selection of a user ID. Users can optionally provide an email address, and they can fill out demographic surveys used to study correlations with successful trading, but a user who wants to remain anonymous can easily do so. Unfortunately, this probably contributes to cheating. An individual who is tempted to cheat and knows that SciCast knows nothing about him may see little downside to cheating since his maximum loss is his trading privileges (and he can always open a new account thereafter). If, on the other hand, he had to provide his actual name, then being caught cheating and having his name so-publicized would likely be a strong deterrent to his deciding to cheat. Particularly if cash prizes are awarded, it would be perfectly reasonable to request name, email address, perhaps geographic address, etc., and other information necessary to process awarding of a cash prize.
- **Bots** – as discussed above, bots are quite useful in maintaining efficient and honest markets, assuming they are programmed to quickly reverse long-shot trades (whether such trades are made by naïve traders, risk-taking traders, or by the Loser half of a colluding pair of traders). In the SciCast contest, when bots were active they very successfully acquired points in this manner. While the bots serve a valuable market function, the fruits of their labors were enjoyed by only a small handful of users sufficiently knowledgeable to program and operate bots. If bots were made easily available to users (by SciCast administration), there would have been more bots operating in the market (providing greater hindrance to collusion) and the easy points that bots pick up presumably would have been spread more broadly across the user population.

Question Management

The challenges of maintaining a large volume of high-quality questions and of engaging our collaborators and users in the development and resolution of questions continued to require an intensive question management process, managed by dedicated staff. Software support for this process in Y4 was provided by Inkling's continued development of SciCast Spark, a system for entering, editing, and publishing questions to SciCast Predict.

Please refer to last year's annual report to understand the detailed question management process and guidelines for writing SciCast questions, or refer to the appendices including the QM HOWTO and the question-writers' Style Guide.

In the first 11 months of Y4, the question management team published **713** questions from **36** authors with on average about **540** questions available to forecast each day. On average it took **5** days for a published question to be published. Jill Lu from BAE was the most prolific publisher (**198** questions). A total of **258** people registered in SciCast Spark to participate in the publishing process. Most of our questions came from IARPA FUSE partners BAE (**201** questions) and SRI (**241** questions). Counting since it opened 31-NOV-2013, SciCast has published 1,248 question, of which 663 have come from FUSE. (See the section FUSE Questions, below, for more information about FUSE.)

Streamlining Question Generation

To generate a sufficient number of usable questions with a sufficiently broad range of coverage for SciCast's launch and initial operation, early question generation was largely a manual process carried out by project staff. Techniques included generation of templates from which large numbers of questions could be generated; scanning news stories for question ideas; and focusing on specific topics such as climate change, cybersecurity, and space exploration. Project staff also worked closely with FUSE performers and outside Topic Leaders to help them navigate the question generation and publication process.

During the initial period, calendar time to generate and publish a question was on the order of two weeks. An effort was undertaken to streamline the question management workflow and increase throughput of questions. Changes were made to the publishing process to enable updates to questions that have already been published on SciCast predict. A Style Guide was published and provided to all question developers to help them understand how to correctly write a "good" SciCast question. Information extracted from the style guide is provided in the remainder of this section. We also began collecting metrics on question development time to help identify bottlenecks and suggest improvements.

As a result of these improvements to the question development and publishing process, in addition to experience writing questions for predictive markets, both calendar time and labor hours for question development have been substantially reduced. By April 2015, the average calendar time to publish a question had been reduced to five days.

Question Invalidation

The overall SciCast question invalidation rate was 8%. Many of those came from a single study in Y3 (OY2); excluding Study 1.2, the invalidation rate was 4%. Table 14 summarizes the causes of question invalidation, and what has been done to mitigate that risk.

Table 14: Causes of Question Invalidation

Cause of Invalidation	# Questions	Solution
Data Not Published: <ul style="list-style-type: none"> Authoring multiple questions dependent on a single report without ensuring the report would be published Assuming old data sources would be updated. 	25, almost all in the "Study 1.2" experiment reported last year	<ul style="list-style-type: none"> When possible, list multiple sources to potentially resolve a question in the event one of them does not provide expected data. Avoid authoring >3 questions dependent on a single data source, or verify the publication plans.
Management: <ul style="list-style-type: none"> Vetoed by Sponsor Poorly Written Question Duplicate Question 	9 8 6	<ul style="list-style-type: none"> Dedicated question management staff to review all questions prior to publishing. Wrote the style guide so all question authors have codified parameters for a "good" question. Fixed errors in Spark -> SciCast integration, preventing duplication.
All Forecasts are Aftercasts <ul style="list-style-type: none"> Questions published when the answer is already known. 	4	<ul style="list-style-type: none"> Dedicated QM staff reviews all questions prior to publishing. The instance of this is low, and, in these few cases, forecasters corrected the market by submitting sources that could fulfill the resolution criteria of each question.

As noted in Table 14, repeated sources of invalidation have been addressed with changes to software, procedures, and checklists. Removing these avoidable errors we estimate that the site would still have suffered 5 invalidations as a result of unpublished reports, 4 from poorly-written questions, 2 from duplicates, and 4 from questions where the answer is already known when published. Our target future invalidation rate then is <2%.

General Guidelines for Writing Questions

The general guidelines for writing questions did not change from Y3 to Y4. Please refer to the Y3 annual report for our guidelines for writing questions, or to the QM style guide.

New Question Type: Scaled Continuous

As discussed elsewhere, in Y4 we introduced the Scaled Continuous (Scaled) question which maps a desired numerical range into the usual 0..100% range of a binary question. For example, Figure 33 shows a Scaled question estimating the cost of an MH370 search contract, in millions of (Australian) dollars. Because Scaled re-uses the existing machinery for a binary question, it can only estimate the *expected value* of the desired quantity. The contract pays off at the actual value, or the nearest endpoint if the actual value falls outside the forecast range.



Figure 33: Example of a Scaled question with range from \$25M to \$275M

1. *Inputs on creation.* The question creator declares a scaled continuous question.
 - A. Question creator provides lower and upper bounds on the range.
 - B. Question creator provides prefix (e.g., "\$" and postfix (e.g., "M"). Prefix and postfix can be up to <charlim> characters and can contain spaces and special characters
 - C. Question creator provides the step size. (Note: The slider is only 300 pixels wide, so the minimum effective step size is (max-min)/300.)
 - D. Bins: This option is a very basic version of the scaled continuous question in the safe mode interface, allowing users to bid on bins as opposed to the actual value.
 - a. Each bin must be formatted [value, value2] = "bin name"
 - i. If formatted improperly, the question will not publish to predict
 - ii. No commas can be used in the values of the bins
 - iii. The quotation marks are unnecessary to designate the name
 - E. Tuning: once published, the creator may wish to set the number of significant digits for display.
2. *Phrasing guidelines.* A scaled continuous question is represented internally as a binary question.
 - A. As for any question, the question creator provides question text. The text should follow scaled continuous question wording conventions. Whereas a binary question about a quantity should be phrased as, e.g.: "Will xx quantity exceed xx?", a scaled continuous question is phrased as, e.g.: "Forecast the value of xx."
 - B. The question writer provides text to replace the standard "What is the probability of this happening?" text displayed with standard binary questions. For example: "What will sales be in millions of dollars?"

Formatting Guide and Fine Points

Please refer to last year's annual report to read more about our style guide or to the Style Guide or QM HOWTO manual.

Lessons Learned

The resolution source is by far the most critical portion of a question and must be rigorously crafted to ensure there is no ambiguity when resolving it. Each user who reads a question will have his or her own interpretation of it and what could constitute a resolution; it is the job of question management to minimize these instances. For example with a question about Philae, "When will the Rosetta's small robotic lander, Philae, land successfully on the surface of a comet?" "Successfully" was defined as Philae touching down at



the intended comet with the lander intact and able to execute its first step of firing a harpoon to anchor the lander. While the lander did land on the comet, it was unable to fire the harpoon, and, therefore, the landing was deemed unsuccessful. The fine print allowed site admins to clearly resolve the question without significant complaints from users as the criteria was clearly defined. However, if this stipulation were not made, then the landing of Philae could easily have been debated and would have led to a necessarily arbitrary decision.

Adherence to a consistent style guide is necessary to ensure high-quality questions. Concise grammar, diction, and unambiguous explanations are absolutely necessary.

FUSE Questions

Overview

Current horizon scanning is manual and ad hoc, consumes substantial expert time, receives little systematic validation, and tends to be narrowly focused. The combination of narrow focus and infrequent updating creates vulnerability to technical surprise. The Foresight and Understanding from Scientific Exposition (FUSE) program aims to reduce technical surprise through early detection of emerging scientific and technical capabilities through automatic scanning of full-text scientific, technical, and patent literature.

Both SciCast and FUSE fall under the IARPA Forest program, providing a natural opportunity for collaboration. IARPA directed FUSE performers BAE and SRI to provide SciCast questions inspired by automated horizon scanning. The SciCast question management team coordinates closely with FUSE performers to evaluate, edit, and publish FUSE forecast questions. This collaboration has created the necessary channels used when addressing user questions/comments in addition to providing data and measurements feedback to FUSE performers. As of April 2015, **664** questions from FUSE have been posted on the SciCast prediction market: **322** from BAE and **342** from SRI. That is just over 50% of the **1,248** valid questions published since SciCast began.

Activity

Here we present basic activity statistics for both FUSE and non-FUSE questions. Table 15 shows that more than half our questions came from FUSE, and those questions were, on average, about 1/3 less active than non-FUSE questions. Table 16 breaks activity down by topic; each question counts in all of its topics. Six of the fifteen topics claimed more than 100 questions: Biology & Medicine (378, 43% FUSE), Engineered Technologies (289, 85% FUSE), Business of S&T (204, 27% FUSE), Information Systems (168, 35% FUSE), Global Change (134, 44% FUSE), and Energy (124, 52% FUSE). The smallest topic was Agriculture (26, 11% FUSE).

For a complete list of FUSE questions and related information, please see Appendix I: SRI Copernicus Questions on SciCast and Appendix II: BAE ARBITER Questions on SciCast (Table 20, Table 21, and Table 22) and/or the file QM_FUSE_Questions.csv.

Table 15: FUSE and non-FUSE activity

	FUSE	Non-FUSE
Number of Questions	664	584
Number of Forecasts	53,320	60,675
Avg. Forecasts/Question	80.3	103.9

Table 16: Question activity by topic, FUSE & non-FUSE. Questions count for each of their topics.

		# Questions	Trades	% Trades
Agriculture	Fuse	8	304	10.96%
	Non	18	2470	89.04%
	Subtotal	26	2774	1.53%
Biology & Medicine	Fuse	190	12479	43.11%
	Non	187	16471	56.89%

SciCast Annual Report (2015)

• • •

	Subtotal	377	28950	15.93%
Business of Science	Fuse	85	6156	25.83%
	Non	117	17681	74.17%
	Subtotal	202	23837	13.12%
Chemistry	Fuse	61	5078	74.91%
	Non	12	1701	25.09%
	Subtotal	73	6779	3.73%
Computational Sciences	Fuse	50	2531	36.82%
	Non	37	4343	63.18%
	Subtotal	87	6874	3.78%
Energy	Fuse	107	10571	91.60%
	Non	17	970	8.40%
	Subtotal	124	11541	6.35%
Engineered Technologies	Fuse	173	13669	49.42%
	Non	115	13989	50.58%
	Subtotal	288	27658	15.22%
Global Change	Fuse	54	5699	46.75%
	Non	80	6491	53.25%
	Subtotal	134	12190	6.71%
IEEE Spectrum	Fuse	5	641	6.66%
	Non	30	8990	93.34%
	Subtotal	35	9631	5.30%
Information Systems	Fuse	79	6880	36.11%
	Non	89	12175	63.89%
	Subtotal	168	19055	10.48%
Mathematics	Fuse	24	1462	46.49%
	Non	16	1683	53.51%
	Subtotal	40	3145	1.73%
Physics	Fuse	73	3987	70.28%
	Non	9	1686	29.72%
	Subtotal	82	5673	3.12%
Social Sciences	Fuse	12	1738	24.83%
	Non	14	5262	75.17%
	Subtotal	26	7000	3.85%
Space Sciences	Fuse	76	6744	71.36%
	Non	23	2707	28.64%
	Subtotal	99	9451	5.20%
Transportation	Fuse	21	3050	42.45%
	Non	39	4135	57.55%
	Subtotal	60	7185	3.95%
Total Questions	Fuse	1105	80989	44.56%
	Non	726	100754	55.44%
	Total	1831	181743	

*The %Trades for the subtotals are calculated against the total number of trades

Questions with >100 forecasts tend towards general topics or topics in the popular press, such as hurricanes, nanotechnology, health, and consumer electronics.

BAE ARBITER System

Courtesy of Dr. Olga Babko-Malaya, BAE

Under the IARPA FUSE program, the BAE Systems team has developed ARBITER (Abductive Reasoning Based on Indicators and Topics of EmeRgence), an automated system whose purpose is to identify and characterize emerging technologies and emerging fields in science. ARBITER processes very large collections of scientific publications and patents and identifies trends, associations, and predictions more rapidly than with current methods. The system is extracting information from the metadata and text of publications and patents, identifying authors, their affiliations, addresses, as well as classifying types of organizations and publications. Moreover, it applies natural language processing techniques to extract scientific terminology from the full text of the documents, to identify different types of relationships between citations, authors, terms, and organizations, including contrast, opinion, and related work, and to characterize maturity and other properties of terms based on their contextual patterns.

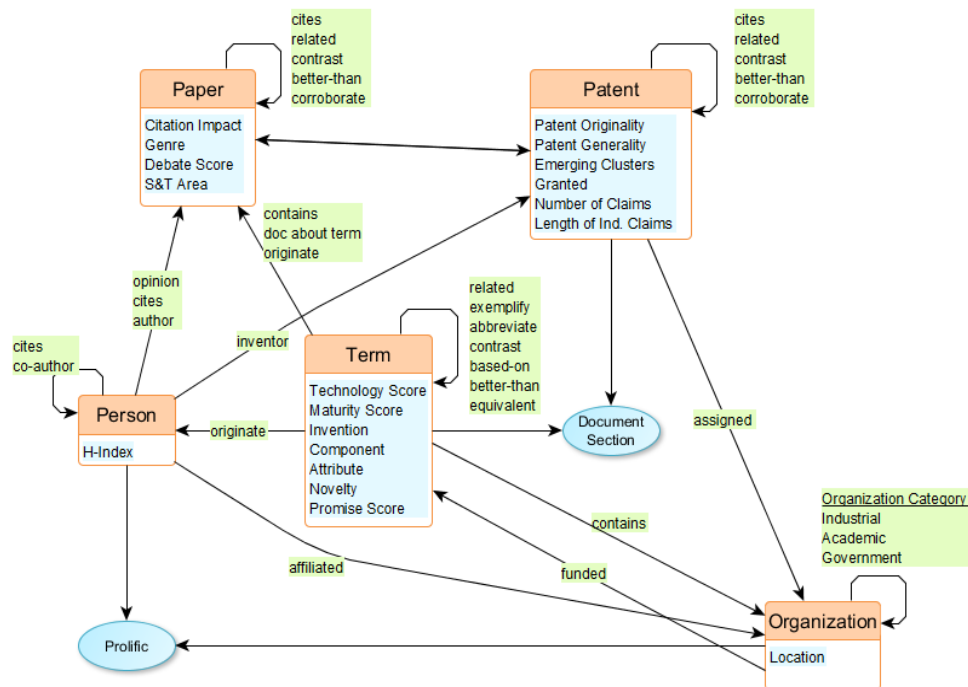


Figure 34: ARBITER Network Extracted from Meta and Text

This heterogeneous network follows actant network theory, which posits that human actors do not interact in a vacuum, but rather in the context of multiple non-human actants, as described in (Brock et al., 2012). Using this network, BAE Systems developed over 250 indicators that measure different characteristics and changes in the network associated with particular technologies and concepts. Examples of these indicators include growth of term usage by prolific people and organizations, slope of term usage as a component of another technology, or frequency of term usage as an invention. The forecasting system then combines these indicators via Bayesian networks to predict future prominence or scientific and technology terms, as measured by a significant



increase in term usage. To support an analysis of indicator hypotheses, BAE Systems has developed an infrastructure for automated model generation and optimization, which enables rapid development and validation of extensive lists of indicators, as well as generation of a large space of models that apply across different technologies, entities, and forecast gaps.

The results of applying indicators to forecast prominence of technology terms are described in (Babko-Malaya et al., 2015) and the BAE Indicator Report. In prior work, ARBITER indicators were also applied to characterize communities of practice (Babko-Malaya, Thomas, et al., 2013), identify the presence of the debate in the community (Babko-Malaya, Meyers, Pustejovsky, & Verhagen, 2013), as well as determine whether practical applications exist for research fields (Thomas, Babko-Malaya, Hunter, Meyers, & Verhagen, 2013).

Summary Statistics

As of May 8, 2015, BAE Systems published 321 questions with the following trading statistics:

Number of Questions Published: 321
Number of Questions Resolved: 162
Average # Forecasts per Question: 95.6
Average # Unique Forecasts per Question: 36.32
Average # Forecasts/Day by Question: 0.59

Lessons Learned

Our approach to question generation is a mixed-initiative framework consisting of two steps: (a) the ARBITER system automatically identifies terms that are potentially interesting; and (b) human analysts explore the space of options to associate these terms with anticipated scientific breakthroughs in order to generate questions. In our experience, the use of ARBITER in step (a) is very useful for focusing the efforts of analysts on interesting technologies, especially technologies that may not otherwise have drawn their attention. However, step (b) is still a significant task, particularly in developing interesting questions with sensible targets, so the concept of end-to-end automation of question generation is still some way off.

A number of components of ARBITER proved to be particularly useful in step (a) of the question generation process. Using our terminology extraction tool, we generated terms that are characteristic of over 20 scientific disciplines and technology areas, a total of over 120,000 terms. We learned that 15-20% of terms have positive prominence (i.e. are forecast to have increased future usage) and these terms are generally good candidates for potential question topics. Beyond this, we also found that terms that score highly for certain ARBITER indicators, notably indicators based on semantic relations, are particularly likely to be useful for question generation (e.g. 95% of terms with an abbreviation attached were found to be useful). We also used our related term generation tool to identify terms that are related to a given topic. We learned that when the tool was available to analysts, they used related terms in 57% of questions generated.

SRI Copernicus System

Courtesy of Dr. John Byrnes, SRI

SRI's Copernicus system tracks both content and metadata from tens of millions of published scientific papers and US patents. The system computes a number of indicators specific to the use of scientific terminology within a discipline, including information about volume of citation to documents containing the terminology, use of the terminology over time, and associations to other terms. Features are derived from time series of these indicators, and historical values of these features are used to train a statistical model that estimates future increases in the use of the term, computed as "prominence".

The system presents a list of terms to the user, sorted by prominence. Users can select terms from the list or enter arbitrary terms into the user interface, and they can then view the indicators used to compute the terms, see scores indicating how much each indicator contributed to the prominence score, and browse documents relevant to the term being investigated.

The Copernicus system is described in detail in the SRI Indicator Report delivered to the FUSE program.

Summary Statistics

As of May 1, 2015, 350 SRI questions have been published to SciCast Predict, with the following activity statistics (activity as of May 8).

- Total questions published: 350
- Average number of forecasts per question all time: 76
- Median number of forecasts per question all time: 36
- Average unique forecasters per question all time: 26
- Median unique forecasters per question all time: 15
- Questions with 15 or more unique forecasters: 178
- Number of questions that have resolved: 110

Lessons Learned

We learned that a system like Copernicus allows a question writer to very quickly arrive at a specific technology area to research for question writing. It does not speed the process of question writing beyond that point, however. Thus, the use of the system in the context of ForeST is primarily as a discovery tool more than as a research tool. Once the core terms and relevant documents are known, searching for more information on the area is very different from predicting future prominence, which is the primary task of Copernicus.

The ability to do custom search on a corpus, such as searching for mentions of scientific measurements or experiment results, was deemed likely to accelerate the question writing process. In part this is because in order to ask about a future metric, one needs to know what the current value of the metric in the community is. We were not able to automate extraction of such mentions during the ForeST project, so we have not tested this hypothesis.



We also found that writing broad, generally accessible questions was much easier than writing questions that were specific to a narrow scientific specialization. We do not anticipate the Copernicus system to be a strong discovery tool in the practice of writing broad questions, as a typical news website seemed to work just as well.

User Experience (UX) Design

In Y4, we continued to iterate on the SciCast User Experience. Key developments focused on conditional edits, user-added links, network visualization, and a new look and feel for possible transition sites. We also made various modifications to SciCast Spark to account for additional capabilities in Predict.

Conditional Edits

We made two significant changes to conditional edits (forecasts) this year: re-ordering when a user selected a conditional edit and consolidating input for multiple edits. With these two changes, the number of conditional edits increased this year (independent of incentive programs.)

Conditional Edit Re-Ordering

Initially, we asked a user to complete a conditional edit by following the logic: “If X, then Y” where X was the base question and Y was the conditions related to X. Here’s a screenshot of a user making a conditional edit *after* making a forecast in a base question:

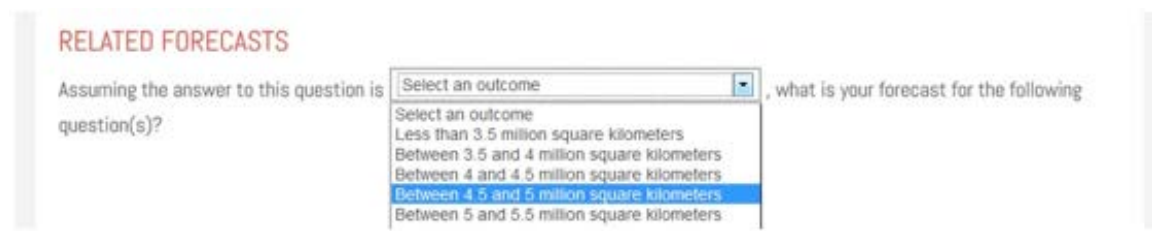


Figure 35: Initial SciCast conditional edits screen.

In an effort to increase conditional edits, we switched the order to first make the assumption, then make a forecast to a base question:

- 1 If: the answer to one of these related questions is:

Your Assumption	Question
<i>TIP: Forecast different scenarios using these assumptions. If an answer occurs, what will happen in the question asked below?</i>	
Select One: ▾	When will a vertical takeoff-vertical landing rocket booster that had previously delivered a payload into orbit successfully be reused in a second orbital launch?

- 2 Then: When will SpaceX successfully recover a Falcon 9 rocket booster on a solid sea platform after a mission launch?

POSSIBLE ANSWERS AND CURRENT CHANCE

Before April 30, 2015	0%
Between May 1, 2015 and June 30, 2015	30%

Figure 36: Redesigned conditional edits screen.

Choice of Conditional Edits

Another improvement we sought to make was to allow a user to easily select which condition to assume. We consolidated any possible conditional edit in to a table. This layout will support multiple conditions when we choose to enable that in the UX. For the moment the UX only supports a single condition at a time. This was done both to keep forecasters on higher-probability scenarios, and because the engine can only entertain multiple assumptions if they are in the same clique, and that introduces a complication for the UX. However, the advent of user-added links opens the path to a previously-unavailable solution: expand the clique on demand, for a cost.

- 1 If: the answer to one of these related questions is:

Your Assumption	Question
<i>TIP: Forecast different scenarios using these assumptions. If an answer occurs, what will happen in the question asked below?</i>	
Select One: ▾	Will a solar-powered plane circumnavigate Earth before the end of 2015?
Select One: ▾	Will Solar Impulse 2 complete leg 9 of its circumnavigation by May 31st, 2015?

- 2 Then: How many legs of its journey around the globe will Solar Impulse 2 have completed by May 31st, 2015?

POSSIBLE ANSWERS AND CURRENT CHANCE

6 or no change	25%
----------------	-----

Figure 37: Showing a choice of assumptions. Although the engine allows multiple assumptions, the UX has for now been deliberately held to one at a time.

User Added Links

This year we gave users the ability to create links. To avoid too many errant additions, we only grant this capability to users who choose to make edits in “power mode.” We also require a certain sized trade to accompany the additional link to confirm the “seriousness” of the user to make the addition. Here is how it works:

First, the user is presented with the ability to add a link:

- 1 If: the answer to one of these related questions is:

Your Assumption	Question
<i>TIP: Forecast different scenarios using these assumptions. If an answer occurs, what will happen in the question asked below?</i>	
+ Add another assumption from our existing questions OR suggest a new related question	

Figure 38: Adding links, Step 1, “Add another assumption”

Then they are presented with a list of recommended questions. Searching allows the user to access all possible questions.

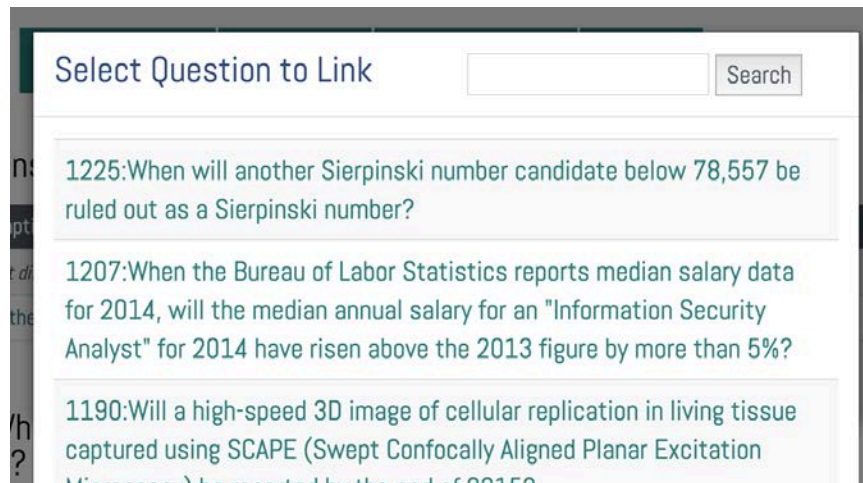


Figure 39: Adding links Step 2, search/select question

Once a question has been selected, it shows up in the list of possible assumptions, just as it would appear otherwise. We highlight it however, to show that it is one of the assumptions that has been newly added.

1 If: the answer to one of these related questions is:

Your Assumption	Question
<i>TIP: Forecast different scenarios using these assumptions. If an answer occurs, what will happen in the question asked below?</i>	
Select One: ▾	When will another Sierpinski number candidate below 78,557 be ruled out as a Sierpinski number?
+ Add another assumption from our existing questions OR suggest a new related question	

Figure 40: Add links step 3, new item available for assuming

The user can then make an edit using this assumption but they are warned they must make a trade of a certain side before the assumption will be permanently added to the network.

Say a few words about why you're making this forecast:

Forecast must be less than 0% or greater than 7% to keep the new assumption you added.

SUBMIT YOUR FORECAST

Figure 41: Add links step 5, make a large enough change

Network Visualization

In addition to listing direct links, we wanted to show broader network relationships. We also wanted to make the diagram interactive, so a user could click on any node and make a forecast in that node. Figure 42 shows what the network diagram looks like for a particular (large) subnet.



Figure 42: The network diagram for the flu subnet.

The red dot indicates the currently-selected question. Each node is clickable; clicking a node makes that question the active forecasting question.

Users can also choose to see where they are in the context of the entire network by clicking on a checkbox to show the full network. In the full network diagram, one can hover over any node to see the question name. Once again, the current question is highlighted red.

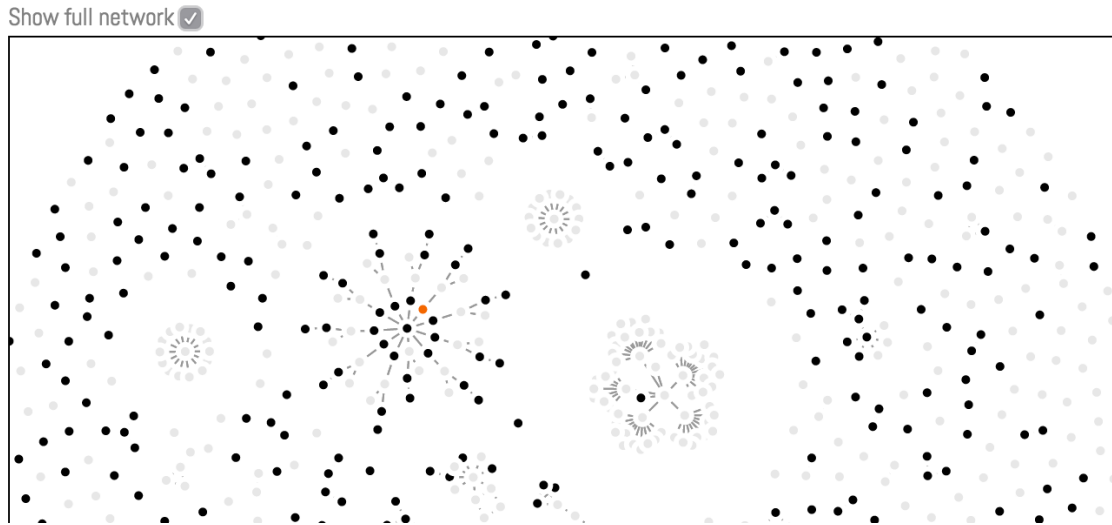


Figure 43: Portion of the full network diagram, featuring the flu subnet.

New Look and Feel for Transition Sites

As part of our transition efforts, we created a new look and feel for future SciCast sites. We chose a new, more muted color palette and changed the fonts to give us more flexibility in typography. We believe the new look and feel could be adopted across a broad range of potential topic areas.

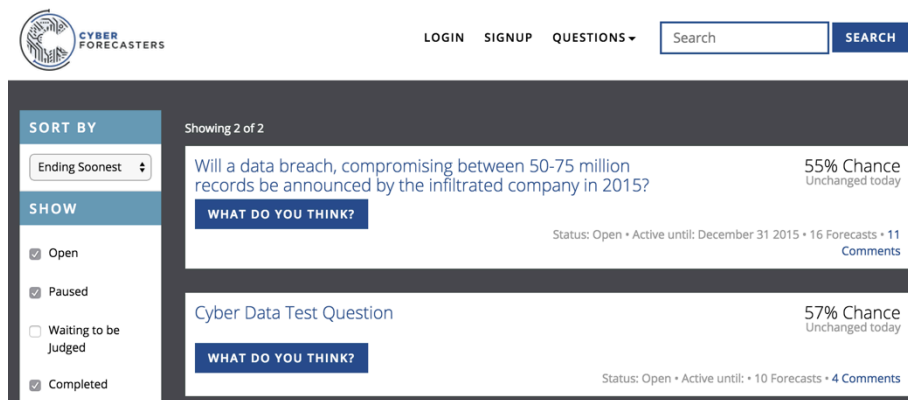


Figure 44: Cyberforecasters site with the new look and feel

Executive Dashboard

The Executive Dashboard (Figure 45) is a central place to understand the performance of the prediction market over time. The interface is clickable and selectable so the user can either analyze the entire market or deep dive into specific questions, categories, traders, or time periods.

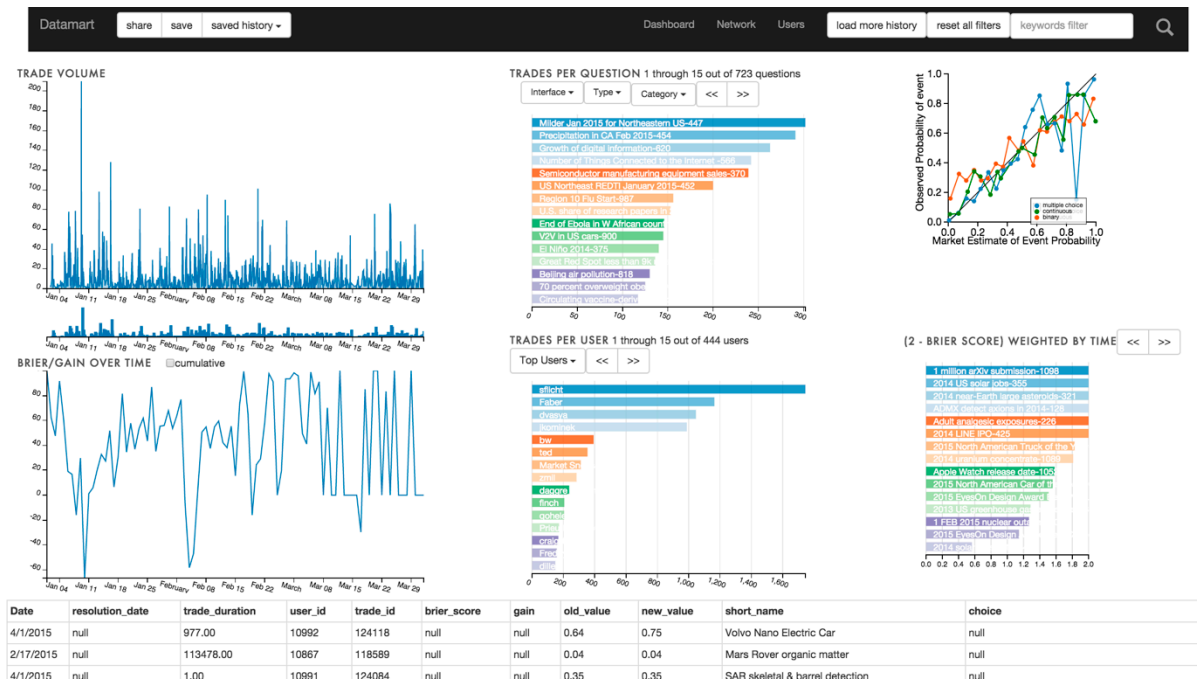


Figure 45: Executive Dashboard main screen showing (clockwise from top left) activity, most active questions, calibration curves, question accuracies, most active users, and Brier gain/loss.

For instance to understand what the top 10 users in the market were trading on in the month of February:

1. select Top 10 from the drop down
2. select and drag the date range to cover the month of February

Suppose the Top 10 users are very active in a specific bitcoin question. To zoom back out and focus on that one question, simply:

- click on the bitcoin question in the trades per question graph.
- set the "Top Users" drop down to all users
- adjust the date range slider as needed

The graphs update to reflect the current filters. The combination of filters gives you a very large set of ways to slice and dice the market data.

Network The network tab (Figure 46) provides a spatial layout of all the questions in the market. Current links are shown as solid lines, with link strength overlayed as a red line of varying thickness: thicker lines show stronger links. The dashboard relies on the link strengths reported by the Markov Engine (see next section); right now it uses mutual information. Links to resolved questions, or manually deleted links are shown as dashed gray lines. Hovering over a node shows the question name, and clicking on a node loads that trading

page in Predict. Future work includes merging this display with the one shown to forecasters on the question's own "Network" tab, and adding toggles to hide resolved questions.

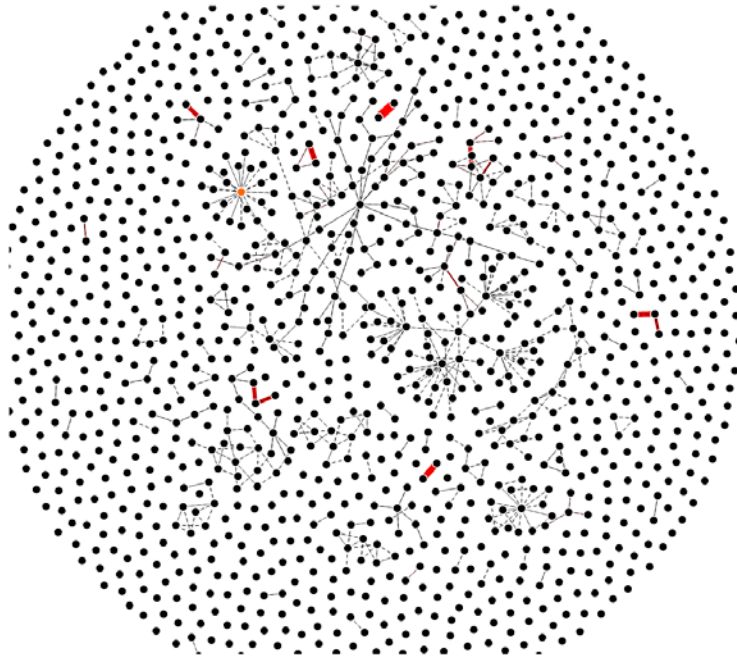


Figure 46: Network graph with strong links in red.

User Interaction: It is both interesting and potentially important to track forecaster interactions. For example, especially during contests it is important to discover if multiple users are artificially colluding with each other or are in fact controlled by the same person. To facilitate this kind of analysis the dashboard has a user interaction report, shown in Figure 47.

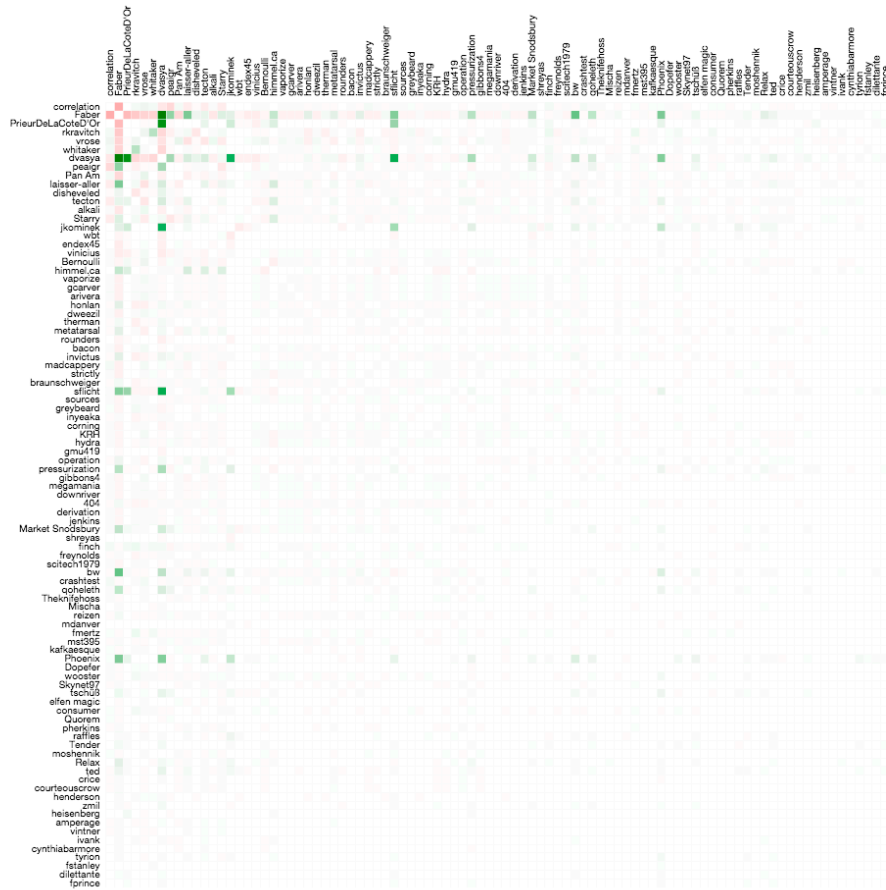


Figure 47: User Interaction Matrix with the most suspicious interactions in red at the top left

Red cells indicate that there was a large transfer of points between the two respective users on questions they both traded on. Green cells indicate both users benefited or lost on their intersecting questions. White cells indicate that the users tended to not trade on the same questions.

Thus red cells are a possible indicator of fraud and should be investigated. Each cell is clickable and has a detail report, shown in Figure 48 This report shows common questions for the two forecasters along with total gains and losses, number of trades, and some timing information.

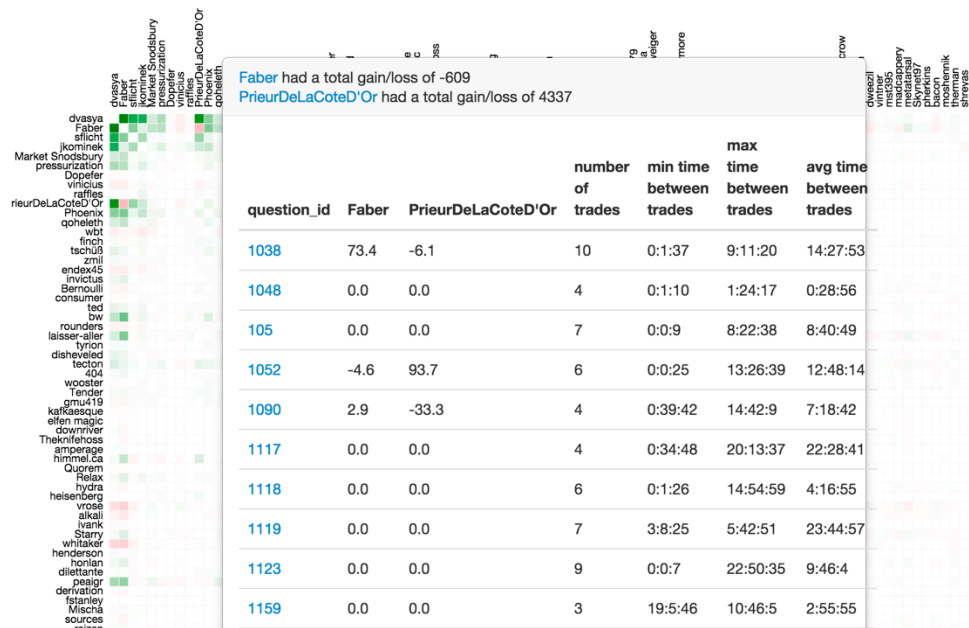


Figure 48: User Interaction Detail showing potential interactions between @Faber and @Prieur....

Software Tools and Data Resources

SciCast involves many separate processes. Figure 49 shows the high-level view. Figure 50 shows a more detailed formal architecture for the core and key dependencies of the core. The following list provides a brief explanation for each component, and the chapter expands key items like Predict, Spark, the Datamart, and the Recommender.

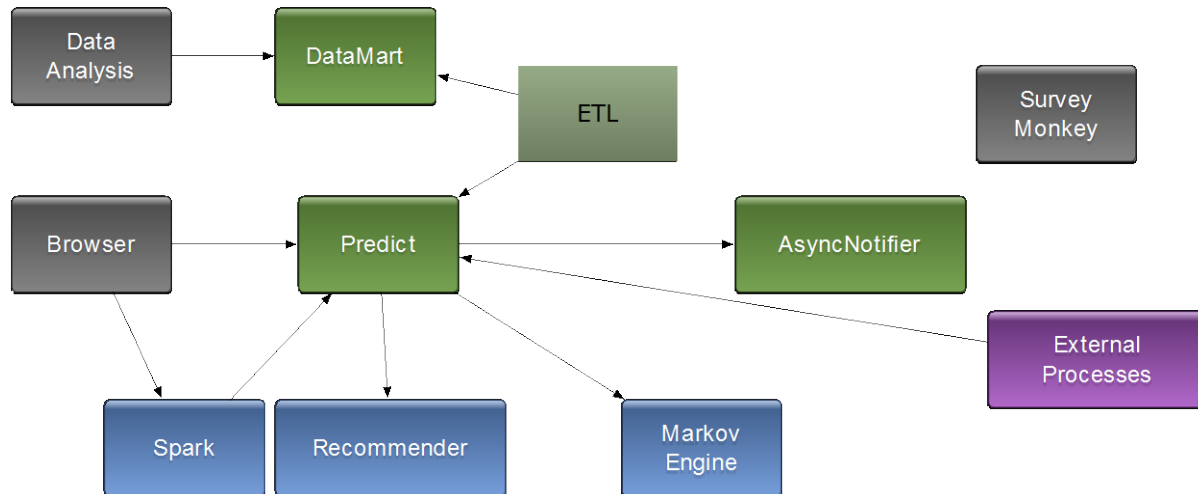


Figure 49: SciCast high-level architecture diagram. Green = Core; Drab Green = ETL nightly processes; Blue = Packaged Services; Purple = External SciCast processes; Gray = Foreign (not maintained as part of SciCast codebase)

1. **Predict** – Core services that provide basic capabilities to run the prediction site.
2. **Spark** – Question Management Capabilities that interact with end users and the Predict site
3. **Recommender** – A recommendation engine that allows selection and ordering of questions as appropriate to a user.
4. **Markov Engine** – The core inference computation engine. An inference engine (UnBBayes or other) that is used by the daggre server to compute probabilities of questions. Communication between Predict and the Inference Engine is done using RabbitMQ which is a AMPQ message broker.
5. **Datamart/ETL** – A datamart and Extract/Transform/Load (ETL) layer that provides a layer useful for data analysis without impacting the core system performance. Additionally, it provides time intensive services such as trend graph generation and leaderboard scoring. The ETL process copies from Predict to the data mart. The ETL processes have their own copy of the Predict and the Markov Engine so they can utilize the computational engine support.
6. **Asynchronous Notifier** – External Process that sends email notifications from rabbitMQ requests.
7. **Survey Monkey** – External service used for survey completion.
8. **External Processes** – Several external processes used to support system development, including:
 - a. **Incentive leaderboard** – A secondary leaderboard generator that is external to the system and run on demand.
 - b. **Digest emailer** – An external process which generates and send the (weekly) digests to relevant users.

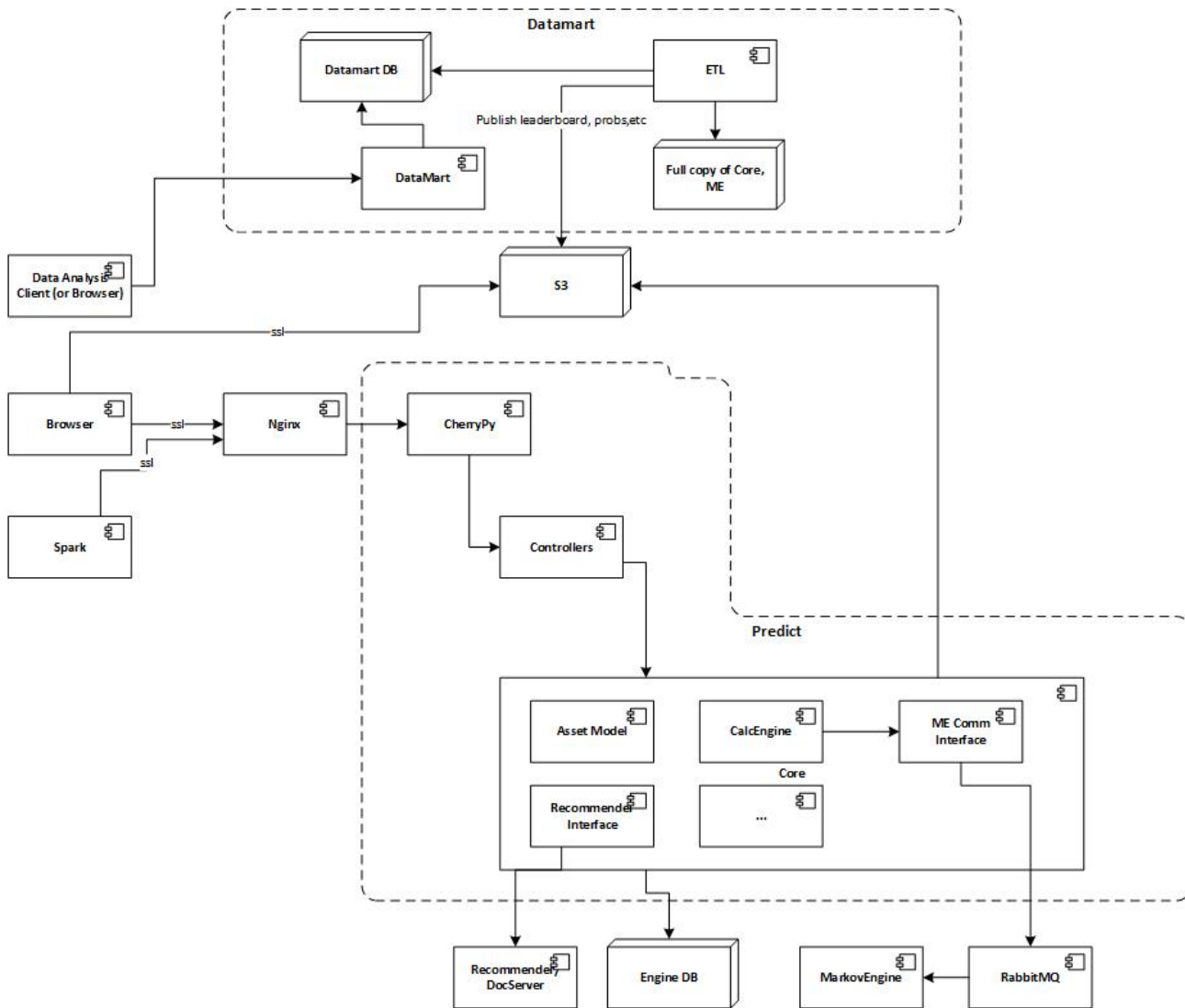


Figure 50: SciCast Architecture

Predict

Predict (formerly Daggre Server) is the core prediction market and web site. It has gone thru multiple iterations and significant restructuring as it evolved to meet the project demands. For details on the history of this evolution, see the document “Daggre Software Architectures”. A high level view of the evolutions is shown below, with “(Y1)” indicating Year 1 (the base year), and similarly for other years. The evolution is

shown as a series of captioned figures without additional narrative. Details for previous years can be found in the respective annual reports.

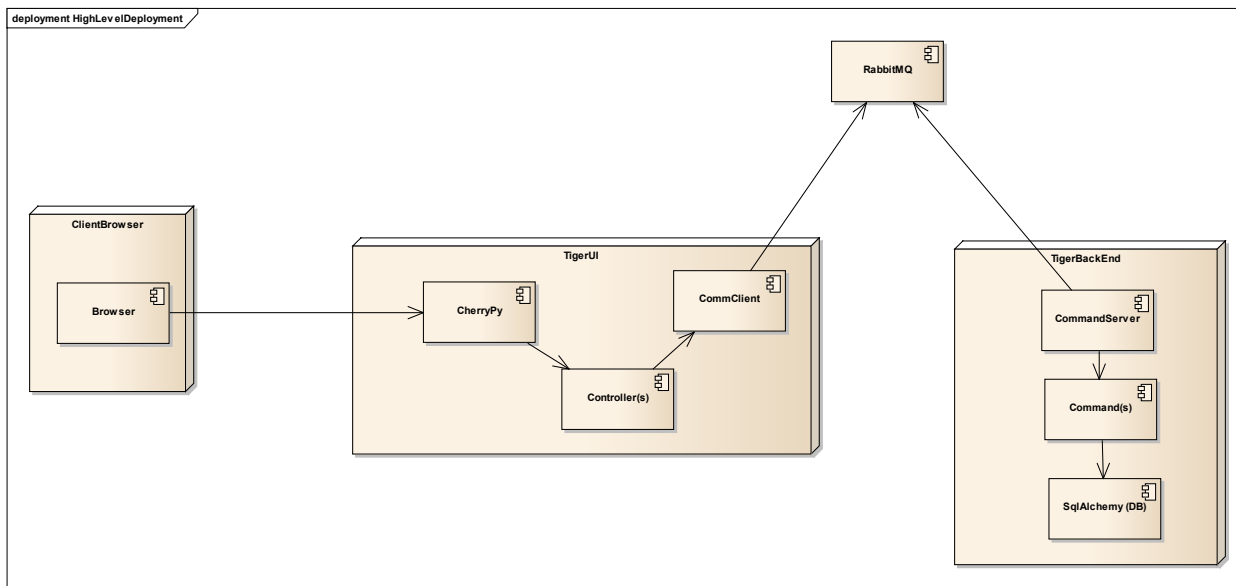


Figure 51: Initial architecture (Y1)

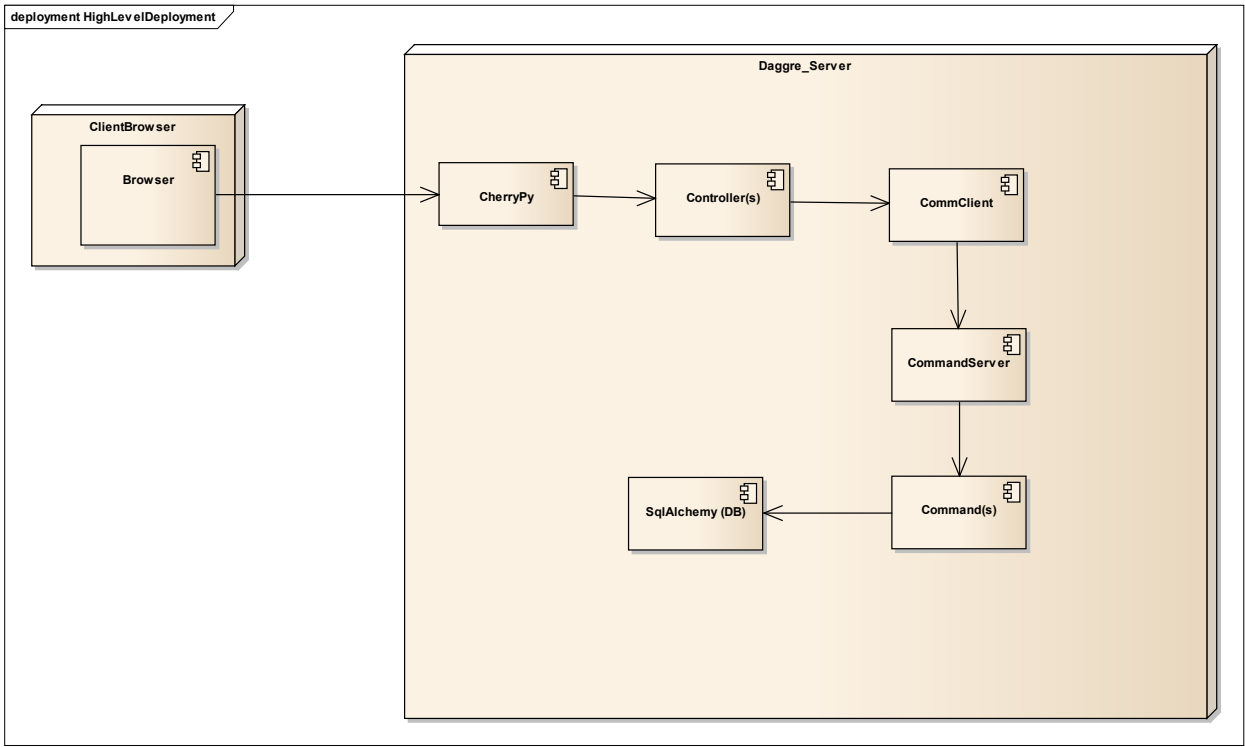


Figure 52: Code simplification / rewrite (Y1)

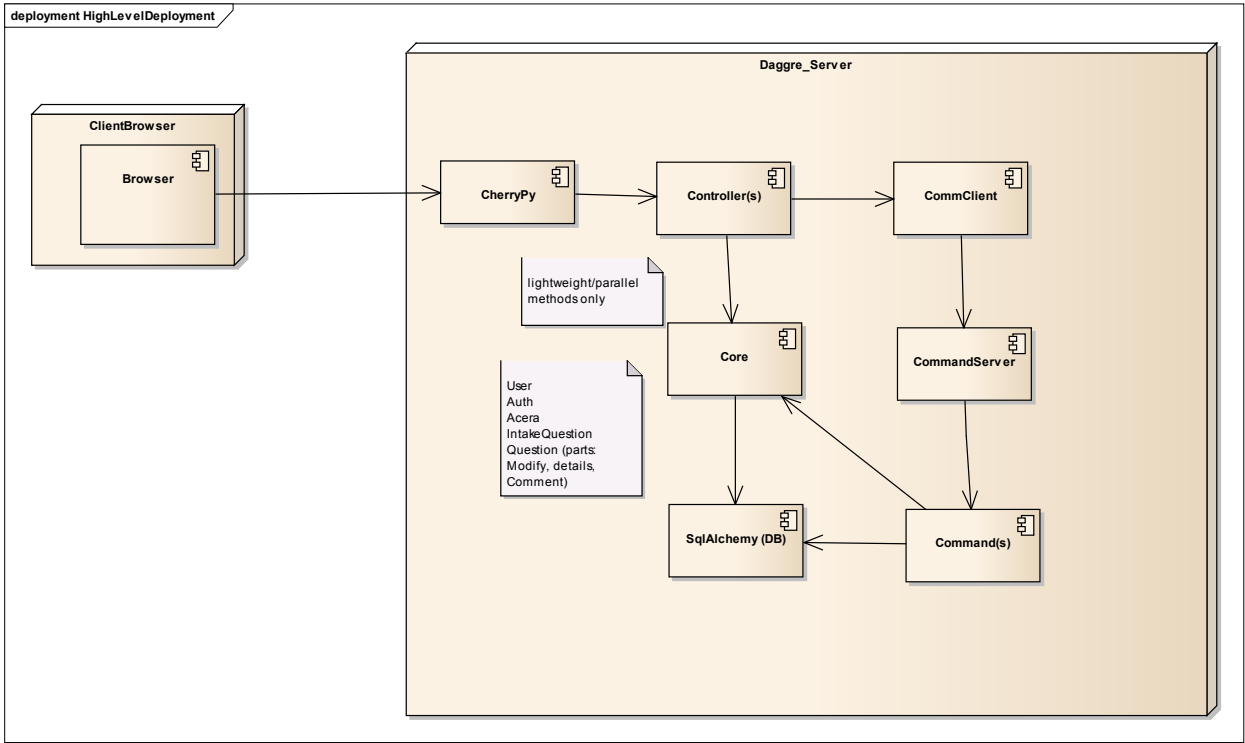


Figure 53: Development of Core objects and continued refactoring (scoring time reduced from 2.1 hours to 20 seconds) (Y1)

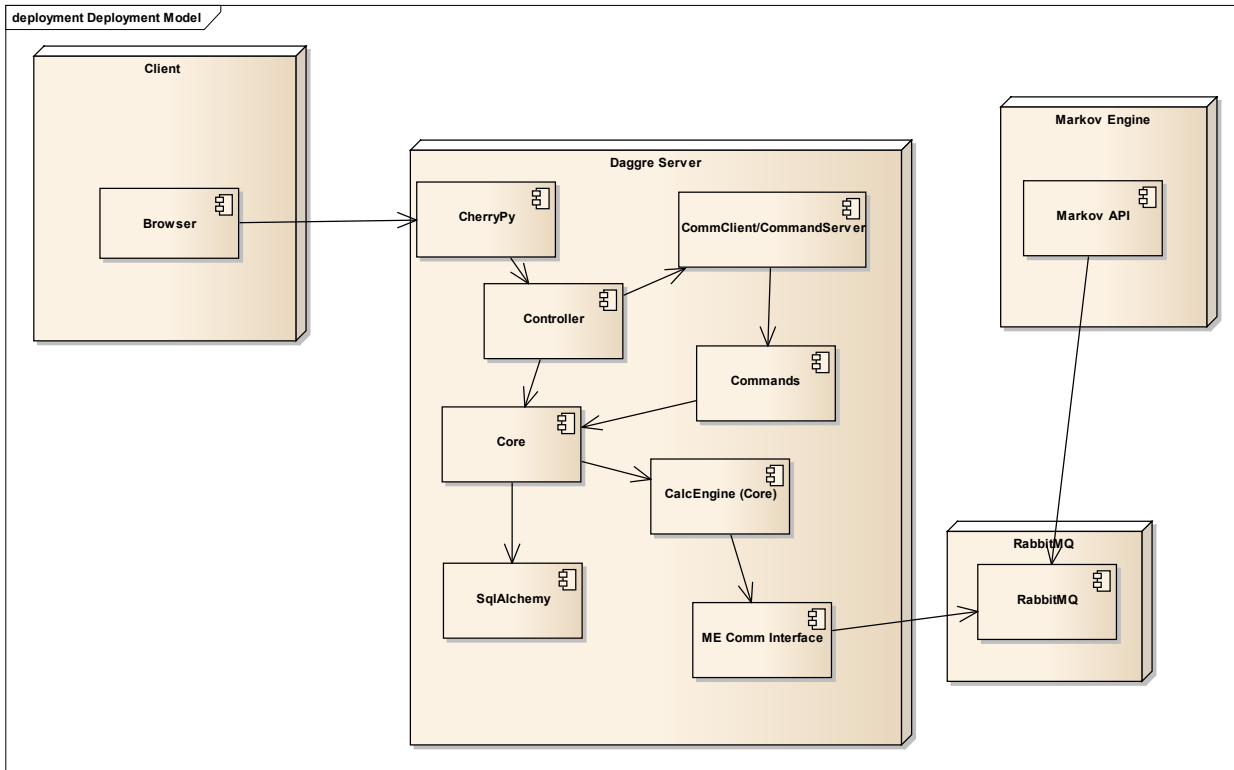


Figure 54: Markov Engine (UnBBayes extension) (Y2)

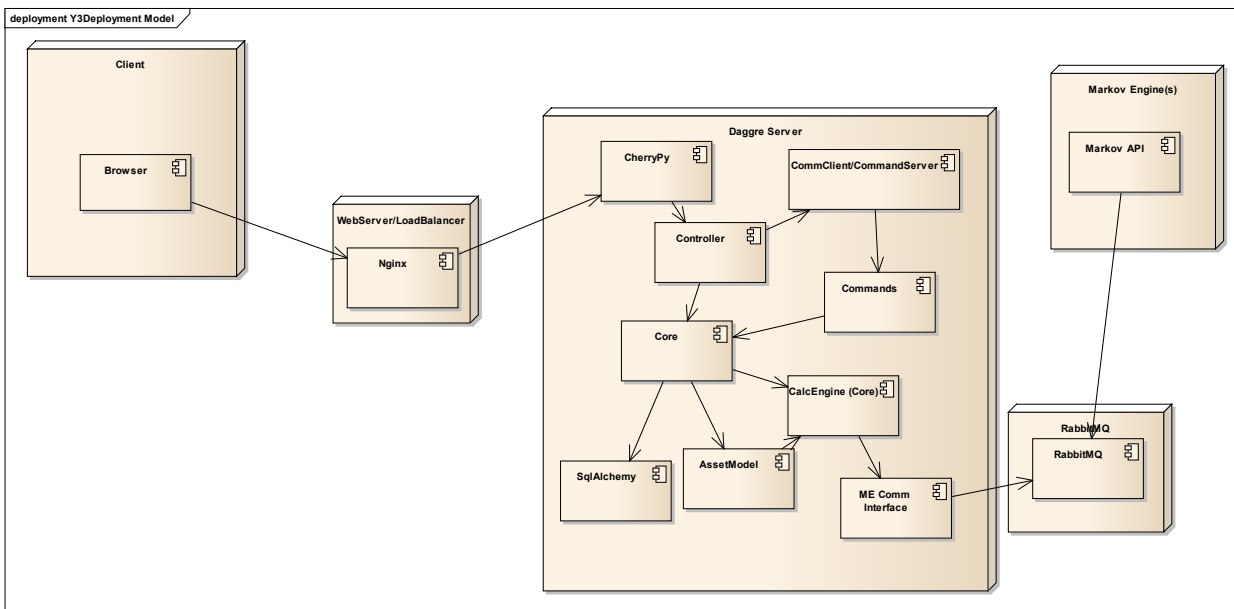


Figure 55: Redesign of Asset Model, Markov Engine, Performance Improvements (Y3-Y4)



For details on the installation of Predict please see the [Software Installation Guide](#) (separate enclosure, also available in Google Docs at

<https://docs.google.com/document/d/1B2Zy3uIep00wb9qWaBbx9vxtLGcmcSvz5pewLmnGbUI/>). SciCast has been successfully installed on Windows, Mac OSX, and Ubuntu Linux. Installation involves the core components of Predict as well as the 3rd party modules shown below.

3rd party modules used

System modules

1. rabbitMq
2. Erlang
3. Java
4. Python 2.7
5. Postgres
6. Nginx
7. NewRelic (optional)

Javascript modules

1. Grunt
2. Gulp
3. Angular 1.3+
4. Angularartics
5. Bootstrap
6. C3
7. D3
8. Es5-shim
9. JQuery
10. Json
11. Json3
12. ngAnimate
13. smart-table
14. underscore

Python modules

- sqlalchemy (<http://www.sqlalchemy.org/>) (0.7.8)
- python-dateutil version 1.5 (<http://labix.org/python-dateutil>)
- easy_install python-dateutil==1.5
- jinja2 (<http://jinja.pocoo.org/>) (2.5.5)
- formencode (<http://www.formencode.org/en/latest/index.html>) (1.2.4)
- cherrypy (<http://cherrypy.org/>) (3.2.2)
- py-pretty (<http://pypi.python.org/pypi/py-pretty>) (1)
- psycpg2
- sqlalchemy-migrate
- fixture



- RobotFramework
- boto
- requests
- pika
- futures-request
- pytz
- mock
- pysqlite
- bleach
- numpy
- oauthlib
- requests_oauthlib
- html2text
- twill
- httpretty
-

DataMart

The goal of the datamart has grown from simply providing an analytical copy of the Predict data to also housing the executive dashboard. The original rationale for a physically separate datamart was to ensure that large analytical queries did not impact the transactional performance of the Predict system. For this reason, the Datamart has a fully separate database instance and is designed to be off loaded to other servers, replicated or otherwise scaled if needed.

Enhancements introduced in Y4, including Shadow Trades (which record the side effects of trades on other questions), mean some of the additional tables added in the datamart schema could now be eliminated. One area for future consideration is to merge the schemas (and in low volume cases consider merging instances) with the Predict schema. The separation between Predict and Datamart schema and database instances introduces a tradeoff between the additional overhead of development and maintenance vs the scaling potential

For details on Datamart installation please see the [Software Installation Guide](#).

ETL (Extract, Transport, Load)

The ETL process is a (nightly) process with two responsibilities:

1. Copy data from Predict to the Datamart
2. Using a separate instantiation of the Markov Engine and Calc Engine compute leaderboard scores and other important statistics and publish them to Amazon S3 for other systems to use.

For details on ETL installation please see the [Software Installation Guide](#).

Spark

Spark is a Ruby on Rails application built to support the crowdsourced question development process. Spark currently runs on two AWS EC2 instances ("m1.small") and also requires Amazon S3 to store assets uploaded through the question development process.

Figure 56 shows Spark's architecture diagram.

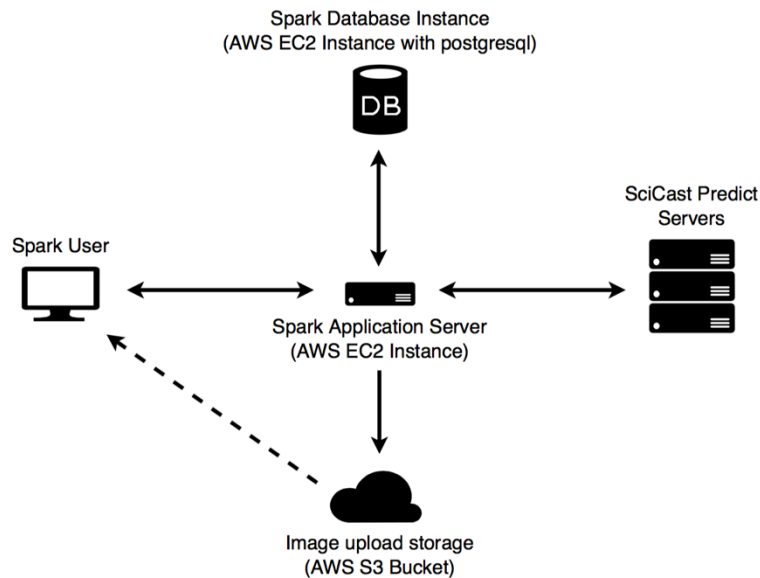


Figure 56: Spark architecture

SciCast/iOS

SciCast/iOS is an experimental native iPhone app compatible with ios >7.0 (best >=8.0) and devices iPhone 5S, iPhone 4S, iPhone 6 and 6 Plus, iPad Mini and iPad 2.0. The purpose of SciCast/iOS was to make the SciCast API more available to users. It delivers alerts to their mobile device, and helps them build API queries. Users can build and store as many queries as needed and run those queries on demand. A deployed version should facilitate automated forecasting and commenting, possibly helping with engagement and retention.

The following figures present sample screens. We remind the reader that this app remains in the experimental stages and was not released to users.

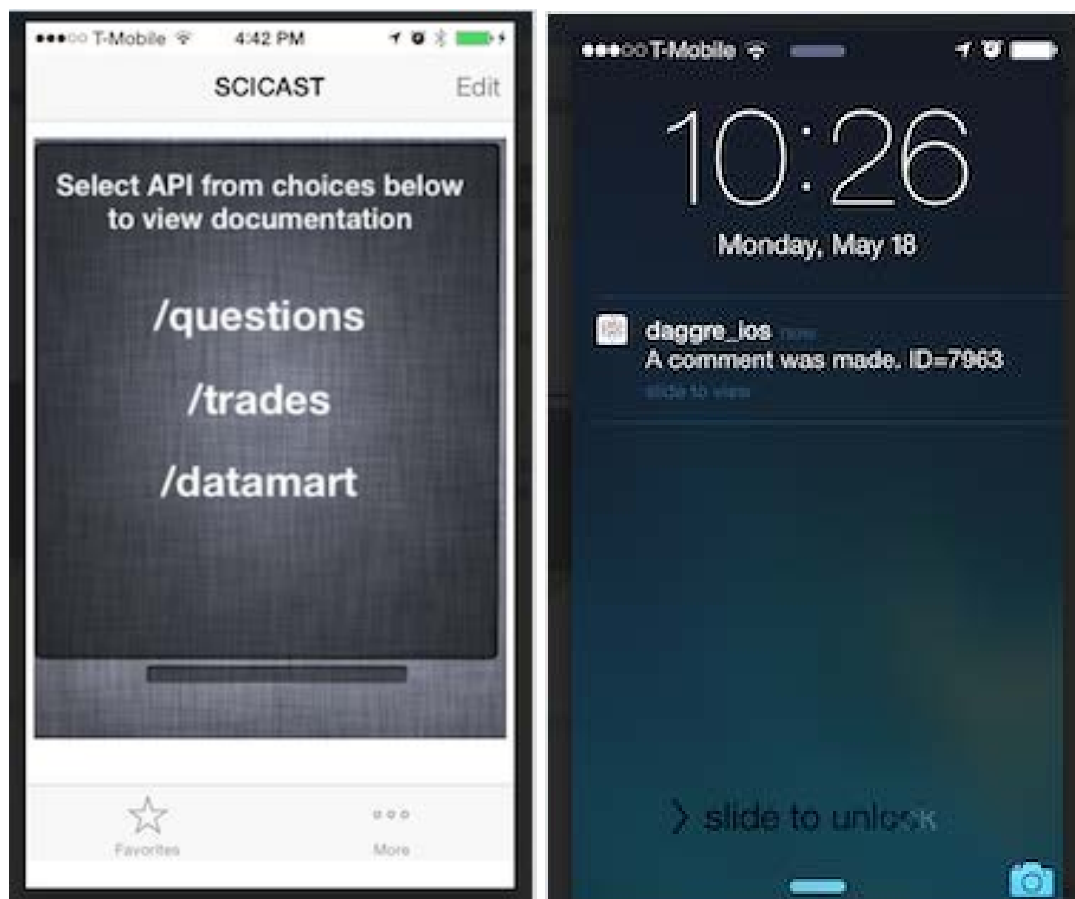


Figure 57: SciCast/iOS home screen and notification screen

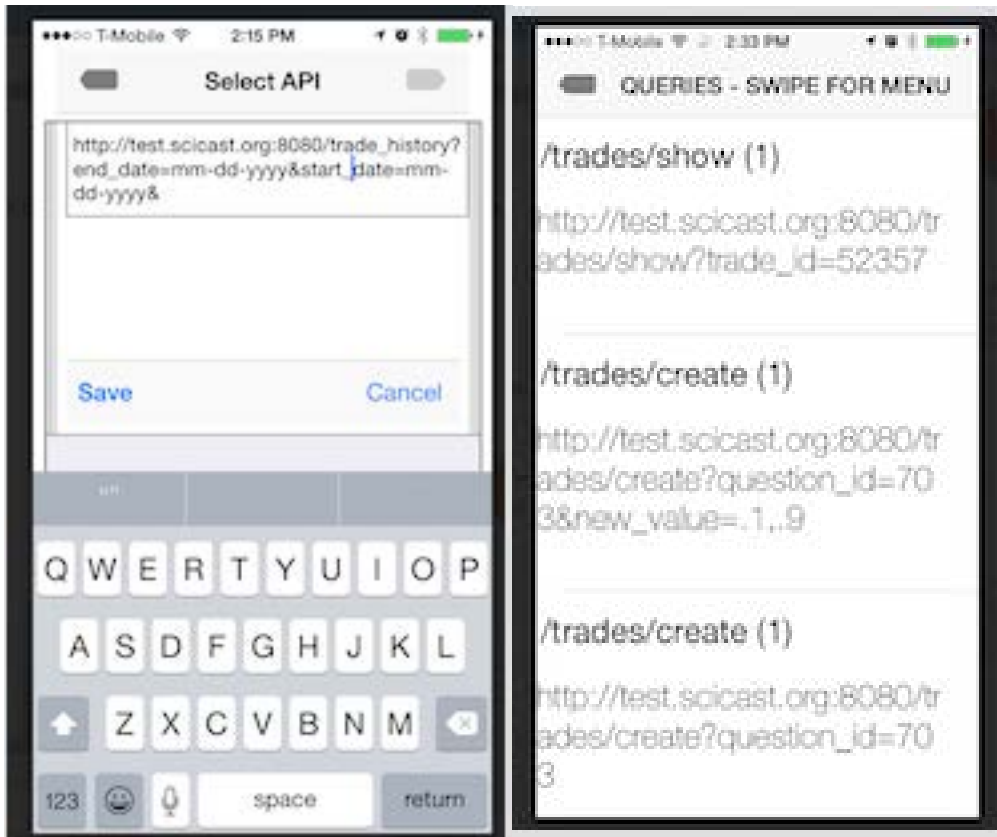


Figure 58: SciCast/iOS Query Builder and Query Selector

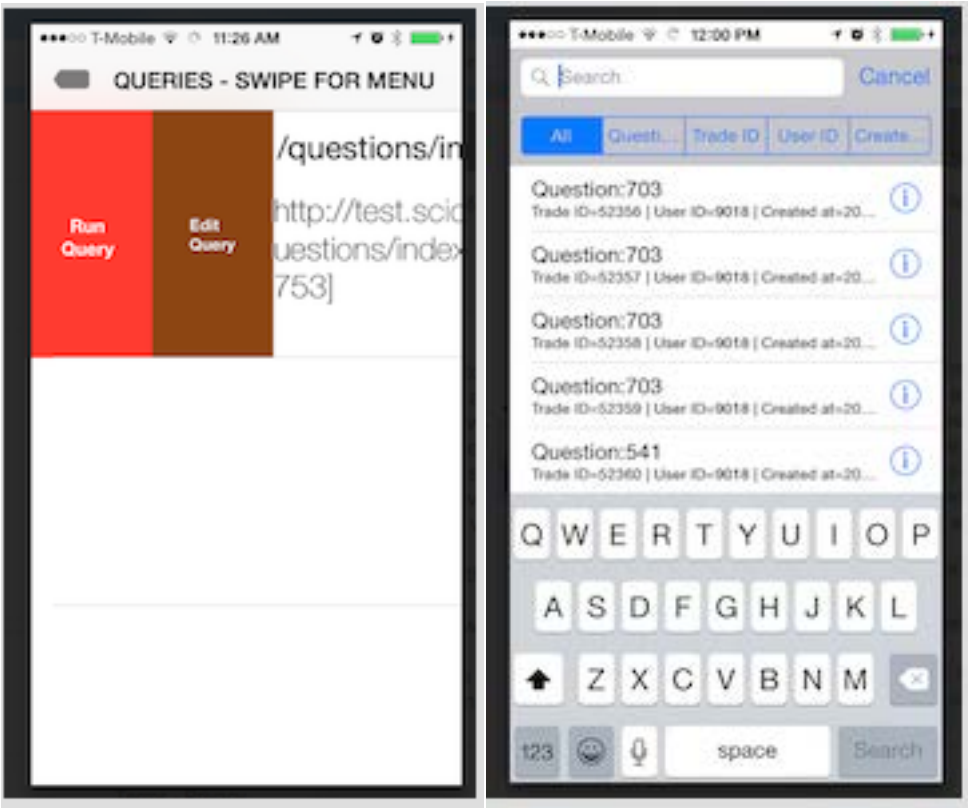


Figure 59: SciCast/iOS Query Manager and Search

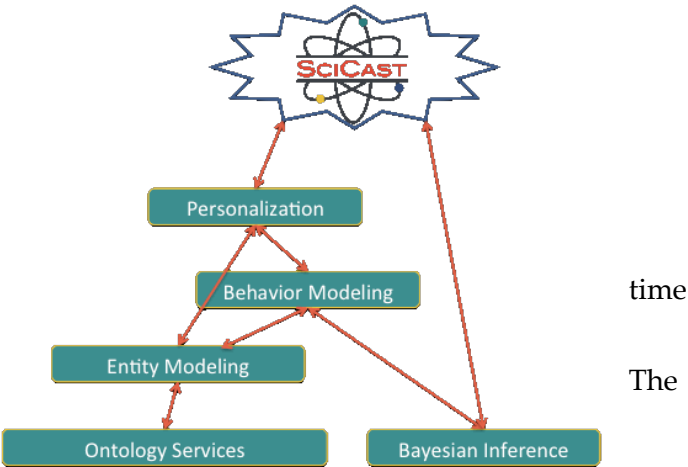
Recommender

SciCast uses Tuuyi’s recommendation services to learn user preferences and suggests other questions judged likely to interest a forecaster. Like any prediction market, SciCast requires an active user community. Recommendations can help encourage activity and sense of community by personalizing the presentation of questions, comments, and/or other users to follow. The current integration focuses on question recommendation in three places:

- when viewing the main page,
- after making a forecast, and
- in sorting any list of questions after a search or query.

Technical Approach

The Tuuyi recommendation system operates as a separately hosted SAAS service responding to real-requests from SciCast. An option to configure a local, private version of the recommender is also available. service is structured as three separate, publically



accessible, APIs (Ontology Services, Entity Modeling, and Personalization), as shown in the figure at right, together with two support libraries providing Behavioral Modeling and Inference. The state-of-the-art personalization engine integrates social (users who traded on this question also traded on that question) and semantic (this question is about medicine and technology) information with current user activity to identify, according to request parameters, relevant questions, trades, or other users.

Ontology Services provides basic named entity recognition and relationship services, recognizing over six million named entities, and provides the backbone conceptual namespace for "understanding" entities of interest. For example, the screenshot below (Figure 60 shows the result of an ontology search for the term "Higgs Boson". The search results indicate that the term participates in categories "Particle_physics", "Bosons", "Standard_Model", "Electroweak_theory", "Hypothetical_elementary_particles", and so on, and also indicates that Peter_Higgs and others are known for their work on the Higgs_Boson. Note that the indicated categories exist in a semi-lattice, so that Particle_physics, for example, is a sub-category of Physics (categories can be subcategories of multiple broader categories). The semi-lattice over categories allows the recommender to recognize that those interested in the Higgs Boson, or questions involving it, are more likely to be interested in questions involving Baryons, or Electroweak interaction than questions concerning Ebola.

The screenshot displays a web interface for ontology services. It includes input fields for 'Id', 'SimpleName' (containing 'Higgs Boson'), 'Phrase', and 'SciCast'. There are buttons for 'Fetch', 'Match', and 'PhraseLookup'. Below these are fields for 'Required Categories', 'Preferred Categories', 'Disfavored Categories', 'Excluded Categories', 'Reqd Types', and 'Excluded Types'. The search results are displayed as follows:

```
{id: 3047494 depth: 4 descendants: 0 Higgs boson}
relations:
  subject(-1) : [ Category:Particle_physics(7) Category:Bosons(5) Category:Standard_Model(7)
                 Category:Electroweak_theory(6) Category:Hypothetical_elementary_particles(5) Category:Mass(4)
                 Category:Article_Feedback_5_Additional_Articles(3) ]
inverse:
  knownFor(-1) : [ Peter_Higgs(4) Tom_W._B._Kibble(4) Gerald_Guralnik(4) C._R._Hagen(4)
                  Robert_Brout(7) ]
```

Figure 60 Tuuyi Ontology Services result for "Higgs boson"

Questions, as well as Users and Trades, are managed by the *DocumentServer*, which provides entity-modeling logi services for each of these items. It stores each of the three as a *Document* with associated ontological and social information. The ontological information stored is essentially the set of ontology terms extracted from



the unstructured text associated with each item (e.g., title, description, tags, and statement of interest for a user). Ontological information is unary – that is, it is static information associated with each item, and can be used by the recommender even when the item has not participated in any interactions (e.g., a new user with no trades, a new question no users have yet viewed, etc). Social information maintained by the DocumentServer, by contrast, is primarily *relational*. For example, trade pairs, that is, records of users who have executed a trade on one question and have also executed a trade on another question, are important social information maintained by the DocumentServer. At the Entity Modeling level the system remains largely agnostic with respect to the specifics of any entity, providing general storage capabilities for arbitrary entity-specific key-value data. However, significant customization and adaptation was needed to incorporate role information into the core retrieval services provided at this level. For example, questions reside in groups, and are only visible to users who share a group with the question.

Behavior Modeling uses a SciCast-specific user behavior model, operating at both the individual and community levels, to relate the current context (user, location within the SciCast website, etc) to projected future activity and potential interests. This model is built from stochastic modeling elements in the inference toolkit, including anytime abstraction, projection, and inference elements.

Finally, the *Recommender* service uses a "collection of experts" paradigm, integrating recommendations and assessments from both an ontological and social perspective to personalize browsing suggestions for the user.

Tuuyi Inference Engine

Objective

Tuuyi was tasked with working with Mason to understand the various accuracy and scaling inference challenges, analyze them, and provide proposed solutions as possible. Two challenges were identified as needing near-term attention: (1) conditional trades, and the resulting network complexity; and (2) structured outcomes. Tuuyi proposed, and Mason accepted, the use of its wholly owned software as part of an inference solution for SciCast, for both the challenge of inference on complex networks of questions related by conditional trades and that of structured domains.

Tuuyi implemented an API to integrate the Tuuyi inference engine with SciCast and demonstrated the potential for attractive performance scaling. According to Tuuyi internal tests, load time for large question/trade graphs, as well as queries of posteriors for all questions in the network were an order of magnitude faster than UnBBayes. In further development, Tuuyi successfully demonstrated efficient representation and evaluation of both conditional trades and structured outcome spaces (e.g., hierarchical outcome spaces, with both mutually exclusive (e.g. calendars) and non-mutually-exclusive leaves). Project priorities did not allow for further integration and evaluation of the approach. More details can be found in the Ordered Questions subsection under Experiments and Studies, above.

Technical Approach

The overall architecture of the Tuuyi approach to probabilistic inference relies on the minimization of the evaluation complexity of queries. It does this through application of algebraic operations of association and distribution on a query represented as an algebraic expression³. This approach has the advantage of being able to represent and exploit local structure within probability distributions described at the variable level, a capability not available in more traditional approaches. It also provides direct representation of complex structured outcome spaces such as occur in questions over calendars, organizations, or conceptual categories (e.g. scientific discipline). The implementation used in the integration and assessment includes:

- Support for the existing SciCast/UnBBayes API.
- Multiplicative local expressions (support for noisy and/or and logical and/or, as well as hierarchical domain representation (see section on Hierarchical domains).
- Support for serialization / deserialization of state.
- Support for incremental hard and soft observation
- Support for query of arbitrary sub-joints and conditionals.
- Support for pre-compiled computational structure, with incremental adaptation/extension as graph or query-set is extended.

UnBBayes Inference Engine

The UnBBayes Inference Engine is an instance of Markov Engine – the software component responsible for the computation of SciCast's market probability distribution. It was developed through customization and extension of UnBBayes, a plug-in based Java open source framework and GUI for modeling, learning and reasoning upon probabilistic networks. UnBBayes has been developed as a collaborative effort between University of Brasília and George Mason University. Prior to Year 3, it was also used as a software tool and library for managing the user's asset structures. Further description of UnBBayes can be found at its home page: <http://unbbayes.sourceforge.net/>.

In fact, there were multiple candidate probabilistic reasoning software packages that seemed suitable for DAGGRE/SciCast use cases, and we studied four of them: Netica (www.norsys.com/netica), SMiLE (dslpitt.org/genie/), libDAI (staff.fnwi.uva.nl/j.m.mooij/libDAI/), and UnBBayes. Three factors played the major driving force for choosing UnBBayes: being available under an open-source license, allowing easy access and customization of low-level data structures (so that they can be easily adapted to manage numbers other than probabilities – e.g. assets), and providing good exact inference time for networks with reasonable tree width. The following table summarizes some important observations about the four packages we studied:

Table 17: A short comparison between some probabilistic reasoning software packages -- candidates for Markov Engine

	Source-code availability	Access to low-level structures	API languages	Notes
Netica	No (proprietary)	Limited	Java, C, C++, C#, VB, Matlab, CLisp	Multiple language support by offering different APIs.
SMiLE	Yes, if signed in to their website	Limited	C++	Fast exact inference.
libDAI	Yes	Yes	C++	Fast approximate inference support. Requires POSIX libraries.
UnBBayes	Yes	Yes	Java	Binary level portability (Java virtual machine).

Due to source-code availability and support for low-level data structures, UnBBayes and libDAI were tentatively selected and submitted to a computational performance comparison. The performance test was designed to run a large number of trades – probability updates – in the same machine for different networks with 191 nodes (this was the number of questions present in the system in Y1), and then to compare the average time for a trade. The following list describes the test procedure, and results:

- In the network with 191 disconnected nodes (*i.e.* all questions present in DAGGRE at the time the test was performed), run 21914 trades (*i.e.* number of trades performed in DAGGRE until this time). Repeat procedure 20 times and estimate the average time (of 20 replications) to finish all 21914 trades. A CSV file specifying the trades can be obtained at svn.code.sf.net/p/unbbayes/code/trunk/MarkovEngine/src/test/resources/DAGGRE.csv.
 - Average time to finish 21914 trades in UnBBayes: 134 seconds.
 - Average time to finish 21914 trades in libDAI: <1 second.
- In the network with 191 nodes, split the nodes in sets of 5 nodes each (the last set can have less than 5 nodes). Fully connect the 5 nodes in each set. Run the 21914 trades. Repeat procedure 20 times and estimate the average time (of 20 replications) to finish all 21914 trades.
 - Average time to finish 21914 trades in UnBBayes: 99 seconds.
 - Average time to finish 21914 trades in libDAI: 153 seconds.
- For the network used in the 1st test, randomly generate arcs, but asserting that the maximum number of parents per node won't exceed 20 nodes. Using the same network for both software packages, perform 20 trades, calculate the average time per trade, and multiply it with 21914 to estimate the approximate time to finish all trades. This simplified procedure was used in this test because finishing all 21914 trades would take too long. The UnBBayes network file used in this experiment is available at svn.code.sf.net/p/unbbayes/code/trunk/MarkovEngine/src/test/resources/bn191_treewidth_20.net. The same network, but saved as a factor graph file for libDAI is available at svn.code.sf.net/p/unbbayes/code/trunk/MarkovEngine/src/test/resources/bn191_treewidth_20.fg.
 - "Estimated" time to finish 21914 trades in UnBBayes: 43 hours.
 - "Estimated" time to finish 21914 trades in libDAI: 67 hours.



The tests suggested that UnBBayes was not well calibrated to run with a fully-disconnected network at that time (this issue was actually fixed later as part of an ancillary optimization effort), but it was already faster for exact inference in networks with connections. For this reason, UnBBayes was chosen as the main framework or library for DAGGRE/SciCast's Markov Engine.

The following sections describe features and extensions developed for UnBBayes during the DAGGRE/SciCast project. As the name implies, section “Features developed prior to Year 4” presents observable software features that were developed for UnBBayes in DAGGRE/SciCast project before Year 4. Similarly, the section “Features developed in Year 4” presents the new features developed in Year 4. Finally, the section “UnBBayes project architecture” presents the general architecture of the UnBBayes Inference Markov Engine, with a description of where the features presented in the previous sections were implemented, and where to find the main software modules.

Starting from this section, the name “UnBBayes” will be used to indicate the UnBBayes framework, and the name “Markov Engine” will be used to indicate the wrapper of UnBBayes, a facilitator directly accessed by the DAGGRE/SciCast system. The word “UnBBayes Markov Inference Engine” will be generally used to indicate the combination of UnBBayes and Markov Engine together.

Features developed prior to Year 4

This section presents some features developed in UnBBayes Markov Inference Engine before Year 4 of DAGGRE/SciCast project.

Conditional soft evidence

Two features were developed during DAGGRE project and integrated to UnBBayes' core regarding evidence handling: a capability to insert evidences that will set a random variable's current probability to any desired value in between 100% and 0%; and the same capability (to set the probability to a desired value) for a probability involving multiple variables, particularly for a conditional probability of a variable given a set of other variables.

In contrast to “hard” evidence, which are indications or beliefs that some outcome of questions will either happen for sure (100%) or to never happen (0%), trades performed by users in DAGGRE/SciCast are “soft” evidences, because they represent user's support for an outcome, with any probability in between 0% and 100%. Technically, this type of revision of probabilities can be seen as an application of Jeffrey's rule of conditioning.

UnBBayes already had support for such “non-hard” evidence. However, its implementation was based on a mechanism known as likelihood evidence, in which new beliefs were expressed in terms of likelihood ratios, which represent how much the new information favors each of the question outcomes. For instance, a likelihood ratio of 3/2 for a question to be “true” means the likelihood of the new information given a “true” answer versus a “false” answer has ratio 3:2.



On the other hand, trades in DAGGRE/SciCast are expressed directly as probabilities, not as likelihood ratios. Fortunately, estimating the likelihood ratio to be multiplied by current probabilities in order to reach some desired probability is a straightforward algebraic calculation. For this reason, UnBBayes' likelihood evidence methods could be reused with minor changes.

For example, the likelihood ratio that will move a probability of “true” from 75% to 80% (and “false” from 25% to 20%) is 4/3. This is because:

- Initially, the likelihood ratio of the probability of true divided by probability of false was $75\%/25\% = 3/1$.
- Similarly, the likelihood ratio of the desired probability is $80\%/20\% = 4/1$.
- The ratio r to be multiplied to the initial likelihood ratio in order to obtain the desired likelihood ratio can be obtained by solving $3r/1 = 4/1$ ~~the ratio 4/3 can be obtained faster by~~ calculating $(P_{\text{desired}}(\text{true})/P_{\text{current}}(\text{true})) / (P_{\text{desired}}(\text{false})/P_{\text{current}}(\text{false})) = (80\%/75\%)/(20\%/25\%) = 4/3$.
- Consequently, we can obtain 80% for true, and 20% for false by multiplying the initial probability of true (75%) with 4, and the initial probability of false (25%) by 3, and then performing normalization.

Soft evidence for conditional probabilities were implemented based on the same principles. The following observations were considered:

- A soft evidence to a joint probability $P(Q,A)$ of question Q and a set of assumptions A is equivalent to a soft evidence to a “large” question whose outcomes are the combination of all possible outcomes of Q and A . Therefore, soft evidence involving multiple questions can be implemented in the same manner as in soft evidence to a single “large” node.
- The objective is to change only the conditional probability $P(Q|A)$. This is possible by changing the joint probability $P(Q,A)$ without changing the probability of the assumptions $P(A)$. This is because $P(Q,A) = P(Q|A)P(A)$; therefore, soft evidence in $P(Q,A)$ without changing $P(A)$ will result in changes only to $P(Q|A)$.
- The ratio $P_{\text{desired}}(Q,A) / P_{\text{current}}(Q,A)$ when fixing $P(A)$ is equivalent to $P_{\text{desired}}(Q|A) / P_{\text{current}}(Q|A)$. Consequently, conditional soft evidence could be translated to likelihood evidences with the likelihood ratio at $P_{\text{desired}}(Q|A) / P_{\text{current}}(Q|A)$.

The translation of trades to likelihood evidences is sensitive to the question's current probability distribution. As a consequence, trades cannot be performed in parallel (unless questions are independent each other), because the likelihood ratio to be multiplied in order to obtain some desired probability is obviously different if we are starting from different probabilities.

Another limitation of this type of conditional soft evidence is that changes in conditional probabilities that will break constraints of conditional (in)dependence expressed in the market's Bayesian network structure are not allowed. Simply stated, the only permissible conditional probabilities are those that can be represented in the network's structure. In practice, this limits conditional soft evidence to be applicable only to questions in the

same clique of the junction tree managed by UnBBayes. Of course, new arcs can be added to the Bayes net in order to permit previously forbidden conditional evidence.

Bayesian network management facades

A facade is an object, module, or a function that provides a simplified interface to a larger body of code. A set of facilitator methods for the management of the Bayesian network of the market's probability distribution was implemented in order to avoid the exposition of unnecessary details of UnBBayes to consumers of its functionalities.

For instance, all basic operations, namely nodes/arc inclusion, probability/asset updating, and queries to probabilities/assets were exposed as simplified functions/methods. Arc deletion was not supported before Year 4, because a monotonic property of network complexity was required at this time to archive maintainability.

Asset representation (deprecated)

The Parallel Junction Tree algorithm (Sun et al., 2012) is based on the observation that assets factor according to the same decomposition of the market's probability distribution, and asset updating could be performed by defining a parallel asset junction tree for each user and through modifying the junction tree algorithm to find the user's minimum and expected assets.

Since UnBBayes' basic representation of the factorized probability distribution was also a junction tree, the Parallel Junction Tree algorithm was implemented as a plug-in for UnBBayes, mainly by adding support for a junction tree with non-normalized values (i.e. values not between 0% and 100%). (The Markov Engine wraps UnBBayes and all the plug-ins necessary for DAGGRE/SciCast use cases together, so plug-ins of UnBBayes are invisible for DAGGRE/SciCast system.)

Asset management in UnBBayes is currently a deprecated feature, because it was decided in SciCast that assets and probabilities should be managed independently to allow parallel evolution and maintenance.

Estimation of balancing trades (deprecated)

The balancing trade feature calculates the changes in probability that would minimize the impact (on user's assets) of a given question, once the question is resolved. Basically, when the user chooses to "balance" (or "equalize") a question Q, the UnBBayes Markov Inference Engine will try to estimate and execute trades which will make the user's asset not depend on Q (but may still depend on questions other than Q). As a consequence, if all other questions eventually resolve, then the user's assets position for all the states of Q will be equal (or close enough). Similarly, if the user chooses to "balance" question Q given set of assumptions A, then the UnBBayes Markov Inference Engine will try to estimate and execute trades which will make the user's asset not to depend on Q, but only on cases where assumptions A are valid. In other words, if the assumptions A resolves to the indicated values, and all other questions eventually resolves, then the assets for all states of Q will be equal (or sufficiently close).

Notice that if the assumption A is not empty, then all other worlds (incompatible with the assumptions) will not be balanced. Therefore, the user's asset may still have some dependency with question Q in such case.

Technically, this operation is calculated in terms of the values present in the asset tables (clique tables containing asset values of the user given combination of states of the questions) of the user. Suppose that the asset's clique table is structured as the following:

Table 18: A simplified example of an asset clique table

	c1	c1	c2	c2
	b1	b2	b1	b2
a1	x1	x3	x5	x7
a2	x2	x4	x6	x8

In the above table, a1 and a2 are the states of a question A. Similarly, c1 and c2 are states of question C, and b1 and b2 are states of B. The values x_i with $i = 1...8$ are the asset that the user invested for that particular combination of states of the questions A, B and C.

If we are balancing question A (*i.e.* balancing states a1 and a2) given assumption $C=c1$, then the Engine shall make a trade which will set $x1 = x2$ AND $x3 = x4$ and all other values remain unchanged. This is equivalent to balancing question A given $B=b1$ and $C=c1$, and then balancing A given $B=b2$ and $C=c1$ (because we are changing these 2 columns in order to set $x1 = x2$ and $x3 = x4$).

Now, suppose $P(A=a1, B=b1, C=c1) = p1$ and $P(A=a2, B=b1, C=c1) = p2$ are the current joint probabilities (stored in the clique table of the junction tree of the market's Bayesian network).

In order to calculate a trade which balances question A given $B=b1$ and $C=c1$ (*i.e.* balance one of columns in order to set $x1 = x2$), we need to solve a system of equations which estimates the posterior probabilities $P_{\text{post}}(A=a1, B=b1, C=c1) = P1$ and $P_{\text{post}}(A=a2, B=b1, C=c1) = P2$ which also satisfies the following constraints:

- $P1 + P2 = 1$;
- $x1 + 100 \cdot \log(P1/p1) = x2 + 100 \cdot \log(P2/p2)$

For simplicity, the second equation can be reduced as follows:

- $x1/100 + \log(P1/p1) = x2/100 + \log(P2/p2) \rightarrow$
 - $\rightarrow 2^{(x1/100 + \log(P1/p1))} = 2^{(x2/100 + \log(P2/p2))}$
 - $\rightarrow 2^{(x1/100)} \cdot 2^{(\log(P1/p1))} = 2^{(x2/100)} \cdot 2^{(\log(P2/p2))}$
 - $\rightarrow 2^{(x1/100)} \cdot P1/p1 = 2^{(x2/100)} \cdot P2/p2$

By setting $2^{(x1/100)} = q1$, and $2^{(x2/100)} = q2$, we obtain the following linear equation (because only P1 and P2 are unknown values):

- $P1 + P2 = 1$;
- $q1 \cdot P1/p1 = q2 \cdot P2/p2$

In a generic case whose questions have more than 2 states, balancing/equalizing a question (given all the other questions in the same clique) also reduces to a linear equation system of calculating the probabilities P_1, P_2, \dots, P_N (N is the quantity of states of the given question), in which:

- $P_1 + P_2 + \dots + P_N = 1$
- $q_1 * P_1 / p_1 = q_2 * P_2 / p_2 = \dots = q_N * P_N / p_N$

Note: for $1 \leq i \leq N$; P_i is the solution (posterior probability), p_i is the current (prior) probability, and q_i is the q value – *i.e.* $2^{(x_i / 100)}$ – of the i -th state.

A closed form solution of the above equation is:

$$P_i = (q_1 * q_2 * \dots * q_{i-1} * q_{i+1} * \dots * q_N * p_i) / ((q_2 * q_3 * q_4 * \dots * q_N * p_1) + (q_1 * q_3 * q_4 * \dots * q_N * p_2) + \dots + (q_1 * q_2 * \dots * q_{i-1} * q_{i+1} * \dots * q_N * p_i) + \dots + (q_1 * q_2 * \dots * q_{N-1} * p_N))$$

For $1 \leq i \leq N$.

The balancing trade feature of Markov Engine simply solves the above equation and performs the trade. Again, please notice that asset management by Markov Engine is deprecated; therefore, this feature is also deprecated. In addition, it should be said that the feature confused users – while technically correct, it appears not to have been what they expected when equalizing or balancing.

Support for value trees (deprecated)

As previously stated in this report, algorithms and structures appropriate to variables with a large number of possible states/outcomes were explored. One approach considered was the value tree.

Data structures and algorithms for the case of a single lone value tree were implemented in UnBBayes' asset representation plug-in, wrapped and made available by the Markov Engine. This feature is also currently deprecated in UnBBayes Markov Inference Engine, though it could be enabled later.

Support for exporting/importing snapshots of current probability distribution

The system start-up of DAGGRE was traditionally processed by re-running all trades stored in the system history. While this was convenient for keeping assets and probability synchronized (this was a requirement in the Common Junction Tree algorithm), such synchronization soon became unnecessary, because new algorithms allowed assets and probabilities to evolve independently. Besides, the start-up time was increasing substantially as the number of trades in the history increased.

As an attempt to reduce the overall start-up time, functions to create and load snapshots of a probability distribution at a given moment was implemented for the Markov Engine. The client (*i.e.* SciCast system) now enjoys the ability to create milestones (*i.e.* store the snapshots), and in an eventual system reboot, the client can load the snapshots and run only the trades occurred after the time of the snapshot. This feature drastically reduced the system's overall start-up time, and it consequently reduced system's overall downtime.

The snapshots are represented in Hugin NET language version 3, which is the default format supported by UnBBayes for persisting Bayesian networks. More information about the Hugin NET language specification can be found at Chapter 13 of Hugin API reference manual version 8.1 (Hugin Expert A/S, 2014).

Support for queries for network's statistics

This is an ancillary feature designed for administrators. The Markov Engine can be used to retrieve some metrics that are related to the complexity of the market's Bayesian network. Some of the metrics that can be retrieved are:

1. number of nodes (questions), organized by their number of states (possible outcomes);
2. sum of the junction tree's clique and separator table sizes, or sum of clique table sizes only;
3. degrees of freedom;
4. number of cliques and separators;
5. maximum clique table size;
6. maximum number of parents per node;
7. number of arcs;

In SciCast, this is mapped to the “/engine” call. Someone logged in with an admin account can visit <https://scicast.org/engine> to see something like Figure 61, below:

```
▼ {
  "degreeOfFreedom": 21952,
  "isRunningApproximation": false,
  "maxCliqueTableSize": 899,
  "maxNumParents": 4,
  "numArcs": 211,
  "numberOfCliques": 562,
  "numberOfNonEmptyCliques": 562,
  "numberOfSeparators": 561,
  ► "numberOfStatesToNumberOfNodesMap": { ... },
  "sumOfCliqueAndSeparatorTableSizes": 24942,
  "sumOfCliqueAndSeparatorTableSizesWithoutResolvedCliques": 24942,
  "sumOfCliqueTableSizes": 23681,
  "sumOfCliqueTableSizesWithoutResolvedCliques": 23681
}
```

Figure 61: Example of the “/engine” call to SciCast, displaying the network statistics as formatted json data.

Features developed in Year 4

This section specifies features developed for UnBBayes Markov Inference Engine in Year 4.

Dynamic/incremental junction tree compilation feature

As previously stated, DAGGRE/SciCast's market probability distribution is represented as a Bayesian network, and UnBBayes uses the Junction Tree algorithm for calculating and updating the probabilities. This algorithm works basically by converting the Bayesian network to a probabilistically equivalent alternative representation

called Junction Tree, which is a tree structure whose nodes (cliques) represent a joint state of a set of nodes in the original network, and connections (separators) represent intersections – or nodes in common – between cliques.

Once a Junction Tree is compiled, UnBBayes is able to quickly calculate or update the probabilities of nodes in the original Bayesian network by using standard message passing mechanisms, but there is overhead in compiling the Junction Tree, which is required each time the structure of the original network (*e.g.* number of nodes, or arcs) changes. While this overhead is negligible if changes in the Bayesian network's structure are not so frequent, a new feature in Year 4 – the user-edited links – has drastically increased the expected frequency of such changes. Therefore, a faster algorithm for compiling Junction Trees was required.

The solution implemented in UnBBayes, in order to archive faster compilation time, is the dynamic/incremental junction tree compilation algorithm, specified by (Flores, Gámez, & Olesen, 2003). Basically, this algorithm reduces the compilation time by keeping track of changes in the network structure and only re-compiling the subnets affected by the changes, and reusing the cliques and separators that are not affected. Its essence can be summarized into three main ideas:

Compilation time

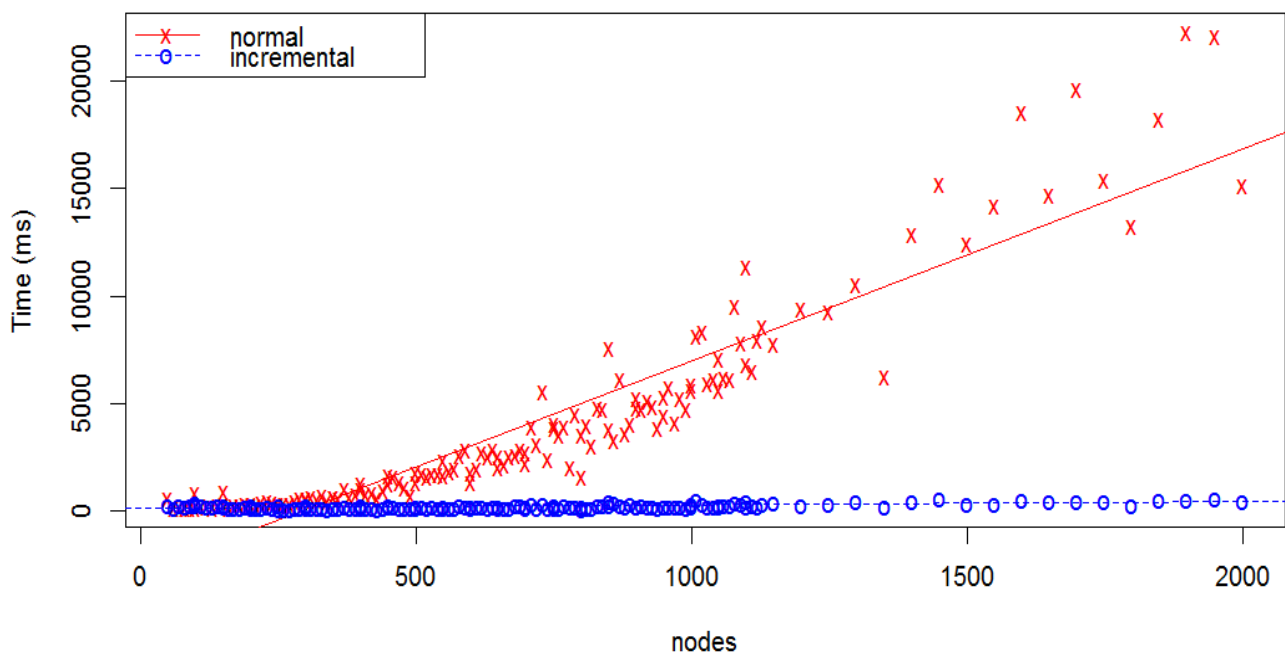


Figure 62: Comparison of (re)compilation time between normal and incremental junction tree compilation, for networks with different sizes

1. “Complete” separators isolate changes. A complete separator is either empty (it contains no nodes from the original Bayesian network), single-node (it contains only one node), or fully connected (it contains only a set of nodes that are fully connected each other, by arcs or by moralization, in the original network).

2. We only need to recompile cliques/separators in an “isolated” area. If a change in the original Bayesian network affects some set X of nodes, then the nodes pertaining to the same cliques of X and all other cliques connected to such cliques by “non-complete” separators will be re-compiled.
3. Reuse everything outside the “isolated” area. Other cliques/separators – those connected to affected cliques by complete separators – will remain unchanged.

The dynamic/incremental junction tree compilation improved the average performance of (re)compilation of SciCast's Bayesian network, especially because most of separators have shown to be “complete” – due to sparseness. Figure 62 illustrates how the compilation time changes as the size of the network (expressed in terms of number of nodes) increases. The arcs in the tested networks were generated randomly, but the arc density (*i.e.* the proportion between the number of arcs and the number of nodes) was kept the same.

The compilation time shown in Figure 62 reflect a typical use case scenario in SciCast. Each time we include a new question (node) or link (arc), we virtually perform a re-compilation. Therefore, the plots actually represent the total time of multiple compilations. As expected, the compilation times of dynamic/incremental junction tree compilation algorithm were considerably faster, because recompilation in dynamic/incremental algorithm was always limited to a smaller subnet.

Approximation with loopy belief propagation between cliques

The execution time of belief propagation in structures of cliques (*e.g.* Junction Tree) is typically dominated by the size of the cliques (*i.e.* number of nodes contained in a clique). If the number of arcs in the Bayesian network increases, the clique sizes are also likely to increase, and consequently the system will be more likely to experience slowdowns. A common approach to avoid slowdown is to use approximation. However, two properties were desirable for approximate algorithms in UnBBayes Markov Inference Engine:

1. The algorithm should allow reuse of software/code and/or data structures of UnBBayes, in order to reduce development effort;
2. The algorithm should trigger approximation only when needed, and only where needed. In other words, we needed the ability to switch back and forth from/to approximation, and to keep exact calculation where approximation is unnecessary.

The approach implemented in UnBBayes (actually, in a plug-in) in order to address this issue was to extend the default junction tree algorithm and disable triangulation. Triangulation is, simply stated, a step in the junction tree compilation process which adds some extra arcs between nodes, just in order to assert that the resulting structure of hyper-nodes (*i.e.* structure of cliques and separators) is a tree. Disabling triangulation generally results in smaller clique sizes, but the structure of cliques will become a graph, instead of a tree.

Standard belief propagation in non-tree structures is known to result in incorrect probabilities. Loopy Belief Propagation was implemented to address this issue. Loopy Belief Propagation is a well known approximate algorithm which simply iterates standard belief propagation until reaching convergence or until exceeding



some time limit. Loopy Belief Propagation applied to Bayesian networks is frequently found in the literature; however, we instead apply it to a graph of cliques. Our approach has some interesting advantages:

1. Exact subnets are kept exact. In other words, subnets with no need for triangulation result in a tree-like structure somewhere in the clique graph, and belief propagation in trees converges immediately, with exact results. Therefore, evidences inserted in cliques that are independent from approximate portions will converge immediately and it will also yield exact results.
2. It is an anytime algorithm, because it is possible to halt the iteration (of belief propagation) and return the best results known so far.
3. Most source-code and data structures of UnBBayes are reused, because we are still using cliques, and Loopy Belief Propagation is simply a traditional belief propagation run multiple times.
4. If used together with dynamic/incremental junction tree compilation, switching back and forth to/from approximation is fast, and it will reuse cliques and separators.

However, the approach also has some particular disadvantages:

1. Loopy Belief Propagation in general is not guaranteed to always converge.
2. It only reduces clique sizes caused by triangulation. It does not reduce clique sizes if they were caused by too many parents per node in the original Bayesian network.

This feature works best when the Bayesian network (from which we generate the clique graph) has a limited number of parents per node, but many (or large) loops. It is already integrated to the dynamic/incremental junction tree compilation feature, and it is transparent to users of Markov Engine. Approximation (in Markov Engine) will be automatically triggered when a threshold of maximum clique size – a configurable parameter in the Markov Engine – is reached, and it will be disabled when the maximum clique size gets below the threshold after a question settlement or arc deletion.

Support for arc removal

This feature removes an arc (link between random variables) from the market's Bayesian network.

Arc removal was not considered in the use case scenarios of DAGGRE/SciCast until Year 4, when user-edited links became a requirement (since users would be able to include arcs, it would be important for administrators to be able to remove unnecessary arcs as well, for completeness of the business logic).

UnBBayes had this feature, but it was not visible from a facilitator function in the Markov Engine. Now, the facilitator is implemented.

This feature is not supported when asset structures used by the deprecated Common Junction Tree algorithm are present. In other words, if the Markov Engine is running with the asset management (a deprecated feature) enabled, then arc removal will be disabled by default, because algorithms for handling asset structures in Common Junction Tree algorithm is not well defined when arcs are removed.

Link strength estimation from mutual information

A set of methods which returns a metric for the strength of arcs in the market's Bayesian network was implemented in the Markov Engine. The implementation in UnBBayes Markov Inference Engine simply calculates the mutual information between nodes connected by the arc. Since mutual information measures how close the joint probability of two nodes is to the product of their marginal distributions, it is a metric on how dependent two variables are. Mutual information closer to zero suggests independence, and it is an indication that the arc/link is weak and should be considered for deletion.

Arc complexity factor estimation

While link strength is a measure of how much a link (arc) contributes to the market's probability distribution, the “complexity factor” is an estimate of how much a link (arc) contributes to computation time in the underlying inference algorithm.

In the UnBBayes Markov Inference Engine, the method for estimating the complexity factor returns the maximum clique size, or the sum of all cliques, depending on some parameters. It will basically estimate the sum/max clique size after including a new arc, or after removing an existing arc. This functionality is useful for administrators to decide which arcs to delete, or to check whether a new arc may compromise system's performance. (User-added arcs relies heavily on this feature.)

Support for compressed snapshots of current probability distribution

This is just an extension of the snapshot feature, which stores the market's probability distribution in Hugin NET language. Now, the same representation can be compressed (zipped) and encoded to base64. The base64 encoding is used to keep backward compatibility, because the non-compressed format was a text, and it is desired that the new format to be also a text. The Markov Engine can load both compressed and non-compressed formats, by automatically detecting whether the provided snapshot is compressed or not (it looks at the first few bytes in the snapshots).

UnBBayes project architecture

This section describes where the features presented in the previous sections were actually implemented (*i.e.* which software assets are associated with the features), and where to find/download the main software components – made available as open-source projects. Components of UnBBayes and the Markov Inference Engine were developed as Apache Maven projects. Maven is a Java software project management tool. (See maven.apache.org/ for details.) Figure 63 illustrates the dependencies of the Maven projects developed for DAGGRE/SciCast. All the dependencies must be solved in order for a Maven project to be compiled.

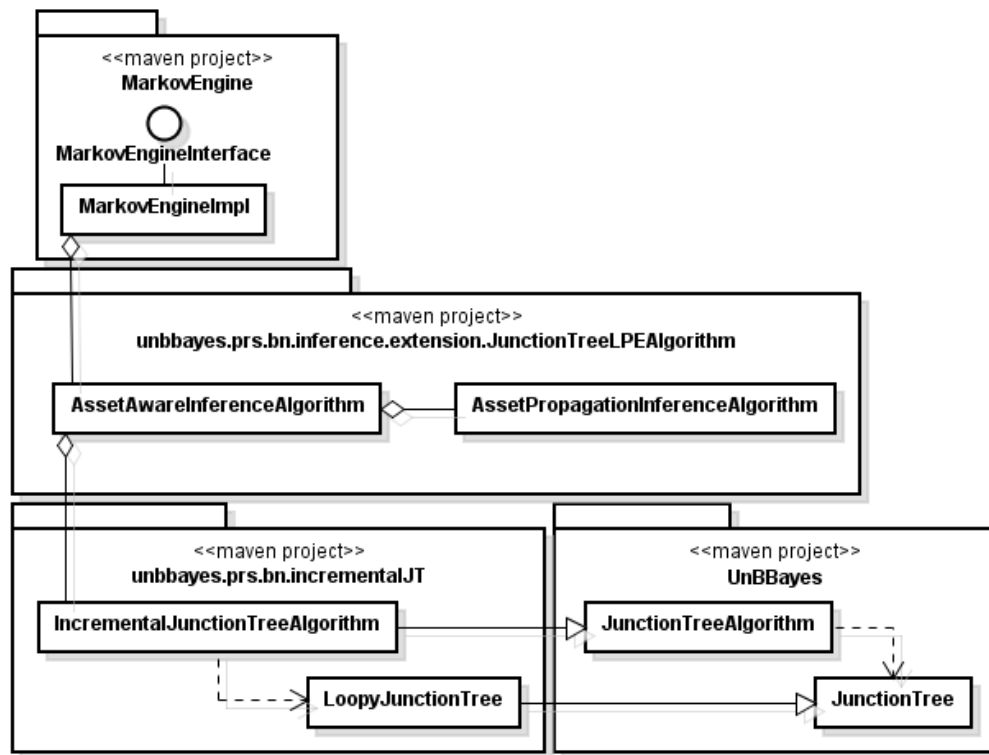


Figure 63: Maven dependencies for the Markov Inference Engine

The following list describes the Maven projects shown in the above picture:

1. *MarkovEngine*: this maven project is the main project of Markov Inference Engine, and it basically contains wrappers and facilitators. It also comprises all features related to user management, network snapshots, trade history (and reverting trades based on such history), transaction control, collection of statistics, and calculation of balancing trades. *MarkovEngineImpl* is the central class of this project. The source code is available in the following subversion repository: <https://svn.code.sf.net/p/unbbayes/code/trunk/MarkovEngine/>.
2. *unbbayes.prs.bn.inference.extension.JunctionTreeLPEAlgorithm*: this maven project contains classes which make a bridge between the Markov Engine (*AssetAwareInferenceAlgorithm*) and other components that manage Bayes nets. It also contains classes for handling user assets (*AssetPropagationInferenceAlgorithm*). This maven project adheres to UnBBayes' plug-in framework; therefore, while this is a library used by *MarkovEngine*, it can also be used as a plug-in for UnBBayes GUI. The source code is available in the following subversion repository: <https://svn.code.sf.net/p/unbbayes/code/trunk/unbbayes.prs.bn.inference.extension.JunctionTreeLPEAlgorithm/>.
3. *unbbayes.prs.bn.incrementalJT*: this maven project contains classes for dynamic/incremental junction tree compilation (*IncrementalJunctionTreeAlgorithm*) and for approximation by loopy belief propagation in cliques (*LoopyJunctionTree*). This maven project adheres to UnBBayes' plug-in framework; therefore, while this is a library used by *MarkovEngine*, it can also be used as a plug-in for UnBBayes GUI. The



source code is available in the following subversion repository:

<<https://svn.code.sf.net/p/unbbayes/code/trunk/unbbayes.prs.bn.incrementalIT/>>.

4. *UnBBayes*: this is the maven project with the source code of original UnBBayes project. It basically contains all classes for managing and manipulating a Bayes net. The classes *JunctionTreeAlgorithm* and *JunctionTree* implement the junction tree algorithm in UnBBayes. The source code is available in the following subversion repository: <<https://svn.code.sf.net/p/unbbayes/code/trunk/UnBBayes/>>.



Recruiting, Outreach, and User Engagement

The Y4 research year saw the total number of registered users on SciCast go above 11,000. The bulk of the growth came in June and July of 2014, when we added several thousand registered users through a broad online advertising campaign. The following is an overview of our recruiting, outreach, and user engagement activities.

Advertising

At the beginning of Y4, we were given a target of 5,000 registered users by August, 2014. In April 2014 we began planning a multi-faceted outreach campaign that focused mainly on online advertising. We created multiple banner ads and placed them on two advertising networks: Millennial Media, and Conversant. We were able to test which ads were performing better and soon chose to use these exclusively in our campaign:



Figure 64: Winning ads for the August 2014 campaign



The online campaign concluded in August, 2014. We resumed it for a brief period in October, 2014, but suspended again around Thanksgiving and did not resume online campaigning for the rest of the research year.

Working with AAAS, we also placed a full-page advertisement in *Science*:

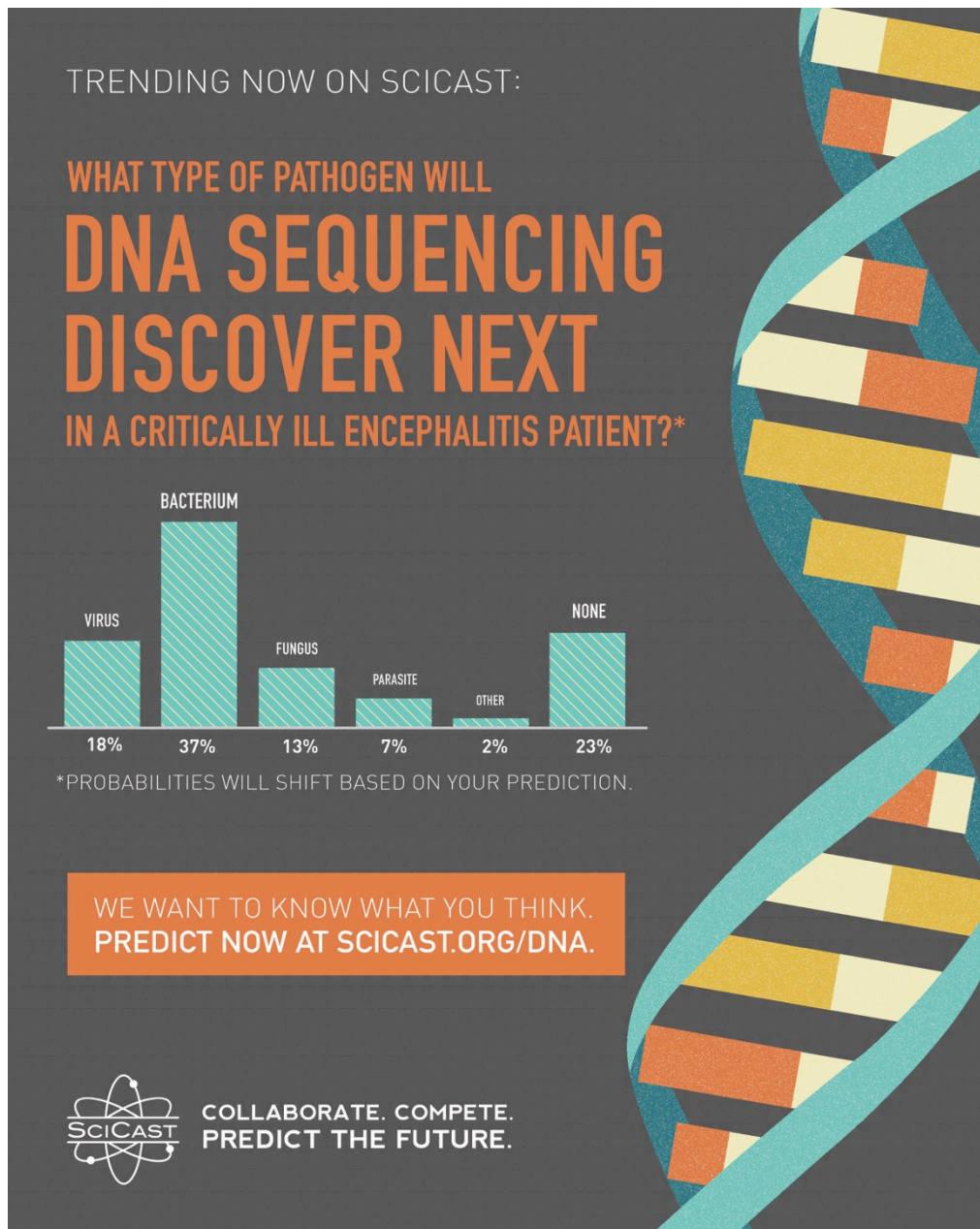


Figure 65: Full-page SciCast ad in *Science*

Social Media

Blog.scicast.org

Our blog is the primary channel of outreach to our forecasting community and other interested parties. We posted 91 blog posts from May 2014 to April 2015. The following are the top 10 posts according to the number of likes and comments they garnered (in order):

1. SciCast Recruitment Announcement
2. Accuracy Contest: Second round questions
3. Calibration Update
4. Market Accuracy and Calibration
5. SciCast Recruitment Announcement
6. Survivor: Future! (And Accuracy Contest Winners Will Be Announced Soon)
7. SciCast WSJ Coverage: U.S. Intelligence Community Explores More Rigorous Ways to Forecast Events
8. HPV questions have resolved on SciCast
9. Help Our Research Efforts and Be Entered to Win an Amazon Gift Card
10. SciCast Accuracy Incentives Contest Has Ended

Facebook: <https://facebook.com/scicast>

During this research year, we made 536 posts to Facebook. The average “reach” per post was 37 people and we reached a total of 19,633 people. The average “engagement” per post, or people who clicked on a link in the post, was 8%.

Here are the posts with the most engagements, and when they were posted:

Will a new free-space distance record for quantum #teleportation...

January 12, 2015 8:45 am

50.00% engagement

We are made of star stuff IMAGE

December 31, 2014 2:46 pm

29.17%

Introducing SciCast Courses

December 16, 2014 11:54 am

20.00%

Predict now How many near-Earth large asteroids will NASA detect in 2014? <http://bit.ly/1xS7GFM>

December 02, 2014 1:59 pm

17.65%

Happy Thanksgiving from SciCast!



November 27, 2014 5:30 am
17.39%

How many 'things' will be connected to the Internet by th...
November 03, 2014 9:48 pm
50.00%

The Hoverboard Is Finally Real <http://on.mash.to/1uA71Ud> ...
October 21, 2014 11:38 am
20.00%

Will scientists create a fully air-transmissible, mammali...
September 29, 2014 2:32 pm
27.27%

Which of these 2016 presidential candidates will accept #bitcoin...
September 18, 2014 9:21 am
21.74%

Attention #mathletes! Check out SciCast's #math questions...
September 09, 2014 9:14 am
50.00%

New on the #SciCast Blog: Meet SciCaster Ted Sanders htt...
July 09, 2014 7:00 pm
25.40%

As the internet continues to spread across the globe, do ...
July 09, 2014 3:05 pm
22.92%

Electric cars are becoming more popular every day. Will N...
July 07, 2014 2:00 pm
20.00%

When will an artificial blood cell product derived from s...
June 25, 2014 10:03 am
20.00%

Trending: Will Google announce the development of a smart...
June 24, 2014 6:30 pm
21.05%

Twitter

During this research year, we made 1,448 Tweets on our twitter account at <https://twitter.com/scicasters>. From those tweets we got 99,574 clicks and 308 direct responses. Over 4 million people saw the tweets. Here were the tweets with the most reach:

Table 19: Top SciCast Tweets

3/30/15 3:51	Will at least 250 data breaches in the US be reported by the ITRC between 1/1/15 and 3/31/15? http://bit.ly/1LseGwR #cybersecurity
4/15/15 3:38	7 easy ways to avoid being #hacked http://bit.ly/1GILiok via @BIUK_Tech #cybersecurity
9/10/14 13:33	How a smartphone-sized gadget could free over 100 million animals http://bit.ly/1tss77Q via @good #animaltesting
1/14/15 13:20	Will a breathalyzer for measuring blood sugar be commercially available before 2017? http://bit.ly/1qX73pK #diabetes http://pic.twitter.com/NTZNInjcn
6/23/14 22:01	When will #3Dprinting Oreo technology become readily available to the public? http://bit.ly/1wprHS3
1/29/15 9:48	@mouselink Perhaps you'd like to create prediction markets on your questions about the future of #science and #tech? http://bit.ly/Q2rK62
1/12/15 19:26	When will the first car equipped with #V2V safety technology be offered for sale to the general public in the US? http://bit.ly/1C0dbC2
11/29/14 16:15	Will @Philae2014 wake up and send data before the end of March 2015? http://bit.ly/1CqRkIC #Rosetta #ESA http://pic.twitter.com/ka26IcRvZt
11/29/14 4:08	Do you think NASA will lose contact with the Mars Curiosity rover before April 2015? http://bit.ly/1CqQtrl
8/20/14 6:13	Join SciCast for a @reddit_AMA and @acswebinars! http://wp.me/p3RWus-gL #science #AMA

SciCast Annual Report (2015)

• • •

7/22/14 11:56	New blog post on our partner site @ISACANews: #SciCast Calls for #ISACA members to make predictions http://bit.ly/1A280m0
11/26/14 12:04	Make real-time predictions on future innovations and events in #technology and #science http://SciCast.org
11/28/14 21:18	Make real-time predictions on future innovations and events in #technology and #science http://SciCast.org
10/31/14 14:47	Will any government officially accept a digital currency for circulation in their country in 2014? http://bit.ly/1tmCGS #bitcoin
9/7/14 9:29	We have hundreds of predictions that are coming true on http://SciCast.org . Join us! #forecasting #science #tech
8/21/14 12:46	15 min until the webinar! Forecast the future of chemistry today at 2pm ET http://bit.ly/1t3sjur
9/14/14 14:20	SciCast: World's Largest Crowdsourced Science and Technology Forecasting Site http://bit.ly/1D84Y19 via @IndustryTap
3/13/15 12:54	@alizasherman Oh, but it IS fun! Check out what @missmetaverse had to say: SciCast Wants You To Be A Futurist - http://go.shr.lc/1Ac4ty2
2/22/15 21:00	Make your forecasts on #cybersecurity questions http://bit.ly/1LseeYt #data http://pic.twitter.com/eneX744DQot
8/18/14 21:42	Join @robinhanson and @ctwardy from SciCast for a #Reddit #Science #AMA on August 20 at 11 a.m. EST. http://www.reddit.com/r/science
2/26/15 12:29	@missmetaverse Thanks for spreading the word about SciCast!
9/29/14 3:34	The power of collective intelligence & why online gaming may just be the future of science http://bit.ly/1A3eP6P via @good
1/28/15 9:43	SciCast talks to @healthmap co-founder @johnbrownstein about the value of #crowdsourced forecasting in #publichealth http://bit.ly/1Dakisc

7/19/14 20:33	At its first launch, will Apple's #iWatch include a sensor to measure blood #glucose level non-invasively? http://bit.ly/1wATGOS
2/28/15 20:07	#Wanderlust? Will the average price of a US domestic plane ticket decline in the first 2 quarters of 2015? #travel http://bit.ly/1D4q26I

Other Outreach Activities

In addition to regular blogging and social media activity, we also conducted several other outreach activities:

- Reddit AMA with Charles Twardy and Robin Hanson on Reddit's Science sub-reddit (https://www.reddit.com/r/science/comments/2e2mvh/science_ama_series_we_are_drs_robin_hanson_and/)
- A webinar sponsored by the American Chemical Society, run by ACS (<http://www.acs.org/content/acs/en/events/upcoming-ac-s-webinars/forecasting-chemistry.html>)
- A feature in IEEE's Spectrum magazine (<http://spectrum.ieee.org/aerospace/space-flight/forecasting-tomorrows-technology-today>)

SciCast in the Media

We conducted public relations activities to drive attention to SciCast. Here is a complete list of media mentions about SciCast.

- [Attitudes to the Future](#) – IEET – 3/23/15
- [Leveraging Forecasting Techniques for Security](#) – Security Intelligence – 3/23/15
- [SciCast Wants You to Be a Futurist](#) – Miss Metaverse – 2/25/15
- [Question Authority: Make Your Own "Top Tech 2015" Predictions – IEEE Spectrum](#) – 1/1/15
- [Special Report: 2015 Top Tech to Watch – IEEE Spectrum](#) – 12/31/14
- [When Will We Have an Exascale Supercomputer? – IEEE Spectrum](#) – 12/19/14
- [BitBet, Fairlay, BetMoose: Meet Bitcoin's Prediction Markets – CoinTelegraph](#) – 11/2/14
- [SciCast: World's Largest Crowdsourced Science and Technology Forecasting Site – Industry Tap](#) – 9/14/14
- [U.S. Intelligence Community Explores More Rigorous Ways to Forecast Events – Wall Street Journal](#) – 9/5/14
- [Science AMA Series: Drs. Robin Hanson and Charles Twardy from George Mason University's SciCast project – Reddit](#) – 8/21/14
- [Forecasting Chemistry – ACS Webinar \[Slides\]](#) – 8/20/14
- [SciCast Calls for ISACA members to make predictions – ISACA Now](#) – 7/21/14
- [AAAS policy fellow makes predictions about tomorrow today – AAAS](#) – 7/15/14
- [SciCast Wants To Crowdfund Forecasting The Future – Science 2.0](#) – 6/25/14
- [SciCast Calls for Science, Technology Experts to Make Predictions](#) – 6/19/14



- [SciCast, Crowdsourcing Science and Technology Forecasting For Policy – Wilson Commons Lab](#) – 5/23/14
- [Text-mining offers clues to success – Nature](#) – 5/20/14
- [Collective intelligence to find the plane MH370 – Huffington Post \(Spanish\)](#) – 5/17/14
- [Dicty World Race – finding the fastest and smartest Dicty cells – Experiment](#) – 5/16/14
- [The game is on – Nature](#) – 5/7/14
- [Big Data Offers Big Challenges and Opportunities for Life Sciences and National Security – AAAS](#) – 4/28/14
- [Robin Hanson: in-depth interview on prediction markets – Sintentia](#) – 4/28/14
- [Scientists, Start Your Dictyostelium: Researchers Rev Up for Slime-Mold Race – Wall Street Journal](#) – 3/10/14
- [Chess under Computational Sciences at Scicast.org – Chess.com](#) – 2/26/14
- [Robin Hanson – SciCast, Prediction Markets & Future Day – Science, Technology, Future](#) [Video] – 2/4/14
- [Robin Hanson – SciCast, Prediction Markets & Future Day – Exponential Times](#) – 2/5/14
- [Science and technology prediction market opens – Adi Gaskell](#) – 1/13/14
- [SciCast Crowdsources Forecasts on Science and Technology Events and Innovations – GMU News](#) – 1/10/14
- [Announcing: SciCast – Overcoming Bias](#) – 1/3/14
- [SciCast: a Crowdsourced Forecasting Platform for Science and Technology – KDnuggets](#) – 1/14
- [Crowdsourcing forecasts on science and technology events and innovations – KurzweilAI](#) – 1/10/14
- [SciCast Launch! Crowdsourced Forecasting Project from George Mason University – Chicago Tribune](#) – 12/16/13

References

- Babko-Malaya, O., Meyers, A., Pustejovsky, J., & Verhagen, M. (2013). Modeling Debate within a Scientific Community. In *Proceedings of Society 2013 (International Conference on Social Intelligence and Technology 2013 (SOCIETY 2013))*.
- Babko-Malaya, O., Seidel, A., Hunter, D., HandUber, J., Torrelli, M., & Barlos, F. (2015). Forecasting Technology Emergence from Metadata and Language of Scientific Publications and Patents. In *ISSI 2015*.
- Babko-Malaya, O., Thomas, P., Hunter, D., Meyers, A., Pustejovsky, J., Verhagen, M., & Amis, G. (2013). Characterizing Communities of Practice in Emerging Science and Technology Fields. In *Proceedings of Society 2013 (International Conference on Social Intelligence and Technology 2013 (SOCIETY 2013))*.
- Baron, J., Mellers, B. A., Tetlock, P. E., Stone, E., & Ungar, L. H. (2014). Two Reasons to Make Aggregated Probability Forecasts More Extreme. *Decision Analysis*. <http://doi.org/10.1287/deca.2014.0293>
- Brock, D. C., Babko-Malaya, O., Pustejovsky, J., Thomas, P., Stromsten, S., & Barlos, F. (2012). Applied Actant-Network Theory: Toward the Automated Detection of Technoscientific Emergence from Full-text Publications and Patents. In *Proceedings of the AAAI Fall Symposium on Social Networks and Social Contagion*.

- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (Second). Hillsdale, NJ: Lawrence Earlbaum Associates.
- Cumming, G. (2009). Inference by eye: Reading the overlap of independent confidence intervals. *Statistics in Medicine*, 28(2), 205–220. <http://doi.org/10.1002/sim.3471>
- Dawes, R., Faust, D., & Meehl, P. (1989). Clinical Versus Actuarial Judgment. *Science*, 243(4899), 1668–74. Retrieved from <http://www.sciencemag.org/cgi/content/abstract/243/4899/1668>
- Flores, M. J., Gámez, J. A., & Olesen, K. G. (2003). Incremental compilation of Bayesian networks. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence* (pp. 233–240). Morgan Kaufmann Publishers Inc. Retrieved from <http://dl.acm.org/citation.cfm?id=2100612>
- Galles, D., & Pearl, J. (1995). Testing identifiability of causal effects. In P. Besnard & S. Hanks (Eds.), *Proceedings of the eleventh conference on uncertainty in artificial intelligence* (pp. 185–95). San Fransisco, CA: Morgan Kaufmann.
- Grove, W. M., Zald, D. H., Lebow, B. S., Snitz, B. E., & Nelson, C. (2000). Clinical Versus Mechanical Prediction: a Meta-Analysis. *Psychological Assessment*, 12(1), 19–30. Retrieved from <http://www.psych.umn.edu/faculty/grove/096clinicalversusmechanicalprediction.pdf>
- Hanson, R. (2002). Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation. *Journal of Prediction Markets*, 1, 2007.
- Hugin Expert A/S. (2014). Hugin API Reference Manual (Version 8.1). Retrieved from <download.hugin.com/webdocs/manuals/api-manual.pdf>
- Ide, J. S., & Cozman, F. G. (2002). Random Generation of Bayesian Networks. In *Brazilian Symposium on Artificial Intelligence* (pp. 366–375). Springer-Verlag.
- Kominek, J. (2014). SciCasting [GitHub]. Retrieved May 26, 2015, from <https://github.com/jkominek/scicasting>
- Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and ANOVAs. *Frontiers in Psychology*, 4(863). <http://doi.org/10.3389/fpsyg.2013.00863>
- Marchese, M. C. (1992). Clinical versus actuarial prediction: a review of the literature. *Perceptual and Motor Skills*, 75(2), 583–94. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/1408625>
- Meehl, P. E. (1954). *Clinical Versus Statistical Prediction: A Theoretical Analysis and a Review of the Evidence*. University of Minnesota Press.
- Mellers, B., Ungar, L., Baron, J., Ramos, J., Gurcay, B., Fincher, K., ... Tetlock, P. E. (2014). Psychological Strategies for Winning a Geopolitical Forecasting Tournament. *Psychological Science*, 0956797614524255. <http://doi.org/10.1177/0956797614524255>
- Mercier, H., & Sperber, D. (2010). *Why Do Humans Reason? Arguments for an Argumentative Theory* (SSRN Scholarly Paper No. ID 1698090). Rochester, NY: Social Science Research Network. Retrieved from <http://papers.ssrn.com/abstract=1698090>
- Neapolitan, R. E. (1990). *Probabilistic Reasoning in Expert Systems*. Wiley & Sons, Inc.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. San Mateo, CA: Morgan and Kaufman.
- Pearl, J. (2000). *Causality: models, reasoning and inference*. New York: Cambridge University Press.

- Satopää, V. A., Baron, J., Foster, D. P., Mellers, B. A., Tetlock, P. E., & Ungar, L. H. (2014). Combining multiple probability predictions using a simple logit model. *International Journal of Forecasting*, 30(2), 344–356. <http://doi.org/10.1016/j.ijforecast.2013.09.009>
- Shwe, M., Middleton, B., Heckerman, D., Henrion, M., Horvitz, E., Lehmann, H., & Cooper, G. (1991). Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base I. The probabilistic model and inference algorithms. *Methods of Information in Medicine*, 30, 241–255.
- Silver, N. (2012). *The Signal and the Noise: Why So Many Predictions Fail — but Some Don't* (1st ed.). Penguin Press HC, The.
- Sullivan, G., & Feinn, R. (2012). Using Effect Size—or Why the P Value Is Not Enough. *Journal of Graduate Medical Education*, 4(3), 279–282. <http://doi.org/10.4300/JGME-D-12-00156.1>
- Sun, W., Hanson, R., Laskey, K. B., & Twardy, C. (2012). Probability and Asset Updating using Bayesian Networks for Combinatorial Prediction Markets. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI-2012)*. Catalina, CA: AUAI Press. Retrieved from <http://mason.gmu.edu/~wsun/publications/uai2012.htm>
- Sun, W., Laskey, K. B., Twardy, C. R., Hanson, R. D., & Goldfedder, B. R. (2014). Trade-based asset model using dynamic junction tree for combinatorial prediction. Presented at the MIT Collective Intelligence 2014, Cambridge, MA.
- Surowiecki, J. (2005). *The Wisdom of Crowds*. Random House Digital, Inc.
- Tetlock, P. (2005). *Expert political judgment: how good is it? how can we know?* Princeton University Press.
- Thomas, P., Babko-Malaya, O., Hunter, D., Meyers, A., & Verhagen, M. (2013). Identifying Emerging Research Fields with Practical Applications via Analysis of Scientific and Technical Documents. In *Proceedings of ISSI 2013*.
- Twardy, C. R., & Laskey, K. B. (Eds.). (2014, May 25). SciCast Annual Report 2014.
- Woodward, J. (2005). *Making Things Happen: A Theory of Causal Explanation*. Oxford University Press US.

