Machine Learning A Probabilistic Perspective

Zhenbang Wang

Summer 2019 - Fall 2020

http://www.stat.cmu.edu/~larry/=sml/

https://cs.uwaterloo.ca/~ppoupart/teaching/cs489-winter18/schedule.html

1 Introduction

Keywords : Supervised (Predictive) Learning : Document classification and email spam filtering, Classifying flowers, Image classification and handwriting recognition, Face detection and recognition and Regression Unsupervised (Descriptive) Learning : Discovering clusters, Discovering latent factors(dimensionality reduction) and Discovering graph structure, Matrix completion(Image inpainting, Collaborative filtering and Market basket analysis) Reinforcement Learning Parametric vs non - Parametric models Curse of dimensionilty Overfitting Model Selection

2 Probability

Keywords : Discrete random variables, Continuous random variables, Probability of a union of two events, Joint probabilities, Conditional probability, Bayes Rule, Independence and Conditional Independence, Quantiles, Mean and variance, Some common discrete distribution and continuous distribution, Joint probability distribution, Covariance and correlation , Transformations of random variables, Monte Carlo approximation

Definition 2.1. *The entropy of a random variable X with distribution p, denoted by $\mathbb{H}(X)$ or sometimes $\mathbb{H}(p)$, is a measure of its uncertainty. In particular, for a discrete variable with K states, it is defined by* $\mathbb{H}(X) \triangleq -\sum_{k=1}^{K} p(X=k)\log_2 p(X=k)$

Usually, we use log base 2, in which case the units are called bits. If we use log base e, the units are called nats.

Definition 2.2. *Kullback - Leibler divergence (KL divergence) or relative entropy (discrete)* $\mathbb{KL}(p||q) \triangleq \sum_{k=1}^{K} p_k \log \frac{p_k}{q_k} = \sum_k p_k \log p_k - \sum_k p_k \log q_k = -\mathbb{H}(p) + \mathbb{H}(p,q)$, *where $\mathbb{H}(p,q)$ is called the cross entropy.*

The KL divergence is not a distance, since it is asymmetric.

Theorem 2.1. *(Information Inequality)* $\mathbb{KL}(p||q) \geq 0$ *with equality iff* $p = q$.

Corollary 2.1. *(Laplace's principle of insufficient reason) Discrete distribution with the maximum entropy is the uniform distribution.* ($\mathbb{H}(X) \leq \log |\mathcal{X}|$), *where* $|\mathcal{X}|$ *is the number of states for X, with equality iff* $p(x)$ *is uniform.*

Definition 2.3. *Mutual Information (MI)*
$\mathbb{I}(X;Y) \triangleq \mathbb{KL}(p(X,Y)||p(X)q(Y)) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)q(y)}$

Definition 2.4. *Pointwise mutual information (PMI) For two events (not random variables) x and y, this is defined as* $PMI(x,y) \triangleq \log \frac{p(x,y)}{p(x)q(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(x|y)}{p(y)}$

3 Generative models for discrete data

How to classify a feature vector x by applying Bayes rule to a generative classifier of the form $p(y = c|\mathbf{x}, \boldsymbol{\theta}) \propto p(\mathbf{x}|y = c, \boldsymbol{\theta})p(y = c|\boldsymbol{\theta})$

4 Gaussian Models

$$\mathcal{N}(\mathrm{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \triangleq \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathrm{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathrm{x} - \boldsymbol{\mu})\right]$$

Consider eigendecomposition of $\Sigma = U\Lambda U^\top$,

$$\Sigma^{-1} = \mathbf{U}^{-T}\Lambda^{-1}\mathbf{U}^{-1} = \mathbf{U}\Lambda^{-1}\mathbf{U}^T = \sum_{i=1}^{D} \frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^T$$

Hence we can rewrite the Mahalanobis distance between a data vector x and the mean vector μ as follows:

$$\begin{aligned}(\mathrm{x} - \mu)^T\Sigma^{-1}(\mathrm{x} - \mu) &= (\mathrm{x} - \mu)^T \left(\sum_{i=1}^{D} \frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^T\right)(\mathrm{x} - \mu) \\ &= \sum_{i=1}^{D} \frac{1}{\lambda_i}(\mathrm{x} - \mu)^T\mathbf{u}_i\mathbf{u}_i^T(\mathrm{x} - \mu) = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}\end{aligned}$$

Theorem 4.1. *(MLE for a Gaussian). If we have N i.i.d samples $x_i \sim \mathcal{N}(\mu, \Sigma)$, then the MLE for the parameters is given by*

$$\hat{\boldsymbol{\mu}}_{mle} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{x}_i \triangleq \overline{\mathrm{x}}$$

$$\hat{\Sigma}_{mle} = \frac{1}{N}\sum_{i=1}^{N} (\mathrm{x}_i - \overline{\mathrm{x}})(\mathrm{x}_i - \overline{\mathrm{x}})^T = \frac{1}{N}\left(\sum_{i=1}^{N} \mathrm{x}_i \mathrm{x}_i^T\right) - \overline{\mathrm{x}}\overline{\mathrm{x}}^T$$

5 Statistics

5.1 Bayesian Statistics

Using the posterior distribution to summarize everything we know about a set of unknown variables is at the core of Bayesian statistics.

$\hat{h}_{\text{MAP}} = \underset{h}{\text{argmax}}\, p(D|h)p(h) = \underset{h}{\text{argmax}}\, [\log p(D|h) + \log p(h)]$

$p(m|\mathcal{D}) = \frac{p(\mathcal{D}|m)p(m)}{\sum_{m\in\mathcal{M}} p(m,\mathcal{D})}$

$\hat{m} = \text{argmax}\, p(m|\mathcal{D})$ is called Bayesian model selection.

If we use a uniform prior over models, $p(m) \propto 1$, $\hat{m} = \text{argmax}\, p(\mathcal{D}|m) = \text{argmax} \int p(\mathcal{D}|\theta)p(\theta|m)d\theta$. This quantity is called the marginal likelihood, the integrated likelihood, or the evidence for model m.

BIC $\triangleq \log p(\mathcal{D}|\hat{\theta}) - \frac{\text{dof}(\hat{\theta})}{2}\log N \approx \log p(\mathcal{D})$ where dof stands for degree of freedom and $\hat{\theta}$ is the MLE for the model.

Example 5.1. *Consider Linear regression, $\hat{\sigma}^2 = \frac{\sum(y_i - \hat{\theta}_i^T x_i)^2}{N}$, BIC $\triangleq -\frac{N}{2}\log(\hat{\sigma}^2) - \frac{D}{2}\log N$*

$p_\theta(\theta) = I^{\frac{1}{2}}(\theta)$

$\boldsymbol{\eta} \to \boldsymbol{\theta} \to \mathcal{D}$

Two-Level model : $p(\boldsymbol{\eta}, \boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\eta})p(\boldsymbol{\eta})$ $\hat{\eta} = \text{argmax}\, p(\mathcal{D}|\boldsymbol{\eta}) = \text{argmax}\left[\int p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\eta})\right]$

Method	Definition
ML	$\hat{\boldsymbol{\theta}} = \text{argmax}_\theta\, p(\mathcal{D}|\theta)$
ML - II (Empirical Bayes)	$\hat{\boldsymbol{\eta}} = \text{argmax}_\eta \int p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\eta})d\boldsymbol{\theta} = \text{argmax}_\eta\, p(\mathcal{D}|\boldsymbol{\eta})$
MAP estimation	$\hat{\boldsymbol{\theta}} = \text{argmax}_\theta\, p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\eta})$
MAP - II	$\hat{\boldsymbol{\eta}} = \text{argmax}_\eta \int p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\eta})p(\boldsymbol{\eta})d\boldsymbol{\theta} = \text{argmax}_\theta\, p(\mathcal{D}|\boldsymbol{\eta})p(\boldsymbol{\eta})$

We can formalize any given statistical decision problem as a game against nature. In this game, nature pick $y \in \mathcal{Y}$(Unknown), and then generates an observation, $x \in \mathcal{X}$, which we get to see. Then we have to choose an action $a \in \mathcal{A}$. Finally, we incur some loss $L(y, a)$ which measures how compatible our action a is with nature's hidden state y. Our goal is to devise a decision procedure $\delta : \mathcal{X} \to \mathcal{A}$, which specifies the optimal action for each possible input.

$$\delta(x) = \underset{a \in \mathcal{A}}{\operatorname{argmin}} \mathbf{E}[L(y, a)]$$

In the Bayesian approach, we consider minimizes the posterior expected loss

$$\rho(a|x) = \mathbf{E}_{p(y|x)}[L(y, a)] = \sum_y L(y, a)p(y|x)$$

Hence the Bayes estimator(Bayes decision rule), is given by

$$\delta(x) = \underset{a \in \mathcal{A}}{\operatorname{argmin}} \rho(a|x)$$

5.2 Frequentist statistics

In frequentist statistics, a parameter estimate $\hat{\theta}$ is computed by applying an estimator δ to some data $\mathcal{D}$, so $\hat{\boldsymbol{\theta}} = \delta(\mathcal{D})$. Having chosen an estimator, we define its expected loss or risk as follows :

$$R(\theta^*, \delta) \triangleq \mathbf{E}_{p(\mathcal{D}|\theta^*)}\left[L(\theta^*, \delta(\mathcal{D}))\right] = \int L(\theta^*, \delta(\mathcal{D}))p(\mathcal{D}|\theta^*)d\mathcal{D}$$

6 GLM and Exponential family

6.1 Logistic Regression

Generative Approach (From joint model $p(y, \mathbf{x})$ to conditional model $p(y|\mathbf{x})$) VS Discriminative Approach (Fit a model $p(y|\mathbf{x})$)

Model (Binary - Class) : $p(y|\mathbf{x}, \mathbf{w}) = \text{Bernoulli}(y|sigmoid(\mathbf{w}^T\mathbf{x}))$

Fitting : minimize the negative log-likelihood

1. Steepest Descent
2. Newton Method
3. IRLS
4. Quasi - Newton

Model (Multi - Class) : $p(y|\mathbf{x}, \mathbf{W}) = \frac{\exp(\mathbf{w}_c^T\mathbf{x})}{\sum_{c'=1}^{C}\exp(\mathbf{w}_{c'}^T\mathbf{x})}$

Why Exponential family

1. Under certain regularity conditions, the exponential family is the only family of distributions with finite-sized sufficient statistics(**Pitman-Koopman-Darmois Theorem**), meaning that we can compress the data into a fixed-sized summary without loss of information.
2. The exponential family is the only family of distributions for which conjugate priors exist, which simplifies the computation of the posterior.

3. The exponential family is the only family of distributions for which conjugate priors exist, which simplifies the computation of the posterior (**Maximum Entropy**).
4.
5.

7 Directed graphical models (Bayes Net)

Definition 7.1. *Conditional Independence (CI)* $X \perp\!\!\!\perp Y|Z \iff p(X,Y|Z) = p(X|Z)p(Y|Z)$

Graphical Model is a way to represent a joint distribution by making CI assumptions.

Graph Keywords : Graph, nodes, vertices, edges, adjacency matrix, undirected, directed, self loops, parent, child, family, root, leaf, ancestors, descendants, neighbors, degree, cycle, directed acyclic graph(DAG), topological ordering, path, tree, forest, subgraph, clique

Definition 7.2. *A directed graphical model or DGM is a GM whose graph is a DAG which is also known as Bayesian networks, belief networks and causal networks.*

Example 7.1. *1. Naive Bayes Classifiers :* $p(y, \mathbf{x}) = p(y)\prod_{i=1}^{D} p(x_j|y)$

2. (First-order) Markov and hidden Markov models : $\mathbf{x}_t$ *is the observed variable,* z_t *is the hidden variable. The* $p(z_t|z_{t-1})$ *is the transition model,* $p(\mathbf{x}_t|z_t)$ *is the observation model.*

Inference

Learning

8 Mixture models and The EM Algorithm

Definition 8.1. *(Mixture Models) Define latent variable* $z_i \in \{1, ..., K\}$ *and* $p(z_i) = Cat(\boldsymbol{\pi})$*. Denote* $p(\mathbf{x}_i|z_i = k) = p_k(\mathbf{x}_i)$*, where* P_k *is the k-th base distribution. Then* $p(\mathbf{x}_i|\boldsymbol{\theta}) = \sum_{i=1}^{K} \pi_k p_k(\mathbf{x}_i|\boldsymbol{\theta})$ *is called the Mixture Models, where* $\sum_{k=1}^{K} \pi_k = 1$*.*

Mixtures of Gaussians, Multinoullis, Clustering and Experts

8.1 EM(Expectation maximization) algorithm

The goal is to maximize the log likelihood of the observed data : $\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p(\mathbf{x}_i|\boldsymbol{\theta}) = \sum_{i=1}^{n} \log[\sum_{z_i} p(\mathbf{x}_i, z_i|\boldsymbol{\theta})]$. Complete data log likelihood : $\ell_c(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p(\mathbf{x}_i, z_i|\boldsymbol{\theta})$.

$$\mathbf{E-Step}: Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) = \mathbf{E}\left[\ell_c(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{t-1}\right]$$
$$\mathbf{M-Step}: \boldsymbol{\theta}^t = \underset{\theta}{\text{argmax}}\ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1})$$

Theoretical basis for EM :

9 Latent Linear Models

9.1 Factor Analysis(FA)

Real-valued latent variables $\mathbf{z}_i \in \mathbb{R}^L$. Gaussian prior $p(\mathbf{z}_i) = \mathcal{N}(\mathbf{z}_i|\mu_0, \Sigma_0)$, If $\mathbf{x}_i \in \mathbb{R}^D$, so the "linear regression" model $p(\mathbf{x}_i|\mathbf{z}_i, \theta) = \mathcal{N}(\mathbf{W}\mathbf{z}_i + \mu, \Psi)$ where $\mathbf{W}$ is a $D \times L$ matrix, known as the factor loading matrix, and Ψ is a $D \times D$ diagonal matrix.

$p(\mathbf{z}_i|\mathbf{x}_i, \theta) = \mathcal{N}(\mathbf{z}_i|m_i, \Sigma_i)$, $\Sigma \triangleq (\Sigma_0^{-1} + \mathbf{W}^T\Psi^{-1}\mathbf{W})^{-1}$, $m_i = \Sigma_i(\mathbf{W}^T\Psi^{-1}(\mathbf{x}_i - \mu) + \Sigma_0^{-1}\mu_0)$

9.2 Principle Component Analysis(PCA)

10 Sparse Linear models

Useful Applications of feature selection(Sparsity) :

1. Small N, Large D problem [fatter data, gene microarrays]
2. Signal Processing, sparse representation of the signals in terms of a small number of wavelet basis functions

1. **Bayesian variable selection** Let $\gamma_j = 1$ if feature j is "relevant" and let $\gamma_j = 0$ otherwise. Goal : compute the posterior over models $p(\gamma|\mathcal{D}) = \frac{e^{-f(\gamma)}}{\sum_{\gamma'} e^{-f(\gamma')}}$ where $f(\gamma) \triangleq -[\log p(\mathcal{D}|\boldsymbol{\gamma}) + \log p(\boldsymbol{\gamma})]$

   Interpreting the posterior over a large number of models is quite difficult, so we will seek various summary statistics.

   Posterior mode, MAP estimate: $\hat{\gamma} = \text{argmax}\, p(\boldsymbol{\gamma}|\mathcal{D}) = \text{argmax}\, \frac{e^{-f(\gamma)}}{\sum_{\gamma'} e^{-f(\gamma')}} = \text{argmin}\, f(\boldsymbol{\gamma})$

   Median model : $\hat{\gamma} = \{j : p(\gamma_j = 1|\mathcal{D}) > 0.5\}$
2. The spike and slab model From the Bernoulli - Gaussian model to ℓ_0 regularization
3. Since there are 2^D possible models (bit vectors), it will be impossible to compute the full posterior in general, and even finding summaries, such as the MAP estimate or marginal inclusion probabilities will be intractable. We will therefore focus on algorithmic speedups. Since there are 2^D models, we cannot explore the full posterior or find the globally optimal model. All of the methods we will discuss involve searching through the space of models and evaluating the cost $f(\gamma)$ at each point. (**Wrapper method**) In order to make wrapper methods efficient, it is important that we can quickly evaluate the score function for some new model, γ', given the score of a previous model, γ.

Greedy Search : Single best replacement, Orthogonal least squares, Orthogonal matching pursuits, Matching pursuits, Backwards selection, Forwards - backwards algorithm, Bayesian Matching pursuit

Stochastic search : MCMC

10.1 ℓ_1 regularization

Consider a prior of the form $p(w|\lambda) = \prod_{j=1}^{D} \text{Lap}(\omega_j|0, 1/\lambda) \propto \prod_{j=1}^{D} e^{-\lambda|\omega_j|}$

The penalized negative log likelihood has the form $f(w) = -\log p(\mathcal{D}|w) - \log p(w|\lambda) = \text{NLL}(w) + \lambda||w||_1$. Note this can be thought of as a convex approximation to the non-convex ℓ_0 objective.

In the case of linear regression, ℓ_1 objective becomes $f(\omega) = \sum_{i=1}^{N} -\frac{1}{2\sigma^2}(y_i - (\omega_0 + \omega^T x_i))^2 + \lambda||\omega||_1$

This method is known as basis pursuit denoising. The BPDN objective is the following non - smooth objective function: $\min_w \text{RSS}(w) + \lambda||w||_1$.

Definition 10.1. *least absolute shrinkage and selection operator (lasso)*
$\min_w RSS(w)$ *s.t.* $||w||_1 \leq B$

Definition 10.2. *Ridge regression :* $\min_w RSS(w)$ *s.t.* $||w||_2^2 \leq B$

10.1.1 Why does ℓ_1 regularization yield sparse solutions

From the theory of constrained optimization, we know that the optimal solution occurs at the point where the lowest level set of the objective function intersects the constraint surface (assuming the constraint is active). It should be geometrically clear that as we relax the constraint B, we "grow" the ℓ_1 "ball" until it meets the objective; the corners of the ball are more likely to intersect the ellipse than one of the sides, especially in high dimensions, because the corners "stick out" more. The corners correspond to sparse solutions, which lie on the coordinate axes. By contrast, when we grow the ℓ_2 ball, it can intersect the objective at any point; there are no "corners", so there is no preference for sparsity.

The lasso objective has the form $f(\theta) = \text{RSS}(\theta) + \lambda||w||_1$. Since $||w||_1$ term is not differentiable whenever $w_j = 0$.

Definition 10.3. *subderivative (subgradient) of a convex function* $f : \mathcal{I}^n \to \mathbb{R}$ *at* $\boldsymbol{\theta}_0$ *to be* $\boldsymbol{g}$ *such that* $f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}_0) \geq (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^T \boldsymbol{g}\ \forall \boldsymbol{\theta} \in \mathcal{I}^n$ *where* $\mathcal{I}^n$ *containing* $\boldsymbol{\theta}_0$*. Note* g *is a linear lower bound to the function* f *at* θ_0*.*

Definition 10.4. *The set* $[a, b]$ *of all subderivatives is called the* **subdifferential** *of the function* f *at* θ_0 *and is denoted* $\partial f(\theta)|_{\theta_0}$*, where* $a = \lim_{\theta \to \theta^-} \frac{f(\theta) - f(\theta_0)}{\theta - \theta_0}$*,* $b = \lim_{\theta \to \theta^+} \frac{f(\theta) - f(\theta_0)}{\theta - \theta_0}$*.*

For example, in case of $f(\theta) = |\theta|$, the subderivative is given by $\partial f(\theta) = \begin{cases} \{-1\} & \text{if } \theta < 0 \\ [-1, 1] & \text{if } \theta = 0 \\ \{+1\} & \text{if } \theta > 0 \end{cases}$

10.1.2 Optimality conditions for lasso

10.1.3 Comparison of LS, Lasso, Ridge and subset selection

Suppose $\mathbf{X}$ are orthonormal (e.g $\mathbf{X}^T\mathbf{X} = I$),

$$\text{RSS}(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 = \mathbf{y}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{y} = C + \sum_k w_k^2 - 2\sum_k \sum_i w_k x_{ik} y_i$$

1. MLE
2. Ridge
3. Lasso
4. Subset selection

10.1.4 Regularization Path

Definition 10.5. *Plot of $\widehat{\omega}_j(\lambda)$ vs λ for each feature j;*

LARS(least angle regression and shrinkage)

10.1.5 Algorithms

1. Coordinate Descent
2. LARS and other homotopy methods
3. Proximal and gradient projection method
4. EM for lasso

10.2 ℓ_1 regularization : extensions

10.2.1 Group Lasso

10.2.2 Fused Lasso

10.2.3 Elastic net

10.3 Non - convex regularizers

10.4 Sparse Coding

11 Kernels

Definition 11.1. *A real-valued function of two arguments, $k(\mathbf{x}, \mathbf{x}') \in \mathbb{R}$, for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$.*

Note that the function can be symmetric and non-negative so it can be interpreted as measure of similarity.

Example 11.1. *1. Radial basis function(RBF) kernel* $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$*, squared exponential kernel (Gaussian kernel)* $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^T\Sigma^{-1}(\mathbf{x}-\mathbf{x}')\right)$

2. *Cosine similarity* $k(\mathbf{x}_i, \mathbf{x}_{i'}) = \frac{\mathbf{x}_i^T\mathbf{x}_{i'}}{\|\mathbf{x}_i\|_2\|\mathbf{x}_{i'}\|_2}$

3. *Mercer kernel (positive definite kernel)* $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')$ *where ϕ depends on the eigen functions of k (D is potentially infinite dimensional space).*

   Gram matrix $\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) \cdots k(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) \cdots k(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}$

   Theorem 11.1. *(Mercer theorem) If the gram matrix is positive definite, we can compute an eigenvector decomposition,* $\mathbf{K} = \mathbf{U}^T\mathbf{\Lambda}\mathbf{U}$*, where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues $\lambda_i > 0$. Then $k_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ where $\phi(\mathbf{x}_i) = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}_{:i}$*

4. *Linear kernels* $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$

5. *Matern kernel* $k(r) = \frac{2^{1-v}}{\Gamma(v)}\left(\frac{\sqrt{2v}r}{\ell}\right)^v K_v\left(\frac{\sqrt{2v}r}{\ell}\right)$ *where $r = \|\mathbf{x}-\mathbf{x}'\|$, $v > 0$, $\ell > 0$ and K_v is a modified Bessel function.*

6. *String Kernels*

7. *Pyramid match kernels*

8. *Kernels derived from probabilistic generative models*

11.1 Using kernels inside GLM

Definition 11.2. *Kernel machine : the input feature vector has the form* $\phi(\mathbf{x}) = [k(\mathbf{x}, \mu_1), \cdots, k(\mathbf{x}, \mu_K)]$ *where* $\mu_k \in \mathbf{X}$ *are a set of K centroids.*

For logistic regression : $p(y|\mathbf{x}, \theta) = \text{Ber}(\mathbf{w}^T \phi(\mathbf{x}))$. For linear regression : $p(y|\mathbf{x}, \theta) = \mathcal{N}(\mathbf{w}^T \phi(\mathbf{x}), \sigma^2)$.

How do we choose the centroids $\boldsymbol{\mu}_k$? Simpler approach is $\phi(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), \cdots, k(\mathbf{x}, \mathbf{x}_N)]$

Sparse vector machine : L1VM (ℓ_1 - regularized vector machine), L2VM (ℓ_2 - regularized vector machine), RVM(relevance vector machine) and SVM(support vector machine)

11.2 Kernel trick

Definition 11.3. *Instead working with the original feature vectors* $\mathbf{x}$*, but modify the algorithm so that it replaces all inner products of the form* $\langle \mathbf{x}, \mathbf{x}' \rangle$ *with a call to the kernel function* $k(\mathbf{x}, \mathbf{x}')$*.*

11.2.1 Kernelized nearest neighbor classification

11.2.2 Kernelized K-medoids clustering

11.2.3 Kernelized ridge regression

11.2.4 Kernel PCA

11.3 Support vector machines

Consider ℓ_2 regularized empirical risk function

$$J(\mathbf{w}, \lambda) = \sum_{i=1}^{N} L(y_i, \hat{y}_i) + \lambda \|\mathbf{w}\|^2$$

where $\hat{y}_i = \mathbf{w}^T \mathbf{x}_i + w_0$. Notes : If L is quadratic loss, this is equivalent to ridge regression, and if L is the log-loss, this is equivalent to logistic regression.

Definition 11.4. *Support vector machine is combination of the kernel trick (replace* $\mathbf{x}^T \mathbf{x}'$ *by* $k(\mathbf{x}, \mathbf{x}')$ *) and modified loss function (to ensure the solution is sparse)*

11.3.1 Regression

Definition 11.5. *Epsilon insensitive loss function* $L_\epsilon(y, \hat{y}) = \begin{cases} 0 & \text{if } |y - \hat{y}| < \epsilon \\ |y - \hat{y}| - \epsilon & \text{otherwise} \end{cases}$

The corresponding objective is

$$J = \frac{1}{\lambda}\sum_{i=1}^{N} L_\epsilon(y_i, \hat{y}_i) + \frac{1}{2}\|\mathbf{w}\|^2$$

Note that the objective is convex and unconstrained but not differentiable. So we introduce slack variable $\xi_i^+ \geq y_i - \hat{y}_i - \epsilon$ and $-\xi_i^- \leq y_i - \hat{y}_i + \epsilon$.

So we can rewrite the objective as follows :

$$J = \frac{1}{\lambda}\sum_{i=1}^{N}(\xi_i^- + \xi_i^+) + \frac{1}{2}\|\mathbf{w}\|^2$$

with the passivity constraints $\xi_i^- \geq 0$ and $\xi_i^+ \geq 0$. This is a standard quadratic program in $2N + D + 1$ variables. The soultion is given by $\hat{\mathbf{w}} = \sum_i \alpha_i \mathbf{x}_i$. The $\mathbf{x}_i$ for which $\alpha_i > 0$ are called the support vectors.

11.3.2 Classification

Definition 11.6. *(Hinge Loss)* $L_{hinge}(y, \eta) = \max(0, 1 - y \cdot \eta) = (1 - y \cdot \eta)_+$

The objective has the form

$$\min_{\mathbf{w}, w_0} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\lambda}\sum_{i=1}^{N}(1 - y_i(\mathbf{w}^T\mathbf{x} + w_0))_+$$

Since this is not differenetiable, by introducing slack variable ξ_i, the object has the form

$$\min_{\mathbf{w}, w_0, \boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\lambda}\sum_{i=1}^{N}\xi_i \quad \text{s.t} \quad \xi_i \geq 0,\ y_i(\mathbf{x}_i^T\mathbf{w} + w_0) \geq 1 - \xi_i,\ i = 1, \cdots, N$$

This is a quadratic program in $N + D + 1$ variables subject to $O(N)$ constraints.

The solution is $\hat{\mathbf{w}} = \sum_i \alpha_i \mathbf{x}_i$ where $\alpha_i = \lambda_i y_i$ and $\boldsymbol{\alpha}$ is sparse. The prediction is $\hat{y} = \text{sgn}(\hat{w}_0 + \hat{\mathbf{w}}^T\mathbf{x})$

11.4 Kernels for building generative models