Linear Classifiers

$f(x) = \Theta \cdot x + \Theta_0 = \sum_{i=1}^{d} \Theta_i x_i + \Theta_0$, $\hat{y} = \begin{cases} 1 & \text{if } f(x) \geq 0 \\ -1 & \text{if } f(x) < 0 \end{cases}$ $x \in \mathbb{R}^d, \Theta \in \mathbb{R}^d, \Theta_0 \in \mathbb{R}$

Decision Boundary: $H = \{x \in \mathbb{R}^d : f(x) = 0\} = \{x \in \mathbb{R}^d : \Theta \cdot x + \Theta_0 = 0\}$

normal to Θ: For any $x, \tilde{x} \in H$ $\Theta \cdot (x - \tilde{x}) = 0$, Θ_0 as offset: $-\frac{\Theta_0}{\|\Theta\|}$ (0,0) to H

by $0 = f(c\Theta) = c\Theta \cdot \Theta + \Theta_0 = c\|\Theta\|^2 + \Theta_0$ where $c\|\Theta\| = -\Theta_0 \frac{\|\Theta\|}{\|\Theta\|^2} = -\Theta_0 / \|\Theta\|$

Can augment x and Θ w/ 1 and Θ_0 to have $f(x) = \Theta \cdot x$, $\hat{y} = \begin{cases} 1 & \text{if } f(x) \geq 0 \\ -1 & \text{if } f(x) < 0 \end{cases}$, $H = \{x : \Theta \cdot x = 0\}$

Perceptron algorithm

Input: $(x_1, y_1) \ldots (x_n, y_n) \in \mathbb{R}^d \times \{\pm 1\}$

while some $y^i \neq \text{sign}(\Theta \cdot x_i)$ (improve Θ)

pick some misclassified (x_i, y_i)

$\Theta \leftarrow \Theta + y_i x_i$ (update Θ)

return Θ

Convergence Theorem

Given linearly separable data, for any choice of updates, the perceptron algorithm terminates with all data correctly classified.

No more than R^2/γ^2 updates

(radius) $R = \max_i \|x^i\|$ $\gamma = \min_i \frac{y^i (\Theta \cdot x^i)}{\|\Theta\|}$ (margin)

Since $\Theta = \sum_i \alpha^i y^i x^i$, $\Theta \cdot x = \sum_i \alpha^i y^i x^i \cdot x$ so just need inner product space

View as stochastic gradient descent, where $J = \sum_i (-y^i (\Theta \cdot x^i))_+$, $(\alpha)_+ = \begin{cases} \alpha & \text{if } \alpha > 0 \\ 0 & \text{otherwise} \end{cases}$

Converges only if data are separable, time to convergence depends on margin, solution depends on starting point

Support Vector Machines

Optimize linear classification by choosing classifier maximizing $y^i \frac{\Theta \cdot x^i}{\|\Theta\|}$

Find a separating Θ, $\delta \leq y^i \Theta \cdot x^i$ and scale Θ by $1/\delta$ $\rightarrow \min_i y^i \frac{\Theta \cdot x^i}{\|\Theta\|} = 1/\|\Theta\|$

$\max_\Theta \frac{1}{\|\Theta\|} \Rightarrow \min_\Theta \|\Theta\|^2$ s.t. $y^i \Theta \cdot x^i \geq 1$ (quadratic program)

Support vectors are points satisfying constraints with equality (compressed training data)

$\Theta = \sum_i \alpha^i y^i x^i$ ($\alpha^i \neq 0$ for s.v.), $\Theta \cdot x = \sum_i \alpha^i y^i x^i \cdot x$, $\|\Theta\|^2 = \sum_{i,j} \alpha^i \alpha^j y^i y^j x^i \cdot x^j$

(just inner products, so we can work in any inner product space)

Hard Margin SVM

$\min_\Theta \|\Theta\|^2$ s.t. $y^i \Theta \cdot x^i \geq 1$ (data linearly separable)

Soft Margin SVM

Relax inequalities $y^i \Theta \cdot x^i \geq 1$ by introducing slack variables $\xi_i \geq 0$, enforcing milder constraint $y^i \Theta \cdot x^i \geq 1 - \xi_i$ and adding ξ_i to cost function

$\min_\Theta \|\Theta\|^2 + C \sum_{i=1}^n \xi_i$ s.t. $y^i \Theta \cdot x^i \geq 1 - \xi_i$, $\xi_i \geq 0$ ($\xi_i = (1 - y^i \Theta \cdot x^i)_+$)

$\hookrightarrow \min_\Theta \|\Theta\|^2 + C \sum_{i=1}^n (1 - y^i \Theta \cdot x^i)_+$

Regularization: small $C \rightarrow$ small $\|\Theta\|^2$, large $C \rightarrow$ small misclassification

Feature Selection

Linear classifier: $f(x) = \Theta \cdot x + \Theta_0$, quadratic: $f(x) = \|x - c\|^2 - r^2$ $\begin{pmatrix} 1 \\ 1 \\ -2c_1 \\ -2c_2 \\ c_1^2 + c_2^2 - r^2 \end{pmatrix} \cdot \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix}$

Quadratic is simply linear classifier with features $\phi(x)$, so $f(x) = \Theta \cdot \phi(x)$

want to balance feature complexity and degree of fit

Kernels

We can work in inner product space, so use $K(x^i, x^j) = x^i \cdot x^j$ or $\phi(x^i) \cdot \phi(x^j)$

Solve optimization problem given $K(x^i, x^j)$ to determine α^i for $\Theta = \sum_j \alpha^j y^j \phi(x^j)$ and then compute classifier for x via $\Theta \cdot \phi(x) = \sum_j \alpha^j y^j (\phi(x^j) \cdot \phi(x)) \rightarrow K(x_j, x)$

Degree-m polynomial kernel given by $K_m(x, \tilde{x}) = (1 + x \cdot \tilde{x})^m$

Radial basis function kernel (sim to least squares) $K_{rbf}(x, \tilde{x}) = \exp(-\gamma \|x - \tilde{x}\|^2)$

Scaling parameter γ affects smoothness (small $\rightarrow$ large [illegible] $\rightarrow$ few points influence)

Kernels add modularity to classifier training since same optimization procedure can be used (need kernel matrix $K_{ij} = K(x_i, x_j)$)

Degree-m polynomial kernel on $\mathbb{R}^d$ is inner product with $\binom{m+d}{m}$ features

Convex Optimization and SVMs

Convex problems are those in which it is possible to draw line, and stay within

Consider the convex optimization problem $p^* = \min_{x \in \mathbb{R}^n} f_0(x)$ s.t. $f_i(x) \leq 0$, $i = 1, 2, \ldots, m$

Rewriting constraints as penalties $p^* = \min_{x \in \mathbb{R}^n} f_0(x) + \begin{cases} 0 & \text{if all } f_i(x) \leq 0 \\ +\infty & \text{otherwise} \end{cases}$

Replace constraint penalty with something smaller:

Introduce Lagrange multipliers (dual variables) $\lambda_1, \ldots, \lambda_m \geq 0$ and define

Lagrangian $L : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$ as $L(x, \lambda) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x)$

λ_i cost of violating constraint $f_i(x) \leq 0$

L defines a saddle point game: one player (MIN) chooses x to minimize L and other player (MAX) chooses λ to maximize L. If MIN violates $f_i(x) > 0$, MAX $L \rightarrow \infty$.

primal problem $p^* = \min_x \max_{\lambda \geq 0} L(x, \lambda)$ (infeasible $x \rightarrow \infty$, feasible $x \rightarrow \lambda_i f_i(x) = 0$)

dual problem $d^* = \max_{\lambda \geq 0} \min_x L(x, \lambda)$ (max plays first)

In a zero-sum game, always better to play second: $p^* = \min_x \max_{\lambda \geq 0} L(x, \lambda) \geq \max_{\lambda \geq 0} \min_x L(x, \lambda) = d^*$ (weak duality)

If there is a saddle point (x^*, λ^*) so that for all x and $\lambda \geq 0$, $L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*)$, then $p^* = \min_x \max_{\lambda \geq 0} L(x, \lambda) = \max_{\lambda \geq 0} \min_x L(x, \lambda) = d^*$ (strong duality)

Complementary Slackness

If $p^* = d^*$ and we have primal solution x^* and dual solution λ^*, then for the i^{th} constraint ($f_i(x) \leq 0$), $\lambda_i^* f_i(x^*) = 0$, $\rightarrow f_i(x^*) < 0 \Rightarrow \lambda_i = 0$, $\lambda_i > 0 \Rightarrow f_i(x^*) = 0$

As a result, every term in sum $\sum_{i=1}^m \lambda_i^* f_i(x^*) \geq 0$ since $\lambda_i^* \geq 0$, $f_i(x^*) \leq 0$

Karush-Kuhn-Tucker optimality conditions

Suppose f_0, f_i are convex and differentiable. Then x and λ optimal iff:

① Primal feasibility: $f_i(x) \leq 0$ ② Dual feasibility: $\lambda_i \geq 0$

③ Complementary slackness: $\lambda_i f_i(x) = 0$ ④ Stationarity: $\nabla f_0(x) + \sum_i \lambda_i \nabla f_i(x) = 0$.

Hard Margin SVM

Write $\min_\Theta \|\Theta\|^2$ s.t. $y^i \Theta \cdot x^i \geq 1$ as $L(\Theta, \alpha) = \frac{1}{2}\|\Theta\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i \Theta^T x_i)$, $g(\alpha) = \min_\Theta L(\Theta, \alpha)$

$\nabla L(\Theta, \alpha) = \Theta + \sum_{i=1}^n \alpha_i (-y_i x_i) \Longrightarrow \Theta^* = \sum_{i=1}^n \alpha_i y_i x_i$, $g(\alpha) = \frac{1}{2}\Theta^{*T}\Theta^* + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n y_i \Theta^{*T} x_i = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j$

If $\exists$ feasible Θ, strong duality allows us to express $\Theta^* \rightarrow d^*$ $\max_\alpha \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j$ s.t. $\alpha_i \geq 0$, $i = 1, 2, \ldots, n$

and we know $\alpha_i > 0 \Rightarrow y_i \Theta^* \cdot x = 1$, $y_i \Theta^* \cdot x > 1 \Rightarrow \alpha_i = 0$

Express solution as kernel k: $\hat{y}(x) = \text{sign}(\langle \Theta, \phi(x) \rangle) = \text{sign}(\sum_{i=1}^n \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle) = \text{sign}(\sum_i \alpha_i y_i k(x_i, x))$ where α solves dual $\min_\alpha \frac{1}{2}\alpha^T \text{diag}(y) K \text{diag}(y) \alpha - \alpha^T \mathbf{1}$

Soft Margin SVM

Write $\min_\Theta \frac{1}{2}\|\Theta\|^2 + \frac{C}{n}\sum_{i=1}^n (1 - y^i \Theta \cdot x^i)_+$ as $L(\Theta, \xi, \alpha, \lambda) = \frac{1}{2}\|\Theta\|^2 + \frac{C}{n}\sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i (1 - y_i \Theta^T x_i - \xi_i) - \sum_{i=1}^n \lambda_i \xi_i$

based off of QP $\min_{\Theta, \xi} \frac{1}{2}\|\Theta\|^2 + \frac{C}{n}\sum_{i=1}^n \xi_i$ s.t. $\xi_i \geq 0$, $\xi_i \geq 1 - y^i \Theta \cdot x^i$

$\nabla_\Theta L = \Theta - \sum_{i=1}^n \alpha_i y_i x_i$, $\nabla_\xi L = \frac{C}{n} - \alpha_i - \lambda_i \Rightarrow \Theta = \sum_i \alpha_i y_i x_i$, $\frac{C}{n} = \alpha_i + \lambda_i$

Dual problem given by $\max_{\alpha, \lambda} \sum \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$ s.t. $\alpha_i \geq 0$, $\lambda_i \geq 0$, $\alpha_i + \lambda_i = \frac{C}{n}$

consequences of complementary slackness: ① $\alpha_i^*(1 - y_i x_i \cdot \Theta^* - \xi_i^*) = 0$ ($\alpha_i^* > 0$ implies $y_i x_i \cdot \Theta^* = 1 - \xi_i^* \leq 1$, so s.v. in wrong halfspace), ② $\lambda_i^* \xi_i^* = 0$ ($y_i x_i \cdot \Theta^* < 1 \Rightarrow \xi_i^* > 0 \Rightarrow \lambda_i^* = 0$, $\alpha_i^* = \frac{C}{n}$)

Primal - increasing C penalizes errors more (low margin); Dual - decreasing C forces α_i to be small (outlier [illegible])

Decision Theory

The Prediction Problem: Given a training set $(x_1, y_1) \ldots (x_n, y_n)$, choose function $f : X \rightarrow Y$ for later pairs

$f(x)$ is good predictor of Y by Loss function $\ell : Y \times Y \rightarrow \mathbb{R}$ where $\ell(\hat{y}, y)$ is cost of pred $\hat{y}$ when y

① Assume that (x_i, y_i) and (x, y) chosen iid according to probability distribution $X \times Y$

② A good prediction means small expected loss, choose f with small risk $R(f) = \mathbb{E}\,\ell(f(X), Y)$

Risk is misclassification probability: $R(f) = \mathbb{E}\,\ell(f(X), Y) = \mathbb{E}\,\mathbf{1}[f(X) \neq Y] = \Pr(f(X) \neq Y)$

Two class example: $R(f) = \mathbb{E}\,\ell(f(X), Y) = \mathbb{E}\mathbb{E}[\ell(f(X), Y)|X] = \mathbb{E}[\ell(f(X), +1)P(Y = +1|X) + \ell(f(X), -1)P(Y = -1|X)]$

Bayes Decision Rule $f^*(x) = \begin{cases} 1 & \text{if } P(Y = 1|X) > P(Y = -1|X) \\ -1 & \text{otherwise} \end{cases}$ $R(\alpha_i|x) = \sum_j \lambda_{ij} P(w_j|x)$

Risk is minimized when $f = f^*$, where optimum Bayes risk $R^* = \inf_f R(f) = R(f^*)$

Excess Risk $R(f) - R^* = \mathbb{E}[\mathbf{1}[f(X) \neq f^*(X)]|P(Y = +1|X) - P(Y = -1|X)|] = \mathbb{E}[\mathbf{1}[f(X) \neq f^*(X)]|2P(Y = +1|X) - 1|]$

can be quantified in terms of a certain distance from f^*.

Risk in Regression

Risk is the expected squared error: $R(f) = \mathbb{E}\,\ell(f(X), Y) = \mathbb{E}(f(X) - Y)^2 = \mathbb{E}\mathbb{E}[(f(X) - Y)^2|X]$

Minimize conditional expectation of loss $\mathbb{E}[(f(X) - Y)^2|X]$ by $f^*(X) = \mathbb{E}[Y|X]$

Bias-Variance Decomposition

$R(f) = \mathbb{E}(f(X) - Y)^2 = \mathbb{E}\mathbb{E}[(f(X) - Y)^2|X] = \mathbb{E}\mathbb{E}[(f(X) - \mathbb{E}[Y|X] + \mathbb{E}[Y|X] - Y)^2|X] = \mathbb{E}[(f(X) - \mathbb{E}[Y|X])^2] + \mathbb{E}[(\mathbb{E}[Y|X] - Y)^2]$

$= \mathbb{E}\mathbb{E}[(f(X) - \mathbb{E}[Y|X])^2 + (\mathbb{E}[Y|X] - Y)^2 + 2(f(X) - \mathbb{E}[Y|X])(\mathbb{E}[Y|X] - Y)|X]$ (bias², variance)

Minimizer is $f^*(X) = \mathbb{E}[Y|X]$ so $R(f) = \mathbb{E}[(f(X) - f^*(X))^2] + \mathbb{E}[(f^*(X) - Y)^2] \Rightarrow R(f) - R^* = \mathbb{E}[(f(X) - f^*(X))^2]$

Using randomly chosen training data to choose $\hat{f}$, we would like $\mathbb{E}R(\hat{f})$ to be small:

$\mathbb{E}R(\hat{f}) - R^* = \mathbb{E}[(\hat{f}(X) - f^*(X))^2] = \mathbb{E}[(\hat{f}(X) - \mathbb{E}\hat{f}(X) + \mathbb{E}\hat{f}(X) - f^*(X))^2] = \mathbb{E}[(\hat{f}(X) - \mathbb{E}\hat{f}(X))^2] + \mathbb{E}[(\mathbb{E}\hat{f}(X) - f^*(X))^2]$ (variance, bias²)

Three approaches to choosing classifiers:

① Estimate a generative model using $P(Y)$ and $P(X|Y)$, use Bayes Theorem: $P(Y = +1|X) = \frac{P(X|Y = +1)P(Y = +1)}{P(X|Y = +1)P(Y = +1) + P(X|Y = -1)P(Y = -1)}$ and define plug-in estimator for discriminative model

② Estimate a discriminative model by $P(Y|X)$ and use $\hat{P}(Y|X)$ as $P(Y|X)$ plug-in estimator (only require accuracy around $P(Y = +1|X) = 1/2$)

③ Choose a classifier directly (perceptron, SVM, ...) based on optimization criterion

Generative and Discriminative Models

Recall $\Pr(A) = \mathbb{E}\Pr(A|X)$ · $\mathbb{E}Y = \mathbb{E}\mathbb{E}(Y|X)$ · $\Pr(A) = \sum_i \Pr(A \cap B_i) = \sum_i \Pr(A|B_i)\Pr(B_i) = \mathbb{E}\Pr(A|X)$

$\mathbf{1}_E = \begin{cases} 1 & \text{when } E \text{ occurs} \\ 0 & \text{otherwise} \end{cases}$ so $\Pr(E) = \mathbb{E}(\mathbf{1}_E)$ and $\mathbb{E}\mathbf{1}_E = \mathbb{E}(\mathbb{E}(\mathbf{1}_E|X))$ and $Y = \sum_y y\mathbf{1}[Y = y] \Rightarrow$

$\mathbb{E}Y = \mathbb{E}\sum_y y\mathbf{1}[Y = y] = \sum_y y\,\mathbb{E}\mathbf{1}[Y = y] = \sum_y y\,\mathbb{E}\mathbb{E}[\mathbf{1}_{Y=y}|X] = \mathbb{E}\mathbb{E}[\sum_y y\mathbf{1}_{Y=y}|X] = \mathbb{E}\mathbb{E}(Y|X)$

For Gaussian class conditionals $P(X|Y = +1)$, $P(X|Y = -1)$ (with same variance), the posterior probability is logistic: $P(Y = +1|X) = \frac{1}{1 + \exp(-x \cdot \Theta - \Theta_0)}$

Suppose the class conditional distributions are Gaussian:

$P(X|Y = +1) = N(\mu_+, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp(-\frac{1}{2}(x - \mu_+)^T\Sigma^{-1}(x - \mu_+))$ $P(X|Y = -1) = N(\mu_-, \Sigma)$

Consider the one-dimensional case $x, \mu_+, \mu_- \in \mathbb{R}$, $\Sigma = \sigma^2$: $P(X|Y = +1) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x - \mu_+)^2}{2\sigma^2})$

(apply Bayes' Rule) $P(Y = +1|X) = \frac{P(X|+1)P(+1)}{P(X|+1)P(+1) + P(X|-1)P(-1)} = \frac{P(+1)\exp(-\frac{(x - \mu_+)^2}{2\sigma^2})}{P(+1)\exp(-\frac{(x - \mu_+)^2}{2\sigma^2}) + P(-1)\exp(-\frac{(x - \mu_-)^2}{2\sigma^2})}$

(simplify and substitute) $P(Y = +1|X) = \frac{1}{1 + \frac{P(-1)}{P(+1)}\exp(\frac{(x - \mu_+)^2 - (x - \mu_-)^2}{2\sigma^2})} = \exp(\frac{\mu_- - \mu_+}{\sigma^2}x + \frac{\mu_+^2 - \mu_-^2}{2\sigma^2} + \log(\frac{P(-1)}{P(+1)}))$

$\exp(\Theta \cdot x - \Theta_0)$ $\Theta = \frac{\mu_+ - \mu_-}{\sigma^2}$, $\Theta_0 = \frac{\mu_-^2 - \mu_+^2}{2\sigma^2} - \log(\frac{P(-1)}{P(+1)})$

In d-dimensions, $\Theta = \Sigma^{-1}(\mu_+ - \mu_-)$, $\Theta_0 = \frac{\mu_-^T\Sigma^{-1}\mu_- - \mu_+^T\Sigma^{-1}\mu_+}{2} + \log(\frac{P(-1)}{P(+1)})$

The discriminant direction minimizes Gaussian data overlap

Linear Discriminant Analysis

Model class conditional distributions as Multivariate Gaussians, $P(X|Y = i) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp(-\frac{1}{2}(x - \mu_i)^T\Sigma^{-1}(x - \mu_i))$

$\log \ell = -\frac{1}{2}(x - \mu_i)^T\Sigma^{-1}(x - \mu_i) = -\frac{1}{2}x^T\Sigma^{-1}x + \frac{1}{2}x^T\Sigma^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma^{-1}x - \frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + \log \pi_k$

Using MLE estimates $\hat{\pi}_k = \Pr(Y = k) = \frac{n_k}{n}$, $\hat{\mu}_k = \mathbb{E}[X|Y = k] = \frac{1}{n_k}\sum_{i: y_i = k} x_i$, $\hat{\Sigma} = \text{Var}[X|Y = k] = \frac{1}{n}\sum_k\sum_i (x_i - \mu_k)(x_i - \mu_k)^T$

Predict class that maximizes $\delta_k(x) = \mu_k^T\Sigma^{-1}x - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log \pi_k$

Parameter Estimation Methods

We are interested in estimating Bernoulli and Gaussian RV for classification

Bernoulli Random Variable (biased coin p)

Method of Moments

Choose p so that distribution has same expectation as average of data

Observing n tosses, define +1's in $X_1 \ldots X_n$ and choose $p = \frac{1}{n}\sum_{i=1}^n X_i$

Maximum Likelihood

Count the number of +1 outcomes and write $\ell(p) = p^i(1 - p)^{n-i} \rightarrow$ log likelihood

$\log \Pr(k \text{ of } n) = \log\binom{n}{k} + k\log p + (n - k)\log(1 - p) \Rightarrow \ell'(p) = \frac{k}{p} - \frac{n - k}{1 - p} \rightarrow \hat{p} = k/n$

Penalized Maximum Likelihood

Incorporate prior information such as p is close to some value

$\log(\Pr(k \text{ of } n)) + \log(p(1 - p)) = \log\binom{n}{k} + k\log p + (n - k)\log(1 - p) + \log(p(1 - p))$

$\ell'(p) = k/p - \frac{n - k}{1 - p} + \frac{1 - 2p}{p(1 - p)} \Rightarrow p(n - k) = (1 - p)k + 1 - 2p$ $\hat{p} = \frac{k + 1}{n + 2}$

Bayesian Estimation

Model p as a random variable and belief about p captured by prob distribution

Can start by modeling as uniform if no a priori preference

Update belief using Bayes Theorem $P(p|X_1 = 1) \propto P(X_1 = 1|p)\pi(p)$, normalize by $\int P(X_1 = 1|q)d\pi(q)$

Bayesian approach gives us distribution and thus uncertainty

Assumes that parameters randomly chosen with fixed, known distribution so everything in prediction is just RV with known distribution

Bayesian estimation is just a computation of conditional probability distribution

Maximum a posteriori (MAP) estimate is just mode of the posterior (max likelihood if uniform prior, penalized max likelihood with any other prior)

Gaussian Random Variable

$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(x - \mu)^2}{2\sigma^2})$ and would like to estimate $\Theta = (\mu, \sigma^2)$ from data

Method of Moments

Choose $\Theta = (\mu, \sigma^2)$ to match first moment ($\mathbb{E}X$) and second moment ($\mathbb{E}X^2$) of data

Second moment given by $\mathbb{E}(X - \mu)^2 + \mu^2 = \sigma^2 + \mu^2$ so set $\sigma^2 = \frac{1}{n}\sum_{i=1}^n x_i^2 - \mu^2$

If we have p parameters, can solve p equations and match corresp number of moments

Maximum Likelihood

Choose $\Theta = (\mu, \sigma^2)$ so that distribution gives data high probability

$\ell(\mu, \sigma^2) = \log(\prod_{i=1}^n(\frac{1}{(2\pi\sigma^2)^{1/2}}\exp(-\frac{(x_i - \mu)^2}{2\sigma^2}))) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^n(x_i - \mu)^2$

$\nabla_\mu(\ell) \rightarrow \mu = \frac{1}{n}\sum_{i=1}^n x_i$ $\nabla_\sigma(\ell) \rightarrow \frac{n}{2\sigma^2} = \frac{1}{2\sigma^4}\sum_{i=1}^n(x_i - \mu)^2 \rightarrow \sigma^2 = \frac{1}{n}\sum_{i=1}^n(x_i - \mu)^2$

Yields the same result as method of moments

Penalized Maximum Likelihood

$\ell(\mu, \sigma^2) - \text{penalty}(\mu, \sigma^2) = \log(\prod_{i=1}^n(\frac{1}{(2\pi\sigma^2)^{1/2}}\exp(-\frac{(x_i - \mu)^2}{2\sigma^2}))) - \text{penalty}(\mu, \sigma^2)$

Bayesian Estimation

Prior: $\pi(\Theta) = 1$, Posterior: $P(\Theta|X_1 = x_1) \propto P(X_1 = x_1|\Theta)P(\Theta)$

Penalized Maximum Likelihood and Bayesian estimators effective in high-dimensional settings when number of parameters high compared to amount of data

Multivariate Normal Distribution

Recall the univariate Gaussian density function $P(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(x - \mu)^2}{2\sigma^2})$

$\mu \in \mathbb{R}$, $\sigma^2 \in \mathbb{R}_+$, normalization: $\frac{1}{\sqrt{2\pi\sigma^2}}$, arg of exp: $-\frac{(x - \mu)^2}{2\sigma^2} \leq 0$, concave $\rightarrow$ max at μ

$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp(-\frac{1}{2}(x - \mu)^T\Sigma^{-1}(x - \mu))$ $x \in \mathbb{R}^d$

$\mu \in \mathbb{R}^d$, $\sigma^2 \in \mathbb{R}^{d \times d}$, normalization: $\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}$, arg of exp: $-\frac{1}{2}(x - \mu)^T\Sigma^{-1}(x - \mu) \leq 0$ concave, max at μ

location $\mu \in \mathbb{R}^d$: $(x - \mu)$, scale $\Sigma \in S_+^d$: $(x - \mu)^T\Sigma^{-1}(x - \mu)$

Consider two independent Gaussian random variables X and Y:

$P(X, Y) = P(X)P(Y) = \frac{1}{\sqrt{(2\pi)^2\sigma_x^2\sigma_y^2}}\exp(-\frac{(x - \mu_x)^2}{2\sigma_x^2} - \frac{(y - \mu_y)^2}{2\sigma_y^2}) = \frac{1}{\sqrt{(2\pi)^2|\Sigma|}}\exp(-\frac{1}{2}([x - \mu_x \;\; y - \mu_y]\begin{bmatrix}\sigma_x^2 & 0 \\ 0 & \sigma_y^2\end{bmatrix}^{-1}\begin{bmatrix}x - \mu_x \\ y - \mu_y\end{bmatrix}))$

Covariance Matrices

- Symmetric: $\Sigma_{i,j} = \Sigma_{j,i}$ • Non-negative diagonal entries: $\Sigma_{i,i} \geq 0$ $\Sigma = \begin{bmatrix}\text{Var}(X_1) & \cdots & \text{Cov}(X_1, X_n) \\ \vdots & \ddots & \vdots \\ \text{Cov}(X_n, X_1) & \cdots & \text{Var}(X_n)\end{bmatrix}$
- Positive semidefinite: for all $v \in \mathbb{R}^d$, $v^T\Sigma v \geq 0$

Diagonal Covariance Matrices

Consider $X \sim N(\mu, \Sigma)$ where Σ is diagonal, so components of x uncorrelated

Look at super-level sets of pdf: $\mathcal{E}_r = \{x \in \mathbb{R}^d : (x - \mu)^T\Sigma^{-1}(x - \mu) \leq r^2\}$

$(x - \mu)^T\Sigma^{-1}(x - \mu) \leq r^2 \Longleftrightarrow \sum_{i=1}^d \frac{(x_i - \mu_i)^2}{\sigma_i^2} \leq r^2$

$\mathcal{E}_r$ thus corresponds to axis-aligned ellipsoids in $\mathbb{R}^d$

If all terms except $x_i = \mu_i$, then $x \in \mathcal{E}_r$ when $(x_i - \mu_i)^2 \leq r^2\sigma_i^2 \Longleftrightarrow |x_i - \mu_i| \leq r\sigma_i$

Volume of the ellipsoid $\mathcal{E}_r$ is proportional to $\prod_{i=1}^d\sigma_i = \sqrt{|\Sigma|}$

explains $1/\sqrt{|\Sigma|}$ normalization and shows for fixed r, $\Pr(\mathcal{E}_r)$ is same $\forall \mu, \Sigma$

Non-Diagonal Covariance and Diagonalization

Start with a diagonal Σ and consider transformed vector $Y = \mu_x + Q(X - \mu_x)$ where Q is matrix corresponding to rotation about the origin $\begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}$

Mean unchanged by $\mathbb{E}Y = \mu_x + \mathbb{E}[Q(X - \mu_x)] = \mu_x + Q[\mathbb{E}(X) - \mu_x] = \mu_x$

Covariance is transformed $\Sigma_Y = \mathbb{E}(Y - \mu_x)(Y - \mu_x)^T = \mathbb{E}[Q(X - \mu_x)(X - \mu_x)^TQ] = Q\mathbb{E}[(X - \mu_x)(X - \mu_x)^T]Q^T = Q\Sigma_xQ^T$

Eigenvalues and Eigenvectors

For a square matrix A, λ is an eigenvalue and x an eigenvector if $Ax = \lambda x$ where eigenvalues are the roots of the characteristic polynomial $\det(A - \lambda I)$

Spectral Theorem

For a symmetric real matrix $A \in \mathbb{R}^{n \times n}$, we can find n orthonormal eigenvectors of A ($v_1, \ldots, v_n$) and eigenvalues ($\lambda_1, \ldots, \lambda_n$) are real

Write $AU = U\Lambda$, $A = U\Lambda U^T$ where $U = [v_1, v_2, \ldots, v_n]$, $\Lambda = \text{diag}(\begin{bmatrix}\lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n\end{bmatrix})$

Covariance matrix Σ is symmetric so $\Sigma = U\Lambda U^T$ and positive semidefinite so $\forall \lambda_i \geq 0$: $v_i^T\Sigma v_i = v_i^TU\Lambda U^Tv_i = \lambda_i \geq 0$

Precision matrix can be written as $\Sigma^{-1} = U\Lambda^{-1}U^T$

$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp(-\frac{1}{2}(x - \mu)^T\Sigma^{-1}(x - \mu)) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp(-\frac{1}{2}(x - \mu)^TU\Lambda^{-1}U(x - \mu))$

$U^T(x - \mu)$ computes the representation of $x - \mu$ in eigenvector basis of Σ where covariance is the diagonal matrix Λ^{-1}

Level sets $\mathcal{E}_r = \{x \in \mathbb{R}^d : (x - \mu)^T\Sigma^{-1}(x - \mu) \leq r^2\}$ correspond to eigenvector-aligned ellipsoids in $\mathbb{R}^d$; $(x - \mu)^T\Sigma^{-1}(x - \mu) = (x - \mu)^TU\Lambda^{-1}U^T(x - \mu) = \sum_{i=1}^d\frac{(v_i^T(x - \mu))^2}{\lambda_i^2}$

so if $v_2^T(x - \mu) = \ldots = v_d^T(x - \mu) = 0$, then $x \in \mathcal{E}_r$ when $(v_1(x - \mu))^2 \leq r^2\lambda_1^2 \Longleftrightarrow |v_1(x - \mu)| \leq r\lambda_1$

Volume of the ellipsoid $\mathcal{E}_r$ is proportional to $\prod_{i=1}^d\lambda_i = \sqrt{|\Lambda|} = \sqrt{|U|\cdot|\Lambda|\cdot|U^T|} = \sqrt{|U\Lambda U^T|} = \sqrt{|\Sigma|}$

Properties of Multivariate Gaussians

Probability: $\int_{\mathbb{R}^d}p(x; \mu, \Sigma)dx = 1$ Expectation: $\mathbb{E}[X] = \begin{bmatrix}\mathbb{E}[X_1] \\ \vdots \\ \mathbb{E}[X_d]\end{bmatrix} = \int_{\mathbb{R}^d}x\,p(x; \mu, \Sigma)dx = \mu$

Covariance: $\text{Cov}[X] = \mathbb{E}[(X - \mu)(X - \mu)^T] = \int_{\mathbb{R}^d}(x - \mu)(x - \mu)^Tp(x; \mu, \Sigma)dx = \Sigma$

Sums of independent Gaussians $X \sim N(\mu_X, \Sigma_X)$ and $Y \sim N(\mu_Y, \Sigma_Y) \rightarrow X + Y \sim N(\mu_X + \mu_Y, \Sigma_X + \Sigma_Y)$

We know that $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$, $\text{Cov}(X + Y) = \mathbb{E}[(X + Y)(X + Y)^T] = \mathbb{E}[XX^T + XY^T + YX^T + YY^T]$

$= \mathbb{E}[XX^T] + \mathbb{E}[XY^T] + \mathbb{E}[YX^T] + \mathbb{E}[YY^T] = \mathbb{E}[XX^T] + 0 + 0 + \mathbb{E}[YY^T] = \Sigma_X + \Sigma_Y$

Marginals

Write d-dimensional Gaussian $X \sim N(\mu, \Sigma)$ as $X = \begin{bmatrix}Y \\ Z\end{bmatrix}$, $\mu = \begin{bmatrix}\mu_Y \\ \mu_Z\end{bmatrix}$, $\Sigma = \begin{bmatrix}\Sigma_{YY} & \Sigma_{YZ} \\ \Sigma_{ZY} & \Sigma_{ZZ}\end{bmatrix}$ where $Y \in \mathbb{R}^m$ and $Z \in \mathbb{R}^{d-m}$, then $Y \sim N(\mu_Y, \Sigma_{YY})$

Mean and covariance of Y evident from definitions, show Y is Gaussian by writing density as an integral (over Z), complete squares $\rightarrow N(\mu_Y, \Sigma_{YY})$

Affine Transformations

Given a d-dimensional Gaussian $X \sim N(\mu, \Sigma^2)$, matrix $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$

define $Y = AX + b \Rightarrow Y \sim N(A\mu + b, A\Sigma A^T)$ ~ suppose $m = d$, A full-rank, Y just scaled shifted version of X

$\mathbb{E}Y = A\mu + b$ $\text{Cov}(Y) = \mathbb{E}[(Y - \mathbb{E}Y)(Y - \mathbb{E}Y)^T] = \mathbb{E}[(AX + b - A\mu - b)(AX + b - A\mu - b)^T] = \mathbb{E}[(A(X - \mu))(A(X - \mu))^T]$

Ex: Given a d-dimensional standard normal $X \sim N(0, I)$, we have $Y = AX + \mu$ $= A\mathbb{E}[(X - \mu)(X - \mu)^T]A^T = A\Sigma A^T$ $\sim N(\mu, AA^T)$

Given a d-dimensional normal $X \sim N(\mu, \Sigma)$ with Σ positive definite, $Y = \Sigma^{-1/2}(X - \mu) \sim N(0, I) \rightarrow$ whitening X

Regression

Linear Regression

Recall the quadratic loss function $\ell(\hat{y}, y) = (\hat{y} - y)^2$, risk $R(f) = \mathbb{E}\,\ell(f(X), Y) = \mathbb{E}(f(X) - Y)^2$

For each X, minimize $\mathbb{E}[(f(X) - Y)^2|X]$ using $f^*(X) = \mathbb{E}[Y|X]$

Bias-Variance $R(f) = \mathbb{E}[(f(X) - \mathbb{E}[Y|X])^2] + \mathbb{E}[(\mathbb{E}[Y|X] - Y)^2] = \mathbb{E}[(f(X) - f^*(X))^2] + R^*$

Given $X \in \mathbb{R}^p$, $Y \in \mathbb{R}$, linear (affine) prediction rules $F_{lin} := \{x \mapsto x^T\beta + \beta_0 : \beta \in \mathbb{R}^p, \beta_0 \in \mathbb{R}\}$

① Consider class of linear prediction rules and minimize empirical risk

② Model the process of generating the Y_i's as a linear function of the X_i's plus additive Gaussian noise. Compute max likelihood estimate for coefficients

Both approaches arrive at the normal equations: choice of β corresponds to a projection onto a linear sub-space

Empirical Risk Minimization

Risk $R(f) = \mathbb{E}\,\ell(f(X), Y) = \mathbb{E}(f(X) - Y)^2$ is expected squared error while

Empirical risk $\hat{R}(f) = \hat{\mathbb{E}}_n\ell(f(X), Y) = \frac{1}{n}\sum_{i=1}^n(f(X_i) - Y_i)^2$ is sample average of squared errors

Choose linear prediction rule $f \in F_{lin}$ that minimizes empirical risk by

$\hat{f} := \arg\min_{f \in F_{lin}}\hat{\mathbb{E}}_n\ell(f(X), Y) = \arg\min_{f \in F_{lin}}\sum_{i=1}^n(f(X_i) - Y_i)^2$

use convention of offset term β_0 and constant 1 component in X

Solve $\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p}\sum_{i=1}^n(X_i^T\beta - Y_i)^2 = \arg\min_{\beta \in \mathbb{R}^p}\|X\beta - Y\|^2$, $X = \begin{pmatrix}X_1^T \\ \vdots \\ X_n^T\end{pmatrix}$, $Y = \begin{pmatrix}Y_1 \\ \vdots \\ Y_n\end{pmatrix}$ $X \in \mathbb{R}^{n \times p}$, $Y \in \mathbb{R}^n$, $n > p$

$RSS(\beta) = \frac{1}{2}\|X\beta - Y\| = \frac{1}{2}(X\beta - Y)^T(X\beta - Y) = \frac{1}{2}\beta^TX^TX\beta - Y^TX\beta + \frac{1}{2}Y^TY$

$\nabla_\beta RSS(\beta) = X^TX\beta - X^TY$ $\nabla_\beta^2 RSS(\beta) = X^TX \succeq 0$, so function is convex

$X^TX\beta - X^TY = 0 \Rightarrow \hat{\beta} = (X^TX)^{-1}X^TY$ (normal equations)

View as finding linear combination of columns x_j of $X = (x_1\; x_2 \ldots x_p)$ that minimizes squared Euclidean distance to $Y \in \mathbb{R}^n$ by $Y - X\beta = Y - \sum_{j=1}^p\beta_jx_j$

Optimal approximation $\hat{Y}$ in space spanned by columns of X has error $Y - \hat{Y}$ orthogonal to column space

$(Y - \hat{Y})^TX = 0 \Longleftrightarrow X^T(Y - X\beta) = 0 \Longleftrightarrow X^TY = X^TX\beta$

Risk of empirical risk minimizer $\hat{f}$ close to minimum risk if

① X's come from a compact set ② Y_i's have tails not too heavy (e.g. sub-Gaussian) ③ $\|\Theta\|$ is not too large ④ $n \gg p$

Linear Model with Additive Gaussian Noise

Model conditional distribution of Y given $X = x$ as $P(Y|X = x) = N(x^T\beta, \sigma^2)$ or $Y = x^T\beta + \epsilon$, $\epsilon \sim N(0, \sigma^2)$

Using MLE, $L(\beta) = \prod_{i=1}^n p(Y_i|X_i, \beta) = \frac{1}{(2\pi\sigma^2)^{n/2}}\exp(-\frac{1}{2\sigma^2}\sum_{i=1}^n(Y_i - X_i^T\beta)) \rightarrow \ell(\beta) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^n(Y_i - X_i^T\beta)^2$

and we get the least squares solution

Show $\hat{\beta}$ is unbiased ($\mathbb{E}Y = X\beta$, $\text{Cov}(Y) = \sigma^2 I$) by looking at expectation

$\mathbb{E}\hat{\beta} = \mathbb{E}[(X^TX)^{-1}X^TY] = (X^TX)^{-1}X^T\mathbb{E}Y = (X^TX)^{-1}X^TX\beta = \beta$

$\text{Cov}(\hat{\beta}) = \mathbb{E}[(\hat{\beta} - \beta)(\hat{\beta} - \beta)^T] = \mathbb{E}[((X^TX)^{-1}X^TY - \beta)((X^TX)^{-1}X^TY - \beta)^T] \ldots = \sigma^2(X^TX)^{-1}$

If data generated by a linear model with $N(0, \sigma^2)$ noise, then:

we can compute distributions of parameter estimates $\hat{\beta} \sim N(\beta, (X^TX)^{-1}\sigma^2)$

we can calculate approximate confidence sets for parameters, with standardized coefficient $z_j = \frac{\hat{\beta}_j}{\sigma\sqrt{v_j}}$ (which is normal), and design tests for non-zero values of parameters

Regularization and Regression

Bias-variance trade-off $\mathbb{E}R(\hat{f}) - R^* = \mathbb{E}[(\hat{f}(X) - f^*(X))^2] = \mathbb{E}[(\hat{f}(X) - \mathbb{E}\hat{f}(X))^2] + \mathbb{E}[(\mathbb{E}\hat{f}(X) - f^*(X))^2]$ (variance, bias)

Since we use random training data, prediction rule is random, risk is RV.

Increase in bias decreases variance and overall complexity

Linear model complexity depends on number of non-zero coeff of $\hat{\beta}$ and size of $\hat{\beta}$

Subset Selection

Idea: Consider 2^p subsets of variables, fit linear model, decide which models to use

Decide on model complexity using cross validation

We can use greedy computational shortcuts that get good approximate solutions

Forward-stepwise Selection: Iterate through all fittings starting with 1 param, include best in next iteration only if better

Backward-stepwise selection: Start with all and keep removing least contributing

Shrinkage Methods

Encourage linear predictor coefficients to be small using penalty

Ridge Regression

Punish using penalty term $\hat{\beta} = \arg\min_\beta(\sum_{i=1}^n(Y_i - X_i^T\beta)^2 + \lambda\sum_{j=1}^p\beta_j^2)$ or

forbid using constraint $\hat{\beta} = \arg\min_\beta\sum_{i=1}^n(Y_i - X_i^T\beta)^2$ s.t. $\sum_{j=1}^p\beta_j^2 \leq B$

Do not penalize β_0, $RSS = \|Y - X\beta\|^2 + \lambda\|\beta\|^2 \rightarrow \hat{\beta}^{ridge} = (X^TX + \lambda I)^{-1}X^TY$

Even if X^TX is singular, this ensures a unique solution $\hat{\beta}$

Solution depends on scaling of covariates (common to standardize to variance = 1)

Lasso

Penalize using L1-norm $\hat{\beta} = \arg\min_\beta(\sum_{i=1}^n(Y_i - X_i^T\beta)^2 + \lambda\sum_{j=1}^p|\beta_j|)$ or

forbid using constraint $\hat{\beta} = \arg\min_\beta\sum_{i=1}^n(Y_i - X_i^T\beta)^2$ s.t. $\sum_{j=1}^p|\beta_j| \leq B$

Ridge regression leads to reduced coefficients, lasso pushes to zero

Bayesian View of Linear Regression

Given linear model $P(Y|X = x) = N(x^T\beta, \sigma^2)$

Model β as a random variable ($\beta = N(0, \tau^2 I)$) then compute posterior distribution

$P(\beta|X, Y) \propto P(Y|\beta, X)P(\beta) \propto \exp(-\frac{(Y - X^T\beta)^2}{2\sigma^2} - \frac{\|\beta\|^2}{2\tau^2}) = \exp(-\frac{1}{2}(\sum_{i=1}^n\frac{(Y_i - X_i^T\beta)^2}{\sigma^2} + \frac{1}{\tau^2}\|\beta\|^2))$

MAP estimates β as the mode of distribution

Ridge regression $\Longleftrightarrow$ MAP w/ Gaussian prior, Lasso $\Longleftrightarrow$ MAP Laplace prior

Logistic Regression

$P(Y = 1|X) = \frac{1}{1 + \exp(-\beta^Tx)}$ Given data $(X_1, Y_1) \ldots (X_n, Y_n) \in \mathbb{R}^p \times \{0, 1\}$ estimate β using MLE

Log likelihood: $\ell(\beta) = \log P(Y_1, \ldots Y_n|X_1, \ldots X_n; \beta) = \sum_{i=1}^n Y_i\log\mu_i(\beta) + (1 - Y_i)\log(1 - \mu_i(\beta))$

where $\mu_i(\beta) = P(Y = 1|X = x_i, \beta) = \frac{1}{1 + \exp(-\beta^Tx_i)}$, $\nabla_\beta\mu_i(\beta) = \mu_i(\beta)(1 - \mu_i(\beta))x_i$

first derivative: $\nabla_\beta\ell(\beta) = \sum_{i=1}^n((\frac{Y_i}{\mu_i(\beta)} - \frac{1 - Y_i}{1 - \mu_i(\beta)})\nabla_\beta\mu_i(\beta)) = \sum_{i=1}^n(Y_i - \mu_i(\beta))x_i$

second derivative: $\nabla_\beta^2\ell(\beta) = \sum_{i=1}^n -\mu_i(\beta)(1 - \mu_i(\beta))x_ix_i^T$

MLE yields $\sum_{i=1}^n Y_ix_i = \sum_{i=1}^n\mu_i(\beta)x_i$ System of p coupled linear equations with no closed form solution, concave

Gradient Ascent

$\beta^{(t+1)} = \beta^{(t)} + \eta\nabla_\beta\ell(\beta^{(t)}) = \beta^{(t)} + \eta\sum_{i=1}^n(Y_i - \mu_i(\beta^{(t)}))x_i$

Each gradient calculation takes $O(np)$ time

Stochastic Gradient

Instead compute approximation of $\nabla_\beta\ell(\beta^{(t)}) = (Y_{it} - \mu_{it}(\beta^{(t)}))x_{it}$ ($O(p)$ time) and update via $\beta^{(t+1)} = \beta^{(t)} + \eta(Y_{it} - \mu_{it}(\beta^{(t)}))x_{it}$

Minimizing RSS in linear regression can also be accomplished using gradient/stochastic with convex regularization terms $\|\beta\|_2^2$, $\|\beta\|_1$

Newton's Method

Use 1st order Taylor approximation $f(x) = f(x_0) + f'(x_0)(x - x_0)$ and solve for $f(x) = 0$: $x_1 = x_0 - f(x_0)/f'(x_0)$

Solve $\nabla f(x) = 0$ using $x_{t+1} = x_t - [\nabla^2 f(x_t)]^{-1}\nabla f(x_t)$

$\ell(\beta) = \sum_{i=1}^n Y_i\log\mu_i(\beta) + (1 - Y_i)\log(1 - \mu_i(\beta))$

$\nabla_\beta\ell(\beta) = \sum_{i=1}^n(Y_i - \mu_i(\beta))x_i = X^T(Y - \mu)$

$\nabla_\beta^2\ell(\beta) = \sum_{i=1}^n -\mu_i(\beta)(1 - \mu_i(\beta))x_ix_i^T = -X^T\text{diag}(\mu(1 - \mu))X$

$\beta^{(t+1)} = \beta^{(t)} + [X^T\text{diag}(\mu(1 - \mu))X]^{-1}X^T(Y - \mu)$

Each iteration is least squares with $x_ix_i^T$ entries weighted by $\mu_i(1 - \mu_i)$ (iteratively reweighted least squares), distance from 0/1

MLE

View MLE as minimizing log loss for $\hat{p}(Y|X) = \begin{cases} f(x) & \text{if } Y = 1 \\ 1 - f(x) & \text{if } Y = -1 \end{cases}$

Define loss $\ell_{log}(\hat{p}(\cdot|X), Y) = -\log(\hat{p}(Y|X)) = \begin{cases} -\log f(x) & \text{if } Y = 1 \\ -\log(1 - f(x)) & \text{if } Y = -1 \end{cases}$

Write logistic log loss $= \begin{cases} \log(1 + \exp(-\beta^Tx)) & \text{if } Y = 1 \\ \log(1 + \exp(\beta^Tx)) & \text{if } Y = -1 \end{cases} = \log(1 + \exp(-Y\beta^Tx))$

MLE minimizes sample average $\frac{1}{n}\sum_{i=1}^n\log(1 + \exp(-Y_i\beta^Tx_i))$ (binomial deviance)

Nearest Neighbor Classification

Key idea: Store all training examples $\langle x_i, f(x_i) \rangle$

k-NN - given query instance x_q, take a vote among its k nearest neighbors (if discrete target), take mean of k-NN f values (real valued)

Non-parametric, number of parameters grows with N

Training is fast, learn complex target functions easily, don't lose information but slow at query time, requires lots of storage, easily fooled by irrelevant attr

Behavior at the limit: $k=1$, $k=n$ = error of opt/NN $\lim_{n\to\infty} k=\infty$, $\lim_{n\to\infty} k/n=0$ $\Rightarrow \varepsilon^* \le \varepsilon_{NN} \le 2\varepsilon^*$

Curse of Dimensionality

Easy to mislead NN in high dimensions (points on hyper-grid, hypercube ...)

① Get more data (if $D \gg N$, make $D \ll N$)

② Reduce D to get better features (compress pixels, use bag-of-words model), compare histograms using Intersection kernel $K(x,z) = \sum_{i=1}^{n} \min(x_i, z_i)$

③ Use a better distance metric

Distance Metrics

Minkowski distance $Disp(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{1/p} = \|x-y\|_p$

1-norm (Manhattan/cityblock distance) $Dis_1(x,y) = \sum_{i=1}^{d} |x_i - y_i|$

2-norm (Euclidean distance) $Dis_2(x,y) = \sum_{i=1}^{d} (x_i - y_i)^2)^{1/2}$

0-norm (number of nonzero elements) $Dis_0(x,y) = \sum_{i=1}^{d} 1[x_i \neq y_i]$

Hamming distance for binary strings, or edit/Levenshtein distance

Mahalanobis distance $Dis_\Sigma(x,y|\Sigma) = \sqrt{(x-y)^T \Sigma^{-1} (x-y)}$

Decorrelate and renormalize features, Euclidean special case with $\Sigma = I$

Similarities

Inner $(x,y) = \langle x, y \rangle$, symmetric, output $\in \mathbb{R}$

$\cos Sim(x,y) = \frac{\langle x,y \rangle}{\|x\|\|y\|}$, symmetric, output $\in [-1,1]$, normalized inner product

$Corr(x,y) = \frac{\langle x-\bar{x}, y-\bar{y} \rangle}{\|x-\bar{x}\|\|y-\bar{y}\|}$, symmetric, output $\in [-1,1]$, invariant to shift in input

Criteria

Given an instance space X, a distance metric is function $Dis: X \times X \to \mathbb{R}$ such that for any $x, y, z \in X$:

$Dis(x,x) = 0$ (point and itself 0) • $D(x,y) \ge 0$ $(x \neq y)$

$Dis(x,y) = Dis(y,x)$ (symmetric) • $Dis(x,z) \le Dis(x,y) + Dis(y,z)$

Problems

Easy to misled L2 NN in template matching, normalized cross-correlation

Euclidean distance for digit recognition sensitive to thickness, rotation

Strategies

① Augment the dataset with transform copies of the digit

② Build invariance into the feature vector (orientation histograms)

③ Build invariance into classification strategy (tangent dist, conv NN)

Overfitting → set k by cross-validation, form prototypes, remove noisy data

Reducing Computational Cost

k-d tree

Binary tree data structure for organizing set of points in k-dim space

Internal node associated with axis-aligned hyper-plane, splitting subtrees

Dimensions with high variance chosen first

Position of splitting hyperplane taken as median of projected points

Query for NN may require backtracking

Indexing local features: approximate NN search

Best-Bin First (BBF), variant of k-d trees, uses priority queue to examine most promising branches first

Locality-sensitive Hashing (LSH): randomized hashing technique to map similar points to same bin with high probability

We want a high degree of collisions and hence each hash function must satisfy $\Pr[h(x_i) = h(x_j)] = sim(x_i, x_j) = 1 - \frac{\theta(x_i, x_j)}{\pi}$

Decision Trees

Utilize labels y_i (unlike k-d trees) in performing classification

Internal nodes test value of features x_j and branch accordingly

Leaf nodes specify a class $h(x)$

Allow for great feature selection, interpretable results, decision regions looks similar to kNN/linear classifiers

Learning/Tree construction

Choosing Best Attribute: 1-step greedy lookahead, does not account for "progress"

Split on entropy $H(v) = \sum_{v \in V} -P(y=v) \log P(y=v)$

Information gain = entropy(parent) - average entropy(children)

Mutual information is a correct measure (?) vs. absolute error

Non Binary features: Multiple discrete values - multiway split (normalize info gain)

one vs. all split (binary), group into two disjoint subsets

Real-valued features - consider threshold split at each storage

Unknown Attributes

If node n tests A, assign most common value of A among other examples

Assign most common value of A among other examples w/ same target val

Assign prob p_v to each value v_v of A, assign frac p_v to each descendant

Classification vs. Regression Trees

$X \to \{0,1,2\}$ vs. $X \to Y \in \mathbb{R}$, minimize sum of variances

$\sum_{X \in leaf(l,s)} (Y_i - \hat{Y}_{R_1})^2 + \sum_{X \in leaf_2(l,s)} (Y_i - \hat{Y}_{R_2})^2$

Overfitting

Don't wait until absolute purity since never happen - noise, data scarcity

Leads to diminishing returns/depth, deep trees slow and huge)

Stop splitting earlier and make leaf node with (average/majority class)

Early termination based on max tree depth, min # points at node, tree complexity penalty, validation error monitoring

Reduced-Error Pruning

Split data into training and validation set, do until harmful:

① Evaluate impact of pruning each possible node on validation

② Greedily remove node that most improves validation accuracy

Random Forests

Ensemble Methods combine several "weak learner" models by Averaging (randomize each model), Bagging (random dataset given to model), Boosting (specialize each model for subset of examples), Rand Trees (A_{Tree}, B_{Tree})

Bagging still leads to correlated trees, limit to subset of p splits

Use each result as a vote towards overall classification

Boosting

Define classifier using additive model $F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \dots$

Example: Each data point starts with weight 1. Draw weak learner, then reweight misclassified points by $w_t = w_t \exp(-y_t h_t)$

AdaBoost Algorithm

Given m examples $(x_1, y_1) \dots (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$, Initialize $D_1(i) = 1/m$

for $t = 1$ to T: ① Train learner h_t with min error $\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$

② Compute the hypothesis weight $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$

③ For each example (x_i, y_i) update $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$

Output: $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Neural Networks

Input nodes, connected to hidden layers, then output via sigmoid

Each node outputs a linear combination of its inputs, sig activation

Two layer network equivalent to logistic regression, use more complicated learning scheme to extract better features

use one-vs-all classification for multi-class

Learning

Randomly perturbing weights similar to reinforcement learning, but can be very inefficient, changing just one is not effective

Backpropagation uses error derivatives with respect to the hidden activities

Numerical and analytic gradient slow to calculate, Backprop similar to cached analytic gradient

Using squared error, cannot enforce sum to one constraint

use Softmax $y_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$, $\frac{\partial y_i}{\partial z_j} = y_i(1-y_i)$, $C = -\sum_j t_j \log y_j$

Combine ideas of logistic regression, template matching/NN, decision forests/ensembles of weak learners, stochastic gradient descent, and feature learning

Apply regularization using L2, L1, both, max norm constraint

Convolutional Neural Networks

Alternating convolutional and pooling layers

Convolution: Apply element wise convolution using filters to produce many layered-output

Note: locally connected patches (not fully connected), and share weights across each depth slice

Pooling: Filter responses at each location to extract features

Max (take max response), Average, L2 $\sqrt{\sum x^2}$, L2 over features

Unsupervised Learning

Without labels, discover something interesting about x (data), assume/infer some kind of underlying structure (existing data may also be wrong)

Clustering (partitioning into tight clusters), Dimensionality Reduction (discover low-D manifolds, good features), Mode Seeking, (discover frequent patterns)

Clustering

Applications in: image segmentation, super pixels, market segmentation...

Agglomerative: Starting with n points, keep merging closest two clusters

Similarity metrics: $dist\{s_i, s_j\}$

Single link: $dist\{s_i, s_j\} = \min_{ij} d_{ij}$ (elongated, sensitive to outliers)

complete link: $dist\{s_i, s_j\} = \max_{ij} d_{ij}$ (compact, sensitive to outliers)

average link: $dist\{s_i, s_j\} = \frac{\sum_i \sum_j d_{ij}}{n_i n_j}$ (in between, group decision)

centroid link: $dist\{s_i, s_j\} = \|\bar{x}_i - \bar{x}_j\|$

Divisive: Starting with single cluster, recursively divide until each cluster only has one object or members overlap

K-means

Specify there are k means and minimize distance of points to means

Number of distinct assigns exponential in k and N

Use greedy algorithm alternating between mean/cluster assignment

① Randomly initialize cluster centroids $\mu_1, \dots \mu_k$

② Repeat until convergence:

- For every point i, assign to closest cluster k: $c_i = \arg\min_k \|x_i - \mu_k\|^2$
- For every cluster k, recompute mean $\mu_k = \frac{\sum_{i=1}^{n} w_{ik} x_i}{\sum_{i=1}^{n} w_{ik}}$

Repeated coordinate descent on $J(c, \mu) = \sum_{i=1}^{N} \|x_i - \mu_{c_i}\|^2$

Avoid bad local minima by ① initialization: choose one point among data randomly, then others by weighted probability distribution of $D(x)^2$

If only given distance matrix, use k-medoids algorithm

For each point x (row), find closest center m and move towards x a little; repeat

Works well if clusters spherical, well separated, of similar volumes and have similar number of points

Vector Quantization involves finding closest clusters (often by k-means) and encoding vectors to closest neighbor

Mixture Models

$P(x) = \sum_{c=1}^{k} P(c_c) P(x|c_c)$ Objective function, log likelihood of data

Clustering might use $x|cluster_c \sim N(\mu_c, \sigma_c^2)$

EM Algorithm

Initialize parameters w/o missing info, repeat:

E: compute expected values of unobserved variables assuming (current) M: compute new parameters to maximize prob of data

For Gaussians,

Initialization: Choose means at random

E: For all examples x_n:

$P(\mu_i|x_n) = \frac{P(\mu_i) P(x_n|\mu_i)}{P(x_n)} = \frac{P(\mu_i) P(x_n|\mu_i)}{\sum_j P(\mu_j) P(x_n|\mu_j)}$

M: For all components $P(c_i)$:

$P(c_i) = \frac{1}{n_e} \sum_{n=1}^{n_e} P(\mu_i|x_n)$, $\mu_i = \frac{\sum_{n=1}^{n_e} x_n P(\mu_i|x_n)}{\sum_{n=1}^{n_e} P(\mu_i|x_n)}$, $\sigma_i^2 = \frac{\sum_{n=1}^{n_e} (x_n - \mu_i)^2 P(\mu_i|x_n)}{\sum_{n=1}^{n_e} P(\mu_i|x_n)}$ (current)

Non-Parametric Density Estimation

Histogram method: use fixed box widths

Kernel Density Estimation

$P(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i)$ function of finite number of data points

• Normalized: $\int_R^d K(x)dx = 1$ • Symmetric: $\int_R^d K(x)x\,dx = 0$ • Exponential decay: $\lim_{\|x\| \to \infty} K(x) = 0$

Epanechnikov kernel: $K_E(x) = \begin{cases} c(1 - \|x\|^2) & \|x\| \le 1 \\ 0 & \text{otherwise} \end{cases}$

Uniform kernel: $K_U(x) = \begin{cases} c & \|x\| \le 1 \\ 0 & \text{otherwise} \end{cases}$

Normal kernel: $K_N(x) = c \exp\left(-\frac{1}{2}\|x\|^2\right)$

Mode-seeking

"Association Rules", "Frequent Itemsets", "Basket Analysis"

Association Rules

Rule $X \to Y$: "Body → Head [support, confidence]"

Support: $supp(X)$ = frequency i.e. $P(X)$

Confidence: $conf(X \to Y) = \frac{supp(X \cup Y)}{supp(X)}$ i.e. $P(Y|X)$

Lift $(X \to Y) = supp(X \cup Y) / supp(X) * supp(Y)$

Principle Component Analysis

Given a set of feature vectors x_i, we can represent as pattern matrix

Ex: Measurement vectors, original images, text documents, user rating data

Problems → too large/complicated, missing/noisy entries, lack of structure

Extract latent structure of matrix using factorization: $A_{2 \times 2} = L_{2 \times R} \cdot R$

Can perform dimensionality reduction, compressing and keeping only directions of highest variance

PCA: ① Subtract mean from every point

② (sometimes) scale each dimension by its variance

③ Compute covariance matrix $S = X^T X$

④ Compute k largest eigenvectors of $S = VDV^T$

Singular Value Decomposition

① Subtract mean from each data point

② Decompose $X = USV^T$, $X^T X = VS^2V^T$, $XX^T = US^2U^T = UDU^T$

③ $u * s$ matrix provides coefficients: $x_1 = u_{1,1} s_{11} v_1 + u_{1,2} s_{22} v_2 + \dots$