MLE estimate

Consistent, asymptotically normal ($\sqrt{n}(\theta_{ml} - \theta_0) \sim N(0, J^{-1}IJ^{-1})$ as n->∞)

Efficient: ((θ_{Ml} minimize $E((\theta_{Ml} - \theta_0)^2)$ namely variance as n->∞)

Equivariance: if $\hat{\theta}_n$ is MLE of θ, $g(\hat{\theta}_n)$ is MLE of $g(\theta)$

Cramer Rao lower bound

For unbiased estimator

$$E\left[\left(\hat{\theta} - \theta_0\right)^2\right] \geq \frac{1}{I_n(\theta_0)}$$

Where $I_n(\theta_0)$ is the Fisher information $Var[(\frac{\partial}{\partial\theta} log P(\boldsymbol{X}_{1,\dots,n}|\theta))]$

LS estimate

MLE when noise follows iid Gaussian with constant σ

$\mathbf{RSS(\beta) = (y - X\beta)^T(y - X\beta)}$

$\hat{\boldsymbol{\beta}} = (X^TX)^{-1})X^Ty$

$\hat{\beta} \sim N(\beta, (X^TX)^{-1}\sigma^2))$

Optimality: smallest variance among all linear unbiased estimates

Expected prediction error:

$$E_D E_{(Y|X)}(\hat{f}(x) - Y)^2$$
$$= E_D\left(\hat{f}(x) - E(Y|X)\right)^2 + E_{(Y|X)}\left(Y - E(Y|X)\right)^2$$
$$= E_D\left(\hat{f}(x) - E_D\hat{f}(x)\right)^2 + E_D\left(E_D\hat{f}(x) - E(Y|X)\right)^2 + E_{(Y|X)}\left(Y - E(Y|X)\right)^2$$

Expected risk

Total: $R(f) = E_{X,Y}[Q(Y, f(X))] = \int_y \int_x Q(Y, f(X))P(X, Y)dXdY$

Conditional: $R(f, X) = \int_y Q(Y, f(X))P(Y|X)dY$

Connect to test error: The test error can be used to approximate the expected risk. The expected value of the test error is the expected

Ridge and Lasso

Reg: $\lambda\beta^T\beta$; $\lambda||\beta||_1$

$\hat{\boldsymbol{\beta}}^{ridge} = (X^TX + \lambda I)^{-1}X^Ty$

both assume

$\mathbf{Y}|(\mathbf{X}, \beta) \sim N(X^T\beta, \sigma^2\mathbf{I})$

Difference:

1. **Prior** on β

$\beta^{ridge} \sim N(0, \frac{\sigma^2}{\lambda\mathbf{I}})$

$P(\beta^{Lasso}) = \frac{\lambda}{4\sigma^2}\exp\left(-\frac{|\beta|\lambda}{2\sigma^2}\right)$

2. **Cost function**

Ridge is strictly convex while Lasso is just convex. Not differentiable when |β|=0

Spline regression

$$RSS(f, \lambda) = \sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\int(f''(x))^2dx$$

Gaussian process

Assumption:

1. $P(\mathbf{Y}|\mathbf{X}, \beta, \sigma) \propto \exp\left(-\frac{1}{2\sigma^2}(Y - X^T\beta)^2\right)$
2. $P(\beta|\Lambda) \propto \exp\left(-\frac{1}{2}\beta^T\Lambda\beta\right)$

Posterior distribution of β

$N(\beta|\mu_\beta, \Sigma_\beta)$ where

$\mu_\beta = \left(X^TX + \sigma^2\Lambda\right)^{-1}X^Ty$

$\Sigma_\beta = \sigma^2\left(X^TX + \sigma^2\Lambda\right)^{-1}$

$Cov[y] = X\Lambda^{-1}X^T + \sigma^2\mathbf{I}$

distribution of y

$$p\left(\begin{bmatrix}\mathbf{y} \\ y_{n+1}\end{bmatrix}\middle| x_{n+1}, \mathbf{X}, \sigma\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{y} \\ y_{n+1}\end{bmatrix}\middle| \mathbf{0}, \begin{bmatrix}\mathbf{C}_n & \mathbf{k} \\ \mathbf{k}^T & c\end{bmatrix}\right),$$

where $\mathbf{C}_n = \mathbf{K} + \sigma^2\mathbf{I}$,
$c = k(x_{n+1}, x_{n+1}) + \sigma^2$,
$\mathbf{k} = k(x_{n+1}, \mathbf{X})$,
and $\mathbf{K} = k(\mathbf{X}, \mathbf{X})$.

$\mu_{y_{n+1}} = k^TC_n^{-1}y$

$\sigma^2_{y_{n+1}} = c - k^TC_n^{-1}k$

Model averaging

$\mathbf{Bias[\hat{f}(x)]}$= $\frac{1}{B}\sum_{i=1}^{B} E_D\left(\hat{f}_i(x)\right) - E(Y|X) = \frac{1}{B}\sum_{i=1}^{B} Bias[\hat{f}_i(x)]$

$\mathbf{V[\hat{f}(x)]} = \frac{1}{B^2}\sum_{i=1}^{B} V[\hat{f}_i(x)] + \frac{1}{B^2}\sum\sum_{i\neq j} Cov\left(\hat{f}_i(x), \hat{f}_j(x)\right)$

K-fold CV: tend to underfit

LOOCV: unbiased but the variance can be very large due to highly correlated training sets

Bootstrap: too optimistic due to overlap of training and test set

$\hat{R}^* = \frac{1}{B}\frac{1}{n}\sum_{b=1}^{B}\sum_{i=1}^{n} l(y_i, \hat{f}^{*b}(x_i))$

The chance of a sample to have appeared in the bootstrap is $1 - \left(1 - \frac{1}{n}\right)^n \approx 0.632$

The "Jackknife" method

With LOOCV

$bias^{JK} = (n-1)(\tilde{S}_n - \hat{S}_n)$

$\hat{S}^{JK} = \hat{S}_n - bias^{JK}$

with $\tilde{S}_n = \frac{1}{n}\sum_{i=1}^{n}\hat{S}_{n-1}^{-i}$

$E\left[bias[\hat{S}_n]\right] = \frac{a_1}{n} + \frac{a_2}{n^2} + \cdots$

$E[bias^{JK}] = \frac{a_1}{n} + O(n^{-2})$

With bootstrap (debiasing)

$bias_{boot} = \frac{1}{B}\sum_{b\leq B}\hat{S}^*(b) - \hat{S}_n$

$\bar{S} = \hat{S}_n - bias$

$= 2\,\hat{S}_n - \frac{1}{B}\sum_{b\leq B}\hat{S}^*(b)$

Gradient Descent procedure

Algorithm: Initialize α, threshold $\theta, \eta(0), k \leftarrow 0$

Repeat

$\alpha \leftarrow \alpha - \eta(k)\nabla J(\alpha)$

$k \leftarrow k + 1$

Until $|\eta(k)\nabla J(\alpha)| < \theta$

Optimal learning rate

$H = \frac{\partial^2 J}{\partial\alpha_i\partial a_j}$

$J(a(k+1))$
$\approx J(a(k)) + \nabla J$^T $(\alpha(k+1)$
$- a(k)) + 1/2((\alpha(k+1)$
$- a(k))$^T $H((a(k+1) - a(k))$
$= J(a(k)) - \eta(k)\nabla J$^T ∇J
$+ 1/2\ \eta$^2 $(k)\nabla J$^T $H\nabla J$

$\frac{\partial}{\partial\eta}J(a(k+1)) = 0 \Rightarrow$

$\eta^{opt} = \frac{||\nabla J||^2}{\nabla J^T H\nabla J}$

Newton's algorithm

$\frac{\partial}{\partial a(k+1)}J\left(a(k+1)\right) = 0 \Rightarrow$

$a(k+1) = a(k) - H^{-1}\nabla J$

Perceptron

$$J_p(a) = \sum_{x\in\tilde{X}_{mc}}(-ya^T x)$$

Update rule: $a(k+1) = a(k) + \eta(k)\sum_{x\in X_{mc}} yx$

WINNOW algorithm

Performs better when many dimensions are irrelevant. Search for 2 weight vectors a^+, a^-(for each class). If a point is misclassified: $a_i^+ \leftarrow \alpha^{+x^{ki}}a_i^+$; $a_i^- \leftarrow \alpha^{-x^{ki}}a_i^-$ (if $y_k = +1$); $a_i^+ \leftarrow \alpha^{-x^{ki}}a_i^+$; $a_i^- \leftarrow \alpha^{+x^{ki}}a_i^-$(if $y_k = -1$)

Fisher's LDA

$\tilde{m}_\alpha = w^Tm_\alpha = w^T\frac{1}{|X_\alpha|}\sum_{x\in X_\alpha} x$

$\Sigma_\alpha = \Sigma_{x\in X_\alpha}(x - m_\alpha)(x - m_\alpha)^T$

$\Sigma_W = \Sigma_1 + \Sigma_2$;

$\tilde{\Sigma}_1 + \tilde{\Sigma}_2 = w^T\Sigma_W w$

Fisher's Separation Criterion

$$J(w) = \frac{||\tilde{m}_1 - \tilde{m}_2||^2}{\tilde{\Sigma}_1 + \tilde{\Sigma}_2}$$
$$= \frac{w^T(m_1 - m_2)(m_1 - m_2)^Tw}{w^T\Sigma_W w}$$
$$= \frac{w^T\Sigma_B w}{w^T\Sigma_W w}$$

$\frac{d}{dw}J(w) = 0 \Rightarrow \Sigma_B w$

$= \Sigma_w w\frac{w^T\Sigma_B w}{w^T\Sigma_W w} \Rightarrow \Sigma_w^{-1}\Sigma_B w$

$= \lambda w$ with $\lambda = \frac{w^T\Sigma_B w}{w^T\Sigma_W w}$

$\hat{\mathbf{w}} = \mathbf{\Sigma_w^{-1}(m_1 - m_2)}$

complexity: O(nd^2)

Logistic regression

$R(w) = \sum_{i=1}^{n}\log(1 + \exp(-y_iw^Tx_i))$

$P(Y = y|x) = \frac{1}{1 + \exp(-yw^Tx)}$

Kernel

RBF: $k(x,z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$

Sigmoid: k(x,z) = tanhkxz-b

k(x,z) = $\exp(-\alpha||x - z||)$

k((x,h),(y,h'))=p(h|x)p(h'|y)

Properties:

$k(x,z) = k_1(\phi(x), \phi(z))$

$k(x,z) = exp(k_1(x,z))$

k(x,z) =f(x)f(y) (f is real-valued function)

k(x,z)=p($k_1(x,z)$) (p(x) is polynomial with positive coefficients)

SVM

Hard margin

Minimize $\frac{1}{2}w^Tw$

s.t $\forall i: z_i(w^Ty_i + w_0) \geq 1$

Lagrange Function:

$$L(w, w_0, \alpha) = \frac{1}{2}w^Tw - \sum_{i=1}^{n}\alpha_i[z_i(w^Ty_i + w_0) - 1]$$

$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i\leq n}\alpha_i z_i y_i$

$\frac{\partial L}{\partial w_0} = 0 \Rightarrow -\sum_{i\leq n}\alpha_i z_i = 0$

Dual problem:

max $W(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} z_iz_j\alpha_i\alpha_j y_i^Ty_j$ s.t $\forall i\ \alpha_i \geq 0 \wedge \sum_{i\leq n}\alpha_i z_i = 0$

$w^* = \sum_{i=1}^{n}\alpha_i^* z_i y_i$

$w_0^* = -\frac{1}{2}(\min_{i:z_i=1} w^{*T}y_i + \max_{i:z_i=-1} w^{*T}y_i)$

KKT conditions require

$\alpha_i[z_i(w^Ty_i + w_0) - 1] = 0$

$\alpha_i \geq 0$

$z_i(w^Ty_i + w_0) - 1 \geq 0$

Soft margin

Min $\frac{1}{2}w^Tw + C\sum_{i=1}^{n}\mathcal{E}_i$

s.t $\forall i: z_i(w^Ty_i + w_0) \geq 1 - \mathcal{E}_i$

$\mathcal{E}_i \geq 0$

$L(w, w_0, \mathcal{E}, \alpha, \beta)$

$$= \frac{1}{2}w^Tw + C\sum_{i=1}^{n}\mathcal{E}_i - \sum_{i=1}^{n}\alpha_i[z_i(w^Ty_i + w_0) - 1 + \mathcal{E}_i] - \sum_{i=1}^{n}\beta_i\mathcal{E}_i$$

$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow C - \alpha_i - \beta_i = 0$

Dual problem is same, only different in $\forall i\ C \geq \alpha_i \geq 0\ \wedge \sum_{i \leq n} \alpha_i z_i = 0$

KKT conditions require

$\alpha_i[z_i(w^T y_i + w_0) - 1 + \mathcal{E}_i] = 0$

$\mathcal{E}_i(C - \alpha_i) = 0$

Multiclass

$(w_{z_i}^T y_i + w_{z_i,0}) - \max_{z \neq z_i} w_z^T y_i + w_{z,0} \geq m$

Structured SVMs

Joint feature map $\Psi(z, y)$

Scoring function $f_w(z, y) = w^T \Psi(z, y)$

Classification: $h(y) = \mathrm{argmax}_{z \in \mathbb{K}} f_w(z, y)$

Objective:

Min $\frac{1}{2} w^T w + C \sum_{i=1}^n \xi_i$

s.t $\xi_i \geq 0$

$w^T \Psi(z_i, y_i) - w^T \Psi(z, y_i)$

$\geq \Delta(z, z_i) - \mathcal{E}_i\ \forall y_i \in \mathcal{Y}, z \neq z_i$

$\Leftrightarrow w^T \Psi(z_i, y_i) - \max_{z \neq z_i} [\Delta(z, z_i)$

$+ w^T \Psi(z, y_i)] \geq -\mathcal{E}_i\ \forall y_i \in \mathcal{Y}$

Lagrangian: $\frac{1}{2} w^T w + C \sum_{i=1}^n \mathcal{E}_i - \sum_{i=1}^n \sum_{z_j \in \mathbb{K}_i} \alpha_{ij_i} [w^T \Psi_i(z_j) - \Delta_i(z_j) + \mathcal{E}_i] - \sum_{i=1}^n \beta_i \mathcal{E}_i$ where $\mathbb{K}_i = \mathbb{K} \backslash \{z_i\}$

Ensemble methods

Bagging

For b=1 to B do

Z^{*b} = b-th bootstrap sample from Z

Construct classifier c_b based on Z^{*b}

end for

return $\hat{c}_B(x) = \mathrm{sgn}(\sum_{i=1}^B c_i(x))$

Random Forest

For b=1 to B do

Z^{*b} = b-th bootstrap sample from Z

Repeat

1. Select m variables at random from p variables ($m \approx \sqrt{p}$)
2. pick the best variable/split-point among selected variables
3. split the node into two daughter node

Until node size n_{min} is reached

End for

Return the ensemble of trees $\{\hat{c}_b(x)\}_{b=1}^B$

Adaboost

Initialize $w_i = 1/n$

For b = 1 to B do

1. Fit a classifier $c_b(x)$ using weight w_i
2. Compute $\epsilon_b \leftarrow \sum_{i=1}^n w_i^{(b)} \mathbb{I}_{\{c_b(x_i) \neq y_i\}} / \sum_{i=1}^n w_i^{(b)}$
3. Compute $\alpha_b \leftarrow \log \frac{1 - \epsilon_b}{\epsilon_b}$
4. Set $\forall i$: $w_i \leftarrow w_i \exp(\alpha_b \mathbb{I}_{\{c_b(x_i) \neq y_i\}})$

End for

Return $\hat{c}_B(x) = \mathrm{sgn}(\sum_{i=1}^n \alpha_b c_b(x))$

Comparison

(a) same training set on every iteration; varies the training sets using resampling.

(b) weights according to its accuracy; same weight

(c) Both are good at reducing variance but only Boosting tries to reduce bias

PAC learning

Efficiently PAC learnable: if an algorithm A receives as input a sample Z of size $n \geq \mathrm{poly}(\frac{1}{\epsilon}, \frac{1}{\delta})$, then A outputs $\hat{c}$ such that

$P(R(\hat{c}) \leq \epsilon) \geq 1 - \delta$

Axis-aligned rectangles

$P(R(\hat{c}) \leq \epsilon) \geq P(\widehat{R}IG)$

$\geq 1 - 4e^{-\frac{n\epsilon}{4}}$

$P(\neg \widehat{R}IG) = P(\widehat{R}$ doesn't intersect T_i^ϵ for some i$) \leq \sum_i P(\widehat{R} \cap T_i^\epsilon = \phi) = \sum_i \prod_{j \leq n} P(x_j \notin T_i^\epsilon) = 4\left(1 - \frac{\epsilon}{4}\right)^n \leq 4e^{-\frac{n\epsilon}{4}}$

$1 - 4e^{-\frac{n\epsilon}{4}} \geq 1 - \delta \Leftrightarrow n \geq \frac{4}{\epsilon} \ln \frac{4}{\delta}$

$\left[\frac{4}{\epsilon} \ln \frac{4}{\delta}\right]$ is polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$

When $\widehat{R}_n(\hat{c}) = 0$ and $|\mathcal{H}|$ is finite

$n \geq \frac{1}{\epsilon}(\log|\mathcal{H}| + \log 1/\delta)$

Then $P(R(\hat{c}) \leq \epsilon) \geq 1 - \delta$

Proof: $P(R(\hat{c}) > \epsilon) = P(\widehat{R}_n(\hat{c}) = 0\ \wedge R(\hat{c}) > \epsilon) \leq \Pr(\exists h \in H: \widehat{R}_n(h) = 0\ \wedge R(h) > \epsilon) \leq \sum_{h \in H} P(\widehat{R}_n(h) = 0\ \wedge R(h) > \epsilon) = \sum_{h \in H} P(\widehat{R}_n(h) = 0\ |R(h) > \epsilon) P(R(h) > \epsilon) \leq \sum_{h \in H} (\widehat{R}_n(h) = 0\ |R(h) > \epsilon) \leq \sum_{h \in H} \prod_{j \leq n} P(x_j \notin h\Delta c | \Pr(h\Delta c)) < \epsilon) \leq |H|(1 - \epsilon)^n$

Under the stochastic setting

$P(R(\hat{c}) - \inf_{c \in C} R(c) \leq \epsilon) \geq 1 - \delta$

If $\mathcal{C}$ is finite,

$P(R(\hat{c}) - \inf_{c \in C} R(c) > \epsilon) \leq 2|\mathcal{C}| \exp(-2n\epsilon^2)$

VC_c is VC dimension of a concept class $\mathcal{C}$ (complexity measure)

$P(R(\hat{c}) - \inf_{c \in C} R(c) > \epsilon) \leq 9\, n^{VC_c} \exp(-\frac{n\epsilon^2}{32}) \rightarrow 0$ when $n \rightarrow \infty$ if VC_c is finite

Require uniform convergence

$R(\hat{c}) - \inf_{c \in C} R(c) \leq 2 \sup_{c \in C} |\widehat{R}_n(c) - R(c)|$

Hoeffding's Inequality

a. $P(X \geq \epsilon) \leq \frac{E[X]}{\epsilon}$ when $X \geq 0$

b. $E[\exp(sX)] \leq \exp\left(\frac{s^2(b-a)^2}{8}\right)$ when $E[X] = 0$, $a \leq X \leq b, s > 0$

c. $P(S_n - ES_n \geq t) \leq \exp(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2})$ when $S_n = \sum_{i=1}^n X_i$, X_i is independet, $t > 0$

The Hoeffding bound

$P(\tilde{S}_n - E\tilde{S}_n \geq \epsilon) \leq \exp\left(-\frac{2n\epsilon^2}{\sum_{i=1}^n \frac{(b_i - a_i)^2}{n}}\right)$

Where $\tilde{S}_n = \frac{S_n}{n}$ $t = n\epsilon$

Union Bound

$P(\max_i X_i > \epsilon) \leq \sum_i P(X_i > \epsilon)$

ERM for Hyperplanes

$P(R(\hat{c}) - \inf_{c \in C} R(c) > \epsilon) \leq (1 + 2\binom{n}{d}) e^{2d\epsilon} e^{-\frac{n\epsilon^2}{2}}$

If $R(c^*) = 0$ then $P(R(\hat{c}) \leq \epsilon) \leq 2\binom{n}{d} \exp(-\epsilon(n - d))$

Nonparametric Bayesian Methods

$\mathrm{Beta}(x|a, b) = \frac{1}{B(a, b)} x^{a-1}(1 - x)^b$

Stick–Breaking Process(GEM)

Repeatedly draw β_i from $\mathrm{Beta}(1, \alpha)$ $\rho_k = [\prod_{i=1}^{k-1}(1 - \beta_i)] \beta_k$

Connect to DP

$G(\theta) = \sum_{k=1}^\infty \rho_k \delta_{\theta_k}(\theta)$

Chinese Restaurant Process

P(customer n+1 joins table t)

$= \begin{cases} \frac{|t|}{\alpha + n} \\ \frac{\alpha}{\alpha + n} \end{cases}$

DP mixture model

Probability of clusers

$\rho \sim GEM(\alpha)$

Center of clusters

$\mu_k \sim N(\mu_0, \sigma_0)$

Assignment of data points

$z_i \sim Categorical\ (\rho)$

Coordinates of data points

$x_i \sim N(\mu_{z_i}, \sigma)$

Asymptotics of $S(n) = \sum_{i=1}^n \frac{\alpha}{\alpha + i - 1} \rightarrow \alpha \ln(n)\ as\ n \rightarrow \infty$

$S(n) < I(n) = \int_1^{n+1} \frac{\alpha}{\alpha + x - 1} dx$

$< \sum_{i=2}^{n+1} \frac{\alpha}{a + i - 1}$

$= S(n) - 1 + \frac{\alpha}{a + n}$

Gibbs sampling

$P(z_i = k | z_{-i}, x, \alpha, \mu)$

$= \begin{cases} \frac{N_{k,-i}}{\alpha + N - 1} p(x_i | x_{-i,k}, \mu) \\ \frac{\alpha}{\alpha + N - 1} P(x_i | \mu) \end{cases}$

for $i = 1$ to N in random order **do**
  Remove x_i's sufficient statistics from old cluster z_i;
  for $k = 1$ to K **do**
    Compute $p_k(x_i) = p_k(x_i | \boldsymbol{x}_{-i,k})$;
    Set $N_{k,-i} = |\boldsymbol{x}_{-i,k}|$;
    Compute $p(z_i = k | \boldsymbol{z}_{-i}, \boldsymbol{x}) = \frac{N_{k,-i}}{\alpha + N - 1}$;
  end for
  Compute $p_\star(x_i) = p(x_i | \boldsymbol{\mu})$;
  Compute $p(z_i = \star | \boldsymbol{z}_{-i}, \boldsymbol{x})$;
  Normalize $p(z_i | \cdot)$;
  Sample $z_i \sim p(z_i | \cdot)$;
  Add x_i's sufficient statistics to new cluster z_i;
  If any cluster is empty, remove it and decrease K;
end for

Extra

Tayler expansion

$e^x = \sum_{n=0}^\infty \frac{x^n}{n!}$; $\frac{1}{1 - x} = \sum_{n=0}^\infty x^n$

$In(1 - x) = -\sum_{n=1}^\infty \frac{x^n}{n!}$; $In(1 + x) = \sum_{n=1}^\infty (-1)^{n+1} \frac{x^n}{n!}$;

Gaussian

$(2\pi)^{-\frac{k}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \mu)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right)$

AIC : $-2 \log\left(\hat{p}(X | \hat{\theta}_k)\right) + 2k$

BIC: $-2 \log\left(\hat{p}(X | \hat{\theta}_{k}, M_k)\right) + k' logn$