Math

Expectation

$\mathbb{E}[X] = \int_\Omega x p(x)\,\mathrm{d}x = \int_\omega x \mathbb{P}[X{=}x]\,\mathrm{d}x$

$\mathbb{E}_{Y|X}[Y] = \mathbb{E}_Y[Y|X]$

$\mathbb{E}_{X,Y}[f(X,Y)] = \mathbb{E}_X \mathbb{E}_{Y|X}[f(X,Y)|X]$

Variance & Covariance

$\mathbb{V}(X){=}\mathbb{E}[(X{-}\mathbb{E}[X])^2]{=}\mathbb{E}[X^2]{-}\mathbb{E}[X]^2$

$\mathbb{V}[X{+}Y]{=}\mathrm{Var}[X]{+}\mathrm{Var}[Y]$ for X, Y iid

$\mathbb{V}(AX) = A\mathbb{V}(X)A^T$

$\mathbb{V}[\alpha X] = \alpha^2 \mathrm{Var}[X]$

$\mathrm{Cov}(X,Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$

Conditional Probabilities & Bayes

$\mathbb{P}[X|Y] = \frac{\mathbb{P}[X,Y]}{\mathbb{P}[Y]} = \frac{\mathbb{P}[Y|X]\mathbb{P}[X]}{\mathbb{P}[Y]}$. Posterior: $P(\text{model}|\text{data})$, likelihood: $P(\text{data}|\text{model})$, prior: $P(\text{model})$, evidence: $P(\text{data})$

Distributions

$\mathcal{N}(x|\mu,\sigma^2) = \frac{e^{-(x-\mu)^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}}$

$\mathcal{N}(x|\mu,\Sigma) = \frac{e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}}{(2\pi)^{D/2}|\Sigma|^{1/2}}$

$\mathrm{Ber}(x|\theta) = \theta^x(1-\theta)^{1-x} \quad 0 \le \theta \le 1$

Chebyshev & Consistency

$\mathbb{P}(|X - \mathbb{E}[X]| \ge \varepsilon) \le \frac{\mathbb{V}[X]}{\varepsilon^2}$

Jensen Inequality

$f(\sum_{i=1}^n \lambda_i x_i) \le \sum_{i=1}^n \lambda_i f(x_i)$, f convex

$f(E[X]) \le E[f(X)]$

Cramer Rao lower bound

$\mathrm{Var}[\hat\theta] \ge \mathcal{I}_n(\theta)^{-1}(\frac{\partial}{\partial\theta}b_\theta + 1)^2 + b_\theta^2$

$\mathcal{I}_n(\theta) = -\mathbb{E}[\frac{\partial^2 \log p[\mathcal{X}_n|\theta]}{\partial\theta^2}]$, $b_\theta :=$ bias

Efficiency of unb. $\hat\theta$: $e(\theta_n) = \frac{1}{\mathrm{Var}[\hat\theta_n]\mathcal{I}_n(\theta)}$

$e(\theta_n) = 1$ (efficient)

$lim_{n\to\infty} e(\theta_n) = 1$ (asymp. efficient)

Derivatives

$\frac{\partial}{\partial \mathbf{x}}(\mathbf{b}^\top\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top\mathbf{b}) = \mathbf{b}$

$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top\mathbf{x}) = 2\mathbf{x}$

$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top\mathbf{A}\mathbf{x}) = (\mathbf{A}^\top + \mathbf{A})\mathbf{x}$

$\frac{\partial}{\partial \mathbf{x}}(\mathbf{b}^\top\mathbf{A}\mathbf{x}) = \mathbf{A}^\top\mathbf{b} \qquad \frac{\partial}{\partial \mathbf{X}}(\mathbf{c}^\top\mathbf{X}\mathbf{b}) = \mathbf{c}\mathbf{b}^\top$

$\frac{\partial}{\partial \mathbf{x}}(\|\mathbf{x}-\mathbf{b}\|_2) = \frac{\mathbf{x}-\mathbf{b}}{\|\mathbf{x}-\mathbf{b}\|_2}$

$\frac{\partial}{\partial \mathbf{x}}(\|\mathbf{x}\|_2^2) = \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top\mathbf{x}) = 2\mathbf{x}$

$\frac{\partial}{\partial \mathbf{X}}(\|\mathbf{X}\|_F^2) = 2\mathbf{X} \qquad \frac{\partial}{\partial \mathbf{x}}\|\mathbf{x}\|_1 = \frac{\mathbf{x}}{|\mathbf{x}|}$

$\frac{\partial}{\partial \mathbf{x}}(\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2) = \mathbf{2}(\mathbf{A}^\top\mathbf{A}\mathbf{x} - \mathbf{A}^\top\mathbf{b})$

$\frac{\partial}{\partial \mathbf{X}}(|\mathbf{X}|) = |\mathbf{X}|\cdot\mathbf{X}^{-1} \qquad |X| = 1/|X^{-1}|$

$\frac{\partial}{\partial x}(\mathbf{Y}^{-1}) = -\mathbf{Y}^{-1}\frac{\partial \mathbf{Y}}{\partial x}\mathbf{Y}^{-1}$

Constrained Convex Optimization

$\min_w f(w)$ s.t. $g_i(w) = 0, h_j(w) \le 0$.
$\mathcal{L}(w,\lambda,\alpha) = f(w) + \sum_i \lambda_i g_i(w) + \sum_j \alpha_j h_j(w), \quad \alpha_j \ge 0$. Slater: Is w s.t. $g_i(w) = 0$ and $h_j(w) < 0$? If yes, strong duality and compl. slackness: $\alpha_j h_j(w) = 0$

Matrix PSD

Det(2x2): $ad - bc$ Det(3x3): $aei + bfg + cdh - gec - hfa - idb$ Symm. matrix is psd if det of all principal minors (removing k-th row & column, including matrix itself) are nonnegative.

Parametric Density Estimation

Maximum Likelihood (MLE)

Likelihood: $\mathbb{P}[\mathcal{X}|\theta] = \prod_{i\le n} p(x_i|\theta)$

Find: $\hat\theta \in \arg\max_\theta \mathbb{P}[\mathcal{X}|\theta]$

Procedure: solve $\nabla_\theta \log \mathbb{P}[\mathcal{X}|\theta] \equiv 0$

Consistent (conv. θ_0 in prob.) & asymp. eff. & asymp. $\mathcal{N}$ & equivariant ($g(\hat\theta)$ MLE)

MLE for Gaussian:

1) $\mathcal{L} \propto N log|\Sigma| + \sum_i^N (x_i-\mu)^T\Sigma^{-1}(x_i-\mu)$

2) $\hat\mu = \frac{1}{N}\sigma_i^N x_i$

3) Trick: derive wrt. $\Sigma^{-1} \to \hat\Sigma = \frac{1}{N}\sum_i^N (x_i-\mu)(x_i-\mu)^T$

4) Show $\hat\Sigma$ is biased: $\mathbb{E}(\hat\Sigma) = \frac{1}{N}\sum_i^N \mathbb{E}(x_i x_i^T) - \frac{1}{N^2}\sum_i^N\sum_j^N \mathbb{E}(x_i x_j^T) - \frac{1}{N^2}\sum_i^N\sum_j^N \mathbb{E}(x_i x_j^T) + \frac{1}{N^2}\sum_i^N\sum_j^N \mathbb{E}(x_i x_j^T) = \frac{1}{N}\sum_i^N \mathbb{E}(x_i x_i^T) - \frac{1}{N^2}\sum_i^N\sum_j^N \mathbb{E}(x_i x_j^T) = \frac{1}{N}N(\Sigma + \mu\mu^T) - \frac{1}{N^2}(N^2\mu\mu^T + N\Sigma) = \Sigma - \frac{1}{N}\Sigma \ne \Sigma$

Tricks used here:

$\Sigma = \mathbb{E}[(x_i-\mu)(x_i-\mu)^T] = \mathbb{E}[x_i x_i^T] - \mu\mu^T \to \mathbb{E}[x_i x_i^T] = \Sigma + \mu\mu^T$ and for $i \ne j$ $\mathbb{E}[x_i x_j^T] = \mu\mu^T$ Also useful: $\frac{1}{N}\sum_{i=1}^N \mathbb{E}[x_i\hat\mu^T] = \frac{1}{N^2}\sum_{i=1}^N\sum_{j=1}^N \mathbb{E}[x_i x_j]$

Maximum A Posteriori (MAP)

Assume prior $\mathbb{P}(\theta)$

Find: $\hat\theta \in \arg\max_\theta P(\theta|\mathcal{X}) = \arg\max_\theta P(\mathcal{X}|\theta)P(\theta)$

Solve $\nabla_\theta log P(\mathcal{X}|\theta)P(\theta) = 0$

Note: $P(\mathcal{Y}|\mathcal{X},\beta) \sim \mathcal{N}(X^T\beta,\sigma^2) \to \arg\max_\theta log(P(\mathcal{Y}|\mathcal{X},\beta)) = \arg\max_\theta -\frac{1}{2\sigma^2}||Y - X^T\beta||^2$

Bayesian learning

$p(X = x|data) = \int p(x,\theta|data)d\theta = \int p(x|\theta)p(\theta|data)d\theta$

Estimate Gaussian: $X \sim \mathcal{N}(\mu,\sigma^2)$,

$P(\mu) \sim \mathcal{N}(\mu_0,\sigma_0^2)$ then $\sigma_n^2 = \frac{\sigma^2\sigma_0^2}{n\sigma_0^2+\sigma^2}$,

$\mu_n = \frac{n\sigma_0^2}{n\sigma_0^2+\sigma^2}\hat\mu_n + \frac{\sigma^2}{n\sigma_0^2+\sigma^2}\mu_0$ with $\hat\mu_n = \frac{1}{n}\sum_{i=1}^n x_i$

Bayesian density learning

Prior Knowledge of $p(\theta)$,
Find Posterior Density: $p(\theta|\mathcal{X})$.
$\mathcal{X}^n = \{x_1,\cdots,x_n\}$

$p(\theta|\mathcal{X}^n) = \frac{p(x_n|\theta)p(\theta|\mathcal{X}^{n-1})}{\int p(x_n|\theta)p(\theta|\mathcal{X}^{n-1})d\theta}$

Frequentist vs Bayesian

Bayes (MAP): allows priors, provides distribution when estimating parameters, requires efficient integration methods when computing posteriors, prior often induces regularization term. Frequentist method (MLE): does not allow priors, provides a single point when estimating parameters, requires only differentiation methods, MLE estimators are consistent, equivariant, asymptotically normal, asymptotically efficient (for finite samples not necessarily efficient).

Optimization

Gradient Descent

$\theta^{\text{new}} \leftarrow \theta^{\text{old}} - \eta\nabla_\theta\mathcal{L}(\theta^{\text{old}})$

Conv. isn't guaranteed. Less zig-zag by adding momentum: $\theta^{(l+1)} \leftarrow \theta^{(l)} - \eta\nabla_\theta\mathcal{L} + \mu(\theta^l - \theta^{(l-1)})$.

Robbins-Monro alg.: Compute θ^* s.t. $\mathbb{E}_Z[f(Z;\theta)] = 0$ by updating $\theta^{(k)} = \theta^{(k-1)} - \eta(k)f(z_k;\theta^{(k-1)})$. Batch GD: Higher gradient precision, larger gen. error. SGD: Large train sets, faster improvements, escapes local min.

Newton's Method (opt. grad. descent)

Use 2nd order derivation. (Hessian) $\theta^{\text{new}} \leftarrow \theta^{\text{old}} - \nabla_\theta\mathcal{L}/\nabla_\theta^2\mathcal{L}$ GD depends $\eta(k)$, but comp. easier. NM requires H^{-1}, but better updates and no learn rate.

Data Types

monadic: $X : O \to \mathbb{R}^d$

dyadic: $X : O^{(1)} \times O^{(2)} \to \mathbb{R}^d$

pairwise: $X : O^{(1)} \times O^{(1)} \to \mathbb{R}^d$

polyadic: $X : O^{(1)} \times O^{(2)} \times O^{(3)} \to \mathbb{R}^d$

nominal = qualitative (sweet, sour ...),
ordinal = absolute order of items,
quantitative (interval: differences, ratio: zero value, absolute: values) = numbers

Risks and Losses

Expected Risk:

$R(\hat c) = \sum_{y\le k} P(y)\mathbb{E}_{P(x|y)}[\mathbb{I}_{\hat c(x)\ne y}|Y = y]$

or $R(\hat c) = P(\hat c(X) \ne y)$

Empirical Risk Minimizer (ERM) $\hat f$:

$\hat f \in \arg\min_{f\in\mathcal{C}} \hat R(\hat f, Z^{train})$

$\hat R(\hat f, Z^{train}) = \frac{1}{n}\sum_{i=1}^n Q(Y_i, \hat f(X_i))$

$\hat R(\hat f, Z^{test}) = \frac{1}{m}\sum_{i=n+1}^{n+m} Q(Y_i, \hat f(X_i))$

Linear Regression

Model of data: $\mathbf{Y} = \mathbf{X}^T\beta + \varepsilon$

$\mathbf{X} \in \mathbb{R}^{(d+1)\times n}, \beta \in \mathbb{R}^{d+1}$

additive Gaussian noise $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbb{I}\sigma^2)$

$\hat{\mathbf{Y}} = \mathbf{X}\hat\beta \quad \hat\beta \sim \mathcal{N}(\beta, (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2)$

and $p(\mathbf{Y}|\mathbf{X},\beta,\sigma^2) = \mathcal{N}(\mathbf{Y}|\mathbf{X}^T\beta, \mathbb{I}\sigma^2)$

(1) Ordinary Least Squares

Setting: Minimize RSS.

$\mathcal{L} = \mathrm{RSS}(\beta) = \sum_{i=1}^n (y_i - x_i^T\beta)^2 = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$

Solution: differentiate $\mathcal{L}$ w.r.t β

$\hat\beta^{\text{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

Is an orthogonal projection with lowest variance of all unbiased estimates.

Prediction: $\hat y{=}\mathbf{X}\hat\beta{=}\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ Gau-Mary: $\mathbb{V}[\hat\beta] \le \mathbb{V}[\tilde\beta], \tilde\beta = C^T y, \mathbb{E}[\tilde\beta] = \beta$

(2) Ridge Regression (L^2 penalty)

Setting: Penalize energy in β.

$\mathcal{L} = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^T\beta$

Solution: differentiate $\mathcal{L}$ w.r.t β

$\hat\beta^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbb{I})^{-1}\mathbf{X}^T\mathbf{y}$

Bayesian view: $Y|(X,\beta) \sim \mathcal{N}(X^T\beta, \sigma^2)$ $\beta \sim \mathcal{N}(0, \sigma^2/\lambda)$

(3) Lasso (L^1 penalty)

Setting: Penalize full β.

$\mathcal{L} = \sum_{i=1}^n (y_i - x_i^T\beta)^2 + \lambda\sum_{j=1}^d |\beta_j| = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda||\beta||_1$

Bayesian view: $Y|(X,\beta) \sim \mathcal{N}(x^T\beta, \sigma^2 I)$

Laplace prior: $p(\beta_i) = \frac{\lambda}{4\sigma^2}exp(-|\beta|\frac{\lambda}{2\sigma^2})$

Lasso has no closed form.

(4) Bayesian Linear Regression

Setting: Define a prior over β.

e.g. Ridge: Assume β distributed as:

$p(\beta|\Lambda){=}\mathcal{N}(\beta|\mathbf{0}, \frac{\sigma^2}{\lambda}\mathbb{I}){\propto}\exp(-\frac{\lambda}{2\sigma^2}\beta^T\beta)$

For $\Lambda = \frac{\sigma^2}{\lambda}\mathbb{I}$. Linear for $\sigma = 1$.

Then, given observed $\mathbf{X}, \mathbf{y}$, use Bayes' theorem to find the posterior

$p(\beta|\mathbf{X},\mathbf{y},\Lambda,\sigma) = \mathcal{N}(\mu_\beta, \Sigma_\beta)$

$\mu_\beta = \sigma^2(\mathbf{X}^T\mathbf{X} + \sigma^2\Lambda)^{-1}\mathbf{X}^T\mathbf{y}$

$\Sigma_\beta = \sigma^2(\mathbf{X}^T\mathbf{X} + \sigma^2\Lambda)^{-1}$

Nonlinear Regression

Idea: Feature space transformation

Model: $\mathbf{Y} = f(\mathbf{X}) = \sum_{m=1}^M \beta_m h_m(\mathbf{X})$

Transformation $h_m(\mathbf{X}) : \mathbb{R}^d \to \mathbb{R}$

Cubic Spline

e.g. for $d = 1$ with knots at ξ_1 and ξ_2

$h_1(X){=}1 \quad h_3(X){=}X^2 \quad h_5(X){=}(X{-}\xi_1)_+^3$

$h_2(X){=}X \quad h_4(X){=}X^3 \quad h_6(X){=}(X{-}\xi_2)_+^3$

Smoothing Spline

Add penalty for second deriv.: $RSS(f,\lambda) = \sum_{n=1}^n (y_i - f(x_i))^2 + \lambda\int(f''(x))^2 dx$

Gaussian Process Regression

joint Gaussian over all outputs

$\mathbf{y} = \mathbb{X}\beta + \varepsilon \quad \varepsilon \sim \mathcal{N}(0, \sigma^2\mathbb{I}_n)$

We can rewrite the distribution

$P(\begin{bmatrix}\mathbf{y}\\ y_*\end{bmatrix}) = \mathcal{N}(\begin{bmatrix}\mathbf{y}\\ y_*\end{bmatrix} | \mathbf{0}, \begin{bmatrix}\mathbf{K} + \sigma^2\mathbb{I} & \mathbf{k}\\ \mathbf{k^T} & k(x_*,x_*) + \sigma^2\end{bmatrix})$

k is the kernel function. Lengthscale: how far can we reliably extrapolate.

Gaussian Process Prediction

$p(y_*|x_*,\mathbf{X},\mathbf{y}) = \mathcal{N}(\mu_{y_*}, \sigma_{y_*}^2)$,

$\mu_{y_*} = \mathbf{k}^T(\mathbf{K} + \sigma^2\mathbb{I})^{-1}\mathbf{y}$,

$\sigma_{y_*}^2 = k(x_*,x_*){+}\sigma^2 - \mathbf{k}^T(\mathbf{K} + \sigma^2\mathbb{I})^{-1}\mathbf{k}$

$\mathbf{k} = k(x_*,\mathbf{X}) \quad \mathbf{K}_{ij} = k(x_i,x_j)$

General Conditional Gaussian Prediction

$P(\begin{bmatrix}\mathbf{a1}\\ \mathbf{a2}\end{bmatrix}){=}\mathcal{N}(\begin{bmatrix}\mathbf{a1}\\ \mathbf{a2}\end{bmatrix} | \begin{bmatrix}\mathbf{u1}\\ \mathbf{u2}\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix})$

$p(\mathbf{a2}|\mathbf{a1}) = \mathcal{N}(\mathbf{a2}|\mathbf{u2} + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{a1} - \mathbf{u1}), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12})$

Bias-Variance tradeoff

$\mathrm{Bias}(\hat f){=}\ \mathbb{E}[\hat f] - f$

$\mathrm{Var}(\hat f){=}\ \mathbb{E}[(\hat f - \mathbb{E}[\hat f])^2]$

Squared Error Decomposition

$\mathbb{E}_D\mathbb{E}_{X,Y}[(\hat f(X) - Y)^2] =$
$\mathbb{E}_{X,Y}[(\mathbb{E}_{Y|X}[Y] - Y)^2]$ (noise var)
$+\mathbb{E}_X\mathbb{E}_D[(\hat f_D(X) - \mathbb{E}_D[\hat f(X)])^2]$ (var.)
$+\mathbb{E}_X[(\mathbb{E}_D[\hat f_D(X)] - \mathbb{E}_{Y|X}[Y])^2]$ (bias^2)

With $\mathbb{E}_{Y|X}[Y]$ the expected label and $\mathbb{E}_D[\hat f(X)]$ the expected classifier.

Classification

Perceptron

Compute w s.t. $w^T x_i > 0$ if $y_i = 1$.

$L(y, c(x)) = |w^T x|$ if class. wrong, 0 oth.

Algorithm: Pick a sample x_i, update weights by $-\eta(k)y_i x_i$ if misclassified (unt. thresh.).

Fisher's Linear Discriminant Analysis

Maximize distance of the means of the projected classes and minimize var per class.

proj mean: $\tilde\mu_\alpha{=}\frac{1}{n_\alpha}\sum_{x\in\mathcal{X}_\alpha} w^T x{=}w^T\mu_\alpha$

Dist of proj means: $|w^T(\mu_1 - \mu_2)|$ Classes proj. var: $\tilde\Sigma_1{+}\tilde\Sigma_2{=}w^T(\Sigma_1{+}\Sigma_2)w$

Fishers Criterion (maximized):

$\frac{||w^T(\mu_1-\mu_2)||^2}{\tilde\Sigma_1+\tilde\Sigma_2} = \frac{w^T(\mu_1-\mu_2)(\mu_1-\mu_2)^T w}{w^T(\Sigma_1+\Sigma_2)w}$

LDA/QDA

Assume $p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y)$. When $\Sigma_0 = \Sigma_1$, $p(y|x) = \sigma(w^T x + w_0)$ (LDA), in general $\sigma(x^T W x + x^T w + w_0)$ (QDA).

Support Vector Machine (SVM)

Generalize Perceptron with margin and kernel. Find plane that max. margin m s.t.:
$z_i g(\mathbf{y}) = z_i(\mathbf{w}^T\mathbf{y} + w_0) \geq m, \forall \mathbf{y}_i \in \mathcal{Y}$
$z_i \in \{-1, +1\}$ $\mathbf{y_i} = \phi(\mathbf{x_i})$ Equiv. $\min_{w,w_0} 1/2\|w\|^2$ s.t. $z_i(\mathbf{w}^T\mathbf{y} + w_0) \geq 1$

Functional Margin Problem:
minimizes $||\mathbf{w}||$ for m=1: $L(\mathbf{w}, w_0, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^n \alpha_i[z_i(\mathbf{w}^T\mathbf{y}_i + w_0) - 1]$
where $\alpha \geq 0$ are Lagrange multipliers.
$\frac{\partial L}{\partial w}$=0 and $\frac{\partial L}{\partial w_0}$=0 give us constraints
$\mathbf{w}^* = \sum_{i=1}^n \alpha_i z_i \mathbf{y_i}$ $\quad 0 = \sum_{i=1}^n \alpha_i z_i$
Replacing these in L we get (max α)
$\tilde{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j z_i z_j \mathbf{y_i}^T\mathbf{y_j}$
with $\alpha_i \geq 0$ and $\sum_{i=1}^n \alpha_i z_i = 0$
This is the dual (tractable for transform.) representation. Optimal hyperplane:
$\mathbf{w}^* = \sum_{i=1}^n \alpha_i^* z_i \mathbf{y_i}$
$w_0^* = -\frac{1}{2}(\min_{z_i=1}\mathbf{w}^{*T}\mathbf{y_i} + \max_{z_i=-1}\mathbf{w}^{*T}\mathbf{y_i})$
where α maximize the dual problem. Only Support Vectors ($\alpha_i \neq 0$) contribute to the evaluation, dual is quad. optim. in simplex
Optimal Margin: $\mathbf{w}^T\mathbf{w} = \sum_{i \in SV} \alpha_i^*$
Discrim.: $g^*(\mathbf{x}) = \sum_{i \in SV} z_i \alpha_i \mathbf{y_i}^T\mathbf{y_i} + w_0^*$
Comp Slack $\alpha_i^*(1 - z_i(w^{*T}y_i + w_0^*)) = 0$

Soft Margin SVM

Introduce slack to relax constraints
minimize $\frac{1}{2}w^T w + C\sum_{i=1}^n \xi_i$ such that: $z_i(\mathbf{w}^T\mathbf{y}_i + w_0) \geq 1 - \xi_i$
Lagrangian: $L(\mathbf{w}, w_0, \xi, \alpha, \beta) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i[z_i(\mathbf{w}^T\mathbf{y}_i + w_0) - 1 + \xi_i] - \sum_{i=1}^n \beta_i\xi_i$
C controls margin maximization vs. constraint violation. Dual Problem same as usual SVM but with supplementary constraint: $C \geq \alpha_i \geq 0$, $\xi_i^* = \max(0, 1 - z_i(w^{*T}y_i + w_0^*))$ **Kuhn-Tucker Conditions:** $\alpha_i^*(z_i(w^T y_i + w_0) - 1 + \xi) = 0, \xi_i(\alpha_i - C) = 0$

Non-Linear SVM

Use kernel in discriminant function:
$g(\mathbf{x}) = \sum_{i,j=1}^n \alpha_i z_i K(\mathbf{x_i}, \mathbf{x})$
E.g solve the XOR Problem with:
$K(x, y) = (1 + x_1 y_1 + x_2 y_2)^2$

Structured SVM

Each sample $\mathbf{y}$ is assigned to a structured output label z
Output Space Representation:
joint feature map: $\psi(z, \mathbf{y})$
Scoring function: $f_\mathbf{w}(z, \mathbf{y}) = \mathbf{w}^T\psi(\mathbf{z}, \mathbf{y})$
Classify: $\hat{z} = h(\mathbf{y}) = \arg\max_{z \in \mathbb{K}} f_{\mathbf{w}(z,\mathbf{y})}$
SVM objective (loss Δ): $\min_{w,\xi \geq 0} \frac{1}{2}w^T w + C/n\sum_{i=1}^n \xi_i$ s.t. $w^T\psi(z_i, \mathbf{y_i}) - \Delta(z, z_i) - w^T\psi(z, \mathbf{y_i}) \geq -\xi_i \forall z \neq z_i \forall i$
Training algo: Check inequalities, add where not met by largest amount to constraints until tolerance threshold reached.

Advantages/Disadvantages SVM

+: Allows ∞-dim. representations, adapted to structured classific., formulated as QP (efficient). -: Requires careful selection of kernel and feature engineering, use of kernels make susceptible to curse of dim.

Kernels

Similarity based reasoning
Gram Matrix $K = K(\mathbf{x}_i, \mathbf{x}_j), 1 \leq i, j \leq n$
$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')$ $\quad K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$
$K(\mathbf{x}, \mathbf{x}')$ pos.semi-def. ($x^t K x \geq 0$)
Contin.: $\int_\Omega k(x, x') f(x) f(x') dx dx' \geq 0$
If K_1 & K_2 are kernels K is too:
$K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}')K_2(\mathbf{x}, \mathbf{x}')$
$K(\mathbf{x}, \mathbf{x}') = \alpha K_1(\mathbf{x}, \mathbf{x}') + \beta K_2(\mathbf{x}, \mathbf{x}')$
$K(\mathbf{x}, \mathbf{x}') = K_1(h(\mathbf{x}), h(\mathbf{x}'))$ $\quad h: \mathcal{X} \to \mathcal{X}$
$K(\mathbf{x}, \mathbf{x}') = h(K_1(\mathbf{x}, \mathbf{x}'))$ $\quad h$: polynomial with positive coefficients / exp function
$K(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})K_1(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')$ or $K_3(\phi(\mathbf{x}), \phi(\mathbf{x}'))$ with $\phi: \mathcal{X} \to \mathbb{R}^m$
Kernel Function Examples:
$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$ $\quad K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + 1)^p$
$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T A\mathbf{x}'$ (symm psd), RBF(Gauss):
$K(\mathbf{x}, \mathbf{x}') = \exp(-||\mathbf{x} - \mathbf{x}'||_2^2/h^2)$
Sigmoid: $K(\mathbf{x}, \mathbf{x}') = \tanh(\alpha\mathbf{x}^T\mathbf{x}' + c)$
not p.s-d e.g: $\mathbf{x} = [1, -1], \mathbf{x}' = [-1, 2]$
Size of feature space for kernel $(c + x * y)^d$, $x, y \in \mathbb{R}^N$: (n+d choose d)

Bayes Decision Theory

Define $L(y, c(x)), \min_c \mathbb{E}_{X,Y}[L(Y, c(X))] \approx \min_c 1/m\sum_{i \leq m} L(y_i, c(x_i))$, opt. classifier: $c^*(x) = \arg\min_a \sum_y p(y|x)L(y, a)$

Metrics

precision: $\frac{\text{tp}}{\text{tp+fp}}$, recall: $\frac{\text{tp}}{\text{tp+fn}}$, F1: $\frac{2}{\text{p}^{-1}+\text{r}^{-1}}$

Types

Discriminative trains $c: \mathcal{X} \to \mathcal{Y}$, prob. discr. also computes $P(Y|X)$ (degrees of belief), prob. gen infers $P(X, Y)$ (better understanding, new samples, outlier detection)

Ensemble Methods

Combining Regressors

Set of estimators: $\hat{f}_1(x), \cdots, \hat{f}_B(x)$
simple average: $\hat{f}(x) = \frac{1}{B}\sum_{i=1}^B \hat{f}_i(x)$
$\text{Bias}[\hat{f}(x)] = \frac{1}{B}\sum_{i=1}^B \text{Bias}[f_i(x)]$
$\mathbb{V}[\hat{f}(x)] \approx \frac{\sigma^2}{B}$ if the estimators are uncorrelated.

Combining Classifiers

Input: classifiers $c_1(x), \cdots, c_B(x)$
Infer $\hat{c}_B(x) = \text{sgn}(\sum_{b=1}^B \alpha_b c_b(x))$ with weights $\{\alpha_b\}_{b=1}^B$
Requires diversity, ind. of the classifiers.

Bagging

Train on bootstrapped subsets.
Sample: $\mathcal{Z} = \{(x_1, y_1), \cdots (x_n, y_n)\}$
$\mathcal{Z}^*$: chose i.i.d from $\mathcal{Z}$ w. replacement.
Covariance small, variance similar, bias weakly affected.

Random Forest (Bagging strategy)

Ensemble of trees is trained via bagging. But at each splitting step, m features are chosen at random and splitting is only done with one of those. Reduces corr. between trees.

Boosting

Combine uncorr. weak learners in sequence. (Weak to avoid overfitting).
Coeff. of $\hat{c}_{b+1}$ depend on $\hat{c}_b$'s results

AdaBoost (minimizes exp. loss)

Init: $\mathcal{X} = \{(x_1, y_1), \cdots, (x_n, y_n)\}, w_i^{(1)} = \frac{1}{n}$
Fit $\hat{c}_b(x)$ to $\mathcal{X}$ weighted by $w^{(b)}$
$\varepsilon_b = \sum_{i=1}^n w_i^{(b)}\mathbb{I}_{\{\hat{c}_b(x_i) \neq y_i\}} / \sum_{i=1}^n w_i^{(b)}$
$\alpha_b = \log\frac{1-\varepsilon_b}{\varepsilon_b} > 0$
$w_i^{(b+1)} = w_i^{(b)}\exp(\alpha_b\mathbb{I}_{\{\hat{c}_{b}(x_i) \neq y_i\}})$
return $\hat{c}_B(x) = \text{sgn}(\sum_{b=1}^B \alpha_b\hat{c}_b(x))$
Best approx. at log-odds ratio.
Like stagewise-additive modeling with exp. loss: $\min_{\alpha_M, b^{(M)}, \ldots, \alpha_1, b^{(1)}} 1/n\sum_{i \leq n} L(y_i, \alpha_1 b^{(1)}(x_i) + \ldots + \alpha_M b^{(M)}(x_i))$
with m rounds, $L(y, y') = e^{-yy'}$
Trains a max-margin classifier, i.e. $y_i\sum_{t \leq M}\alpha_t b^{(t)}(x_i) \xrightarrow{M \to \infty} 1$. Trains self-avg., spiky (local. changes) interpolators.

Difference

(1) Boosting keeps identical training data, bagging can vary training data for each classif. (2) Boosting weighs the prediction of each classifier accord. to its accuracy, bagging gives same importance to each.

Notes

AdaBoost gives large weight to samples that are hard to classify: those could be outliers. For bagging, there is a chance that imbalanced data-sets lead to bootstrap samples missing a class alltogether. Fix by making the bootstrap size large enough s.t. at least one point is included.

PAC learning

Learning algorithm $\mathcal{A}$ can learn concept c if for any distribution and $0 < \varepsilon < 1/2$, $0 < \delta < 1/2$ s.t. if $\mathcal{A}$ receives sample of size $\geq poly(1/\varepsilon, 1/\delta, size(c))$, it outputs $\hat{c}$ s.t. $\mathbb{P}_{\mathcal{Z} \sim \mathcal{D}^n}(\mathcal{R}(\hat{c}) \leq \varepsilon) \geq 1 - \delta$ ($\mathcal{R}$:= gen. error). In general (opt. classif. might not be in $\mathcal{C}$): $\mathbb{P}[\mathcal{R}(\hat{c}) - \inf_{c \in \mathcal{C}}\mathcal{R}(c) \leq \varepsilon] \geq 1 - \delta$.
for any class $\mathcal{C}$**:** $\mathcal{R}(\hat{c}) - \inf_{c \in \mathcal{C}}\mathcal{R}(c) \leq 2\sup_{c \in \mathcal{C}}|\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)|$, where $\hat{c}$ is the empirical risk minimizer. For finite $\mathcal{C}$: $\mathbb{P}(\sup_{c \in \mathcal{C}}|\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)| > \varepsilon) \leq 2|\mathcal{C}|\exp(-2n\varepsilon^2)$

Rectangle learning

Pick tight rectangle. Diff. between picked rectangle $\hat{R}$ and true R with few examples. Rectangles are **efficiently PAC learnable**: runs in polynom. $1/\varepsilon$ (error param.) and $1/\delta$ (confidence val.).

Nonparametric Bayesian methods

$\text{Beta}(x|a, b) = B(a, b)^{-1}x^{a-1}(1 - x)^{b-1}$: prob. of Bernoulli proc. after observing $a - 1$ success and $b - 1$ failures. Expended to multivariate case with Dirichlet distr. That will give multivar. probs, *based on finite* counts! But we don't know exactly which multivar. distribution works. With more data, we update the Dirichlet distribution. Is a conjugate prior.

Stick-breaking Dirichl. proc.

Repeatedly draw β_i from $\text{Beta}(x|1, \alpha)$ with fixed α and calculate: $\rho_k = \beta_k(1 - \sum_{i=1}^{k-1}\rho_i)$. Then $G(\theta) = \sum\rho_k\delta_{\theta_k}(\theta)$ ($\theta_k \sim H$) is sample from $DP(\alpha, H)$. The prior (analog. CRP):
$$\mathbb{P}[z_i = k|z_{-i}, \alpha] = \begin{cases} \frac{N_{k,-i}}{\alpha+N-1} & \text{existing } k \\ \frac{\alpha}{\alpha+N-1} & \text{otherwise} \end{cases}$$
Final Gibbs sampler:
$$\mathbb{P}[z_i = k|z_{-i}, \alpha, \mu] = \begin{cases} \frac{N_{k,-i}}{\alpha+N-1}p(x_i|x_{-i,k}, \mu) & \text{existing } k \\ \frac{\alpha}{\alpha+N-1}p(x_i, \mu) & \text{otherwise} \end{cases}$$

Gibbs sampling

Init: assign all data to a cluster, with prior π_i, with $\sum_{k=1}^K \pi_i < 1$ (s.t. new clusters possible). E.g. with stick-breaking.
Then remove x from k and compute new θ_k, then compute Gibbs sampler prob. (CRP), and sample the new cluster assignment $z_i \sim p(z_i|x_{-i}, \theta_k)$. If cluster is empty, remove it and decrease K.

Neural Networks

Output units

Binary (sigmoid): $\sigma(z) = 1/(1 + e^{-z})$
Multiclass (softmax): $y_i = e^{\beta z_i}/\sum_{j \leq I} e^{\beta z_j}$

Backpropagation

With layer l weight matrix $w^{[l]}$, outputs $z^{[l]} = w^{[l]}a^{[l-1]} + b^{[l]}$, activations $a^{[l]} = g^{[l]}\left(z^{[l]}\right)$, cost C: $\frac{\partial C}{\partial w_r^{[l]}} = \frac{\partial C}{\partial z^{[l]}}\frac{\partial z_r^{[l]}}{\partial w_r^{[l]}}$, $\frac{\partial C}{\partial b^{[l]}} = \frac{\partial C}{\partial z^{[l]}}\frac{\partial z^{[l]}}{\partial b^{[l]}}$, $\frac{\partial C}{\partial z^{[l]}} = \frac{\partial C}{\partial z^{[l+1]}}\frac{\partial z^{[l+1]}}{\partial a^{[l]}}\frac{\partial a^{[l]}}{\partial z^{[l]}}$

Regularization

Avoid overfitting on complex nets. **Early Stopping** separate data into train/error/validation sets.
Drop Out Combine thinned nets with removed nodes.
Bayesian priors on w's

Convolutional Neural Network

Modelling invariance. Convolutional Layers (filters on a region) & Pooling Layers (aggregate nodes together).

Autoencoder: explicit density

Data compression purposes, Output should reproduce input. $\Rightarrow$ PCA. *Denoising autoencoders* reproduce data from partial observations (blanking out parts of input), more robust.

GAN: implicit density

Sample from noise $\to$ training data. 2-player game: generator network fools discriminator by creating fake images, discriminator distinguishes between real and fake images. Gen: upsampling network with fractionally strided convs, Disc: convolutional network.

Variational Autoencoders

$\log p(x_i) \geq \text{elbo}(x_i) = (Z \sim q_\phi(z|x_i))$
$\mathbb{E}_Z[\log p_\theta(x_i|z)] + \mathbb{E}_Z[\log p_{\theta'}(z)/q_\phi(z|x_i)]$
Max. mutual info, min. KL divergence between prior $p_{\theta'}(z)$ and post. $q_\phi(z|x_i)$. Likelihood $P_\theta(x|z)$ and post. params (e.g. mean, var) often encoded by NN. Training: Define prior, post., likelihood, max. elbo.

Mixture Models

Gaussian Mixture

EM-Algorithm

Latent Variable: unknown data $\to$ What cluster generated each sample?
EM does ML for unknown parameters.
Latent var. $M_{xc} = \begin{cases} 1 & \text{c generated x} \\ 0 & \text{else} \end{cases}$
$P(\mathcal{X}, M|\theta) = \prod_{x \in \mathcal{X}}\prod_{c=1}^k(\pi_c P(\mathbf{x}|\theta_c))^{M_{xc}}$

E-Step

$\gamma_{xc} = \mathbb{E}[M_{xc}|\mathcal{X}, \theta^{(j)}] = \frac{P(\mathbf{x}|c, \theta^{(j)})P(c|\theta^{(j)})}{P(\mathbf{x}|\theta^{(j)})}$

M-Step

$\mu_c^{(j+1)} = \frac{\sum_{x \in \mathcal{X}}\gamma_{xc}\mathbf{x}}{\sum_{x \in \mathcal{X}}\gamma_{xc}}$
$(\sigma_c^2)^{(j+1)} = \frac{\sum_{x \in \mathcal{X}}\gamma_{xc}(\mathbf{x} - \mu_c)^2}{\sum_{x \in \mathcal{X}}\gamma_{xc}}$
$\pi_c^{(j+1)} = \frac{1}{|\mathcal{X}|}\sum_{x \in \mathcal{X}}\gamma_{xc}$