Probabilities

Expectation

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

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

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

Variance & Covariance

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

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

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

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

Conditional Probabilities & Bayes

$\mathbb{P}[X|Y] = \frac{\mathbb{P}[X,Y]}{\mathbb{P}[Y]} = \frac{\mathbb{P}[Y|X]\mathbb{P}[X]}{\mathbb{P}[Y]}$

Distributions

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

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

$\mathrm{Exp}(x|\lambda){=}\lambda e^{-\lambda x}$, $\mathrm{Ber}(x|\theta){=}\theta^x(1{-}\theta)^{(1-x)}$

Sigmoid: $\sigma(x) = 1/(1+e^{-x})$

Chebyshev & Consistency

$\mathbb{P}(|X-\mathbb{E}[X]| \geq \epsilon) \leq \frac{\mathbb{V}[X]}{\epsilon^2}$

$\lim_{n\to\infty} P(|\hat{\mu}-\mu| > \epsilon) = 0$

Cramer Rao lower bound

$\mathrm{Var}[\hat{\theta}] \geq \mathcal{I}_n(\theta)^{-1}$

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

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

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

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

Matrix Derivations

$\frac{\partial \mathbf{a}^T\mathbf{x}}{\partial \mathbf{x}}{=}\mathbf{a}$ $\quad \frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{b}}{\partial \mathbf{X}}{=}\mathbf{a}\mathbf{b}^T$ $\quad \frac{\partial \mathbf{a}^T\mathbf{X}^T\mathbf{b}}{\partial \mathbf{X}}{=}\mathbf{b}\mathbf{a}^T$

$\frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{a}}{\partial \mathbf{a}}{=}\mathbf{a}^T(\mathbf{X}{+}\mathbf{X}^T)$, $\frac{\partial \mathbf{K}^{-1}}{\partial K} = -\mathbf{K}^{-1}\mathbf{K}'\mathbf{K}^{-1}$

$\frac{\partial}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x})^T\mathbf{g}(\mathbf{x}){=}\mathbf{f}(\mathbf{x})^T\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} + \mathbf{g}(\mathbf{x})^T\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}$

$\mathbf{X}^T\mathbf{X}$: invertible if no no zero eigenvalues. Inversion unstable if ratio from $\mathbf{X}$'s smallest EV to the largest is big.

Parametric Density Estimation

Maximum Likelihood (MLE)

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

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: converges to best θ_0.

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

Bayesian density learning

Prior Knowledge of $p(\theta)$,
Find Posterior Density: $p(\theta|\mathcal{X})$.

$\mathcal{X}^n = \{x_1,\cdots,x_n\}$

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

Optimization

Gradient Descent

$\theta^{\mathrm{new}} \leftarrow \theta^{\mathrm{old}} - \eta\nabla_\theta\mathcal{L}$

Convergence isn't guaranteed.
Less zigzag by adding momentum:

$\theta^{(l+1)} \leftarrow \theta^{(l)} - \eta\nabla_\theta\mathcal{L} + \mu(\theta^l - \theta^{(l-1)})$

Newton's Method

Use 2nd order derivation. (Hessian)

$\theta^{\mathrm{new}} \leftarrow \theta^{\mathrm{old}} - \eta(\nabla_\theta\mathcal{L}/\nabla^2_\theta\mathcal{L})$

$H = \nabla^2_\theta\mathcal{L}$ has to be p.d (convex func).

Risks and Losses

Conditional Expected Risk

$R(f,X) = \int_\mathbb{R} \mathcal{L}(Y,f(X))\mathbb{P}(Y|X)\,\mathrm{d}Y$

Total Expected Risk $R(f) =$
$\mathbb{E}_X[R(f,X)] = \int_\mathcal{X} R(f,X)\mathbb{P}[X]\,\mathrm{d}X =$
$\int_\mathcal{X}\int_\mathbb{R} \mathcal{L}(Y,f(X))\mathbb{P}[X,Y]\,\mathrm{d}X\,\mathrm{d}Y$.

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

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

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

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

$Z^{\mathrm{train}} = (X_1,Y_1),\ldots,(X_n,Y_n)$

$Z^{\mathrm{test}} = (X_{n+1},Y_{n+1}),\ldots,(X_{n+m},Y_{n+m})$

Linear Regression

Data: $Z = (x_i,y_i) \in \mathbb{R}^d \times \mathbb{R} : 1 \leq i \leq n$
X are i.i.d.'s and $Y(X)$.

Model: $\mathbf{Y} = \beta_0 + \sum_{j=1}^d \mathbf{X_j}\beta_j$, $\mathbf{Y} \subset \mathbb{R}$

Introduce $X_0 = 1$ and rewrite

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

additive Gaussian noise $\epsilon \sim \mathcal{N}(0,\sigma^2)$

$\hat{y} = \mathbf{X}\hat{\beta} + \epsilon$

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

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

(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

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

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

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

(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

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

(3) Lasso (L^1 penalty)

Setting: Penalize full β.

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

$= (\mathbf{y}-\mathbf{X}\beta)^T(\mathbf{y}-\mathbf{X}\beta) + \lambda||\beta||_1$

Lasso has no closed form.

(4) Bayesian Linear Regression

Setting: Define a prior over β.

e.g. Ridge: Assume β distributed as:

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

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

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

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

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

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

Nonlinear Regression

Idea: Feature space transformation

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

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

Gaussian Process Regression

joint Gaussian over all outputs

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

We can rewrite the distribution

$P(\begin{bmatrix}\mathbf{y}\\ y_*\end{bmatrix}){=}\mathcal{N}(\mathbf{y}|\mathbf{0}, \begin{bmatrix}\mathbf{C_n} & \mathbf{k}\\ \mathbf{k^T} & c\end{bmatrix})$

Such that for prediction:

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

$\mu_{y_*} = \mathbf{k}^T\mathbf{C}_n^{-1}\mathbf{y} \qquad \mathbf{C}_n = \mathbf{K} + \sigma^2\mathbb{I}$

$\sigma_*^2{=}c{-}\mathbf{k}^T\mathbf{C}_n^{-1}\mathbf{k} \qquad c{=}k(x_*,x_*){+}\sigma^2$

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

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

Gaussian Process Prediction

We then predict new values from:

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

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

$\sigma^2_{y_*} = c - \mathbf{k}^T\mathbf{C}_n^{-1}\mathbf{k}$.

Bias-Variance tradeoff

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

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

$|\mathcal{Z}|\downarrow \quad |\mathcal{F}|\uparrow \quad \Rightarrow \quad \mathrm{Var}\uparrow \quad \mathrm{Bias}\downarrow$

$|\mathcal{Z}|\uparrow \quad |\mathcal{F}|\downarrow \quad \Rightarrow \quad \mathrm{Var}\downarrow \quad \mathrm{Bias}\uparrow$

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^2)

$+\mathbb{E}_X\mathbb{E}_D[(\hat{f}_D(X)-\mathbb{E}_D[\hat{f}(X)])^2]$ (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.

Numerical Est. Techinques

Setting: Estimate $\hat{f}(x) \in \mathcal{F}$ with minimal prediction error.

K-Fold Cross Validation

Initialisation (split training set):

$\mathcal{Z} = \mathcal{Z}_1 \bigcup \mathcal{Z}_2 \bigcup \cdots \bigcup \mathcal{Z}_K, \mathcal{Z}_\mu \bigcap \mathcal{Z}_\nu = \emptyset$

$|\mathcal{Z}_k| \approx n\frac{K-1}{K}$ # of samples

Learning:

$\hat{f}^{-\nu}(x){=}\arg\min_{f\in\mathcal{F}} \frac{\sum_{i\notin\mathcal{Z}_\nu}(y_i-f(x_i))^2}{|\mathcal{Z}-\mathcal{Z}_\nu|}$

Validation:

$\hat{R}^{cv} = \frac{1}{n}\sum_{i\leq n}(y_i - \hat{f}^{-\kappa(i)}(x_i))^2$

Underfits because smaller dataset.

Leave-one-out: $K = n$ (unbiased but var can be large from corr. datasets)

Bootstrapping

Bootstrap samples: $\mathcal{Z}^* = \{\mathcal{Z}_1^*,\cdots\mathcal{Z}_B^*\}$, of same size as original, drawn with replacement. The chance of a sample to have appeared in the bootstrap is:

$1-(1-\frac{1}{n})^n \xrightarrow{n\to\infty} 1-\frac{1}{e} \approx 0.632$. So if we compute the ERM on $\mathcal{Z}$ we could get 63% accuracy by memorization. Over-confident!

Leave-one-out: compensates by computing the ERM where no memorization was for specific sample. E.g., for classification, like cross-validation:

$\hat{\mathcal{R}}(S(\mathcal{Z})) = \frac{1}{B}\sum_{b=1}^B\sum_{z_i\notin\mathcal{Z}^{*b}}\frac{\mathbb{I}_{c(x_i)\neq y_i}}{B-|\mathcal{Z}^{*b}|}$

Jackknife

Estimate of an estimator $\hat{S}_n$'s Bias.

$\hat{S}^{JK} = \hat{S}_n - \mathrm{bias}^{JK}$ is JK Estimator.

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

$\tilde{S}_n{=}\frac{1}{n}\sum_{i=1}^n\hat{S}^{-i}_{n-1}$ avg. LOO Estimator.

Debiased est. can have big variance!

Bootstrap debiased

$\overline{S}{=}2\hat{S}{-}\frac{1}{B}\sum_b\hat{S}^*(b)$

Model selection

Bayesian Information Criterion (BIC): $-2\log\mathbb{P}(\mathcal{X}|\theta_k) + k\log(n)$. Penalizes larger models (larger k with more data n). Tendency to underfit.

Akaike Information Criterion (AIC): $-2\log\mathbb{P}(\mathcal{X}|\theta_k) + 2k$. Tendency to overfit.

Takeuchi Information Criterion (TIC): $-2\log\mathbb{P}(\mathcal{X}|\theta_k) + 2\mathrm{Tr}(\mathcal{I}\mathcal{J}^{-1})$.

Classification

Linear Classifier

$g(x) = a^T\tilde{x} \quad a = (w_0,w)^T, \tilde{x} = (1,x)^T$

$a^T\tilde{x}_i > 0 \Rightarrow y_i = 1, a^T\tilde{x}_i < 0 \Rightarrow y_i = 2$

Normalization: $\tilde{x}_i \to -\tilde{x}_i$ if $y_i = 2$

Find a: $a^T\tilde{x}_i > 0, \forall_i$!

Learning w. Gradient Descent:

$a(k+1) = a(k) - \eta(k)\nabla J(a(k))$

$J(.)$: cost function $\eta(.)$: learning rate

Newton's rule (opt. grad descent):

$a(k+1) = a(k) - H^{-1}\nabla J, H = \frac{\partial^2 J}{\partial a_i\partial a_j}$

Taylor: $f(x) = f(a) + (x-a)f'(a) + \ldots$

Perceptron Criterion

$J_P(a) = \sum_{\tilde{x}\in\mathcal{X}^{\mathrm{msc}}}(-a^T\tilde{x})$,
$\mathcal{X}^{\mathrm{msc}}$: set of misclassified samples.

$\Rightarrow a(k+1) = a(k) + \eta(k)\sum_{\tilde{x}\in\mathcal{X}^{\mathrm{msc}}}\tilde{x}$

Converges if data separable.

Single sample perceptron:

$a(k+1) = a(k) + \tilde{x}^k$ (misclassified).

WINNOW Algorithm

Performs better when many dimensions are irrelevant. Search for 2 weight vectors a^+, a^- (for each class). If a point is misclassified:

$a_i^+{\leftarrow}\alpha^{+\tilde{x}_i}a_i^+, a_i^-{\leftarrow}\alpha^{-\tilde{x}_i}a_i^-$ (class 1 err.)

$a_i^+{\leftarrow}\alpha^{-\tilde{x}_i}a_i^+, a_i^-{\leftarrow}\alpha^{+\tilde{x}_i}a_i^-$ (class 2 err.)

Exponential update.

Fisher's Linear Discriminant Analysis

Maximize distance of the means of the projected classes to find projection plane separating them best.

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

Dist of proj means: $|w^T(\mu_1-\mu_2)|$ Classes proj. cov: $\tilde{\Sigma}_1{+}\tilde{\Sigma}_2{=}w^T(\Sigma_1{+}\Sigma_2)w$

Fishers Criterion:

$J(w) = \frac{||\mu_1-\mu_2||^2}{\tilde{\Sigma}_1+\tilde{\Sigma}_2} = \frac{w^T(\mu_1-\mu_2)(\mu_1-\mu_2)^Tw}{w^T(\Sigma_1+\Sigma_2)w}$

Fishers Crit for Multiple Classes:

$J(W) = \frac{|W^TS_BW|}{W^TS_WW}$

$S_B = \sum_{i=1}^k n_k(\mu_k-\mu)(\mu_k-\mu)^T$

$S_W = \sum_{i=1}^k\sum_{x\in\mathcal{D}_i}(x-\mu_i)(x-\mu_i)^T$

LDA for Multiclasses: Reformulate as $(k-1)$ "class α - not class α" dichotomies. But some area are ambiguous

Support Vector Machine (SVM)

Generalize Perceptron with margin and kernel. Find plane that maximi-

zes margin m s.t.:
$z_i g(\mathbf{y}) = z_i(\mathbf{w}^T\mathbf{y}+w_0) \geq m, \forall \mathbf{y}_i \in \mathcal{Y}$
$z_i \in \{-1,+1\} \quad \mathbf{y_i} = \phi(\mathbf{x_i})$
Vectors $\mathbf{y}_i$ are the support vectors
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 αs 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 $L(\mathbf{w}, w_0, \alpha)$ we get
$\tilde{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j z_i z_j \mathbf{y_i}^T\mathbf{y_j}$
with $\alpha_i \geq 0$ and $\sum_{i=1}^n \alpha_i z_i = 0$
This is the dual representation. The optimal hyperplane is given by
$\mathbf{w}^* = \sum_{i=1}^n \alpha_i^* z_i \mathbf{y_i}$
$w_0^* = -\frac{1}{2}(\min_{z_i=1} \mathbf{w^*}^T\mathbf{y_i} + \max_{z_i=-1} \mathbf{w^*}^T\mathbf{y_i})$
where α maximize the dual problem.
Only Support Vectors ($\alpha_i \neq 0$) contribute to the evaluation.
Optimal Margin: $\mathbf{w}^T\mathbf{w} = \sum_{i \in SV} \alpha_i^*$
Discrim.: $g^*(\mathbf{x}) = \sum_{i \in SV} z_i \alpha_i \mathbf{y_i}^T\mathbf{y_i} + w_0^*$
$\text{class} = \text{sign}(\mathbf{y}^T\mathbf{w}^* + w_0^*)$

Soft Margin SVM

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

Non-Linear SVM

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

Multiclass SVM

$\forall$class $z \in \{1, 2, \cdots, M\}$ we introduce $\mathbf{w}_z$ and define the margin m s.t.:
$(\mathbf{w}_{z_i}^T\mathbf{y}_i + w_{z_i,0}) - \max_{z \neq Z_i}(\mathbf{w}_z^T\mathbf{y}_i + w_{z,0}) \geq m, \forall_{\mathbf{y}_i \in \mathcal{Y}}$

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: $\hat{z} = h(\mathbf{y}) \arg\max_{z \in \mathcal{K}} f_{\mathbf{w}(z,\mathbf{y})}$

Kernels

Similarity based reasoning
Gram Matrix $K = K(\mathbf{x}_i, \mathbf{x}_i), 1 \leq i, j \leq n$
$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}') \quad K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$
$K(\mathbf{x}, \mathbf{x}')$ pos.semi-def. (all EV ≥ 0)
If K_1 & K_2 are kernels K is too:
$K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}')K_2(\mathbf{x}, \mathbf{x}')$
$K(\mathbf{x}, \mathbf{x}') = \alpha K_1(\mathbf{x}, \mathbf{x}') + \beta K_2(\mathbf{x}, \mathbf{x}')$
$K(\mathbf{x}, \mathbf{x}') = K_1(h(\mathbf{x}), h(\mathbf{x}')) \quad h: \mathcal{X} \to \mathcal{X}$
$K(\mathbf{x}, \mathbf{x}') = h(K_1(\mathbf{x}, \mathbf{x}'))$ $\quad h$: poly/exp
Kernel Function Examples:
$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}' \quad K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + 1)^p$
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 eg: $\mathbf{x} = [1, -1], \mathbf{x}' = [-1, 2]$

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 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 unaffected.

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 $w^{(b)}$
$\epsilon_b = \sum_{i=1}^n w_i^{(b)} \mathbb{I}_{\{c_b(x_i) \neq y_i\}} / \sum_{i=1}^n w_i^{(b)}$
$\alpha_b = \log\frac{1-\epsilon_b}{\epsilon_b} > 0$
$w_i^{(b+1)} = w_i^{(b)} \exp(\alpha_i \mathbb{I}_{\{c_b(\hat{x_i}) \neq y_i\}})$
return $\hat{c}_B(x) = \text{sgn}(\sum_{b=1}^B \alpha_b c_b(x))$
Best approx. at log-odds ratio. Like stagewise-additive modeling.

Difference

(1) Boosting keeps identical training data, bagging potentially varies the training data for each classifier. (2) Boosting weighs the prediction of each classifier according 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 datasets 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

Function of interest

The probability of large excess error:
$\mathbb{P}[\hat{c}_N(X) \neq c^{\text{Bayes}}(X)|\mathcal{Z}] < \delta$.
But: could be unlucky with $\mathcal{Z}$, c^{Bayes} not in hypoth. class, $\mathbb{P}[\dots]$ is a r.v., not scalar quality measure. Strat: average. $\mathbb{P}[\mathcal{R}(\hat{c}) - \inf_{c \in \mathcal{C}} \mathcal{R}(c) > \epsilon] < \delta$.
Where $\mathcal{R}[\hat{c}]$ is the generalization error of the trained class., should not exceed the min. generalization error achievable by more than ϵ. Leads to:
$\sum_{c \in \mathcal{C}} \mathbb{P}[|\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)| > \epsilon] \leq 2Ne^{-2n\epsilon^2}$.
Def RHS= δ: $\epsilon = \sqrt{\frac{\log N - \log(\delta/2)}{2n}}$. For N the size of hypothesis class, and for n the num. of samples. The expected error of c thus depends on $1/\sqrt{n}$ and $\log N$!

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. $1/\epsilon$ (error param.) and $1/\delta$ (confidence val.).

Hyperplane learning

Hypothesis: $\sum_{i=1}^d a_i x_i + a_0$ (all possible hyperplanes through d-dim vector) has #-of-possible-classifiers $2\binom{n}{d}$.
In class: the classifiers c and $\hat{c}$ differ for no more than d data points on a plane, IF found with ERM: $\forall_{c \in \mathcal{C}} \hat{\mathcal{R}}_n(c) \geq \hat{\mathcal{R}}_n(\hat{c}) - \frac{d}{n}$.

VC dimension

How effectively a classifier (e.g., interval, unions of intervals, circles) can select subsets. Classifying with a closed interval: $V_C = 2$, for you cannot select the begin-point and end-point but not middle point! For unions, $V_c = 2k$, for unit circles $V_c = 3$, for finite-dimensional vector space $V_c \leq \dim(\mathcal{G})$.

From practical

$\mathcal{R}_n(\hat{c}) - \inf_{c \in \mathcal{C}} \mathcal{R}(c) \leq 2\sup_{c \in \mathcal{C}} |\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)|$ for any class $\mathcal{C}$. Finite class: $\mathbb{P}[2\sup_{c \in \mathcal{C}} |\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)| > \epsilon] \leq 2|\mathcal{C}|e^{-2n\epsilon^2}$. Shattering $s(\mathcal{A}, n)$ is the score... For half-planes $s(\mathcal{A}, 2) = 4$, $s(\mathcal{A}, 3) = 8$, ... For triangles $s(\mathcal{A}, 2) = 4, s(\mathcal{A}, 3) = 8, s(\mathcal{A}, 4) = 16$, powers.

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 from $\text{Beta}(x|1, \alpha)$ with fixed α, but from reducing stick: $\rho_k = \beta_k(1 - \sum_{i=1}^{k-1} \rho_i)$. The prior:
$\mathbb{P}[z_i = k|z_{-i}, \alpha] = \begin{cases} \frac{N_{k,-i}}{\alpha+N-1} & \text{exist. } K \\ \frac{\alpha}{\alpha+N-1} & \text{oth.} \end{cases}$.
Final Gibbs sampler:
$\mathbb{P}[z_i = k|z_{-i}, \alpha, \mu] = \begin{cases} \frac{N_{k,-i}}{\alpha+N-1} p(x_i|x_{-i,k}, \mu) & \text{exist. } K \\ \frac{\alpha}{\alpha+N-1} p(x_i, \mu) & \text{oth.} \end{cases}$

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

Regularizazion

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.

Mixture Models

Gaussian Mixture

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

E-Step

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

M-Step

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

Extra

$\mathcal{N}(\mu_1, \sigma_1^2) + \mathcal{N}(\mu_2, \sigma_2^2) = \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$.