Background

Linear Algebra

$\|\mathbf{x}\|_p = (\sum |x_i|^p)^{1/p}$ $\quad \|\mathbf{x}\|_\infty = \max |x_i|$

$\mathrm{tr}(\mathbf{A}\mathbf{x}\mathbf{x}^T) = \mathbf{x}^T\mathbf{A}\mathbf{x}$

$|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}|$ $\quad |\mathbf{A}^m| = |\mathbf{A}|^m$

$(\mathbf{A}+\mathbf{UCV})^{-}=\mathbf{A}^{-}-\mathbf{A}^{-}\mathbf{U}(\mathbf{C}^{-}+\mathbf{V}\mathbf{A}^{-}\mathbf{U})^{-}\mathbf{V}\mathbf{A}^{-}$

$(\mathbf{A}+\mathbf{B})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}(\mathbf{A}+\mathbf{B})^{-1}\mathbf{A}^{-1}$

$\mathbf{U}(\mathbf{VU}+\mathbf{I})^{-1} = (\mathbf{UV}+\mathbf{I})^{-1}\mathbf{U}$

$\mathbf{I} - \mathbf{A}(\mathbf{I}+\mathbf{A})^{-1} = (\mathbf{I}+\mathbf{A})^{-1}$

Probability

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

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

$p_Y(y) = p_X(g^{-1}(y)) \left|\det \frac{\partial g^{-1}(y)}{\partial y}\right|$

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

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

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

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

$\mathrm{Var}(X){=}\mathbb{E}[\mathrm{Var}(X \mid Y)]{+}\mathrm{Var}(\mathbb{E}[X \mid Y])$

$\mathbb{V}[X{+}Y]{=}\mathrm{Var}[X]{+}\mathrm{Var}[Y]{+}2\mathrm{Cov}(X,Y)$

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

$\mathrm{Cov}(A\mathbf{X}{+}c, B\mathbf{Y}{+}d) = A\mathrm{Cov}(\mathbf{X},\mathbf{Y})B^T$

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

$X = \Sigma^{1/2}\mathcal{N}(0,1) + \mu \sim \mathcal{N}(\mu,\Sigma)$

$Y = MX{+}b \sim \mathcal{N}(M\mu{+}b, M\Sigma M^T)$

$X,Y \overset{\text{iid}}{\sim} \mathcal{N}: \quad X{+}Y \sim \mathcal{N}(\mu{+}\mu', \Sigma{+}\Sigma')$

Conditional Gaussian

$P(\begin{bmatrix}\mathbf{X}\\ \mathbf{Y}\end{bmatrix}){=}\mathcal{N}(\begin{bmatrix}\mathbf{X}\\ \mathbf{Y}\end{bmatrix}; \begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix})$

$p(\mathbf{Y}|\mathbf{X}=\mathbf{x}) = \mathcal{N}(\mu,\Sigma)$

$\cdot\ \mu = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{x}-\mu_{\mathbf{X}})$

$\cdot\ \Sigma = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$

Inequalities and Estimators

Jensen: $\quad log(\sum_i \lambda_i^{(\ge 0)} x_i) \ge \sum_i \lambda_i log(x_i)$

Chebyshev: $\mathbb{P}(|\hat{X}-X| \ge \varepsilon) \le \frac{MSE[\hat{X}]}{\varepsilon^2}$

Estimators: *Unbiased:* $\mathbb{E}[\hat{\theta}] = \theta^\star$

Consistent: $\mathbb{P}(|\hat{\theta}-\theta^\star| < \varepsilon) \to 0$ convP

Asymp Normal: $(\hat{\theta}-\theta^\star)\hat{se}^{-1} \sim \mathcal{N}(0,1)$

Rao-Cra.: $\mathbb{E}_{x|\theta}[(\theta-\hat{\theta})^2] \ge \frac{(\frac{\partial}{\partial\theta}b_{\hat{\theta}}+1)^2}{\mathbb{E}_{x|\theta}[\Lambda^2]} + b_{\hat{\theta}}^2$

$b_{\hat{\theta}} = \mathbb{E}_{x|\theta}[\hat{\theta}] - \theta \quad\quad \Lambda = \frac{\partial}{\partial\theta}\log p(x|\theta)$

$\mathbb{E}_{x|\theta}[\Lambda] = 0 \to \mathbb{E}_{x|\theta}[\Lambda\hat{\theta}] = \frac{\partial}{\partial\theta}b_{\hat{\theta}} + 1$

$\to \mathrm{Cov}(\Lambda,\hat{\theta}) \to$ Cauchy

$\mathrm{Var}[\hat{\theta}] \ge \mathcal{I}_n(\theta)^{-1} = -\mathbb{E}[\frac{\partial^2 \log p[\mathcal{X}_n|\theta]}{\partial\theta^2}]^{-1}$

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

$\hat{\theta}_{JS} = \left(1 - \frac{(d-2)\sigma^2}{\|y\|^2}\right) y$

Derivatives

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

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

$\frac{\partial}{\partial\mathbf{X}}(\mathbf{c}^\top\mathbf{X}\mathbf{b}){=}\mathbf{c}\mathbf{b}^\top \quad\quad \frac{\partial}{\partial\mathbf{X}}(\|\mathbf{X}\|_F^2){=}2\mathbf{X}$

$\frac{\partial}{\partial\mathbf{x}}\|\mathbf{x}\|_2{=}\frac{\mathbf{x}}{\|\mathbf{x}\|_2}$ | $\frac{\partial}{\partial\mathbf{x}}||f(\mathbf{x})||_1{=}\frac{\partial f(\mathbf{x})}{\partial\mathbf{x}}^T \mathrm{sgn}(\mathbf{x})$

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

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

$\frac{\partial}{\partial\mathbf{X}}f(\mathbf{X})^\top{=}\frac{\partial f(\mathbf{X})}{\partial\mathbf{X}}^T$ | $\frac{\partial}{\partial\mathbf{X}}\mathrm{tr}\,f(\mathbf{X}){=}\mathrm{tr}\,\frac{\partial f(\mathbf{X})}{\partial\mathbf{X}}$

$\frac{\partial}{\partial\mathbf{X}}\det f(\mathbf{X}){=}\det f(\mathbf{X})\,\mathrm{tr}(f(\mathbf{X})^{-1}\frac{\partial f(\mathbf{X})}{\partial\mathbf{X}})$

$\frac{\partial}{\partial\mathbf{X}}f(\mathbf{X})^{-1} = -f(\mathbf{X})^{-1}\frac{\partial f(\mathbf{X})}{\partial\mathbf{X}}f(\mathbf{X})^{-1}$

Quadratic Forms

$\mathbf{x}^TA\mathbf{x} + 2\mathbf{b}^T\mathbf{x} + c = (\mathbf{x}+A^{-1}\mathbf{b})^TA$

$(\mathbf{x}+A^{-1}\mathbf{b}) - \mathbf{b}^TA^{-1}\mathbf{b} + c,$

$ax^2+bx+c = (x+\frac{b}{2a})^2 - (\frac{b}{2a})^2 + c$

Information Theory

$\mathrm{H}[p] = \mathbb{E}_{\mathbf{x}\sim p}[-\log p(\mathbf{x})]$

$\mathrm{H}[p\|q] = \mathbb{E}_{\mathbf{x}\sim p}[-\log q(\mathbf{x})]$

$\mathrm{KL}[p\|q] = \mathrm{H}[p\|q] - \mathrm{H}[p]$

$\mathrm{KL}[p\|q] = \mathbb{E}_{\theta\sim p}\left[\log\left(\frac{p(\theta)}{q(\theta)}\right)\right]$

$\mathrm{KL}[p\|q] \ne \mathrm{KL}[q\|p] \ge 0$

$\mathrm{H}[\mathbf{X}] = \mathbb{E}_{\mathbf{X}\sim p}[-\log p(\mathbf{X})]$

$\mathrm{H}[\mathbf{X}|\mathbf{Y}=y] = \mathbb{E}_{\mathbf{X}\sim p(\cdot|y)}[-\log p(\mathbf{X}|y)]$

$\mathrm{H}[\mathbf{X}|\mathbf{Y}] = \mathbb{E}_y[\mathrm{H}[\mathbf{X}|\mathbf{Y}=y]]$

$\mathrm{H}[\mathbf{X}|\mathbf{Y}] = \mathrm{H}[\mathbf{Y}|\mathbf{X}]{+}\mathrm{H}[\mathbf{X}]{-}\mathrm{H}[\mathbf{Y}]$

$\mathrm{H}[\mathbf{X},\mathbf{Y}]{=}\mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim p(\cdot,\cdot)}[-\log p(\mathbf{X},\mathbf{Y})]$

$\mathrm{I}[\mathbf{X};\mathbf{Y}] = \mathrm{H}[\mathbf{X}] - \mathrm{H}[\mathbf{X}|\mathbf{Y}] \ge 0$

$\mathrm{I}[\mathbf{X};\mathbf{Y}|\mathbf{Z}] = \mathrm{I}[\mathbf{X};\mathbf{Y},\mathbf{Z}] - \mathrm{I}[\mathbf{X};\mathbf{Z}]$

$\mathrm{H}(\mathcal{N}(\mu,\Sigma) = \frac{1}{2}\ln(\det(2\pi e\,\Sigma))$

$\mathrm{KL}(\mathcal{N}(a,A)||\mathcal{N}(b,B)) = \frac{1}{2}(\mathrm{tr}(B^{-1}A) + (a-b)^TB^{-1}(a-b) - d + \ln(\frac{\det B}{\det A}))$

Risks

Expected Risk: $R(f) = P(f(X) \ne y)$

$\mathcal{R}(f) = \sum_{y\le k} P(y)\mathbb{E}_{P(x|y)}[1_{f(x)\ne y}|Y=y]$

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

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

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

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

Loss Fcts: $\quad \mathcal{L}(y,z) \quad z{=}w^\top x$

$\mathcal{L}^{0/i} = \mathbb{I}[\mathrm{sign}(z) \ne y]$

$\mathcal{L}^{\mathrm{hinge}} = \max(0, 1-yz)$ for SVM's

$\mathcal{L}^{\mathrm{percep}} = \max(0, -yz)$

$\mathcal{L}^{\mathrm{logistic}} = \log(1+\exp(-yz))$

$\mathcal{L}^{\exp} = \exp(-yz)$ for AdaBoost

$\mathcal{L}^{\mathrm{CE}} = -[y'\log z' + (1-y')\log(1-z')]$

$y' = \frac{1+y}{2}, \quad z' = \frac{1+z}{2}$

Optimization

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

GD: $\quad \theta^{(t+1)}{\leftarrow}\theta^{(t)}{-}\eta\nabla_\theta\mathcal{L}$

SGD: $\theta^{(t+1)}{\leftarrow}\theta^{(t)}{-}\eta\nabla\mathcal{L}(\theta^{(t)}, x_i, y_i)$

NGD: $\theta^{(t+1)}{\leftarrow}\theta^{(t)}{-}\eta(\nabla_\theta^2\mathcal{L})^{-1}\nabla_\theta\mathcal{L}$

$\to f(x{+}t){\approx}f(x){+}tf'(x){+}\frac{1}{2}f''(x)t^2{=}0$

Parametric Density Estimation

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

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

$\hat{\theta}_{MLE}{=}\arg\max_\theta \mathbb{P}[\mathcal{X}|\theta]$

$\hat{\theta}_{MAP}{=}\arg\max_\theta[P(\theta|\mathcal{X}){=}P(\mathcal{X}|\theta)P(\theta)]$

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

1-D Gaussian Bayesian learning

$X|\theta \sim \mathcal{N}(\theta,\sigma^2) \quad\quad \theta \sim \mathcal{N}(m_0, s_0^2)$

$\theta|X{\sim}\mathcal{N}(\mu_n, \sigma_n^2)$

$\sigma_n^2 = \frac{\sigma^2 s_0^2}{ns_0^2+\sigma^2}, \quad \mu_n = \frac{ns_0^2\bar{x}+m_0\sigma^2}{ns_0^2+\sigma^2}$

Recursive Bayesian density learning

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

Frequentist vs Bayesian

Bayes: priors, distributions, needs efficient integration, adds regularization term. Frequentist: no priors, point estimate, requires only differentiation methods. MLE are consistent, equivariant, asymptotically normal, asymptotically efficient (no efficient for finite samples).

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 data: $X{:}O_1{\times}O_2{\times}O_3{\to}\mathbb{R}^d$ nominal = qualitative (sweet, sour ...), ordinal = absolute order, quantitative = numbers

Regression

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

$\mathbf{X}{\in}\mathbb{R}^{(d+1)\times n} \quad \beta{\in}\mathbb{R}^{d+1} \quad \varepsilon{\sim}\mathcal{N}(\mathbf{0}, \mathbb{I}\sigma^2)$

$\mathbf{Y}|\mathbf{X},\beta,\sigma^2 \sim \mathcal{N}(\mathbf{Y}; \mathbf{X}^T\beta, \mathbb{I}_{(d+1)}\sigma^2)$

MLE: Ordinary Least Squares

OLSE is unbiased, orthogonal projection with lowest variance. differentiate wrt β.

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

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

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

MAP: Ridge Regression (L^2 penalty)

Penalize energy in β. *Prior:* $\beta{\sim}\mathcal{N}(0, \lambda^{-}\mathbb{I})$

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

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

MAP: Lasso (L^1 penalty)

Penalize full β. Lasso has no closed form.

$\beta \sim Lapl(0, \lambda^{-1}) = \frac{\lambda}{2}exp(-\lambda|\beta|)$

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

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

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

d-Dim Bayesian Linear Regression

Prior: $\beta \sim \mathcal{N}(\mu_0, \Lambda^{-1})$

Likelihood: $Y|\beta,X,\sigma \sim \mathcal{N}(X\beta, \sigma_n^2\mathbb{I})$

Posterior: $\beta|\mathbf{X},\mathbf{y} \sim \mathcal{N}(\mu, \Sigma)$

$\cdot\ \Sigma = (\sigma_n^{-2}\mathbf{X}^T\mathbf{X} + \Lambda)^{-1}$

$\cdot\ \mu = \Sigma(\Lambda\mu_0 + \sigma_n^{-2}\mathbf{X}^T\mathbf{y})$

Nonlinear Regression

Idea: Add feature space transformation, kernel to compute inner product. Suppose:

$\beta \sim \mathcal{N}(\mathbf{0}, \Lambda^{-1}) \quad\quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma_n^2\mathbb{I}_d)$

$\mathbf{Y}{=}\mathbf{X}\beta{+}\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{X}\Lambda^{-1}\mathbf{X}^T + \sigma_n^2\mathbb{I}_d)$

Kernels

Kernel: $k(x_i,x_j) = \phi(x_i)\Lambda^{-1}\phi(x_j)^T$

Similarity based reasoning.

Gram Matrix: $K = k(\mathbf{x}_i,\mathbf{x}_j), \quad 1{\le}i, j{\le}n$

$\cdot k(\mathbf{x},\mathbf{x}'){=}k(\mathbf{x}',\mathbf{x}) \ \cdot k(\mathbf{x},\mathbf{x}')$ pos.semi-def.

If k_1, k_2 kernels, $c \in \mathbb{R}_{>0}$, $\mathbf{A}^{psd}$, $p_{\text{pos-coeff}}$:

$k(\mathbf{x},\mathbf{x}') = k_1(\mathbf{x},\mathbf{x}'){+}k_2(\mathbf{x},\mathbf{x}') = \mathbf{x}^T\mathbf{A}\mathbf{x}'$

$= k_1(\mathbf{x},\mathbf{x}'){\cdot}k_2(\mathbf{x},\mathbf{x}') = c{\cdot}k_1(\mathbf{x},\mathbf{x}')$

$= p(k_1(\mathbf{x},\mathbf{x}')) = f(\mathbf{x})k_1(\mathbf{x},\mathbf{x}')f(\mathbf{x}')$

$k(\mathbf{x},\mathbf{x}') = \phi(x)^T\phi(x') = (1+\mathbf{x}^T\mathbf{x}')^m$

$= \tanh(\alpha\mathbf{x}^T\mathbf{x}' + c)$

$= \sigma^2\exp(-\frac{2\sin(p^{-1}\pi||\mathbf{x}-\mathbf{x}'||_2^2)}{l^2})$

$= \exp(-||\mathbf{x}{-}\mathbf{x}'||_1\, l^{-1})$

$= \exp(-||\mathbf{x}{-}\mathbf{x}'||_2^2\,(2l^2)^{-1})$

RBF: $\phi_j(x){=}\exp(-\frac{||x||_2^2}{2})\prod_{i=0}^d x^{j_i}(j_i!)^{-\frac{1}{2}}$

$\uparrow$ Lengthscale, smoother fcts.

Gaussian Process Regression

Applying a kernel, we get:

$\mathbf{Y}{=}\Phi\beta{+}\varepsilon \sim \mathcal{N}(\mathbf{0}, \Phi\Lambda^{-1}\Phi^T{+}\sigma_n^2\mathbb{I}_d) =$

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

Gaussian Process Prediction

Given $\mathcal{GP}(\mu, K)$,

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

$\cdot\ \tilde{\mu} = \mu(x_*){+}\mathbf{k}^T(\mathbf{K}+\sigma_n^2\mathbb{I})^{-1}(\mathbf{y}{-}\mu(\mathbf{X})),$

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

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

$\cdot\ \tilde{\sigma}_{ij}^2 = k(x_i,x_j) - \mathbf{k}_i^T(\mathbf{K}+\sigma^2\mathbb{I})^{-1}\mathbf{k}_j$

Causality

Regression models capture correlation (not causality). ie, non-causal features can mislead models (*Spurious Correlations*). Iff Train Test Distribution change (*Domain Shift*). *Counterfactual Invariance:* A function f is invariant if $f(X(w)) = f(X(w'))$ for any w, w', reducing bias from spurious correlations. **Confounding:** A hidden variable influences both W and X, creating a spurious correlation with Y. **Selection Bias:** A hidden variable S filters the training data based on W and X, inducing non-causal associations.

If f is a counterfactually invariant predictor:

In the **anti-causal scenario**: $f(X) \perp W|Y$.

In the **causal scenario** (no selection but confounded): $f(X) \perp W$. In the **causal scenario** (no confounding but selected): $Y \perp X \mid W, X_{\perp W}$ and $f(X) \perp W \mid Y$.

A set of variables Z **d-separates** X and Y in a DAG $\mathcal{G}$ if all paths between X and Y are blocked by Z: $X \perp Y|Z$. A path is blocked if: **Collider:** $A{\to}B{\leftarrow}C$ and neither B nor its descendants are in Z. **Chain:** $A{\to}B{\to}C$. **Fork:** $A{\leftarrow}B{\to}C$ where $B \in Z$.

Algos

K-Means $J{=}\sum_{x\in\mathcal{X}}\|x - \mu_{c(x)}\|^2$

PCA proj. maximum variance subspace. top d eigenv. of $S{=}\frac{1}{n}\sum_{i=1}^n(x_i{-}\overline{X})(x_i{-}\overline{X})^T$

EM fit GMMs $(\sum_{k=1}^K \pi_k\mathcal{N}(x|\mu_k,\Sigma_k))$ by max. likelihood. Reaches local optimum.

Latent variable: $M_{xc} = 1\{c \text{ generated } x\}$

$P(\mathcal{X},M|\theta){=}\prod_x\prod_{c=1}^k(\pi_c P(x|\theta_c))^{M_{xc}}$

$\gamma_{xc}{=}\mathbb{E}[M_{xc}|\mathcal{X},\theta^{(j)}]{=}\frac{\pi_c\mathcal{N}(\mathbf{x};\mu_c,\Sigma_c)}{\sum_{j=1}^K \pi_j\mathcal{N}(\mathbf{x};\mu_j,\Sigma_j)}$

$\mu_c^{(j+1)}{=}\frac{\sum_{c\in\mathcal{X}}\gamma_{xc}x}{\sum_{c\in\mathcal{X}}\gamma_{xc}} \quad\quad \pi_c^{(j+1)}{=}\frac{1}{|\mathcal{X}|}\sum_{c\in\mathcal{X}}\gamma_{xc}$

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

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.

p-value

p-value$=\inf\{\alpha : T(X^n){\in}\{x|T(x){\ge}c\}\}$ likelihood to accept H_0. it is the least probable threshold for rejecting the H_0.

Statistical Learning and Validation

Find $f: X \to Y$ to minimize expected risk by approximation with empirical risk.

K-Fold Cross Validation

Partition data Z into K equally sized, disjoint subsets:
$\mathcal{Z} = \mathcal{Z}_1 \bigcup \mathcal{Z}_2 \bigcup \cdots \bigcup \mathcal{Z}_K$, $\mathcal{Z}_\mu \bigcap \mathcal{Z}_\nu = \emptyset$
$|\mathcal{Z}_k| \approx n\frac{K-1}{K}$ # of training samples. Learn:
$\hat{f}^{-\nu}(x) = \arg\min_{f\in\mathcal{F}} \frac{\sum_{i\in \mathcal{Z}_\nu} \mathcal{L}(y_i, f(x_i))}{|\mathcal{Z}-\mathcal{Z}_\nu|}$
$\hat{R}^{cv}(\mathcal{A}) = \frac{1}{n}\sum_{i\le n} \mathcal{L}(y_i, \hat{f}^{-\kappa(i)}(x_i))$
Underfits because smaller dataset.
Leave-one-out: $K = n$ (unbiased but var can be large from correlated 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 (shows too small bias)!
Leave-one-out/out of bucket error: compensates by computing the ERM where no memorization was for specific sample. E.g., for classification, like cross-validation:
$\hat{\mathcal{R}}(\mathcal{A}) = \frac{1}{B}\sum_{b=1}^B \sum_{z_i \notin \mathcal{Z}^{*b}} \frac{\mathbb{I}_{c(x_i)\neq y_i}}{B-|\mathcal{Z}^{*b}|}$ $\hat{R}_{0.632} = 0.368\hat{R}(A(Z)) + 0.632\hat{R}_{bs}$
Wald Test: $W = \frac{\hat\theta-\theta_0}{\text{s.e.}(\hat\theta)}$

Bayesian Neural Networks (BNN)

NN: no uncertainty quantification, overconfident, adversarial examples, poor generalization for domain shifts. BNN: Using $p(w)$ and $p(D|w)$, approx. poster. by variational infer. (min rev KL).
$\sigma \leftarrow \sigma - \alpha_t \left(\epsilon^\top \frac{\partial}{\partial w}F(w,\theta) + \frac{\partial}{\partial\sigma}F(w,\theta)\right)$

Information-based Transductive Lear.

ITL selects x_n that maximizes mutual information of $y_x = f_x + \epsilon_x$ about f:
$x_n = \arg\max_{x\in S} I(f_A; y_x | D_{n-1})$
If $f \sim \text{GP}(\mu, k)$, then:
$I(f_A; y_x|D_{n-1}) = \frac12 \log\left(\frac{\text{Var}[y_x|D_{n-1}]}{\text{Var}[y_x|f_A, D_{n-1}]}\right)$

Safe Bayesian Optimization

$x_n = \arg\max_{x\in\hat{S}_n = \{x | u_n^g(x)\ge 0\}} u_n^f(x)$

Batch Active Learning | ProbCover

$G=(X,E)$, $E=\{(x,x') \mid \|x-x'\|\le\delta\}$ $L\leftarrow\emptyset$ $\forall i = 1,2,\ldots,b$ $\hat{x} \leftarrow \arg\max_{x\in X} |\{x' \mid (x,x')\in E, x'\in X\}|$
$L \leftarrow L\cup\{\hat x\}$ | $E \leftarrow E\setminus(\{\hat x\}\times(B_\delta(\hat x)\cap X))$

Convex Optimization

Given constrained optimization problem: $\min_{w\in\mathbb{R}^d} f(w):\ g_{1:m}(w)=0, h_{1:n}(w)\le 0$ it is convex if $f, g_{1:m}, h_{1:n}$ are convex and the feasible region is convex.
The Lagrangian with Lagrange multipliers $\eta = (\lambda,\alpha)$: $L(\eta,w) = f(w) + \sum_{i\le m}\lambda_i g_i(w) + \sum_{j\le n}\alpha_j h_j(w)$
Any **optimal solution** W satisfies: $\nabla_w L(\eta, W)=0, g_i(W)=0, h_j(W)\le 0, \alpha_j \ge 0$
the **Dual Problem** is and satisfies $\forall w$: $\max_{\alpha\ge0,\lambda}[\theta(\eta) := \inf_w L(\eta,w)] \le f(w^*)$
strong duality if $\theta(\eta^*) = f(w^*)$
Slater's cond. if $\exists w_0$ feasible: $h_{1:n}(w_0) < 0$ strong duality $\to$ w^*: $f(w^*) = L(\eta^*, w^*)$ and $\alpha_j h_j(w^*) = 0, \quad \forall j\le n$.

Support Vector Machine (SVM)

Convex constrained optimization problem with strong duality (if linearly separable). $\mathbf{x}_i$ support vectors, $y_i \in\{-1,+1\}$.
$\min_{w,w_0 | \forall i\le n: y_i(w^\top x_i + w_0)\ge 1} \quad \frac12\|w\|^2$
Lagrangian: $\mathcal{L}(w,w_0,\alpha) = \frac12\|w\|^2 + \sum_{i=1}^n \alpha_i(1-y_i(w^\top x_i + w_0)) \quad \alpha_i\ge 0$.
KKT: $w^* = \sum_{i=1}^n \alpha_i y_i x_i \quad \sum_{i=1}^n \alpha_i y_i = 0$
Dual: $\max_{\alpha\ge0: \sum_{i=1}^n \alpha_i y_i = 0} \quad L(\alpha)$
$\cdot L(\alpha) = \sum_1^n \alpha_i - \frac12\sum_1^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j x_i^\top x_j$
The optimal hyperplane is given by
$\mathbf{w}^* = \sum_{i=1}^n \alpha_i^* y_i \mathbf{x}_i$
$w_0^* = -\frac12(\min_{y_i=1} \mathbf{w}^{*T}\mathbf{x}_i + \max_{y_i=-1}\mathbf{w}^{*T}\mathbf{x}_i)$
Only Support Vectors ($\alpha_i^*\ne 0$) contribute.
Optimal Margin: $\mathbf{w}^T\mathbf{w} = \sum_{i\in SV}\alpha_i^*$
Discrim.: $g^*(\mathbf{x}) = \sum_{i\in SV} y_i\alpha_i^*\mathbf{x}_i^T\mathbf{x}_i + w_0^*$
class $= \text{sign}(\mathbf{x}^T\mathbf{w}^* + w_0^*)$

Soft Margin SVM

Introduce slack to relax constraints. C controls margin maximization vs. constraint violation.
$\min_{\xi_i\ge0,w,w_0|\forall i\le n: y_i(w^\top x_i+w_0)\ge 1-\xi_i}$
$\frac12\|w\|^2 + C\sum_{i=1}^n\xi_i$
Lagrangian: $L(\mathbf{w},w_0,\xi,\alpha,\beta) = \frac12\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^n\xi_i - \sum_{i=1}^n\alpha_i[z_i(\mathbf{w}^T\mathbf{y}_i + w_0) - 1 + \xi_i] - \sum_{i=1}^n\beta_i\xi_i$
Dual Problem same as usual SVM but with supplementary constraint: $C\ge\alpha_i\ge0$
KTT Conditions: $\alpha_i^*(z_i(w^T y_i + w_0) - 1 + \xi) = 0, \xi_i(\alpha_i - C) = 0$
You should solve α via quadratic optimisation. Optimal hyperplane and classification as normal SVM. Optimal slack: $\xi_i^* = \max(0, 1-y_i(w^{*T}x_i + w_0^*))$
$\xi_i^* = \mathcal{L}^{\text{hinge}}(y_i, w^{*T}x_i + w_0^*)$

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: $K(x,y) = (1+x_1y_1+x_2y_2)^2$

Multiclass SVM

$\forall$class $y\in\{1,2,\cdots,M\}$ we introduce $\mathbf{w}_y$ and define our problem: ($\mathbf{w}$ is v-stacked)
$min_w \frac12 w^T w = min_{\{w_y\}_{n=1}^M}\sum_{y=1}^M w_y^T w_y$
s.t. $(\mathbf{w}_{y_i}^T\mathbf{x}_i + w_{y_i,0}) - \max_{y\ne y_i}(\mathbf{w}_y^T x_i + w_{y,0}) \ge 1, \forall \mathbf{x}_i\in\mathcal{X}$
classification: $\hat y = argmax_y(w_y^T x + w_{y,0})$

Structured SVM

Each $\mathbf{x}$ is assigned to a structured output label y. Output Space Representation: joint feature map: $\psi(y,\mathbf{x})$
Scoring function: $f_\mathbf{w}(y,\mathbf{x}) = \mathbf{w}^T\psi(y,\mathbf{x})$
Classify: $\hat y = h(\mathbf{x}) = \arg\max_{y\in\mathbb{K}} f_{\mathbf{w}(y,\mathbf{x})}$
SVM objective:
$w^T\psi(y_i,\mathbf{x_i}) - max_{y_i\ne y} w^T\psi(y,\mathbf{x_i}) \ge m$
with margin rescaling: $min_{w,\xi\ge0}\frac12 w^T w + C\sum_{i=1}^n\xi_i$ s.t. $w^T\psi(y_i,\mathbf{x_i}) - \Delta(y,y_i) - w^T\psi(y,\mathbf{x_i}) \ge -\xi_i \forall y\ne y_i \forall i$
Lagrangian: let $\mathbb{K}_i = \mathbb{K}\setminus y_i$
$\frac12 w^T w + C\sum_{i=1}^n\xi_i - \sum_{i=1}^n\sum_{y_j\in\mathbb{K}_i}\alpha_{i,j}(w^T\psi(y_i,x_i) - \Delta(y_j,y_i) - w^T\psi(y_j,x_i) + \xi_i) - \sum_{i=1}^n\beta_i\xi_i$ with $\alpha_{i,j}\ge0, \beta_i\ge0$

Ensemble Methods

Combining Regressors

Set of estimators: $\hat f_1(x),\cdots,\hat f_B(x)$
simple average: $\hat f(x) = \frac1B\sum_{i=1}^B\hat f_i(x)$
$\text{Bias}[\hat f(x)] = \frac1B\sum_{i=1}^B\text{Bias}[f_i(x)]$
$\mathbb{V}[\hat f(x)] \approx \frac{\sigma^2}{B}$ 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. Covariance small, variance similar, bias weakly affected. **Random Forest** Collection of uncorr. decision trees. Partition data space recursively. Grow the tree sufficiently deep to reduce bias. (random sample cuts to reduce bias). Prediction with voting.
Boosting (Weak to avoid overfitting) Combine uncorr. weak learners in sequence. Coeff. of $\hat c_{b+1}$ depend on $\hat c_b$'s results

AdaBoost (minimizes exp. loss)
Init: $\mathcal{X} = \{(x_1,y_1),\cdots,(x_n,y_n)\}, w_i^{(1)} = \frac1n$
Fit $\hat c_b(x)$ to $\mathcal X$ weighted by $w^{(b)}$
$\varepsilon_b = \sum_{i=1}^n w_i^{(b)}\mathbb{I}_{\{\hat c_b(x_i)\ne y_i\}} / \sum_{i=1}^n w_i^{(b)}$
$\alpha_b = \log\frac{1-\varepsilon_b}{\varepsilon_b} > 0$
$w_i^{(b+1)} = w_i^{(b)}\exp(\alpha_b\mathbb{I}_{\{\hat c_b(x_i)\ne y_i\}})$
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.

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 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.

Logistic Regression

$log\frac{P(y=1|x)}{P(y=-1|x)} = \sum_{b=1}^B c_b(x) =: F(x)$
$P(y=1|x) = \frac{exp(F(x))}{1+exp(F(x))}$

PAC learning

Function of interest

The probability of large excess error:
$\mathbb{P}[misclassification|C] < \delta$.
But: could be unlucky with C, c^{Bayes} not in hypoth. class.
$\mathbb{P}[\mathcal{R}(\hat c) - \inf_{c\in\mathcal{C}}\mathcal{R}(c) > \varepsilon] < 1-\delta$.
Def R.H.S.$=\delta$: $\varepsilon = \sqrt{\frac{\log N - \log(\delta/2)}{2n}}$.
N = size of hypothesis class, n = num. of samples.expected error of c depends on $1/\sqrt{n}$ and $\log N$!
for any class $\mathcal{C}$: $\mathcal{R}_n(\hat c) - \inf_{c\in\mathcal{C}}\mathcal{R}(c) \le 2\sup_{c\in\mathcal{C}}|\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)|$
for finite class: $\mathbb{P}[2\sup_{c\in\mathcal{C}}|\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)| > \varepsilon] \le 2|\mathcal{C}|e^{-\frac12 n\varepsilon^2}$.

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.).

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: $\forall_{c\in\mathcal{C}}\hat{\mathcal{R}}_n(c) \ge \hat{\mathcal{R}}_n(\hat c) - \frac dn$.

VC dimension

If you can find a set of n points, so that it can be shattered by the classifier (i.e. classify all possible 2^n labelings correctly) and you cannot find any set of n+1 points that can be shattered then the VC dimension is n.
Examples: $(-\infty, a] = 1$ all intervals in R: $V_C = 2$ For unions of k intervals, $V_c = 2k$ half planes in R^2: 3 for unit circles $V_c = 3$ convex polygons in R^2: ∞ convex polygons in R^2 with at most k vertices: $2k+1$

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:
$\mathbb{P}[z_i = k|z_{-i},\alpha] = \begin{cases}\frac{N_{k,-i}}{\alpha+N-1} & \text{existing } k\\ \frac{\alpha}{\alpha+N-1} & \text{otherwise}\end{cases}$
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.