General

$\mathrm{var}(AX) = A\,\mathrm{var}(X)A^T$

$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

$\widehat{\mathrm{cov}}(X) = \frac{1}{n}X^\top X = \frac{1}{n}\sum_i x_i x_i^\top$

Convexity

$f(tx+(1-t)y) \le tf(x)+(1-t)f(y)$

f convex, g affine $\Rightarrow f\circ g$ convex

f non-decreasing, g convex $\Rightarrow f\circ g$ convex

Gaussian

$p(x) = \frac{1}{\sqrt{(2\pi|\Sigma|)}}\exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$

PSD

$M \in \mathbb{R}^{n\times n}$ PSD $\Leftrightarrow \forall x \in \mathbb{R}^n : x^T M x \ge 0$

$\Leftrightarrow$ all principal minors of M have non-negative determinant

$\Leftrightarrow \lambda \ge 0\ \forall \lambda \in \sigma(M)$

Theoretical and Empirical Risk

$R(f) = \mathbb{E}_{(x,y)}\ell(f(x),y)$

$\hat{R}(f) = \frac{1}{n}\sum_i \ell(f(x_i),y_i)$

Gradient Descent

$w_{t+1} = w_t - \eta_t \nabla_w \hat{R}(w_t)$

Gradient Descent with Momentum

$w_{t+1} = w_t + m(w_t - w_{t-1}) - \eta_t \nabla_w \hat{R}(w_t)$

SGD

Pick data point (x,y) u.a.r. and set

$w_{t+1} = w_t - \eta_t \nabla_w l(f(x),y)$

Complexity

Matmul $A \in \mathbb{R}^{n\times k}, B \in R^{k\times d}$: $\Theta(n\times k\times d)$

KL Divergence

$D_{KL}(P\|Q) = \mathbb{E}_p[\log(\frac{p(x)}{q(x)})]$

Regression

Linear Regression

$w^* = \mathrm{argmin}_w \|y - Xw\|^2 = (X^TX)^{-1}X^Ty$

$\nabla_w \hat{R}(w) = -2\sum_{i=1}^n (y_i - w^Tx_i)\cdot x_i$

$= 2X^T(Xw-y)$

Ridge

$w^* = \arg\min_w \|y-Xw\|^2 + \lambda\|w\|_2^2$

$w^* = (X^TX+\lambda I)^{-1}X^Ty$

$\nabla_w \hat{R}(w) = 2X^T(Xw-y)+2\lambda w$

LASSO

$w^* = \arg\min_w \|y-Xw\|^2 + \lambda\|w\|_1$

Classification

Logistic Loss

$\ell_{log}(x,y) = \log(1+e^{-yf(x)})$ (convex)

$\nabla_w \ell_{log}(x,y) = \frac{-y\nabla_w f(x)}{1+\exp(yf(x))}$

Hinge Loss

$l_H(x,y,w) = \max\{0, 1-yw^Tx\}$ (convex)

$\nabla_w l_H = \begin{cases} -yx & \text{, if } yw^Tx < 1 \\ 0 & \text{, if } yw^Tx \ge 1 \end{cases}$

Hard-margin SVM

If data is separable, find

$\min_w \|w\|$ s.t. $y_i w^\top x_i \ge 1\ \forall i.$

Soft-margin SVM

If data is not separable, find

$\min_{w,\xi} \frac{1}{2}\|w\|^2 + \lambda\sum_i \xi_i$ s.t. $\begin{cases} y_i w^\top x_i \ge 1-\xi_i \\ \xi_i \ge 0 \end{cases}$

$= \min_w \frac{1}{2}\|w\|^2 + \lambda\sum_i \max\{0, 1-y_iw^\top x_i\}$

Evaluation

Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$

Precision $= \frac{TP}{TP+FP} = 1-$ FDR

Recall = TPR= $\frac{TP}{TP+FN}$

FPR $= \frac{FP}{TN+FP}$

ROC curve: $x =$ FPR, $y =$ TPR

F1 $= \frac{2TP}{2TP+FP+FN} = \frac{2}{1/Precision+1/Recall}$

Kernels

Polynomial kernel

$k(x,y) = (x^Ty)^m$ all monomials of deg. m

$k(x,y) = (1+x^Ty)^m$ all monomials up to deg. m There are $\binom{d+m}{m} = O_d(d^m) = O_m(m^d)$ monomials of order m in d variables.

Properties $k(x,y) = \phi(x)^T\phi(y)$

1. k must be symmetric.
2. Kernel matrix K must be PSD for all $x_1,\ldots,x_n$.

$$K = \begin{bmatrix} k(x_1,x_1) & \ldots & k(x_1,x_n) \\ \vdots & \ddots & \vdots \\ k(x_n,x_1) & \ldots & k(x_n,x_n) \end{bmatrix}$$

Valid kernels

k_1+k_2, k_1k_2, ck if $c>0$;

$f(k)$, where f is a polynomial/power series with non-negative coefficients;

$k(\binom{x}{y},\binom{x'}{y'}) = k(x,x')k(y,y')$,

$k(\binom{x}{y},\binom{x'}{y'}) = k(x,x')+k(y,y')$

where $\binom{x}{y}$ is concatenation of vectors;

$k(x,y) = g(x)k(x,y)g(y)$ where $g\colon X \to \mathbb{R}$.

Kernelized Ridge

Ansatz: $w^* = \Phi^T\alpha$

$\min_w \|\Phi w - y\|^2 + \lambda\|w\|_2^2$

$= \min_a \|K\alpha - y\|_2^2 + \lambda\alpha^TK\alpha$

$\alpha^* = (K+\lambda I)^{-1}y$

Prediction: $\hat{y} = \Sigma_{i=1}^n \alpha_i^* k(x_i,x)$

Neural Networks

$F(x) = W^L\phi^{L-1}(W^{L-1}\ldots(\phi^1(W^1x)\ldots))$

Activation functions

Sigmoid: $\varphi(z) = \frac{1}{1+\exp(-z)}$; $\varphi' = (1-\varphi)\cdot\varphi$

Tanh: $\tanh(z) = \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$

ReLu: $\max(0,z)$

Backpropagation

For the output layer L:

1. Compute error $\delta^L = \nabla_f \ell$
2. and gradient $\nabla_{W^L}\ell = \delta^L(v^{(L-1)})^T$

For each hidden layer $l = L-1,\ldots,1$:

1. Compute error $\delta^l = \varphi'(z^l)\odot((W^l)^T\delta^{l+1})$
2. and gradient $\nabla_{W^l}\ell = \delta^l(v^{(l-1)})^T$

Regularization

- Weight decay
- Early stopping (validation)
- Dropout
- Batch normalization

CNNs

Output size (per dim) $l = \frac{n+2p-f}{s}+1$

Clustering

k-means

Want μ_i to minimize $\sum_{i=1}^n \min_{j\in\{1,\ldots,k\}}\|x_i - \mu_j\|_2^2$ Non-convex and NP-hard in general. Can be kernelized.

Lloyd's heuristic

While not converged:

$z_i = \arg\min_{j\in\{1,\ldots,k\}}\|x_i - \mu_j^{t-1}\|_2^2$

$\mu_j^{(t)} = \frac{1}{n_j}\sum_{i:z_i=j} x_i$

Monotonically decreases objective and converges to a local optimum. Cost per iteration $O(nkd)$.

k-means++ Initialization:

Start with random data point as center. Add centers 2 to k randomly, proportionally to squared distance to closest selected center.

Dimension Reduction

Principal component analysis (PCA)

Given centered data, the PCA problem is

$$\min_{W^TW=I_k, z_i\in\mathbb{R}^k} \sum_{i=1}^n \|Wz_i - x_i\|_2^2,$$

with solution $W^* = (v_1|\ldots|v_k)$ where v_i are the ordered eigenvectors of $\frac{1}{n}\sum_i x_ix_i^T$ and $z_i = W^{*\top}x_i$.

Kernel PCA

The Kernel Principal Components are given by $\alpha^1,\ldots,\alpha^k \in \mathbb{R}^n$ where $\alpha^i = \frac{1}{\sqrt{\lambda_i}}v_i$ and $K = \sum_{i=1}^n \lambda_i v_i v_i^T$ with ordered λ_i. A point x is projected to $z \in \mathbb{R}^k$: $z_i = \sum_{j=1}^n \alpha_j^{(i)}k(x,x_j)$

Autoencoders

Try to learn identity function $x \approx f(x;\theta) = f_2(f_1(x_1;\theta_1);\theta_2)$. NN Autoencoder with linear activations is equivalent to PCA.

Probabilistic Modeling

MLE

Given a choice of marginal $P(Y|X,\theta)$ take $\theta^* = \arg\max_\theta \prod_{i=1}^n P_\theta(y_i|x_i)$.

Bayes optimality

$\arg\min_f \mathbb{E}_{x,y}[(y-f(x))^2] = \mathbb{E}[Y \mid X]$

$\arg\min_f \mathbb{E}_{x,y}[1_{[y\ne f(x)]}]$

$= \arg\max_y p(Y=y \mid X=x)$

Bias-Variance-noise decomposition

$\mathbb{E}_{x,y}[(\hat{f}_D(x)-y)^2]$

$= \mathbb{E}_x[\mathbb{E}_D[\hat{f}_D(x)] - f^*(x)]^2$

$+\mathbb{E}_x[\mathrm{var}[\hat{f}_D(x)]]$

$+\mathbb{E}_{x,y}[(y-f^*(x))^2]$ where $f^* = \mathbb{E}[Y \mid X]$.

Logistic regression

Parametrize $P(y \mid x)$ by $\frac{1}{1+\exp(-yw^Tx)}$.

MLE is $\mathrm{argmax}_w P(y_{1:n}|w,x_{1:n})$

$= \mathrm{argmin}_w -\sum_{i=1}^n \log P(y_i|w,x_i)$

$= \mathrm{argmin}_w \sum_{i=1}^n \log(1+\exp(-y_iw^Tx_i))$

Gradient for logistic regression

$\ell(w) = \log(1+\exp(-yw^Tx))$

$\nabla_w \ell(w) = \frac{-yx}{1+\exp(yw^Tx)}$

Multiclass Logistic Regression

Parametrize $P(Y=i \mid x)$ by $\frac{\exp(w_i^Tx)}{\sum_j \exp(w_j^Tx)}$.

Kernelized logistic regression

$\min_\alpha \sum_i \log(1+\exp(-y_i\alpha^\top K_i)) + \lambda\alpha^\top K\alpha$

$\hat{P}(y \mid x) = \frac{1}{1+\exp(-y\sum_i \alpha_i k(x_i,x))}$

Bayesian decision theory

1. Conditional distribution over labels $P(y|x)$
2. Set of actions $\mathcal{A}$
3. Cost function $C: Y \times \mathcal{A} \to \mathbb{R}$

Pick action that minimizes the expected cost: $a^* = \mathrm{argmin}_{a\in\mathcal{A}} \mathbb{E}_y[C(y,a)|x]$

Optimal decision for logistic regression

$a^* = \mathrm{argmin}_y \hat{P}(y|x) = sign(w^T x)$ assuming 0-1-cost.

Generative Modeling

Discriminative: Estimate $P(y \mid x)$

Generative: Estimate $P(y,x)$

Typical approach to generative modeling:

1. Estimate prior on labels $P(y)$
2. Estimate conditional distribution $P(x \mid y)$ for each class y
3. Obtain predictive distribution using Bayes' rule: $P(y \mid x) = \frac{P(y)P(x|y)}{P(x)} = \frac{P(x,y)}{P(x)}$

Naive Bayes

Naive Bayes assumes conditional independence:

$$P(X_1,\dots,X_n \mid Y) = \prod_i P(X_i \mid Y).$$

If independence is violated, predictions become overconfident (close to 0 and 1).

Decision rule

$\hat{y} = \mathrm{argmax}_y P(y \mid x)$

$= \mathrm{argmax}_y P(y)\prod_i P(x_i \mid y)$

$= \mathrm{argmax}_y \log P(y) + \sum_i \log P(x_i \mid y)$

QDA/Gaussian Bayes Classifier

$P(Y=y) = p_y$ and $P(x \mid y) = \mathcal{N}(\mu_y, \Sigma_y)$

$\hat{p}_y = \frac{\mathrm{Count}(Y=y)}{n}$

$\hat{\mu}_y = \frac{1}{\mathrm{Count}(Y=y)} \sum_{i:y_i=y} x_i$

$\hat{\Sigma}_y = \frac{1}{\mathrm{Count}(Y=y)} \sum_{i:y_i=y} (x_i - \hat{\mu}_y)(x_i - \hat{\mu}_y)^T$

For two classes $\hat{y} = \mathrm{sign}\left(\log \frac{P(Y=1|x)}{P(Y=-1|x)}\right)$

where $\log \frac{P(Y=1|x)}{P(Y=-1|x)} = \log \frac{\hat{p}}{1-\hat{p}} + \frac{1}{2}\log\frac{|\hat{\Sigma}_-|}{|\hat{\Sigma}_+|}$
$+\frac{1}{2}(x-\hat{\mu}_-)^T\hat{\Sigma}_-^{-1}(x-\hat{\mu}_-) - \frac{1}{2}(x-\hat{\mu}_+)^T\hat{\Sigma}_+^{-1}(x-\hat{\mu}_+)$

Gaussian Naive Bayes

GBC with diagonal Σs. GNB with shared Σs across two classes yields the same predictions as Logistic Regression (if model is true).

Fisher's LDA (Subcase of GBC)

Assume: Two classes, $p = 0.5$, $\Sigma_- = \Sigma_+$

Outlier Detection

Classify x as outlier if $P(x) \le \tau$.

Regularization

- Restricting model (i.e. covariance)
- Prior on parameters.

Mixture Models

Gaussian Mixtures

$P(x \mid z) = \sum_i w_i \mathcal{N}(\mu_i, \Sigma_i)$

MLE is a nonconvex problem $\to$ EM.

Hard-EM

E-step: Compute

$z_i^{(t)} = \mathrm{argmax}_z P(z \mid x_i, \theta^{(t-1)})$

$= \mathrm{argmax}_z P(z \mid \theta^{(t-1)}) P(x_i \mid z, \theta^{(t-1)})$

$\overset{\text{GMM}}{=} \mathrm{argmax}_z w_z^{(t-1)} \mathcal{N}(x;\ \mu_z^{(t-1)}, \Sigma_z^{(t-1)})$

M-step: $\theta^{(t)} = \mathrm{argmax}_\theta P(x_{1:n}, z_{1:n}^{(t)} \mid \theta)$

Hard-EM converges to a local maximum of $P(x_{1:n}, z_{1:n}^{(t)} \mid \theta)$. It tends to do poorly if clusters overlap. Hard-EM for GMM with $w_z = \frac{1}{k}, \Sigma_z = \sigma^2 I$ is equivalent to k-means.

Soft-EM

E-step:

Compute the distribution of $Z \mid x, \theta^{(t-1)}$, i.e. for each x the responsibilities

$$P_{\theta^{(t-1)}}(Z=j \mid x) = \frac{P(Z=j)P(x \mid Z=j)}{P(x)} \overset{\text{GMM}}{=} \frac{w_j \mathcal{N}(x;\ \Sigma_j, \mu_j)}{\sum_k w_k \mathcal{N}(x;\ \Sigma_k, \mu_k)}.$$

M-step:

$$\theta^{(t)} = \arg\max_\theta \mathbb{E}_{Z_{1:n}|x_{1:n},\theta^{(t-1)}}\left[\log P_\theta(x_{1:n}, Z_{1:n})\right]$$

$$\overset{\text{iid\&cond.ind.}}{=} \sum_{i=1}^n \mathbb{E}_{Z_i|x_i,\theta^{(t-1)}}\left[\log P_\theta(x_i, Z_i)\right]$$

$$= \sum_{i=1}^n \sum_{j=1}^k P_{\theta^{(t-1)}}(Z_i = j \mid x_i) \log P_\theta(x_i, Z_i = j)$$

GMM M-step:

$w_j^{(t)} \leftarrow \frac{1}{n}\sum_i \gamma_j^{(t)}(x_i)$;

$\mu_j^{(t)} \leftarrow \frac{\sum_i \gamma_j^{(t)}(x_i)x_i}{\sum_i \gamma_j^{(t)}(x_i)}$

$\Sigma_j^{(t)} \leftarrow \frac{\sum_i \gamma_j^{(t)}(x_i)(x_i-\mu_j^{(t)})(x_i-\mu_j^{(t)})^T}{\sum_i \gamma_j^{(t)}(x_i)} \{+\nu^2\mathbb{I}\}$

The cluster size can be selected via CV. EM converges to a local maximum for GMMs, dependent on initialization.

Semi-Supervised Learning w/ GMMs:

Set $P_{\theta^{(t-1)}}(Z=j \mid x) = 1_{\{j=y\}}$ for labeled points (x,y).

GANs

Goal: Approximate a distribution $X \approx G_w(Z)$ where Z is a simple distribution. Train G and D simultaneously on the objective

$$\min_G \max_D \mathbb{E}_x[\log D(x)] + \mathbb{E}_Z[\log 1 - D(G(Z))].$$

Training requires finding a saddle point. If G, D are complex enough, then the distribution of X minimizes the objective. Cannot compute likelihood on holdout set. If D is globally optimal for some G, then D predicts $\frac{p_X(x)}{p_G(x)+p_X(x)}$.