CS 189 Cheat Sheet

Classification

k-Nearest Neighbors

kNNs are bounded by $\leq$ 2x the Bayes optimal error, $N, k \to \infty, k/N \to 0$.

Edge Case	2 pts w/ same features but diff classes.
Robustness	Generalizes better to test data.
Fit	Better training classification.
Validation	Hold back data subset as validation set. Train multiple times w/ diff hyperparams. Choose what is best on validation set.
Training Set	Used to learn model weights.
Validation Set	Tunes hyperparameters (ex. $k \in k$NN).
Test Set	used as FINAL evaluation of model.
Isocontour of f	$L_c = \{x \mid f(x) = c\}$, with isovalue c.
Isotropic Gaussian	Same var in ea dir: $\Sigma = cI$.
Anisotropic Gaussian	Allows diff amnts of var along diff dirs, $\Sigma \succ 0$.

Perceptron

Model/rule: 1 if $\vec{X_i} \cdot \vec{w} \geq 0$ elif $\vec{X_i} \cdot \vec{w} \leq 0 \implies$ -1.
Loss: $L(z, y_i) = 0$ if $y_i z \geq 0$ else $-y_i z$, (z=pred, y_i=true ans).

$$R(w) = \sum_{i=1}^{n} L(X_i \cdot w, y_i) = \sum_{i \in V} -y_i X_i \cdot w$$

Gives some linear boundary; if data is linearly separable, correctly classifies all data in at most $O\left(\frac{r^2}{\gamma^2}\right)$ iterations.

Support Vector Machines

Hard-Margin: $\min_{\vec{w},b} \|\vec{w}\|_2^2$, s.t. $y_i(\vec{w}^\top \vec{x_i} - b) \geq 1 \ \forall i$

Fails w/ non-linearly sep. data. Margin size $= \frac{1}{\|w\|}$, Slab size $= \frac{2}{\|w\|}$

Hyperplane	$H = \{x : w \cdot x = -\alpha\}$ flat, infinite, $\dim(d-1)$ plane
$x, y \in H$	$\vec{w} \cdot (y - x) = 0$, $\vec{w}$ is normal vec of H.
Support Vectors	Examples needed to find $f(\mathbf{x}) \in$ SVM. Examples with non-0 weight $\alpha_k \in$ SVM.

Soft-Margin

Allows misclassifications: $\min_{\vec{w},b,\xi_i} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^n \xi_i$ s.t.

$$y_i(\vec{w}^\top \vec{x_i} - b) \geq 1 - \xi_i, \quad \forall i; \quad \xi_i \geq 0, \quad \forall i$$

Small C: maximize margin, underfitting, less sensitive, more flat.
Big C: minimize margin, overfitting, very sensitive, more sinuous.
$C \to \infty \implies$ Soft-Margin $\to$ Hard-Margin. Note $C \geq 0$.

Generative

Want to learn **everything** about data before you classify: the **priors** $\hat{\pi}_i = \Pr(Y = C_i)$ and **cond. dist** $\mathbb{P}(X|Y = C_i)$.
Posterior: $\mathbb{P}(Y = C_i|X) = \frac{\mathbb{P}(X|Y=C_i)\cdot\mathbb{P}(Y=C_i)}{\mathbb{P}(X)}$

Logistic Function:	$\frac{1}{1+e^{-h(x)}}$, where $h(x)$ is **linear** in terms of features. True in LDA but not QDA (where $h(x)$ is quadratic).
GDA:	Assumes each class models a Gaussian distribution. $Q_C(x) = -\frac{\|x-\mu_C\|^2}{2\sigma_C^2} - d\ln\sigma_C + \ln\pi_C$
QDA:	Works with any number of classes; $\frac{d(d+3)}{2} + 1$ params.
LDA:	when variances are equal; $d + 1$ params.

Isotropic:
QDA: $\widehat{\sigma}^2 = \frac{1}{dn}\sum_{i:y_i=C}\|x_i - \widehat{\mu_c}\|^2$
LDA: $\widehat{\sigma}^2 = \frac{1}{dn}\sum_C\sum_{i:y_i=C}\|x_i - \widehat{\mu_c}\|^2$
Anisotropic:
QDA: $\widehat{\Sigma}_c = \frac{1}{n_c}\sum_{i:y_i=C}(X_i - \widehat{\mu_c})(X_i - \widehat{\mu_c})^\top$
LDA: $\widehat{\Sigma} = \frac{1}{n}\sum_C\sum_{i:y_i=C}(X_i - \widehat{\mu_c})(X_i - \widehat{\mu_c})^T$

Discriminative

Want to learn a *few* things before trying to classify.
Only tries to model $\mathbb{P}(Y|X)$ from training data.
Logistic Reg (2 classes): For a training point, $P(Y = y_i \mid x) = p^{y_i}(1-p)^{1-y_i}$. Note that $p = s\left(w^T x\right)$ as given by our model of the posterior $P(Y = 1 \mid x)$. MLE on this leads to the cross entropy loss function (which is convex!), namely

$$L(w) = -\sum y_i\left(\ln p_i + (1 - y_i)\ln(1 - p_i)\right)$$

Note: $P(Y = 1 \mid x) = \frac{1}{1+\exp(-w^T x)}$; $s'(\gamma) = s(\gamma)(1 - s(\gamma))$

Decision Boundary: of the form $w^T x > c_1$ thus must be linear. Though probability predictions are non-linear, actual boundary is linear. Log Reg always separates linearly separable points.
Softmax Reg: logistic regression for multiple classes

Probability

Multivariate Gaussian PDF:
$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$

MLE (Maximum Likelihood Estimate)

We have A, B, C, D. $P(A \mid B) > P(A \mid C) > P(A \mid D)$
$\implies B$ is the MLE of A. MLE Estimate of Anisotropic can be
$\hat{\theta}_{MLE}(x) = \arg\max_\theta f(x \mid \theta) = \arg\max_\theta \mathcal{L}(\theta; x)$
Mean is unbiased; Variance is biased (usually underestimate)
Predicts parameter which max the probability of the data.
Implicitly assumes uniform prior

MAP (Maximum a Posteriori)

We have A, B, C, D. $\mathbb{P}(A \mid B) > \mathbb{P}(C \mid B) > \mathbb{P}(D \mid B)$
$\implies A$ is the MAP of B.
$\hat{\theta}_{MAP} = \arg\max_\theta f(\theta \mid x) = \arg\max_\theta f(x \mid \theta) \cdot g(\theta)$
Predicts the parameter which maximizes the conditional probability of the parameter given the data.
Should be used when you have the prior probabilities.
MLE = MAP when all parameters have equal prior probability.
The axis lengths of Gaussian Isocontours are σ_i s.t. $\sigma^2(X) = \mathrm{Var}(X)$. Independent $\iff$ uncorrelated (only for Multivariate Gaussian).

Bayesian Risk

L (loss function) is symmetric: pick class with max posterior prob.
L is asymmetric: minimize $\mathbb{E}[L(\text{true class}, \text{prediction}) \mid \text{data}]$ or pick max loss-weighted posterior prob.
The risk for r is the expected loss over all values of x, y. Equals to 0 when class distros don't overlap or prior prob for one class is 1.

$$R(r) = \mathbb{E}[L(r(X), Y)]$$
$$= \sum_x \left(\sum_{c\in\{-1,1\}} L(r(x), c)P(Y = c \mid X = x)\right) P(X = x)$$
$$= \sum_{c\in\{-1,1\}} \left(P(Y = c)\sum_x L(r(x), c)P(X = x \mid Y = c)\right)$$

$$R(\hat{y} = i \mid x) = \sum_{j=1}^{C} \lambda_{ij} P(Y = j \mid x)$$

The Bayes decision rule aka Bayes classifier is the fn r^* that minimizes functional $R(r)$. Assuming $L(z, y) = 0$ for $z = y$:

$$r^*(x) = \begin{cases} 1 \text{ if } L(-1,1)P(Y = 1 \mid X = x) > L(1,-1)P(Y = -1 \mid X = x) \\ -1 \text{ otherwise} \end{cases}$$

Regression Methods

Model: $y = Xw$, Loss Function: least squares, $n \in N(X)$

Name	Objective	Solution
OLS	$\frac{1}{n}\|Y - Xw\|_2^2$	$w^* = (X^\top X)^\dagger X^\top y \in X^\dagger y + n$
Ridge	$\frac{1}{n}\|Y - Xw\|_2^2 + \lambda\|w\|_2^2$	$w^* = (X^\top X + n\lambda I)^{-1}X^\top y$
LASSO	$\frac{1}{n}\|Y - Xw\|_2^2 + \lambda\|w\|_1$	No closed form

Linear Algebra

Matrix Calculus

$\nabla_{\vec{x}}\vec{w}^\top\vec{x} = \left(\frac{\partial\vec{w}^\top\vec{x}}{\partial\vec{x}}\right)^\top = \vec{w}$	$\nabla_{\vec{x}}(\vec{w}^\top A\vec{x}) = A^\top\vec{w}$
$\nabla_A\vec{w}^\top A\vec{x} = \vec{w}\vec{x}^\top$	$\nabla_{\vec{x}}(\vec{x}^\top A\vec{x}) = (A + A^\top)\vec{x}$
$\nabla_{\vec{x}}^2(\vec{x}^\top A\vec{x}) = A + A^\top$	$\nabla_{\vec{x}}(\alpha\vec{y}) = (\nabla_{\vec{x}}\alpha)\vec{y}^\top + \alpha\nabla_{\vec{x}}\vec{y}$
$\nabla_{\vec{x}}\vec{f}(\vec{y}) = (\nabla_{\vec{x}}\vec{y})(\nabla_{\vec{y}}\vec{f}(\vec{y}))$	$\nabla_{\vec{x}}(\vec{y}\cdot\vec{z}) = (\nabla_{\vec{x}}\vec{y})\vec{z} + (\nabla_{\vec{x}}\vec{z})\vec{y}$
$\nabla_{\vec{x}}g(\vec{y}) = (\nabla_{\vec{x}}\vec{y})(\nabla_{\vec{y}}g(\vec{y}))$	6969

$$\nabla_{\vec{y}}(\vec{y} - A\vec{x})^\top W(\vec{y} - A\vec{x}) = 2W(\vec{y} - A\vec{x})$$
$$\nabla_{\vec{x}}(\vec{y} - A\vec{x})^\top W(\vec{y} - A\vec{x}) = -2A^\top W(\vec{y} - A\vec{x})$$
$$\nabla_{\vec{w}}\left(\|X\vec{w} - \vec{y}\|_2^2 + \lambda\|\vec{w}\|_2^2\right) = 2X^\top X\vec{w} - 2X^\top\vec{y} + 2\lambda\vec{w}$$

Matrix A is Positive Semi-Definiteness iff

(a) $\forall\vec{x} \neq \vec{0} \in \mathbb{R}^n, \vec{x}^\top A\vec{x} \geq 0$.
(b) All eigenvalues of A are non-negative.
(c) $\exists$ unique matrix $L \in \mathbb{R}^{n\times n}$ such that $A = LL^\top$ (Cholesky decomposition).

All diagonal entries of A are non-negative and $\mathrm{Tr}(A) \geq 0$.
Sum of all the entries ≥ 0.
$M \succeq 0, N \succeq 0 \implies M - N \succeq 0 \iff \lambda_{\min}(M) > \lambda_{\max}(N)$.
$A = A^{\frac{1}{2}}A^{\frac{1}{2}}$, $A^{\frac{1}{2}} = U\Lambda^{\frac{1}{2}}U^\top$
A function is convex iff Hessian is PSD. Strict Convexity: $(\forall 0 < t < 1), f(tx_1 + (1-t)x_2) < tf(x_1) + (1-t)f(x_2)$

Covariance Matrix

$$\Sigma = \frac{1}{n}\hat{X}^\top X = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_d) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \cdots & \mathrm{Cov}(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_d, X_1) & \mathrm{Cov}(X_d, X_2) & \cdots & \mathrm{Var}(X_d) \end{bmatrix}$$

$= \mathbb{E}[(X - \mu)^\top(X - \mu)]$ where $X \in \mathbb{R}^{n\times d}$, all diag entries > 0

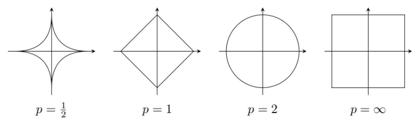
Symmetric, PSD $\implies \exists\Sigma = V\Lambda V^\top$ by Spectral Theorem. PD $\implies$ symmetric in this class. Eigenvectors are orthogonal directions along which points are uncorrelated. $\Sigma^{-1} = V\Lambda^{-1}V^\top = \sum_i \frac{1}{\Lambda_{ii}}v_i v_i^\top$

Spectral Theorem: $A = V\Lambda V^\top$

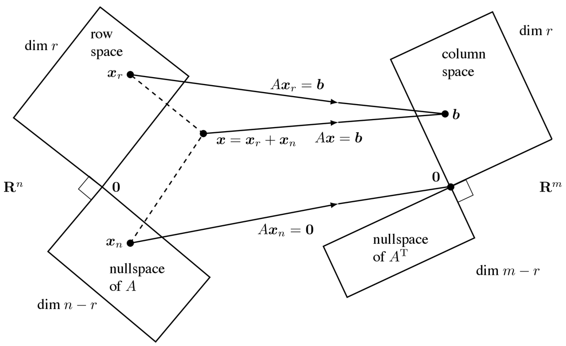
All real+symmetric $n \times n$ matrix has real eigenvalues and n eigenvectors that are mutually orthogonal: $v_i^\top v_j = 0 \quad \forall i \neq j$.

Norm Ball

ℓ_0 and ℓ_1 encourage sparsity (more than ℓ_2).



Fundamental Theorem of Linear Algebra



$(N(A)^\perp = R(A^\top)) \oplus (N(A^\top A) = N(A) = R(A^\top)^\perp) = \mathbb{R}^n$
$(N(A^\top)^\perp = R(A)) \oplus (N(A^\top) = R(A)^\perp) = \mathbb{R}^m$
Rank-nullity Theorem: $\dim(R(A)) + \dim(N(A)) = n$
Jensen's Inequality: If $f(x)$ is strictly convex, $\mathbb{E}[f(x)] > f(\mathbb{E}[x])$.
$\text{rank}(A^\top) = \dim(\text{Row}(X)) = \dim(R(X^\top)) = \text{rank}(X^\top) = \text{rank}(X)$.
$\text{Row}(X^\top X) = R(X^\top X) = \text{Row}(X) = R(X^\top)$

Update Rule

Gradient Descent:	$w \leftarrow w - \epsilon \nabla_w J(w)$
Logistic Reg:	$w \leftarrow w + \epsilon X^\top (y - s(Xw))$

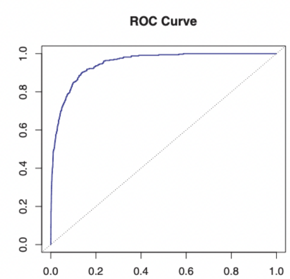
Newton's Method:	$w \leftarrow w - (\nabla_w^2 J(w))^{-1} \nabla_w J(w)$
*** Note:	If J quadratic, Newton's method only needs one step to find exact solution. Newton's Method doesn't work for most nonsmooth functions, and is generally faster than BGD/SGD.

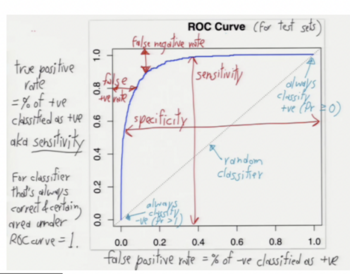
Stochastic GD:	$w \leftarrow w - \epsilon \nabla_w J(w)_i$ for some $i \in U([1, \dots, n])$
Logistic Reg:	$w \leftarrow w + \epsilon (y_i - s(X_i \cdot w)) X_i$

Cost Functions

$y_i = f(X_i) + \epsilon_i$: ϵ_i from Gaussian, all ϵ_i same mean, all y_i same var
General: $J = \sum_{i=1}^n L(X_i \cdot w, y_i)$
Linear: $J = \sum_{i=1}^n (X_i \cdot w + \alpha - y_i)^2 = \|Xw - y\|_2^2$
Logistic: $J = -\sum_{i=1}^n (y_i \ln s(X_i \cdot w) + (1 - y_i) \ln(1 - s(X_i \cdot w)))$
Weight LS: $J = \sum_{i=1}^n w_i (X_i \cdot w - y_i)^2 = (Xw - y)^\top \Omega (Xw - y)$

ROC Curve





Design Matrix

Centering:	subtracting $\mu^\top$ from each row of X: $X \to \dot{X}$
Decorrelating:	Applying rotation $Z = \dot{X}V$ where $\text{Var}(X) = V\Lambda V^\top$. Covariance matrix of Z is Λ (diagonal)
Sphering:	$W = \dot{X}\,\text{Var}(X)^{-1/2}$ ($\Sigma^{-1/2}$:ellipsoid to sphere)
Whitening:	Perform centering, and then sphering

Bias-Variance Tradeoff

Statistical Bias: $\mathbb{E}[\hat{\theta} - \theta] = \mathbb{E}[\hat{\theta}] - \theta$.

Bias: error due to inability of hypothesis h to fit g perfectly e.g., fitting quadratic g with a linear h
Variance: error due to fitting random noise in data e.g., we fit linear g with a linear h, yet $h \neq g$.

Overfitting: Low Bias, High Variance
Underfitting: High Bias, Low Variance.

Adding a feature usually increases variance [don't add a feature unless it reduces bias more]. Adding a feature results in a non-increasing bias.

Forward/Backward stepwise selection aren't guaranteed to find optimal features. Backward stepwise selection looks at $d' - 1$ features at a time, where d' is current num of features (one at a time). Use Forward selection if we think few features important, Backward selection if many features important.

higher residuals $\implies$ higher bias
higher complexity $\implies$ higher variance

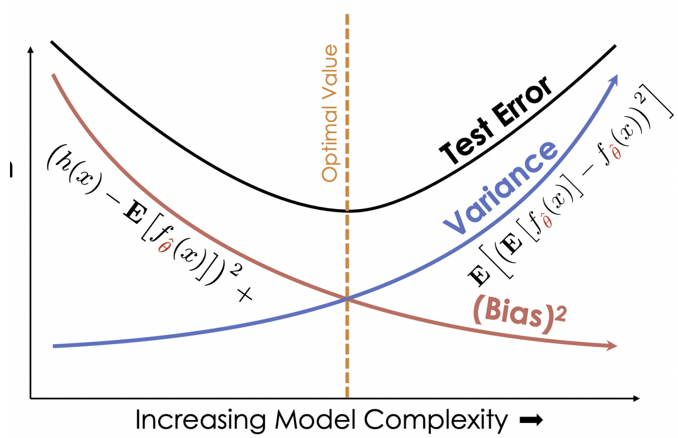
$\text{Var}(h(z)) = E\left[(h(z) - E[h(z)])^2\right] \approx \sigma^2 \frac{d}{n}$

Bias-Variance Decomposition:
Model Risk $= \mathbb{E}[L(h(z), \gamma)] = \mathbb{E}[(h(z) - \gamma)^2]$

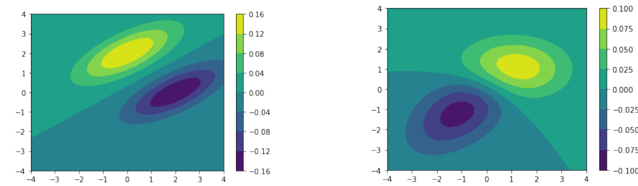
$$= \underbrace{(\text{E}[h(z)] - g(z))^2}_{\text{bias}^2 \text{ of method}} + \underbrace{\text{Var}(h(z))}_{\text{variance of method}} + \underbrace{\text{Var}(\epsilon)}_{\text{irreducible error}}$$

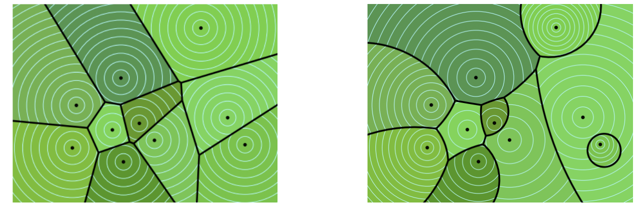
where $\text{E}[\gamma] = g(z)$; $\text{Var}(\gamma) = \text{Var}(\epsilon)$.

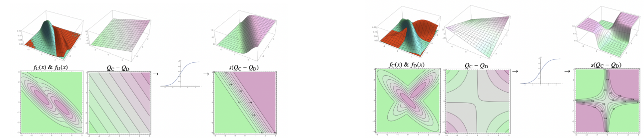
Note: the model determines Bias-Variance Tradeoff, not the algorithm used to solve the model/optimization problem.



Isocontour/Voronoi Diagrams



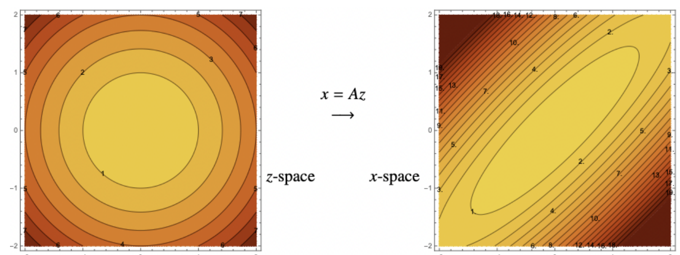




LDA: same variance; decision boundary is linear

QDA: different variance; decision boundary is curved towards class(es) w/ lower variance

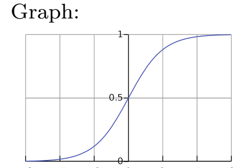
---



Quadratic Form: $x^\top A^{-2} x = \|A^{-1}x\|_2^2$ is an ellipsoid with axes $v_1, v_2, \dots v_n$ (eigenvectors of A) and radii $\lambda_1, \lambda_2, \dots, \lambda_n$ (eigenvalues of A). Note that $A > 0$.

Gaussian with covariance matrix $\Sigma = \frac{1}{n}\hat{X}^\top \hat{X}$ isocontours with radii of length $\sqrt{\lambda_i(\Sigma)} = \sigma_i(X)$

Miscellaneous

Bayes vs. GDA	Bayes uses true mean/variance, while GDA uses sample mean/variance. True mean/variance equal $\not\Rightarrow$ Sample mean/variance equal
Cauchy-Schwarz	$\lvert\langle x, y\rangle\rvert \leq \|x\| \cdot \|y\|$
Sigmoid Function: $s(\gamma) = \frac{1}{1+e^{-\gamma}}$	Graph: 
Unique Optimum	Only ridge regression has one unique optimum (not Least Squares, Lasso, or Logistic).
Training Data:	Training on less data can improve training accuracy, training on more data can improve validation/test accuracy.