Linear decision boundary is a hyperplane
Select hyperparameters with validation
Sample point = feature vector
decision fn: fn $f(x)$ that maps a sample point x to a scalar
$f(x) > 0$, if $x \in$ class c
$f(x) \leq 0$, if $x \notin$ class c
decision boundary is $f(x) = 0$, $x \in \mathbb{R}^d$
$\{x : f(x) = 0\} :=$ isosurface of f for isovalue 0.
Euclidean norm: $|x| = \sqrt{x \cdot x} = \sqrt{x_1^2 + \dots + x_d^2}$
Normalize by: $\frac{x}{|x|}$
Hyperplane $\in \mathbb{R}^d$ has dim $d-1$.
$H = \{x : w \cdot x = -\alpha\}$, given linear decision fn: $f(x) = w \cdot x + \alpha$
w is normal vector of H.

Centroid: ~~doesnt~~ require linear separability
$\mu_c :=$ mean of all vectors $\in C$
$\mu_x :=$ mean of all $\notin C$.
$f(x) = (\mu_c - \mu_x) \cdot x - (\mu_c - \mu_x) \cdot \frac{\mu_c + \mu_x}{2}$
- decision boundary is hyperplane that bisects line segment w endpoints μ_c, μ_x

Perceptron
$y_i = \begin{cases} 1 & \text{if } X_i \in \text{class C} \\ -1 & \text{if } X_i \notin C \end{cases}$
Goal: $y_i X_i \cdot w \geq 0$, i.e. $X_i \cdot w \geq 0$, if $y_i = 1$; $X_i \cdot w \leq 0$, if $y_i = -1$
$L(\hat{y}_i, y_i) = \begin{cases} 0, & \hat{y}_i y_i \geq 0 \\ -\hat{y}_i y_i & \text{ow} \end{cases}$ ⇒ want same sign
$R(w) = \sum_{i \in V} -y_i X_i \cdot w$, $V =$ all i where $y_i X_i \cdot w < 0$.
Goal: minimize risk $= \sum_{i=1}^{n} L(X_i \cdot w, y_i)$

gradient of R wrt W is direction of steepest ascent
SGD: pick 1 misclassified X_i, do GD on $L(X_i \cdot w, y_i)$
- simulate general hyperplane in d dim by using hyperplane thru origin in d+1 dimensions.
Perceptron algorithm will find a soln if possible
Margin: distance from decision boundary to nearest sample point
↳ $w \cdot x + \alpha = 1$ $w \cdot x + \alpha = 0$ (decision boundary)
→ slab of width $\frac{2}{|w|}$ containing no sample points
Opt: Find w, α that minimize $|w|^2$ subject to $y_i(X_i \cdot w + \alpha) \geq 1$ i where $y_i(X_i \cdot w + \alpha) = 1$ lie on margin
:= quadratic program
This is a maximum margin classifier, aka hard margin SVM w is $\perp$ to decision boundary dir of f
Signed distance from hyperplane to X_i is $\frac{w}{|w|} \cdot X_i + \frac{\alpha}{|w|}$
Hard margin SVMs fail if data not linearly separable
sensitive to outliers
Soft Margin SVMs:
- allow some points to violate the margin, w slack variables
$y_i(X_i \cdot w + \alpha) \geq 1 - \xi_i$, $\xi_i \geq 0$ } constraints
Point i has nonzero ξ_i iff it violates the margin
Opt: minimize $|w|^2 + C \sum_{i=1}^{n} \xi_i$ C affects bias variance tradeoff

	small C	big C
desire	maximize margin	Keep most slack var zero or small margin ↓
danger	underfitting	overfitting
outliers	less sensitive	very sensitive
boundary	more "flat"	more sinuous

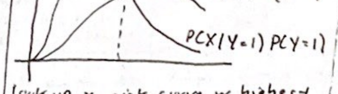
any point on slab/margin considered support vector

Nonlinear decision boundaries - nonlinear features that lift points into higher-d space.
- add features
- can lift points to ϕ space to make them linearly separable
- margin tends to get wider as degree increases
higher degree → overfitting
Edge detection: collect line orientations in local histograms, use histograms as features

Applications/Data
Model
Optimization ~~Algorithm~~ Problem
Optimization Algorithm
Types of Optimization Problems:
- find w that minimizes/maximizes a continuous objective fn $f(w)$
Finding global minimum is hard
For smooth f: unconstrained
- Gradient descent
  - blind
  - w linesearch
  - stochastic
- nonlinear conjugate gradient
- Newton's method
Nonsmooth f:
- Gradient descent
  - blind
  - direct line search
Linesearch finds a local minimum by solving an optimization problem in 1D

Constrained Optimization
- w that optimizes $f(w)$ subject to $g(w) = 0$, g is a smooth fn
→ Lagrange multipliers: transform constrained to unconstrained optimization.
Linear Program - $c \cdot w$, subject to $Aw \leq b$
- linear objective fn + linear inequality constraints
- Set of points that satisfy all constraints is a convex polytope called feasible region
- optimum $\in$ F is point furthest in direction c.
Active Constraints: constraints for which optimum achieves equality.
- set of optimal pts always convex
Linearly separable iff feasible region isn't empty set
Hard: figuring out active constraints
- much more discrete than continuous unconstrained
Quadratic Program
- quadratic, convex objective fn + linear inequality constraint
$f(w) = w^T Q w + c^T w$, subject to $Aw \leq b$
Q is PD ⇒ 1 local minimum
Convex Program
- convex objective fn + convex inequality constraints

Posterior: $P(Y=y \mid X)$
Prior: $P(Y=y)$
Loss fn specifies badness for each incorrect prediction.
$R(r) = E[L(r(X), Y)]$
Bayes Decision Rule: - assumes no loss for correct prediction
$r^*(x) = \begin{cases} 1 & \text{if } L(-1,1) P(Y=1 \mid X=x) > L(1,-1) P(Y=-1 \mid X=x) \\ -1 & \text{ow} \end{cases}$
When L is symmetric, pick class w biggest posterior probability
PDF: $P(x)$
$E[f(x)] = \int_{-\infty}^{\infty} f(x) P(x)\, dx$
$Var = E[(X-\mu)^2] = E[X^2] - \mu^2$
$\mu = E[X] = \int_{-\infty}^{\infty} x P(x)\, dx$
Bayes decision rule



look up x, pick curve w highest probability

Classifiers
1) generative models e.g. LDA
- assume points come from probability distr, different for each class
- fit distr params to class C points, giving $P(X \mid Y=c)$
- estimate $P(Y=c)$
- gives $P(Y \mid X)$

0-1 loss - pick class c that maximizes $P(Y=c \mid X=x)$
→ maximizes $P(X=x \mid Y=c) P(Y=c)$
2) discriminative models e.g. logistic regression model $P(Y \mid X)$ directly
3) Find decision boundary
- model $r(x)$ directly (no posterior)
1+2
$P(Y \mid X)$ tells you probability your guess is wrong
1
- can diagnose outliers: $P(X)$ very small
1
- hard to estimate distr accurately
Generative Model
- most popular when you have phenomena well approximated by normal distr + lots of sample points

Gaussian Discriminant Analysis   models $P(Y=y \mid X)$ as logistic
- each class comes from normal distr
$X \sim N(\mu, \sigma^2)$: $P(x) = \frac{1}{\sqrt{2\pi}\,\sigma^d} \exp\left(-\frac{|x-\mu|^2}{2\sigma^2}\right)$
Bayes decision rule returns class C that maximizes $P(X=x \mid Y=C) P(Y=C)$
$Q_c(x) = \ln\left((\sqrt{2\pi})^d P(x) \pi_c\right) = \frac{-|x-\mu_c|^2}{2\sigma_c^2} - d \ln \sigma_c + \ln \pi_c$
QDA allows you to determine probability that your classification is correct
$P(Y=C \mid X=x) = s(Q_c(x) - Q_D(x))$, $s = \frac{1}{1+e^{-\gamma}}$
LDA:
- linear decision boundaries, less likely to overfit
Assume all Gaussians have same var σ
$Q_C(X) - Q_D(X) = \frac{(\mu_c - \mu_D) \cdot x}{\sigma^2} - \frac{|\mu_c|^2 - |\mu_D|^2}{2\sigma^2} + \ln \pi_c - \ln \pi_D$
$= w \cdot x + \alpha$. } decision boundary
Choose class that maximizes linear discriminant fn
$\frac{\mu_c \cdot x}{\sigma^2} - \frac{|\mu_c|^2}{2\sigma^2} + \ln \pi_c$

MLE to use GDA
- given $X_1, \dots, X_n$, find best fit Gaussian
$\mathcal{L}(\mu, \sigma; X_1, \dots, X_n) = P(X_1) \dots P(X_n)$
- maximize log likelihood.
$\ell(\mu, \sigma) = \ln P(X_1) + \dots + \ln P(X_n)$
$= \sum_{i=1}^{n} \left(\frac{-|X_i - \mu|^2}{2\sigma^2} - d \ln \sqrt{2\pi} - d \ln \sigma\right)$
$\nabla_\mu \ell = \sum_{i=1}^{n} \frac{X_i - \mu}{\sigma^2} = 0$
$\frac{\partial \ell}{\partial \sigma} = \sum_{i=1}^{n} \frac{|X_i - \mu|^2 - d\sigma^2}{\sigma^3} = 0$.
Use mean + variance of points in class C to estimate mean + variance of Gaussian for class C.
QDA: separate conditional mean + variance for each class C
$\hat{\pi}_c = \frac{n_c}{\sum_D n_D}$
LDA: one variance for all classes
↳ within class
→ use each point's distance from its class' mean
$\sigma^2 = \frac{1}{dn} \sum_{c} \sum_{i: y_i = c} |X_i - \hat{\mu}_c|^2$

$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i$
$\hat{\sigma}^2 = \frac{1}{dn} \sum_{i=1}^{n} |X_i - \mu|^2$

Eigenvector $Av = \lambda v$
- v points in same direction after being multiplied by A
$A^k v = \lambda^k v$ $A^{-1} v = \frac{1}{\lambda} v$
Spectral Theorem: every real, symmetric $n \times n$ has real eigenvalues and n eigvec that are mutually orthogonal
- if 2 eigvec have same eigval, every linear combination of those eigvec are also an eigvec
Quadratic Form: shows how applying the matrix affects length of a vector.
$|Ax|^2 = x^T A^2 x$
~~Ellipsoid radii are the reciprocals of eigenvalues; matrix A^{-1} maps spheres to ellipsoids~~
bigger eigval ⇔ steeper hill ⇔ shorter ellipsoid radius.
A is diagonal ⇔ eigvec are coordinate axes ⇔ ellipsoids are axis aligned
Positive definite: $w^T B w > 0$ $\forall w \neq 0$ ⇔ $\lambda > 0$.
PSD: $w^T B w \geq 0$ $\forall w$ ⇔ $\lambda \geq 0$.
Indefinite: $\lambda > 0$, $\lambda < 0$.
Invertible: $\lambda \neq 0$.
+ eigval: curvature goes up
- eigval: curvature goes down
$A = V \Lambda V^T = \sum_{i=1}^{n} \lambda_i v_i v_i^T$
diagonal   ↳ rotate to be axis aligned
$A^2 = V \Lambda V^T V \Lambda V^T = V \Lambda^2 V^T$
$M^{1/2} = A$
1) compute eigvec/val of M
2) take square roots of eigval
3) reassemble A w square roots as eigval, same eigvec

$\boxed{P(x)}_{\text{multivar}} = \frac{1}{(\sqrt{2\pi})^d \sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$, $d = \#$ features
Σ: covar matrix
Σ^{-1}: precision matrix
$P(x) = n(q(x))$, $q(x) = (x-\mu)^T \Sigma^{-1} (x-\mu)$
$q(x)$ is squared distance from $\Sigma^{-1/2} x$ to $\Sigma^{-1/2} \mu$
$Cov(R, S) = E((R - E(R))(S - E(S))^T) = E(RS^T) - \mu_R \mu_S^T$
R_i, R_j independent ⇒ $Cov(R_i, R_j) = 0$.
$Cov(R_i, R_j) = 0$ && multivar normal ⇒ R_i, R_j independent
All features pairwise independent ⇒ $Var(R)$ is diagonal
ow!
$Var(R) = \begin{bmatrix} Var(R_1) & Cov(R_1, R_2) & \dots & Cov(R_1, R_d) \\ Cov(R_2, R_1) & \ddots & & \vdots \\ \vdots & & \ddots & \\ Cov(R_d, R_1) & \dots & & Var(R_d) \end{bmatrix}$

$\Sigma = V \Gamma V^T$. eigval of Σ are variances along the eigvec, $\gamma_{11} = \sigma_1^2$
$\Sigma^{1/2} = V \Gamma^{1/2} V^T$: maps spheres to ellipsoids
- eigval are Gaussian widths / ellipsoid radii / standard deviations
Isocontours of multivariate normal distr are same as isocontours of quadratic form of Σ^{-1}.

Anisotropic Gaussian
QDA: $\hat{\Sigma}_c = \frac{1}{n_c} \sum_{i: y_i = c} (X_i - \hat{\mu}_c)(X_i - \hat{\mu}_c)^T$
LDA: $\hat{\Sigma} = \frac{1}{n} \sum_{c} \sum_{i: y_i = c} (X_i - \hat{\mu}_c)(X_i - \hat{\mu}_c)^T$
QDA: choosing C that maximizes $P(X=x \mid Y=c) \pi_c$ is equivalent to maximizing quadratic discriminant fn
$Q_c(x) = -\frac{1}{2}(x-\mu_c)^T \Sigma_c^{-1} (x-\mu_c) - \frac{1}{2} \ln |\Sigma_c| + \ln \pi_c$

For anisotropic: decision boundary could be hyperbola

* need to apply logistic fn to find decision boundary

LDA: Decision fn is linear, decision boundary is hyperplane.

Maximize linear discriminant fn:

$\mu_c^T \Sigma^{-1} x - \frac{1}{2}\mu_c^T \Sigma^{-1}\mu_c + \ln \pi_c$

For 2 classes:

LDA has d+1 params

QDA has $\frac{d(d+3)}{2}+1$ params

With features, LDA can give nonlinear boundaries, QDA nonquadratic

· LDA/QDA work well when data can only support simple decision boundaries such as linear/quadratic, bc Gaussians provide stable estimates

X: n×d design matrix of sample pts

centering: subtracting μ^T from each row of X $\rightarrow \dot{X}$

$Var(R) = \frac{1}{n}\dot{X}^T\dot{X}$

decorrelating $\dot{X}$: applying $Z = \dot{X}V$, $Var(R) = V\Delta V^T$

↳ transforms sample points to eigvec coordinate system

$\rightarrow Var(Z) = \Delta$

sphering $\dot{X}$: applying transform $W = \dot{X}\,Var(R)^{-1/2}$

whitening X: centering + sphering, $X \rightarrow W$

W has covariance matrix I

whitening: puts features on equal basis

Regression

· given point x, predict a numerical value

Regression Fns: linear: $h(x; w, \alpha) = w \cdot x + \alpha$

polynomial (equivalent to linear w poly features)

logistic $h(x; w, \alpha) = s(w \cdot x + \alpha)$

Loss Fns: z = prediction h(x), y = true val

Squared Error: $L(z, y) = (z-y)^2$

Absolute Error: $L(z, y) = |z - y|$

Cross Entropy: $L(z, y) = -y \ln z - (1-y)\ln(1-z)$

Cost Fns: (minimize)

$J(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(X_i), y_i)$ mean loss

$J(h) = \max_{N} L(h(X_i), y_i)$ max loss

$J(h) = \sum_{i=1}^{n} w_i L(h(X_i), y_i)$ weighted sum

$J(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(X_i), y_i) + \lambda \|w\|^2$ L2 regularized

$J(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(X_i), y_i) + \lambda \|w\|$ L1 regularized

Least Squares Linear Regression

Find w that minimizes $|Xw - y|^2 = RSS(w)$

Soln: $X^T X w = X^T y$ $\quad = \sum_{i=1}^{n}(X_i \cdot w + \alpha - y_i)^2$

Let $X^+ = (X^T X)^{-1} X^T$

$\hat{y}_i = w \cdot X_i \Rightarrow \hat{y} = Xw = XX^+y = Hy$

Xw is subspace of $\mathbb{R}^n$ spanned by columns

$X \in \mathbb{R}^{n \times d}$ at most d+1 dim

y ∈ n dim space

Minimizing $|\hat{y} - y|^2$ finds $\hat{y}$ nearest y on subspace = orthogonal projection.

Advantages:

- easy to compute; just solve a linear system
- unique, stable soln.

Disadvantages:

- sensitive to outliers
- fails if $X^T X$ is singular

Logistic Regression - discriminative

· fits probabilities in range (0,1)

· used for classification

no closed form soln

Find w that minimizes:

$J = -\sum_{i=1}^{n} \big(y_i \ln s(X_i \cdot w) + (1-y_i)\ln(1 - s(X_i \cdot w))\big)$

$s'(\gamma) = s(\gamma)(1 - s(\gamma))$

$\nabla_w J = -X^T(y - s(Xw))$

Gradient Descent Rule: $w \leftarrow w + \epsilon X^T(y - s(Xw))$

SGD: $w \leftarrow w + \epsilon(y_i - s(X_i \cdot w))X_i$

Least Squares Polynomial Regression

- replace each X_i with vector

$\Phi(X_i) = [X_{i1}^2 \; X_{i1}X_{i2} \; X_{i2}^2 \; X_{i1} \; X_{i2} \; 1]^T$

Weighted Least Squares Regression

· assign trusted sample points a higher weight ω

Greater $w_i \rightarrow$ work harder to minimize $|\hat{y}_i - y_i|^2$

Find w that minimizes

$(Xw - y)^T \Omega (Xw - y)$

$= \sum_{i=1}^{n} w_i (X_i \cdot w - y_i)^2$

Solve for w in normal eqn:

$X^T \Omega X w = X^T \Omega y$

Newton's Method - iteratively reweighted least squares

· iterative optimization method for smooth fn J(w)

· faster than gradient descent

Idea: at point v.

Approximate J(w) near v by quadratic fn. Jump to critical pt.

pick starting point w

repeat till convergence

$e \leftarrow (\nabla^2 J(w))e = -\nabla J(w)$

$w \leftarrow w + e$

The closer J is to quadratic, the faster Newton's converges.

· doesn't know difference btwn optima, must start close enough to soln.

Advantages

· tries to find right step length to reach min

· tries to choose better descent direction than just steepest descent

Disadvantages

· computing Hessian is expensive

· doesn't work for nonsmooth fns

Newton's for logistic regression.

$s_i = s(X_i \cdot w)$

$\nabla_w(J) = -X^T(y - s)$

$\nabla_w^2 J(w) = X^T \Omega X$, $\Omega = \begin{bmatrix} s_1(1-s_1) & & 0 \\ & \ddots & \\ 0 & & s_n(1-s_n) \end{bmatrix}$

$e \leftarrow (X^T \Omega X)e = X^T(y - s)$

Ω prioritizes points with $s_i \approx 0.5$, tunes at pts near 0/1

⇒ sample pts near decision boundary have biggest effect on iterations, make big contribution to logistic fit.

If n very large, save time by using a random subsample of pts per iteration, increase sample size as you go.

LDA vs Logistic Regression

LDA:

· stable for well-separated classes

· more accurate when classes nearly normal

Logistic Regression

· more emphasis on decision boundary (LDA gives equal weight)

· less sensitive to most outliers

· easy treatment of partial membership - LDA pts all or nothing

· more robust on non Gaussians

ROC curve evaluates classifier after it's trained

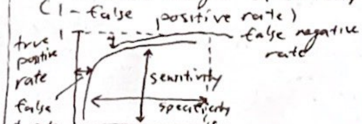
· shows rate of false positive vs ~~false negative~~ true positive

X-axis: false positive rate

Y-axis: true positive rate (sensitivity)

false-negative rate: vertical distance from curve to top (1-sensitivity)

true negative rate: horizontal distance from curve to right "specificity" (1 - false positive rate)



Upper Right Corner: always classify +

Lower Left: always classify −

diagonal: random classifiers

Classifiers' effectiveness = area under curve

Always right = 1.

Random = 1/2.

Model of Reality:

sample pts from unknown prob distr

y-values are sum of unknown, non-random fn + random noise

$\forall X_i,\; y_i = f(X_i) + \epsilon_i,\; \epsilon_i \sim D',\; \sim \mu = 0$

Risk for hypothesis h is expected loss

$R(h) = E[L]$

Empirical Distr: discrete uniform dist over sample pts.

Empirical Risk: expected loss under empirical distr

$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(X_i), y_i)$

~ Max likelihood explains where logistic loss fn comes from

2 sources of error in h

bias: error due to inability of h to fit f perfectly

variance: error due to fitting random noise in data

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

Underfitting: too much bias

Overfitting: too much variance

· variance → 0 as n → ∞

· Adding good feature reduces bias; adding bad feature rarely ~~[illegible]~~ increases

· adding a feature increases variance

· noise in test set affects only $Var(\epsilon)$

· noise in training affects bias + Var(h)

Ridge Regression

Find w that minimizes

$|Xw - y|^2 + \lambda |w'|_2^2$,

w' is w w α replaced by 0.

· regularization terms guarantees PD ⇒ unique soln

· many minima = ill-posed

· regularization reduces variance

Ideally: features "normalized" to have same variance

↳ $(X^T X + \lambda I')w = X^T y$

Subset Selection

· all features increase variance, but not all features reduce bias

· identify poorly predictive features, ignore them

· less overfitting, smaller test error

Alg: try all $2^d - 1$ nonempty subsets of features

→ train one classifier per subset

choose best classifier by cross-validation.

Heuristic 1: Forward stepwise selection

· repeatedly add best feature until validation errors start increasing $O(d^2)$

Heuristic 2: Backward stepwise selection

· start w all d features, remove feature whose removal gives best reduction in validation error $O(d^2)$

3: try to remove features w small weights

Lasso

· often naturally sets some weights to zero

Find w that minimizes $|Xw - y|^2 + \lambda \|w'\|_1$

$\|w'\|_1 = \sum_{i=1}^{d} |w_i|$

λ↑, more + more weights go to 0.

$\nabla f(x) = \frac{\partial f}{\partial x}^T$ $\quad (A^T x)_i = \sum_{j=1} A_{ij}^T x_j$

$(Ax)_i = \sum_{j=1}^{d} A_{ij} x_j$

$f(x + \Delta) = f(x) + \frac{\partial f}{\partial x}\Delta + o(\|\Delta\|)$

Gradient = $\frac{\partial f}{\partial x}^T$

Posterior: $P(Y = y \mid x)$

$\theta_{MLE} = \arg\max_\theta P(X \mid \theta)$

$= \arg\max_\theta \prod_{i=1}^{n} P(x_i \mid \theta)$

$\theta_{MAP} = \arg\max_\theta P(\theta \mid X)$

$= \arg\max_\theta P(X \mid \theta) P(\theta)$

$= \arg\max_\theta \prod_{i=1}^{n} P(x_i \mid \theta) P(\theta)$

Covariance matrix must be PSD

Gaussian prior = L2 reg.

$Cov = \frac{Corr}{\sigma_{x_1}\sigma_{x_2}}$ Laplace prior = L1 reg

determinant = product of eigval

PDF of univariate Gaussian:

$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$

Bayes:

$P(Y \mid X) = \frac{P(X \mid Y) \cdot P(Y)}{P(X)}$

Maximum a posteriori

- maximizing posterior $P(w \mid X, y)$

length of ellipsoid axes for multivariate Gaussian are $\sqrt{\lambda_i}$ of Σ

- train on more data to improve training accuracy, less data to improve test
- Bayes risk is 0 when class distr don't overlap, prior for one class = 1
- add λI to covariance matrix causes isocontours to be more spherical

$s(0) = 1/2$ is decision boundary

No covariance terms: isocontour is axis aligned

- space between contours is $\sqrt{\lambda_i}$ of Σ
- higher SD ⇒ larger gaps between significant isocontours.

· eigenvectors form orthonormal basis

* tell if something is convex by whether its Hessian is positive definite.

$A^T A = |A|^2$

· LDA maximizes between class variance over within class variance

Shrinkage: minimize |w|

Support vectors:

· necessary to compute f(x) in SVM

· have nonzero weight

Hessian: symmetric

$\begin{matrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{matrix}$