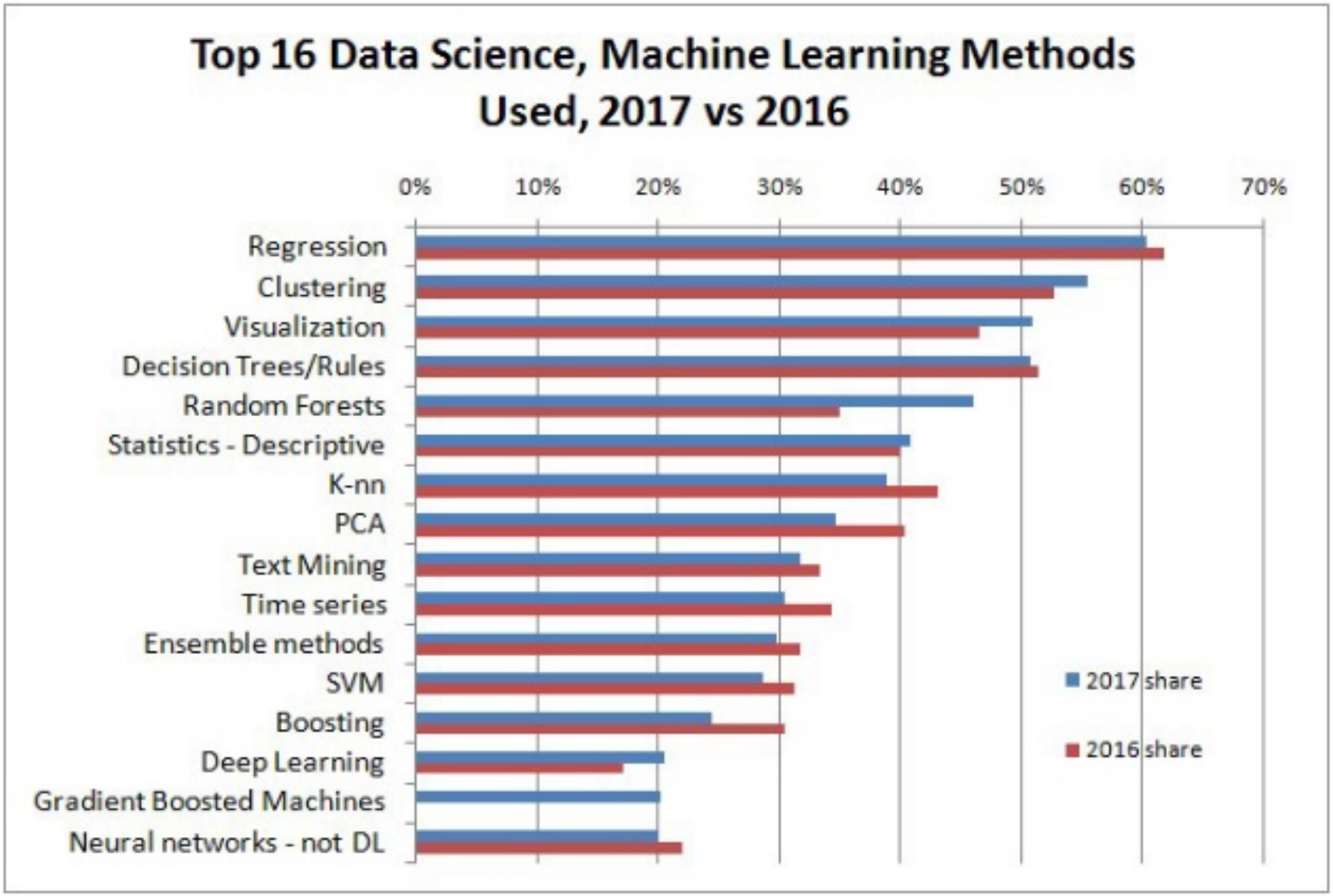
Chapter A. ML Applications

Prof. Jaewook Lee
Statistical Learning and Computational Finance Lab.
Department of Industrial Engineering
jaewook@snu.ac.kr
http://slcf.snu.ac.kr

Top Machine Learning Methods



Top Journals and Conferences in Machine Learning

- **Top journals & conferences in machine learning, pattern recognition, artificial intelligence, and data mining**

Top journals in ML, PR, AI, DM	Field Rating
PAMI - IEEE Transactions on Pattern Analysis and Machine Intelligence	246
NECO - Neural Computation	140
ML - Machine Learning	137
NN - IEEE Transactions on Neural Networks	136
PR - Pattern Recognition	112
TKDE - IEEE Transactions on Knowledge and Data Engineering	109
JMLR - Journal of Machine Learning Research	92

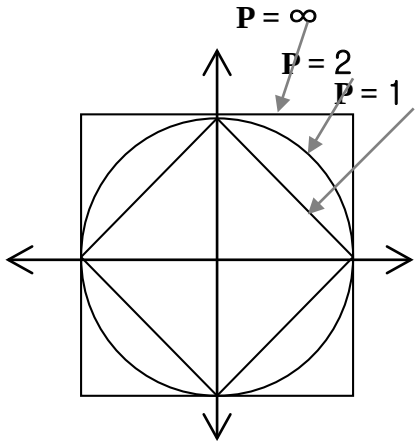
Top conferences in ML, PR, AI, DM	Field Rating
NIPS - Neural Information Processing Systems	135
AAAI - National Conference on Artificial Intelligence	132
ICML - International Conference on Machine Learning	127
KDD - Knowledge Discovery and Data Mining	122

source: http://academic.research.microsoft.com/

DISTANCE & KNN

Measures of Distance, Dissimilarity, and Density

- Distance Measures
  - Euclidean Distance $$d_{ij} = \left[\sum_k (x_{ik} - x_{jk})^2 \right]^{1/2}$$
  - Minkowski p- (or L^p) Metric $$d_{ij}(p) = \left[\sum_k \left| x_{ik} - x_{jk} \right|^p \right]^{1/p}$$
  - Mahalanobis Distance $$D_{ij}^2 = (\mathbf{x}_i - \mathbf{x}_j)' \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j)$$



  - Other Measures

    Caution about Correlation :

    Correlation is a measure of covariation, not necessarily of similarity.

    (1, 2, 1, 2) vs. (9, 10, 9, 10) ➔ corr=1

    (1, 2, 1, 2) vs. (1, 1, 2, 2) ➔ corr=0

Measures of Distance, Dissimilarity, and Density

Other Distance Measures

- Matching Measures: nominal scale.

	Cola flavor	Caffeine	Diet	Manufactured by Coke
Coke	1	1	0	1
Pepsi	1	1	0	0
Diet Coke	1	1	1	1
Caffeine-free diet coke	1	0	1	1

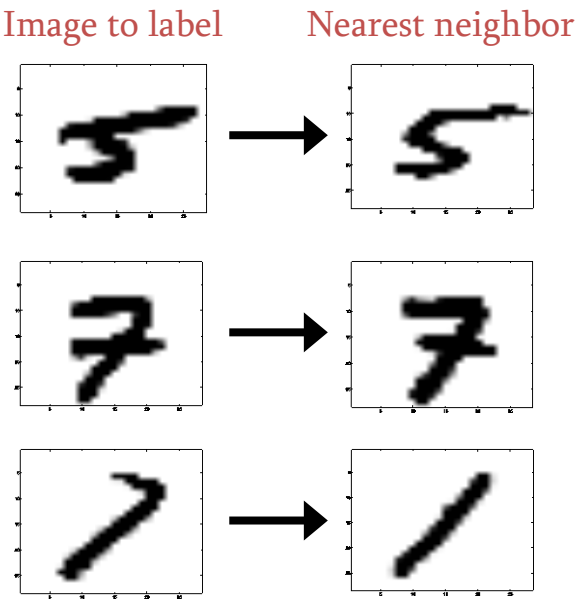
profiles

	Coke	Pepsi	Diet Coke	Caffeine-free diet coke
Coke				
Pepsi	3/4			
Diet Coke	3/4	2/4		
Caffeine-free diet coke	2/4	1/4	3/4	

Similarity measures

K Nearest Neighbor (KNN)

- KNN
  - Training set includes classes.
  - Examine K items near item to be classified.
  - New item placed in class with the most number of close items.
  - O(n) for each tuple to be classified. (Here n is the size of the training set.)

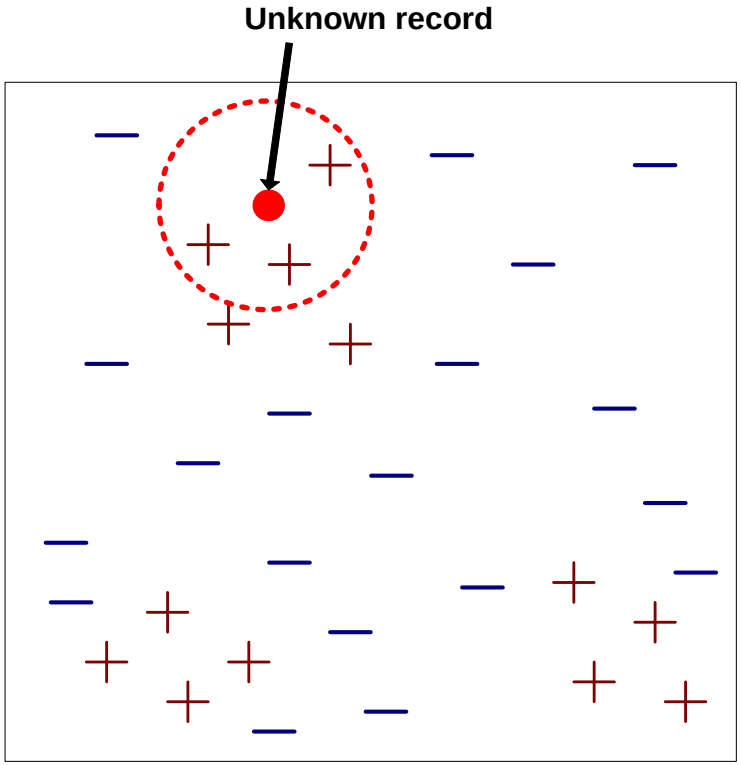


Overall: error rate = 6% (on test set)

Question: what is the error rate for random guessing?

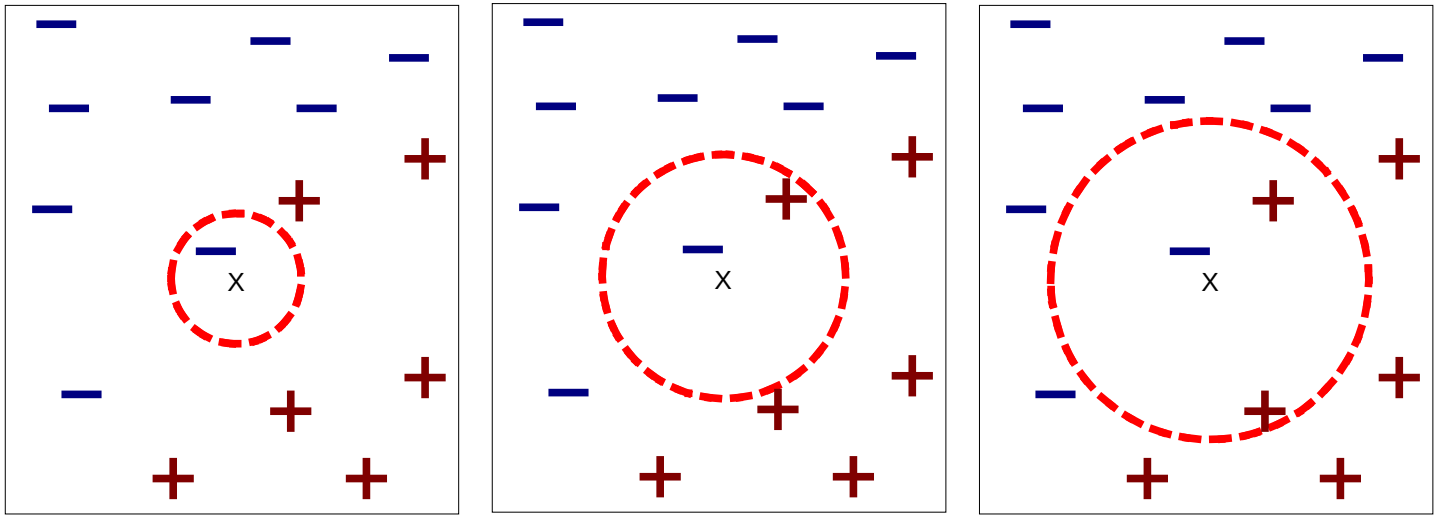
K Nearest Neighbor (KNN)

- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of k, the number of nearest neighbors to retrieve
- To classify an unknown record:
  - Compute distance to other training records
  - Identify k nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)



Definition of Nearest Neighbor

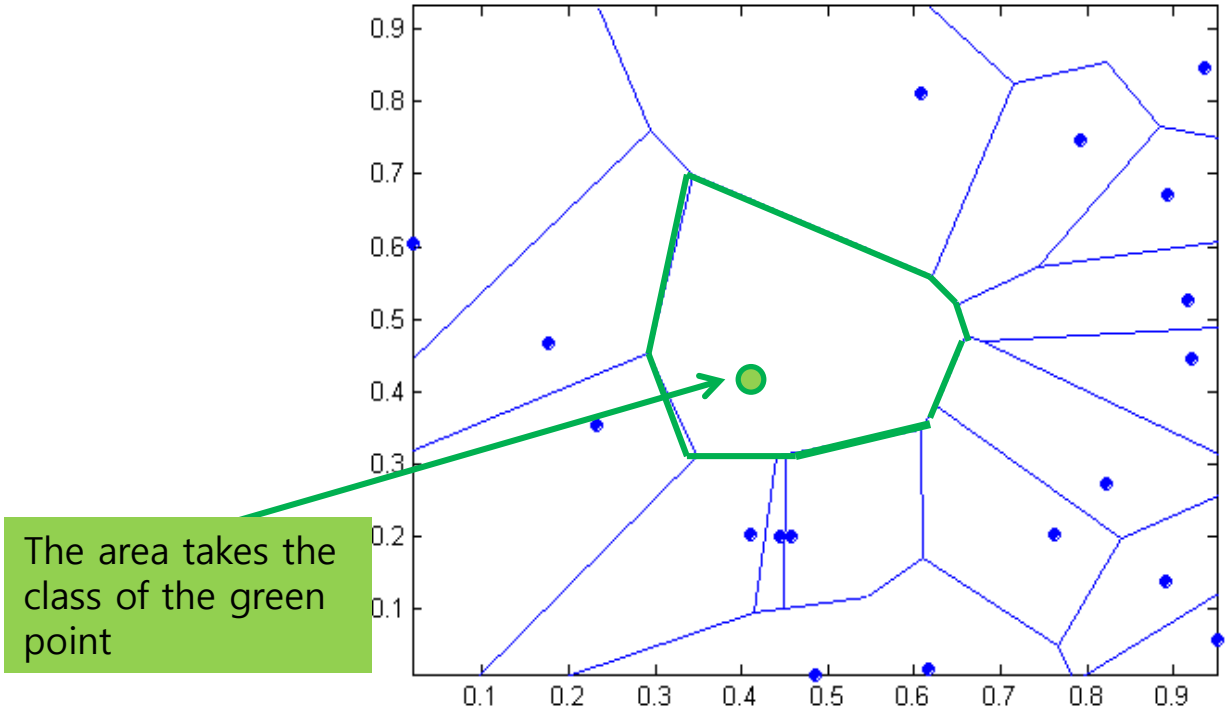
- Definition
  - K-nearest neighbors of a record x are data points that have the k smallest distance to x



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

1 nearest-neighbor

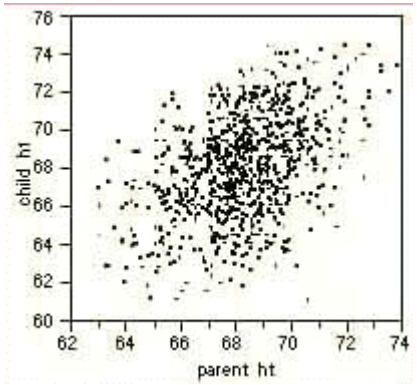
- Voronoi Diagram defines the classification boundary



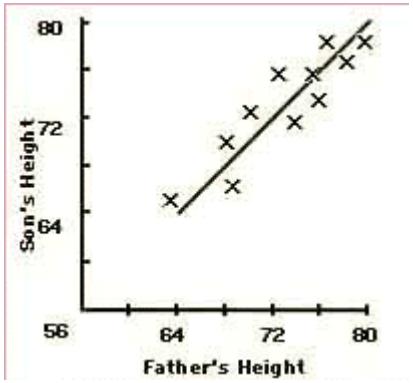
LINEAR REGRESSION

The General Idea - Example

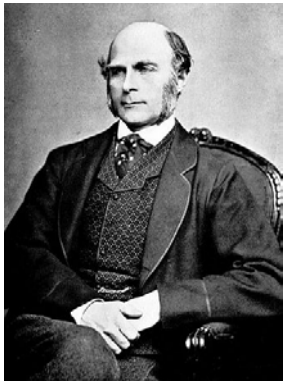
- Regression
  - Empirical – derived from observation rather than theory using a mathematical optimization technique which, when given a series of observed data, attempts to find a function which closely approximates the data ➔ a "best fit".
- Historical Note
  - Sir Francis Galton first used the term regression analysis in a study of the heights of fathers (x) and sons (y). Galton fit a least squares line and used it to predict the son's height from the father's height. He found that if a father's height was above average, the son's height would also be above average, but not by as much as the father's height was. A similar effect was observed for below average heights. That is, the son's height "regressed" toward the average. Consequently, Galton referred to the least squares line as a regression line.



Galton's original data



Hand drawn with regression



The Multiple Regression Model

The Multiple Regression Model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

- More than one regressor or predictor variable.
- Linear in the unknown parameters – the β's; The β_0 - intercept, β_1 - partial regression coefficients, ε – errors.
- Can handle nonlinear functions as predictors, e.g. $X_3 = Z^2$.
- Interactions can be present, e.g. $\beta_{12} X_1 X_2$.

Example – Oakland games won:

13 = β_0 + β_1*2285 + β_2*45.3 +β_3*1903 + ε_1

Team	Games Won	Passing Yds.	% Run Plays	Opp. Rushing Yds.
Oakland	13	2285	45.3	1903
Pittsburgh	10	2971	53.8	1457
Baltimore	11	2309	74.1	1848
Los Angeles	10	2528	65.4	1564
Dallas	11	2147	78.3	1821
Atlanta	4	1689	47.6	2577
Buffalo	2	2566	54.2	2476
Chicago	7	2363	48	1984

Linear Models and Least Squares

Linear Models

- Given a vector of inputs $X = (X_1, X_2, \ldots, X_p)$, we predict the output Y via the model

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

- It is convenient to include the constant variable 1 in X, the intercept ($\hat{\beta}_0$) in the vector of coefficients $\hat{\beta}$, and then write the linear model in vector form as an inner product

$$\hat{Y} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$$

where $\mathbf{X}$ is an $N \times (p+1)$ matrix and β is $(p+1) \times 1$ vector as follows

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^1 & \cdots & x_1^p \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N^1 & \cdots & x_N^p \end{bmatrix} \quad , \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

Linear Models and Least Squares

Least Squares

- To fit the linear model to a set of training data, we use the method of **least squares**. In this approach, we pick the coefficients β to minimize the residual sum of squares

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$$

- or in vector form

$$RSS(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

- Differentiating w.r.t. β we get the *normal equations*

$$\mathbf{X}^T (\mathbf{y} - \mathbf{X}\beta) = 0$$

- If $\mathbf{X}^T\mathbf{X}$ is nonsingular, then the unique solution is given by

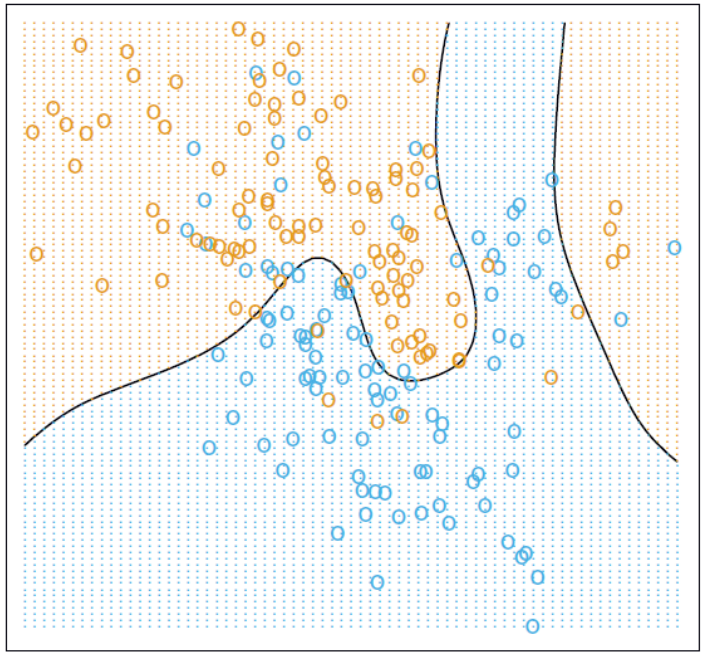
$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- In a classification context, with the response Y coded as 0 or 1, the fitted values $\hat{Y}$ are converted to a fitted class variable $\hat{G}$ according to the rule

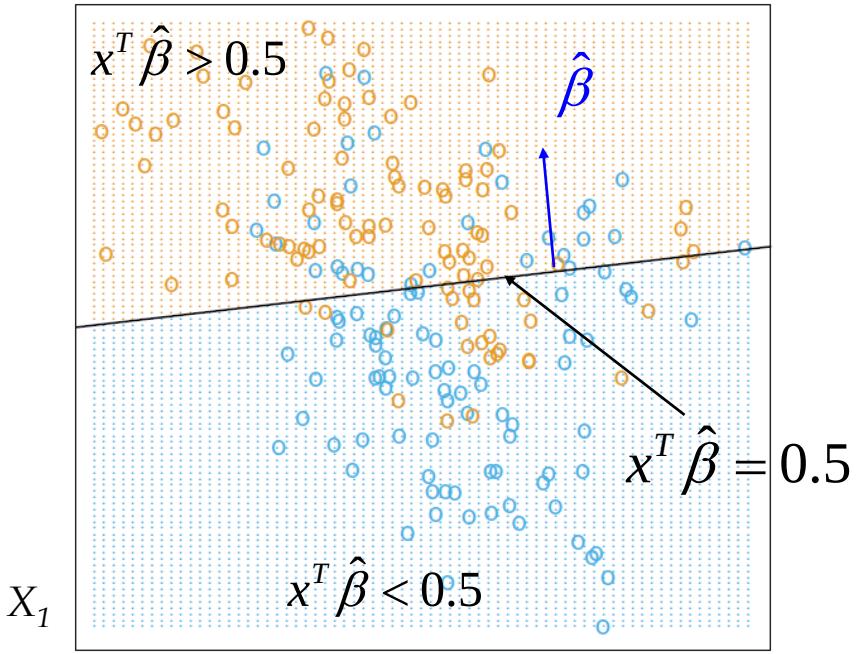
$$\hat{G} = \begin{cases} \text{class1} & if \ \hat{Y} > 0.5 \\ \text{class2} & if \ \hat{Y} \leq 0.5 \end{cases}$$

Result: Linear Regression

Bayes Optimal Classifier



Linear Regression

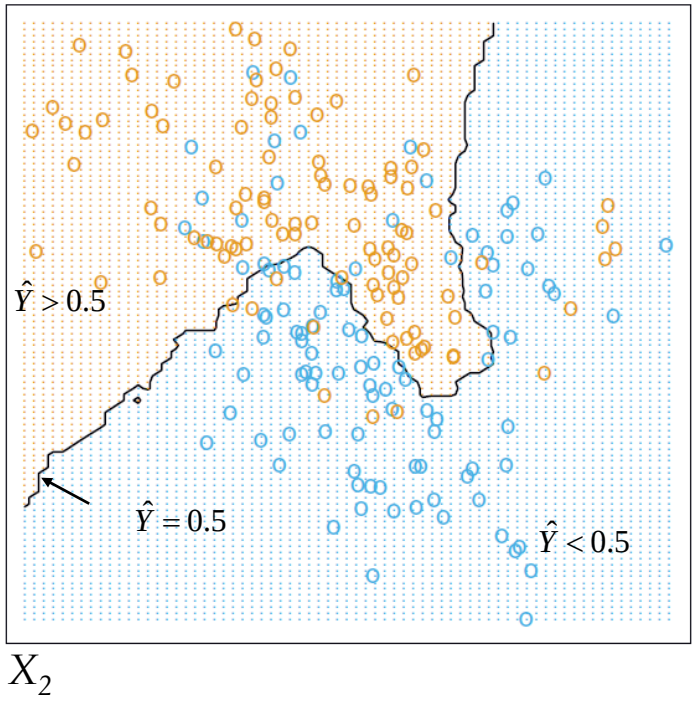


$$f(X) = X^T\beta$$

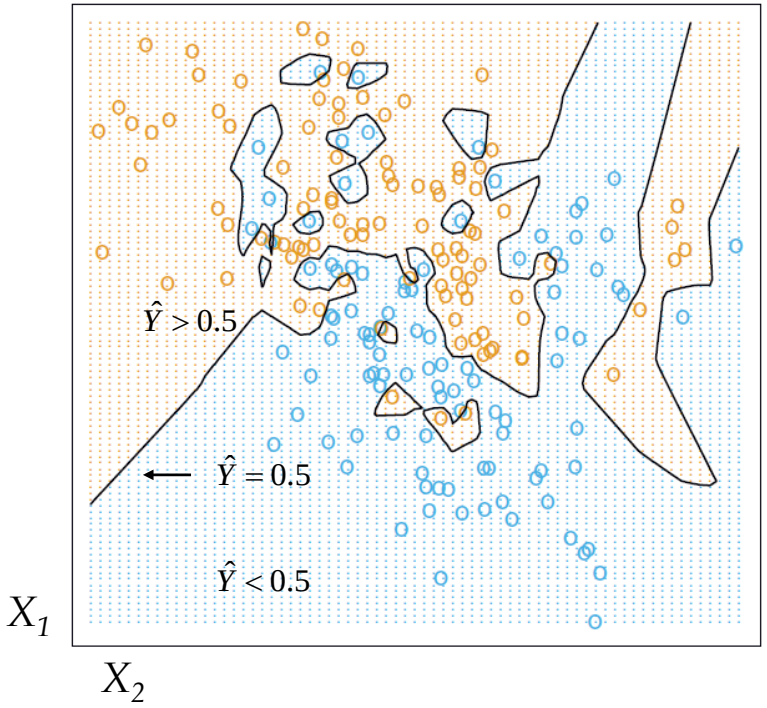
Source: T. Hastie, et al.,
The Elements of Statistical Learning

Result: K-Nearest Neighbors

15-nearest neighbor averaging



1-nearest neighbor averaging



$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

Source: T. Hastie, et al.,
The Elements of Statistical Learning

RIDGE, LASSO, ELASTIC NET

Ridge regression

Ridge regression

- An equivalent way to write the ridge problem is

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s,$$

  which makes explicit the size constraint on the parameters.

  - When there are many correlated variables in a linear regression model, their coefficients can become poorly determined and exhibit high variance. A wildly large positive coefficient on one variable can be cancelled by a similarly large negative coefficient on its correlated cousin. By imposing a size constraint on the coefficients, this phenomenon is prevented from occurring.

- Using centered inputs (i.e. each x_{ij} gets replaced by $x_{ij} - \bar{x}_j$ and β_0 by $\bar{y} = \sum_1^N y_i/N$, $\mathbf{X}$ has p-columns without intercept) and matrix form

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

- The ridge regression solutions are easily seen to be

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

  - Ridge regression solution is again a linear function of $\mathbf{y}$.
  - Ridge estimate is the mode of the posterior distribution; since the distribution is Gaussian, it is also posterior mean.

Lasso regression

- The Lasso estimate is defined by

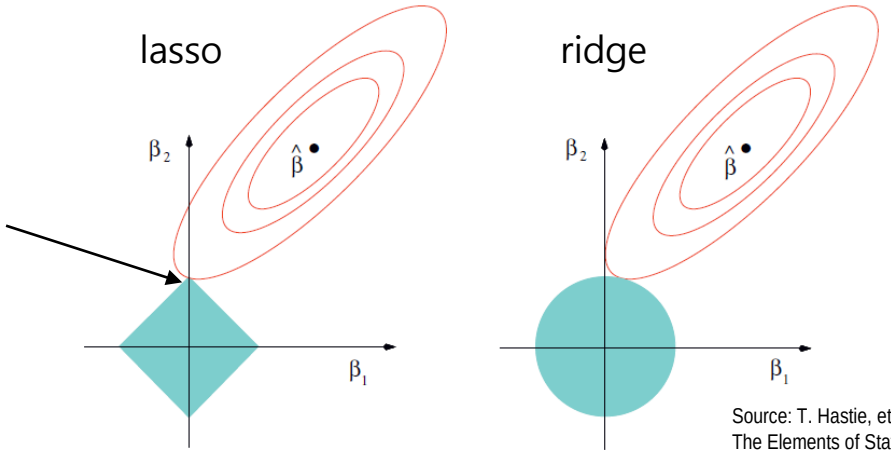
$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \le t$$

  - We can also write the lasso problem in the equivalent *Lagrangian form*

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

    - Just as in ridge regression, we can reparameterize the constant β_0 by standardizing the predictors; the solution for $\hat{\beta}_0$ is $\bar{y}$, and therefore we fit a model without an intercept.
    - The L_1 lasso penalty constraint makes the solutions nonlinear in the y_i, and a quadratic programming algorithm is used to compute them.

- Because of the natures of the constraint, making t sufficiently small will cause some of the coefficient to be exactly zero.
- The lasso does a kind of continuous subset selection. t should be adaptively chosen to minimize an estimate of expected prediction error.



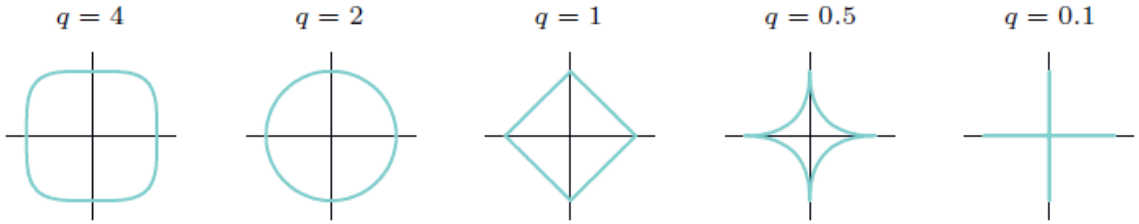
Source: T. Hastie, et al.,
The Elements of Statistical Learning

Elastic net

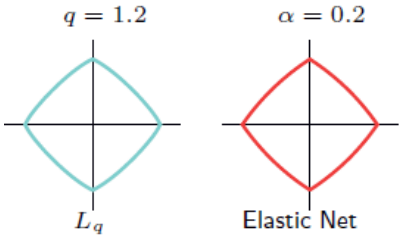
- View ridge regression and the lasso as Bayes estimates.

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q$$

  - Thinking of $|\beta_j|^q$ as the log-prior density for β_j, The value $q = 0$ corresponds to variable subset selection, as the penalty simply counts the number of nonzero parameters; $q = 1$ corresponds to the lasso, while $q = 2$ to ridge regression.



- Elastic net: a compromise between ridge and lasso



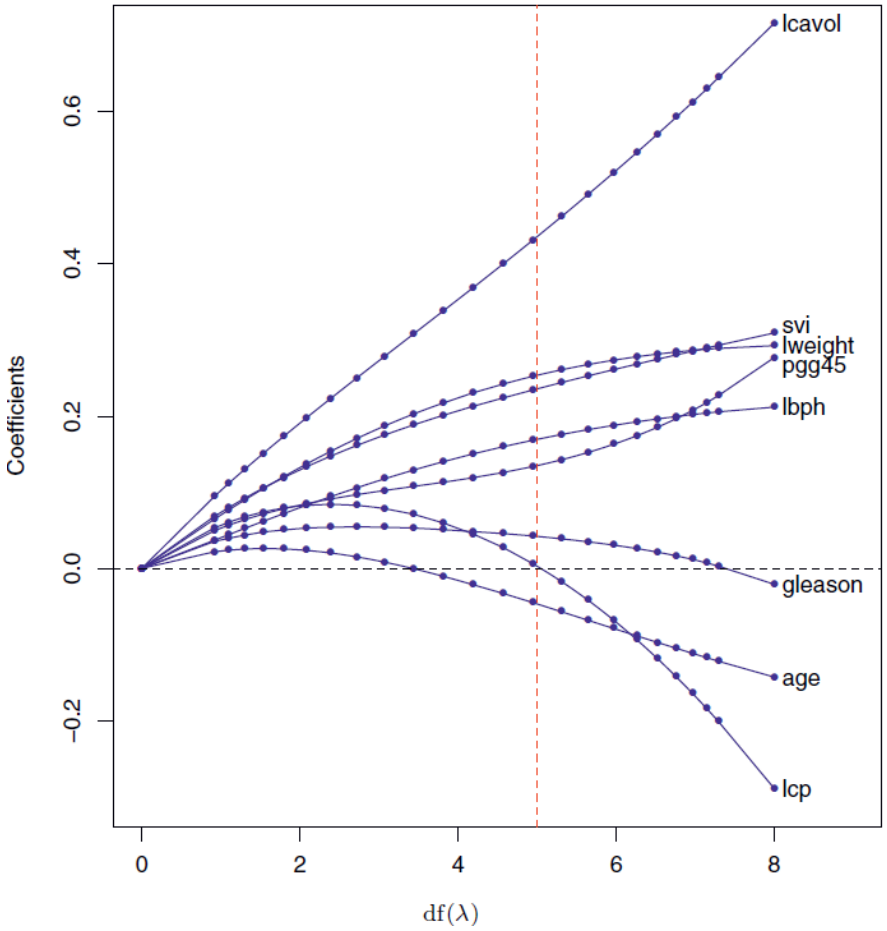
  - Elastic net estimate use the penalty given by

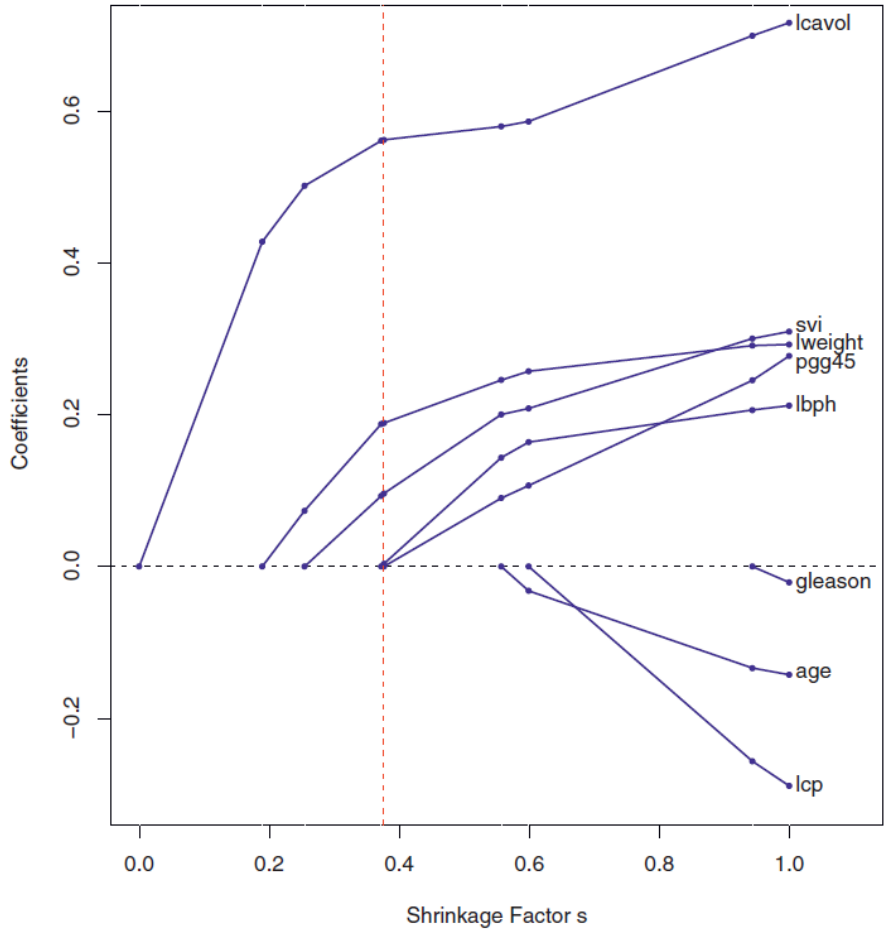
$$\lambda \sum_{j=1}^{p} (\alpha\beta_j^2 + (1-\alpha)|\beta_j|)$$

  - The elastic-net selects variables like the lasso, and shrinks together the coefficients of correlated predictors like ridge. It also has considerable computational advantages over the L_q penalties.

Ridge v.s. Lasso

Ridge v.s. Lasso





$$s = t / \sum_{j=1}^{p} |\beta_j|$$

Source: T. Hastie, et al., The Elements of Statistical Learning

Dimensions

	Scalar	Vector	Matrix
Scalar	$\frac{\mathrm{d}y}{\mathrm{d}x}$	$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x} = \left[\frac{\partial y_i}{\partial x}\right]$	$\frac{\mathrm{d}\mathbf{Y}}{\mathrm{d}x} = \left[\frac{\partial y_{ij}}{\partial x}\right]$
Vector	$\frac{\mathrm{d}y}{\mathrm{d}\mathbf{x}} = \left[\frac{\partial y}{\partial x_j}\right]$	$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\mathbf{x}} = \left[\frac{\partial y_i}{\partial x_j}\right]$	
Matrix	$\frac{\mathrm{d}y}{\mathrm{d}\mathbf{X}} = \left[\frac{\partial y}{\partial x_{ji}}\right]$		

By Thomas Minka. Old and New Matrix Algebra Useful for Statistics

Examples

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

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

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

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

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

http://matrixcookbook.com/

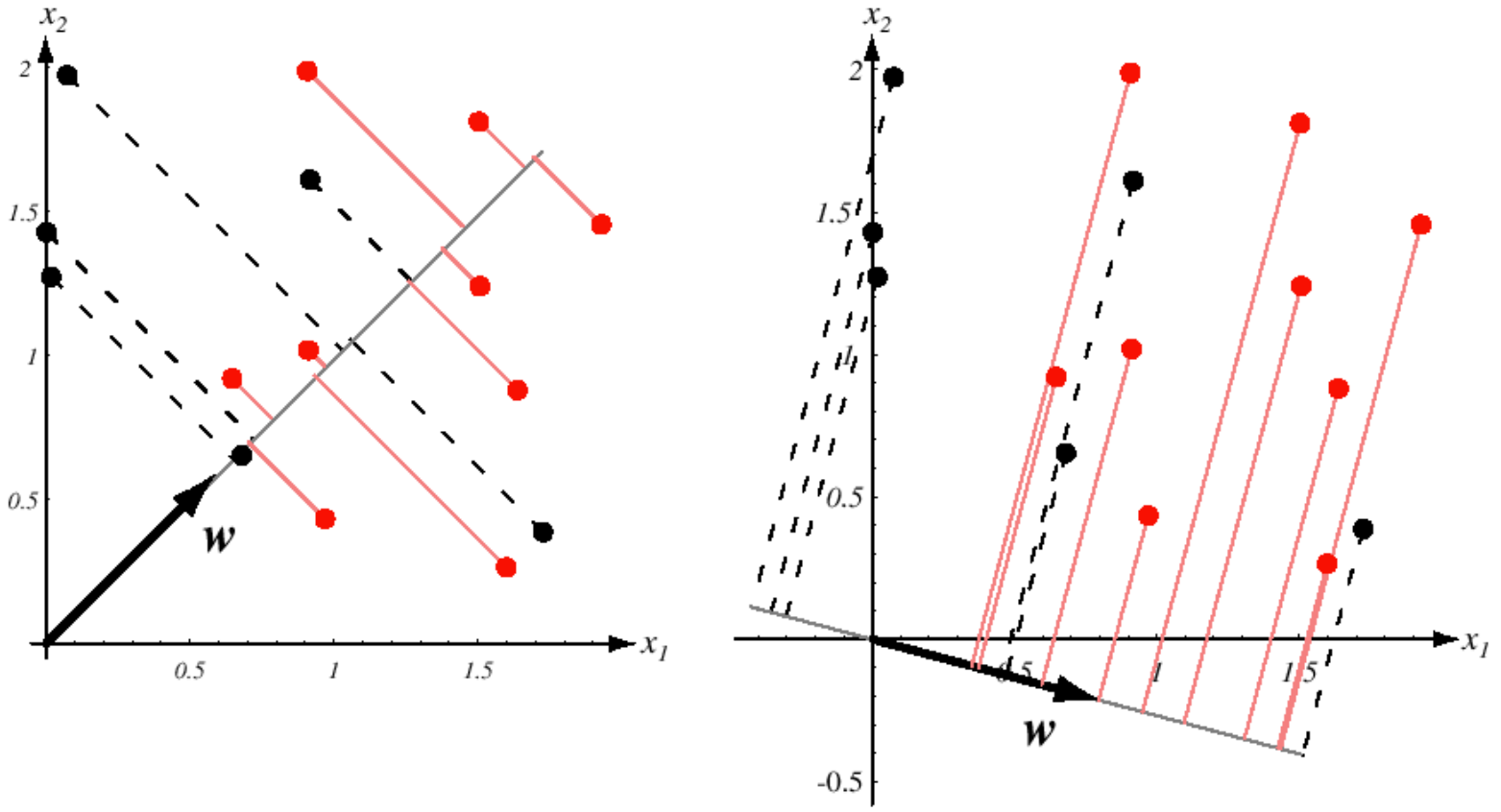
PRINCIPAL COMPONENT ANALYSIS

Principal component analysis: Introduction

- Principal component analysis
  - Method for re-expressing multivariate data.
  - Reorient the data so that the first few dimensions account for as much of the available information as possible
  - Dimension reduction makes visualization of the data more straightforward and subsequent data analysis more manageable
  - Principal component analysis (PCA) is first invented in 1901 by Karl Pearson, but mostly developed (and named) by Harold Hotelling in the 1930s.
  - PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.
  - Each component is uncorrelated with all others, which has the advantage of eliminating multicollinearity.
  - Each principal component is an exact linear combination of the original variables

Principal Component : How it works

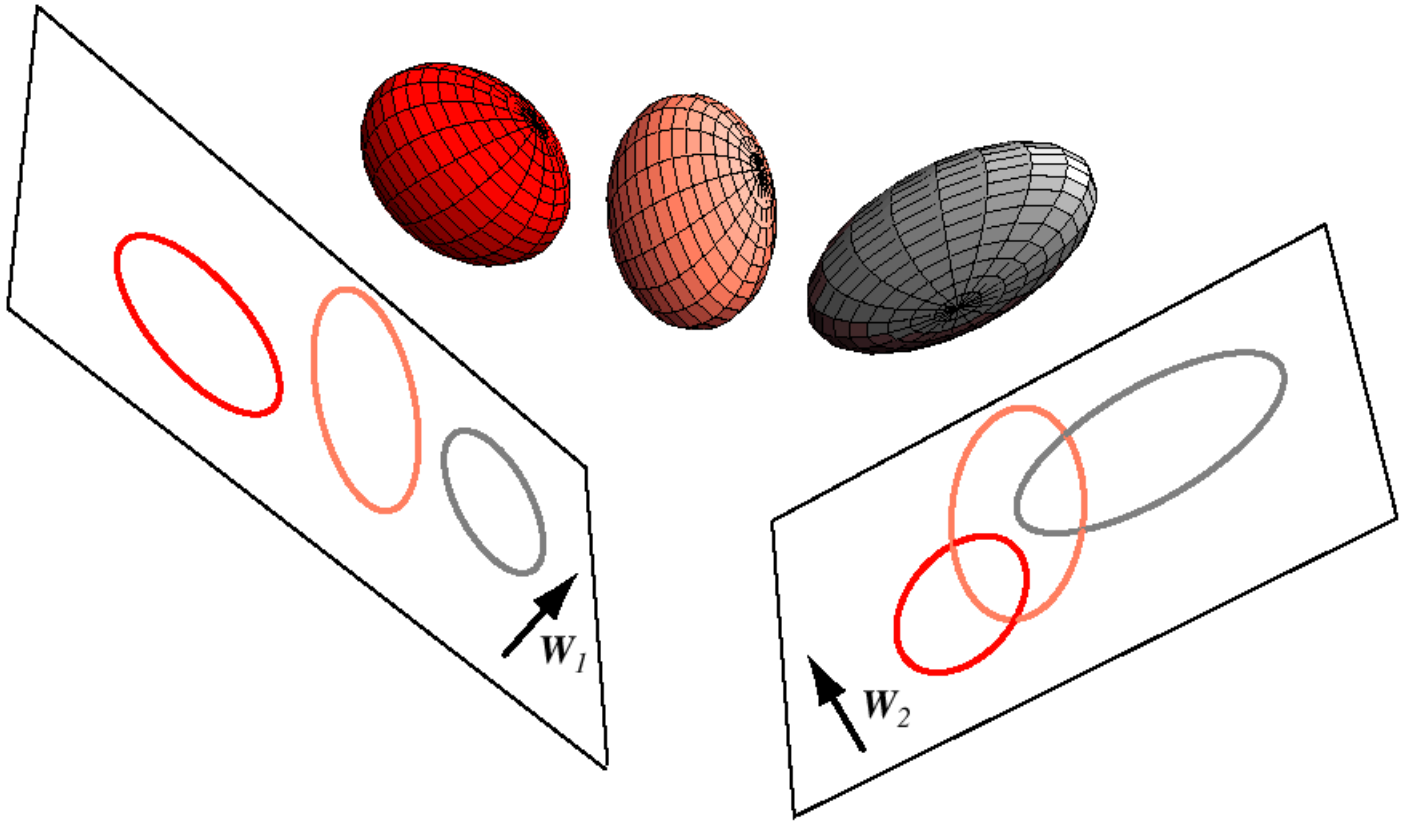
- Intuition



Source: Duda et al. (2001)

Principal Component : How it works

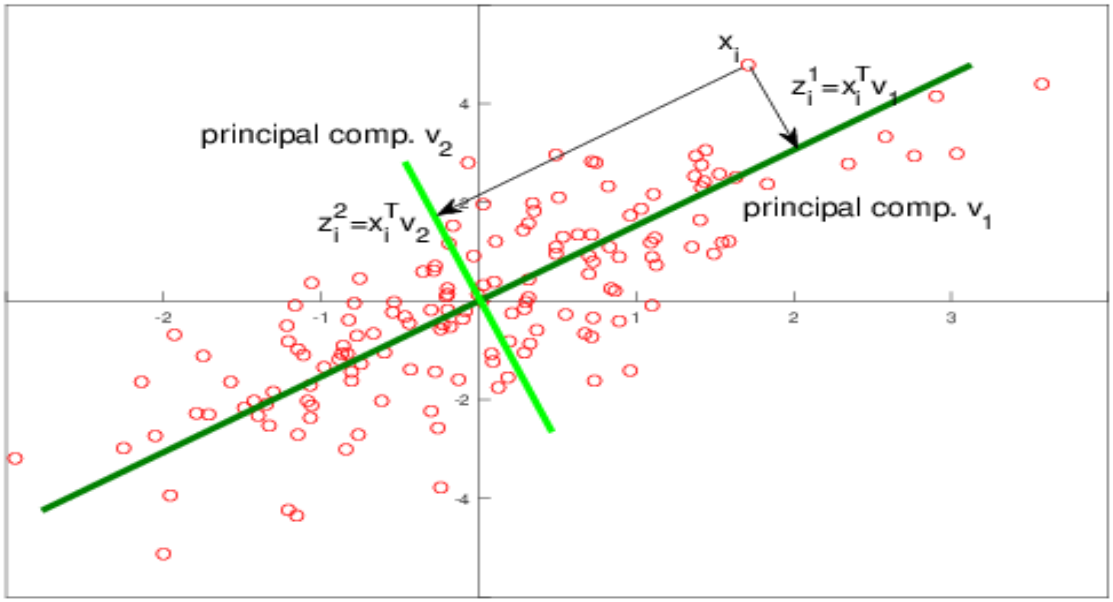
- Intuition



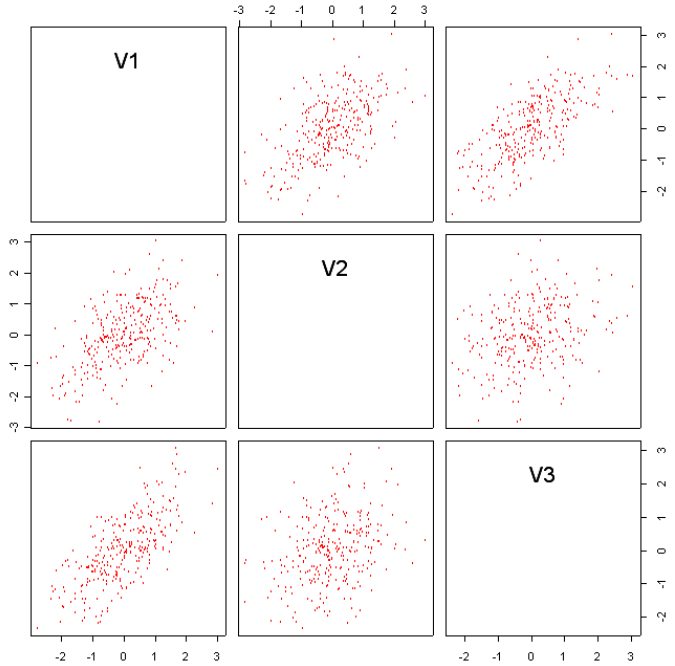
Source: Duda et al. (2001)

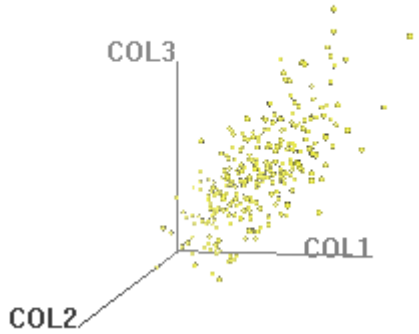
Principal Component : How it works

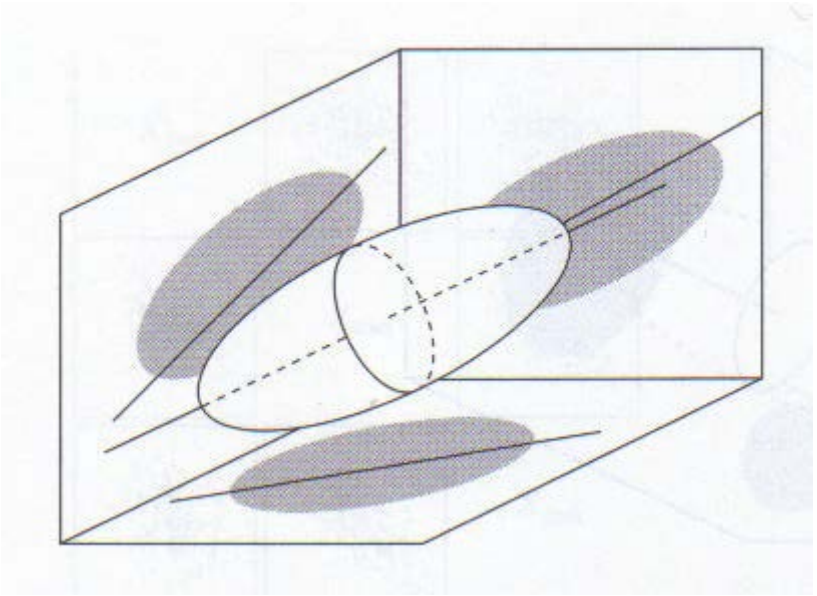
- Intuition
  - PCA : Finding a smaller number of dimensions account for a sufficient amount of the information in the original variables
  - Choose the linear combination with maximal variance, we are accounting for as much of the information contained in $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$



PCA: 3D case







The first component identified

Principal Component : How it works

- How the principal component vectors are determined?
  - Objective : to find a linear combination of the original variables **X** with maximum variance

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

Singular Value Decomposition (SVD)

$$X^T = U\Sigma V^T, \quad XU = V\Sigma = Z$$

  - Goal : to choose $\mathbf{v}$ to maximize the variance of the elements of $\mathbf{z}=\mathbf{Xv}$

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T\mathbf{v} \\ \mathbf{x}_2^T\mathbf{v} \\ \vdots \\ \mathbf{x}_N^T\mathbf{v} \end{bmatrix} = \mathbf{Xv}$$

  - $z_i = P_{\mathbf{v}}\mathbf{x}_i = \mathbf{x}_i^T\mathbf{v}$, $i = 1, \dots, N$, is the orthogonal projection of $\mathbf{x}_i$ onto the subspace spanned by $\mathbf{v}$. The sample variance of the elements of $\mathbf{z}$ ($\sum_{i=1}^N z_i = 0$ since $\mathbf{X}$ is centered) is then

$$\text{var}(\mathbf{z}) = \frac{1}{N-1}\sum_{i=1}^{N} z_i^2 = \frac{1}{N-1}\mathbf{z}^T\mathbf{z} = \frac{1}{N-1}\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v} = \mathbf{v}^T\mathbf{C}\mathbf{v}$$

Principal Component : How it works

- First Principal Component
  - To obtain the first principal component $\mathbf{v}_1$ that maximizes $\mathbf{v}^T\mathbf{C}\mathbf{v}$ subject to the constraint $\mathbf{v}^T\mathbf{v} = 1$, we let $L = \mathbf{v}^T\mathbf{C}\mathbf{v} - \lambda\mathbf{v}^T\mathbf{v}$.
  - $\nabla_{\mathbf{v}} L = 2(\mathbf{C}\mathbf{v} - \lambda\mathbf{v}) = 0$ leads to $\mathrm{var}(\mathbf{z}) = \mathbf{v}^T\mathbf{C}\mathbf{v} = \lambda$
  - λ should be the largest eigenvalue of $\mathbf{C}$ and $\mathbf{v}_1$ be its corresponding eigenvector, called the first **loading vector**.
- Another derivation
  - From $\| \mathbf{x}_i \|^2 = \| \mathbf{x}_i - (\mathbf{x}_i^T\mathbf{v})\mathbf{v} \|^2 + \| (\mathbf{x}_i^T\mathbf{v})\mathbf{v} \|^2$, PC can also be obtained by minimizing the one-dimensional reconstruction error

$$\sum_{i=1}^{N} \| \mathbf{x}_i - (\mathbf{x}_i^T\mathbf{v})\mathbf{v} \|^2 = \| \mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T \|_F^2 .$$

(*) Principal Component : How it works

Further Principal Component

- The next k-th components ($k = 2, \ldots, c$) starts from the reoriented data $\mathbf{x}_i - \sum_{j=1}^{k-1} (\mathbf{x}_i^T \mathbf{v}_j)\mathbf{v}_j$, $i = 1, \ldots, N$
- Maximizes the variance of the elements of $\mathbf{z}_k = \mathbf{X}_{(k)}\mathbf{v}$ from $\mathbf{X}_{(k)} = \mathbf{X} - \sum_{j=1}^{k-1} \mathbf{X}\mathbf{v}_j\mathbf{v}_j^T$

$$\mathrm{var}(\mathbf{z}_k) = \frac{1}{N-1}\mathbf{v}^T\mathbf{X}_{(k)}^T\mathbf{X}_{(k)}\mathbf{v}$$

- Since $\| \mathbf{x}_{(k)i} \|^2 = \| \mathbf{x}_{(k)i} - (\mathbf{x}_{(k)i}^T\mathbf{v})\mathbf{v} \|^2 + \| (\mathbf{x}_{(k)i}^T\mathbf{v})\mathbf{v} \|^2$, this is equivalent to minimizing the k-dimensional reconstruction error

$$\sum_{i=1}^{N} \| \mathbf{x}_i - \sum_{j=1}^{k} (\mathbf{x}_i^T \mathbf{v}_j)\mathbf{v}_j \|^2 = \| \mathbf{X} - \sum_{j=1}^{k} \mathbf{X}\mathbf{v}_j\mathbf{v}_j^T \|_F^2 .$$

$$= \min \sum_{i=1}^{N} \| \mathbf{x}_i - \sum_{j=1}^{k} (\mathbf{x}_i^T \mathbf{v}_j)\mathbf{v}_j \|^2 = \min \| \mathbf{X} - \sum_{j=1}^{k} \mathbf{X}\mathbf{v}_j\mathbf{v}_j^T \|_F^2$$

$$= \min_{\mathrm{rank}(B)=k} \| \mathbf{X} - \mathbf{B} \|_F^2$$

Principal Component : How it works

Principal Component

- Using the SVD $\mathbf{X} = U\Sigma V^T$ where $U \in \Re^{N\times N}$ is orthonormal and $V \in \Re^{p\times p}$ is orthonormal and $\Sigma \in \Re^{N\times p}$ is diagonal with $\sigma_1 \geq \cdots \geq \sigma_p \geq 0$, the k-th loading vector $\mathbf{v}_k$ is the k-th column vector of V, which is also the k-th eigenvectors of the $p \times p$ covariance matrix,

$$\mathbf{C} = \frac{1}{N-1}\mathbf{X}^T\mathbf{X} = \frac{1}{N-1}V\Sigma^2V^T.$$

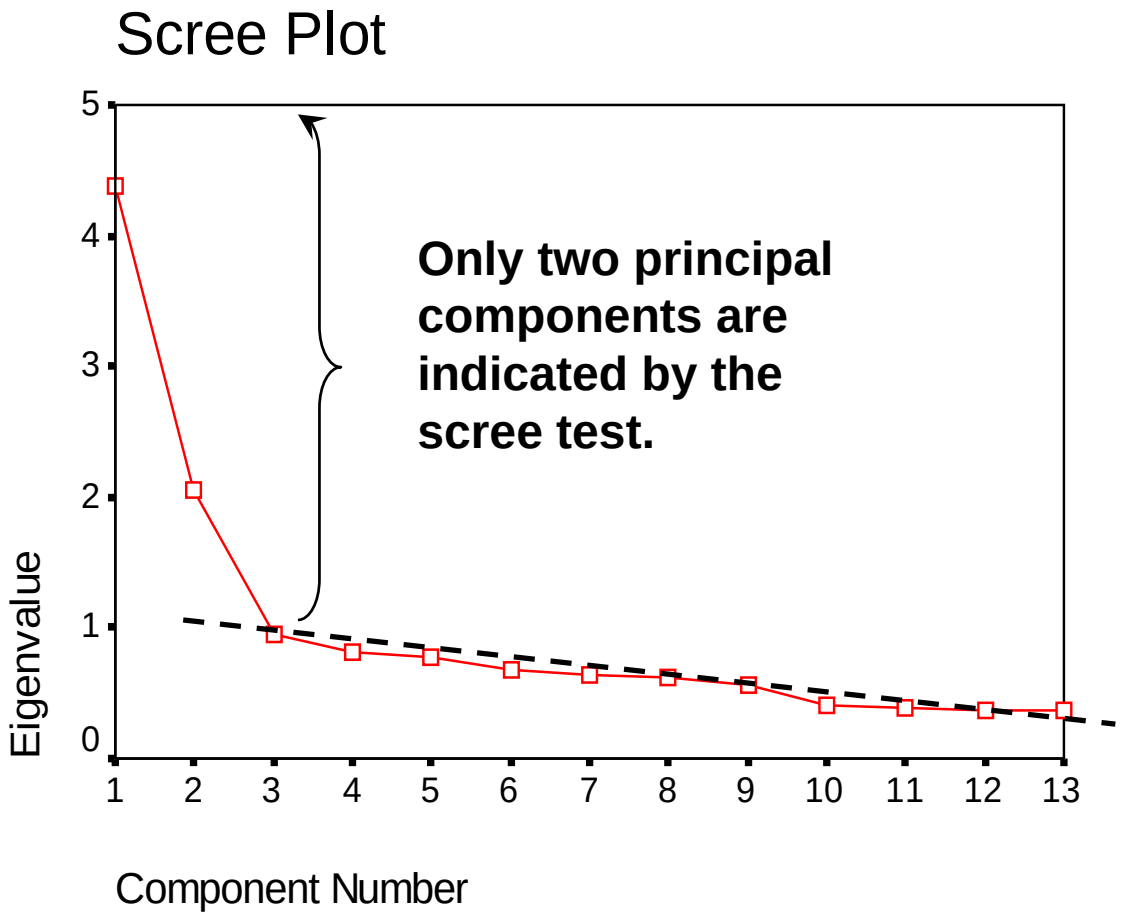
$$\lambda_k\mathbf{v}_k = \mathbf{C}\mathbf{v}_k = \frac{1}{N-1}\sum_{j=1}^{N}(\mathbf{x}_j^T\mathbf{v}_k)\mathbf{x}_j,$$

- All the principal components $\mathbf{v}_k$ with $\lambda_k = \sigma_k^2/(N-1) \neq 0$ can be represented as a linear combination of $\mathbf{x}_1 \ldots \mathbf{x}_N$.
- The c-dimensional representation $\mathbf{z}_i = (z_{i1}, \ldots, z_{ic})^T$ of $\mathbf{x}_i \in \mathbb{R}^n$ has elements

$$z_{ik} = \mathbf{x}_i^T\mathbf{v}_k = (\mathbf{X}\mathbf{v}_k)_i = [(\textstyle\sum_j \sigma_j\mathbf{u}_j\mathbf{v}_j^T)\mathbf{v}_k]_i = \sigma_k(u_k)_i, \quad k = 1, \ldots, c$$

- A large gap between the c^{th} and $(c+1)^{th}$ eigenvalues indicates that the high dimensional input observations in the high dimensional space can be represented approximately as those in a lower c-dimensional space.

How many Components should be retained?

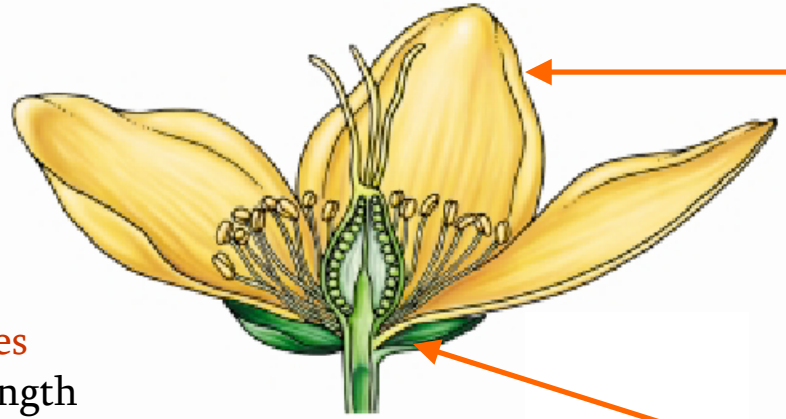


Interpreting Eigenvectors

- Correlations between variables and the principal axes are known as loadings
- Each element of the eigenvectors represents the contribution of a given variable to a component

	1	2	3
Altitude	0.3842	0.0659	-0.1177
pH	-0.1159	0.1696	-0.5578
Cond	-0.2729	-0.1200	0.3636
TempSurf	0.0538	-0.2800	0.2621
Relief	-0.0765	0.3855	-0.1462
maxERht	0.0248	0.4879	0.2426
avERht	0.0599	0.4568	0.2497
%ER	0.0789	0.4223	0.2278
%VEG	0.3305	-0.2087	-0.0276
%LIT	-0.3053	0.1226	0.1145
%LOG	-0.3144	0.0402	-0.1067
%W	-0.0886	-0.0654	-0.1171
H1Moss	0.1364	-0.1262	0.4761
DistSWH	-0.3787	0.0101	0.0042
DistSW	-0.3494	-0.1283	0.1166
DistMF	0.3899	0.0586	-0.0175

Example: Fisher's IRIS data



Petal, a non-reproductive p art of the flower

Sepal, a non-reproductive p art of the flower

Four features
- sepal length
- sepal width
- petal length
- petal width

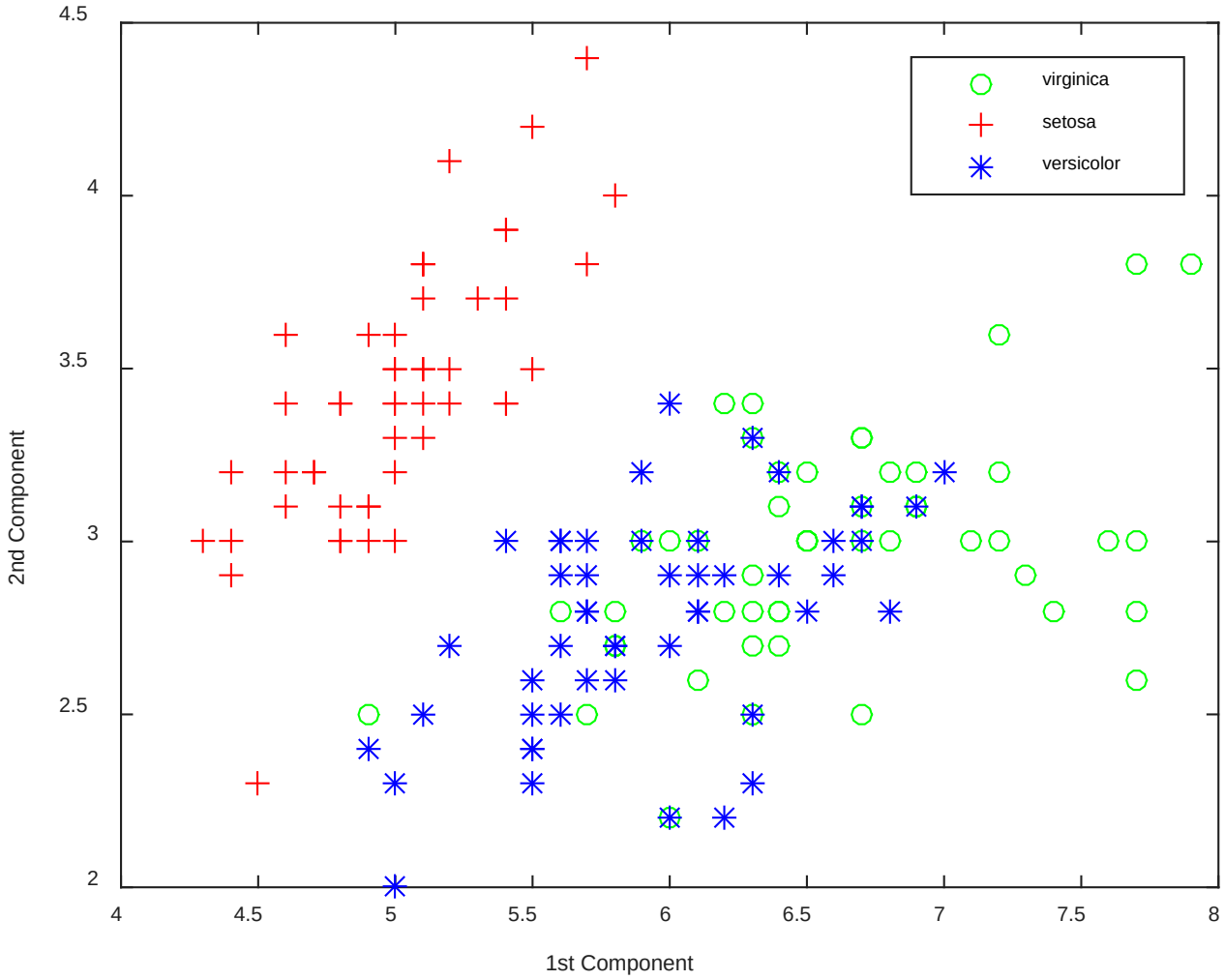
Three classes (species of iris)
- setosa
- versicolor
- virginica

50 instances of each

The famous iris data!

Iris Data: http://archive.ics.uci.edu/ml/datasets/Iris

Example: Features 1 and 2 (sepal width/length)



Example: Fisher's IRIS data Using PCA