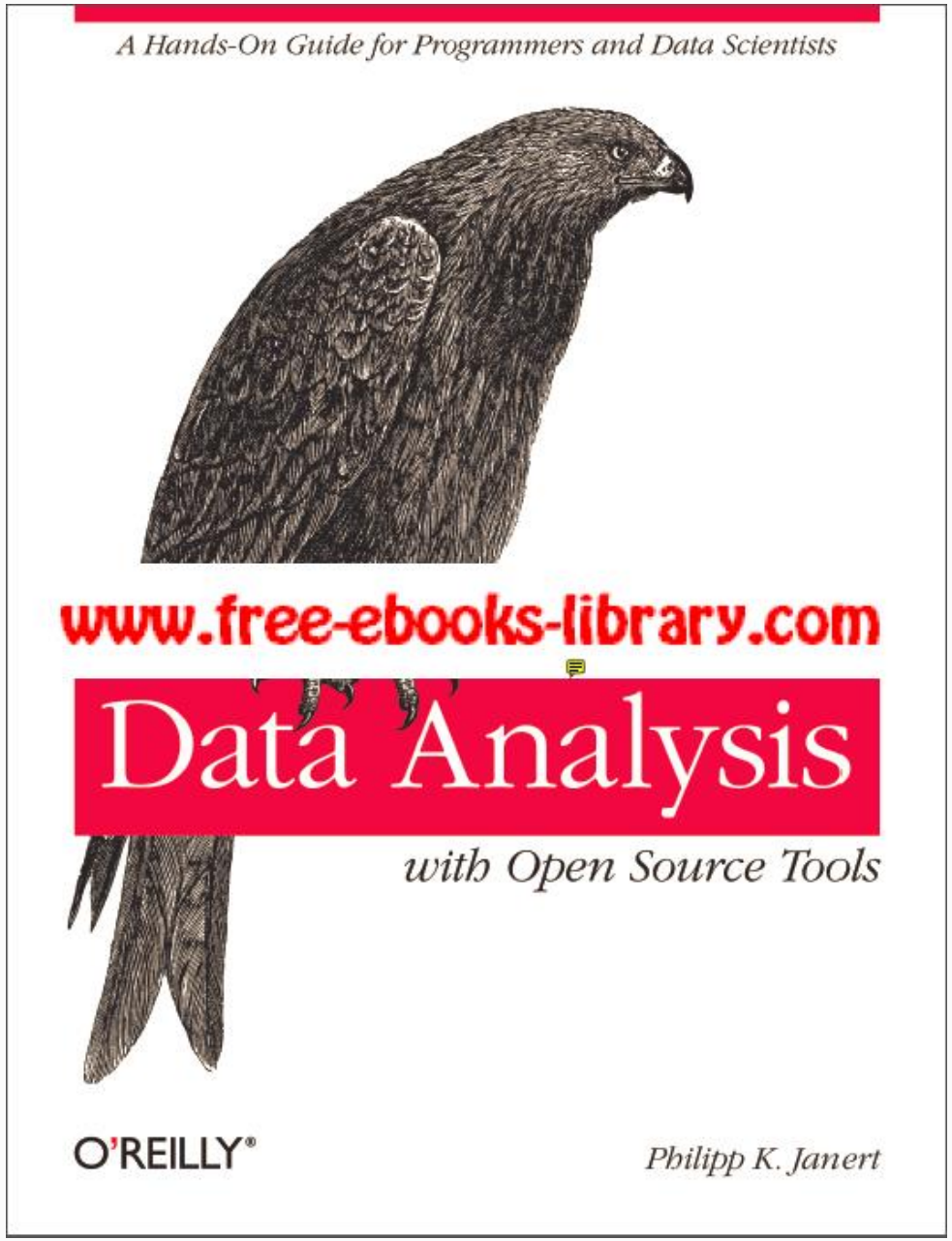
Data Science

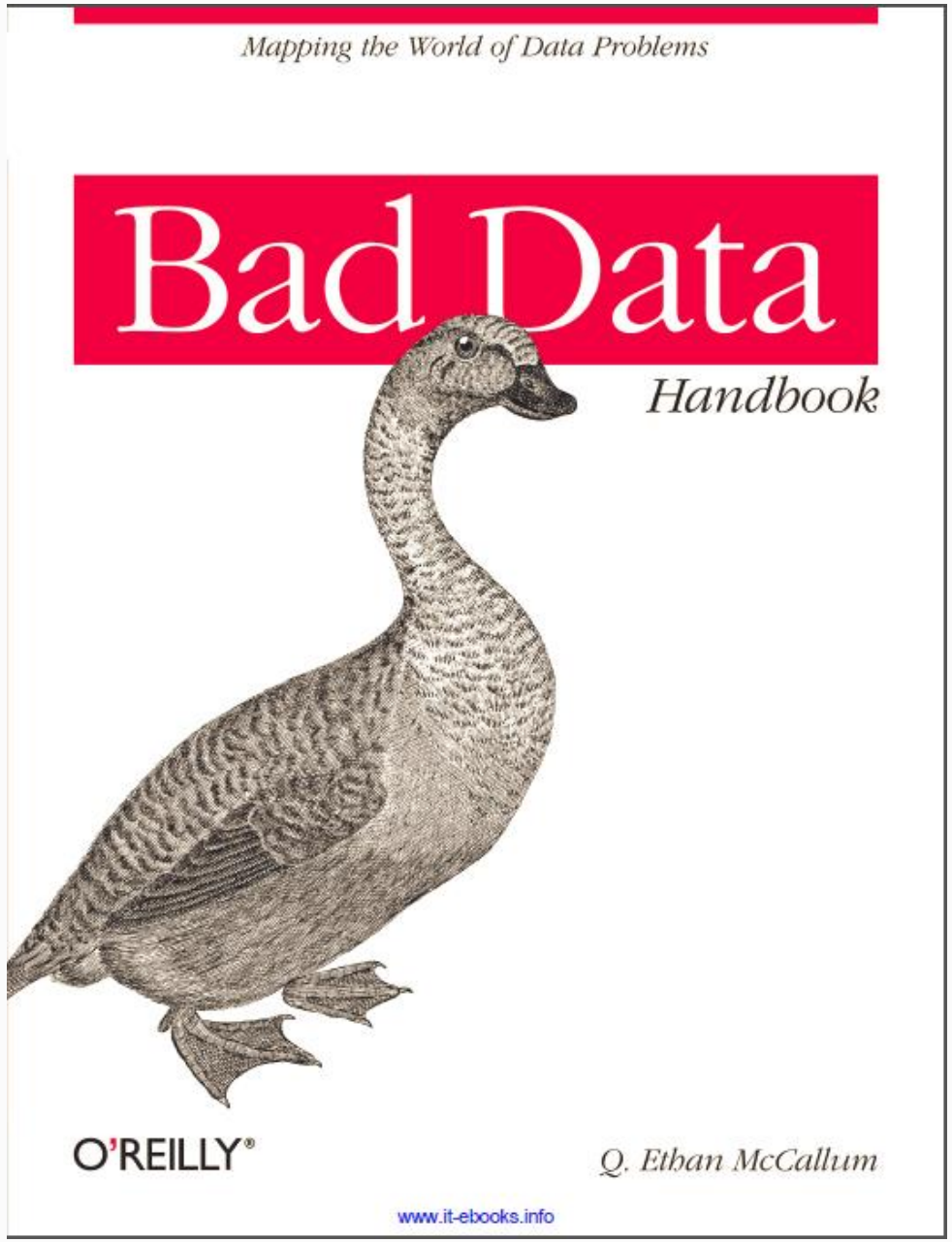
Deriving Knowledge from Data at Scale

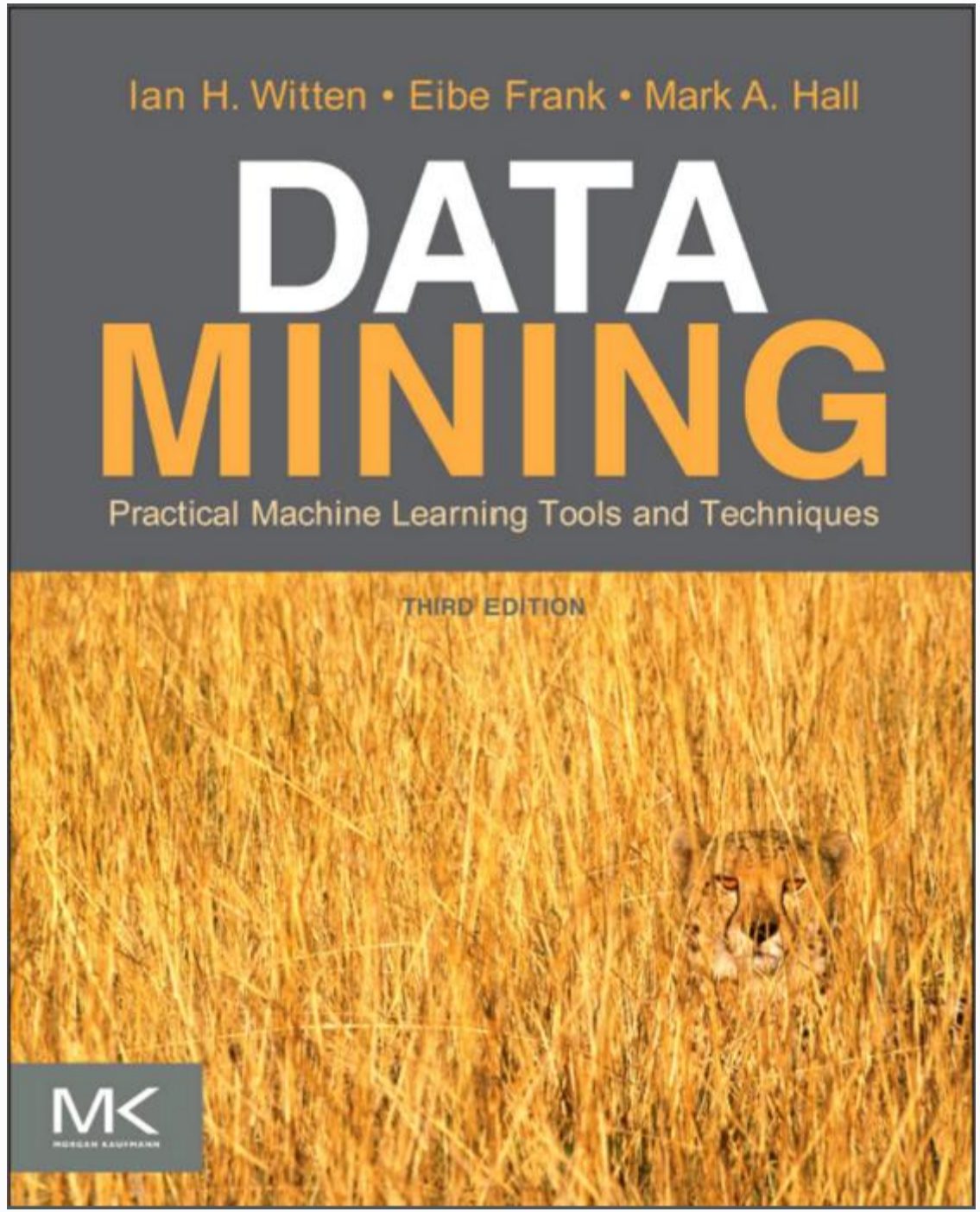
Good data preparation is key to producing valid and reliable models…

Valentine N. Fontama

August 20th , 2015


A Hands-On Guide for Programmers and Data Scientists
www.free-ebooks-library.com
Data Analysis
with Open Source Tools
O'REILLY®
Philipp K. Janert


Mapping the World of Data Problems
Bad Data
Handbook
O'REILLY®
Q. Ethan McCallum
www.it-ebooks.info


Ian H. Witten • Eibe Frank • Mark A. Hall
DATA MINING
Practical Machine Learning Tools and Techniques
THIRD EDITION
MK
MORGAN KAUFMANN

Lecture 7 Agenda

- Support Vector Machines;
- Kaggle, *how the won it…*
- Data Exploration;
- Feature Selection;
- Discuss Class Project;

Heads Up, Lecture 8 Agenda

- Hands-On with SVM;
- Lecture on Linear/Logistic Regression
- Hands-On Data Transformation
  - Dimensionality Reduction via PCA
  - Attribute Selection
- Discuss Course Project

Support Vector Machines

Support vector machines

Similar to linear regression they aim to find a hyperplane that linearly separates data points belong to different classes

In addition SVMs aim to find the hyperplane that is least likely to over fit the training data

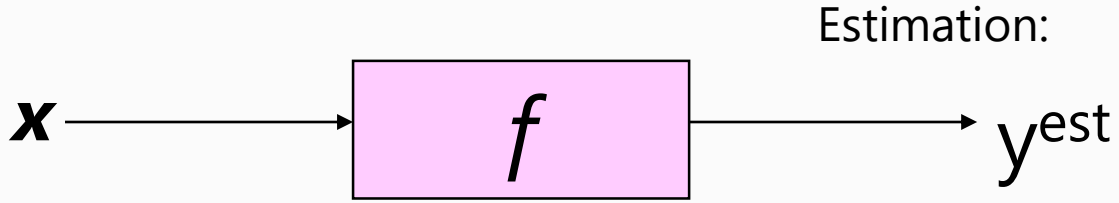
- By design, similar to pruning in Decision Trees, SVMs attempt to regulate the hypothesis space to ensure good performance on validation set…

- Fast in the nonlinear case
  - Use a mathematical trick to avoid creating “pseudoattributes”
  - The nonlinear space is created implicitly

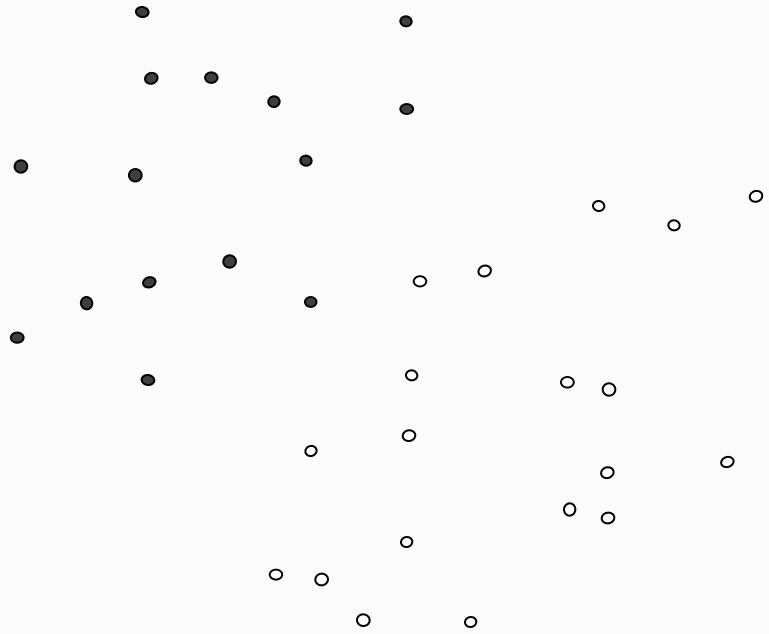
SVM, in a nutshell…

- SVM views the input data as two sets of vectors in an n-dimensional space. It constructs a separating hyperplane in that space, one which maximizes the margin between the two data sets.
- To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating hyperplane.
- A good separation is achieved by the hyperplane that has the largest distance to the neighboring data points of both classes.
- The vectors (points) that constrain the width of the margin are the support vectors.

Linear Classifiers



- denotes +1
- denotes -1

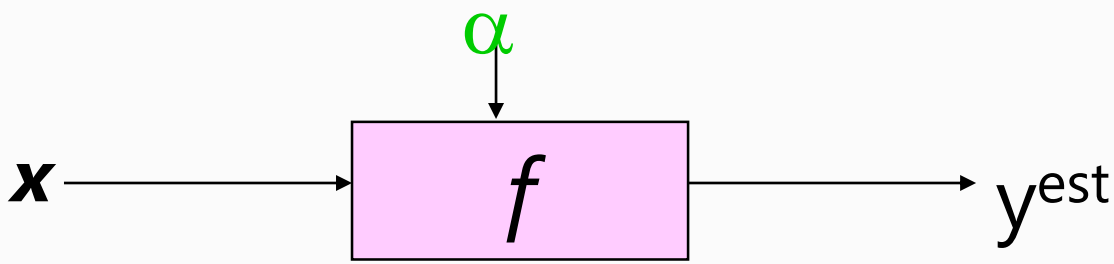


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

w: weight vector
x: data vector

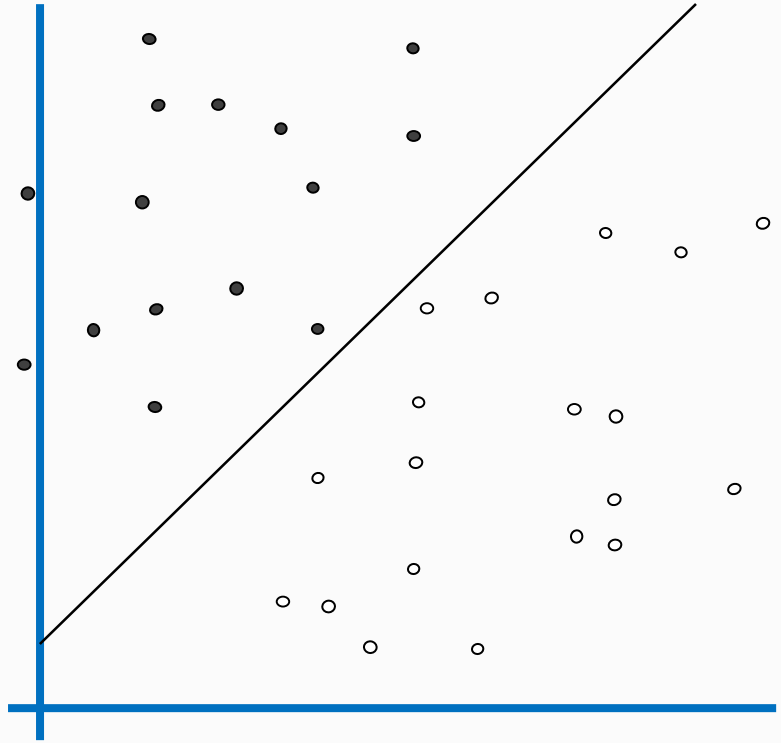
How would you classify this data?

Linear Classifiers



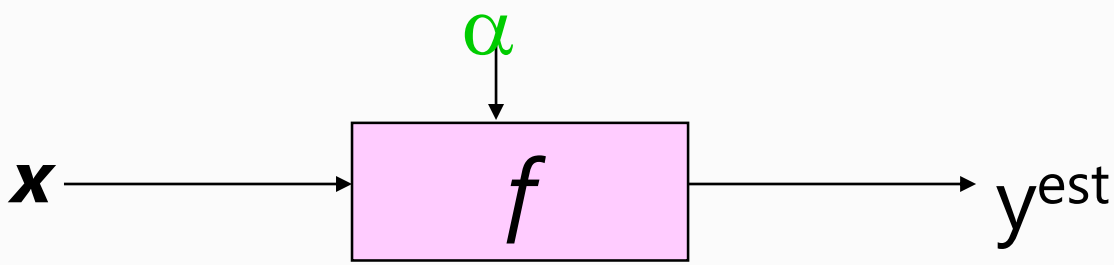
- denotes +1
- denotes -1

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} - b)$



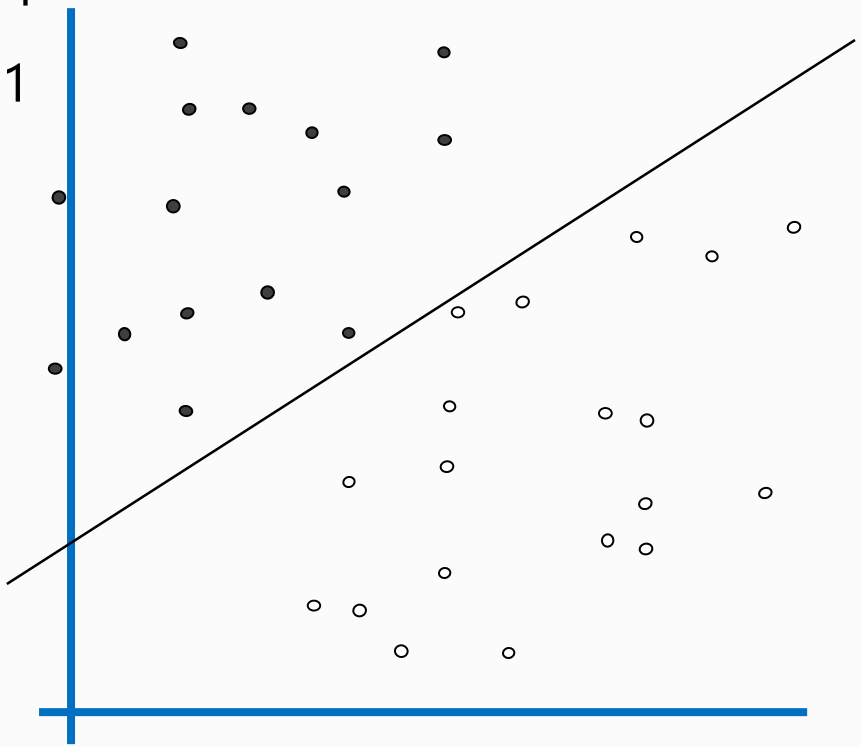
How would you classify this data?

Linear Classifiers



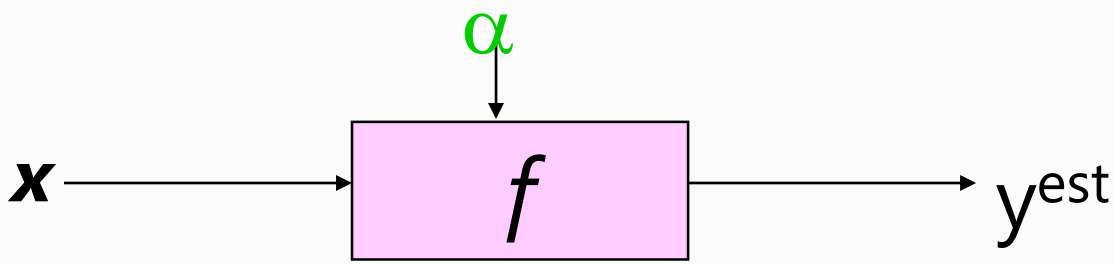
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} - b)$

- denotes +1
- denotes -1

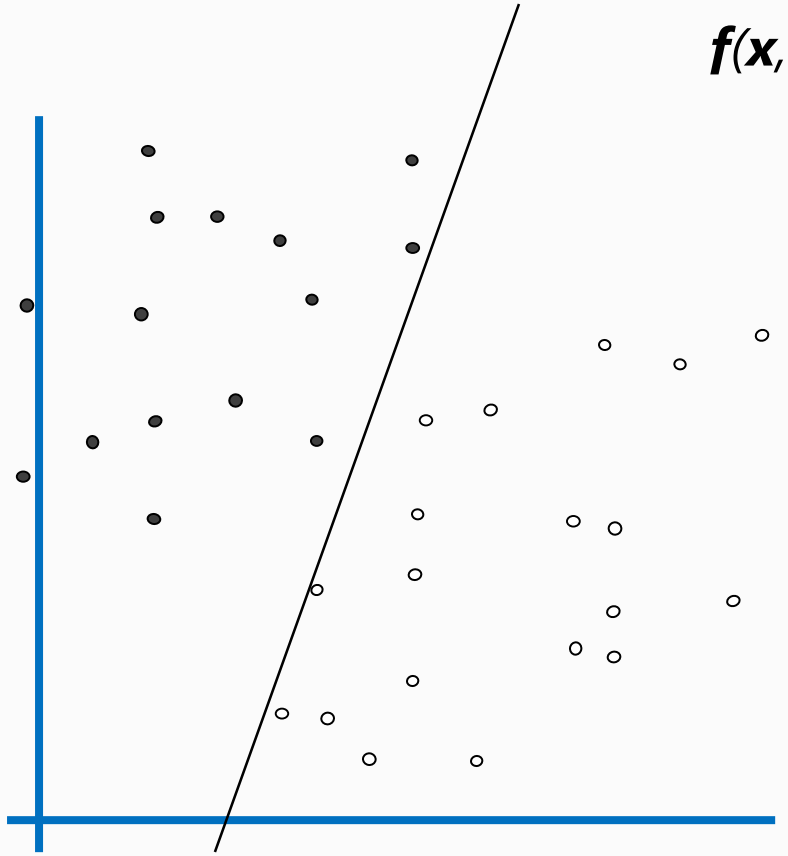


How would you classify this data?

Linear Classifiers



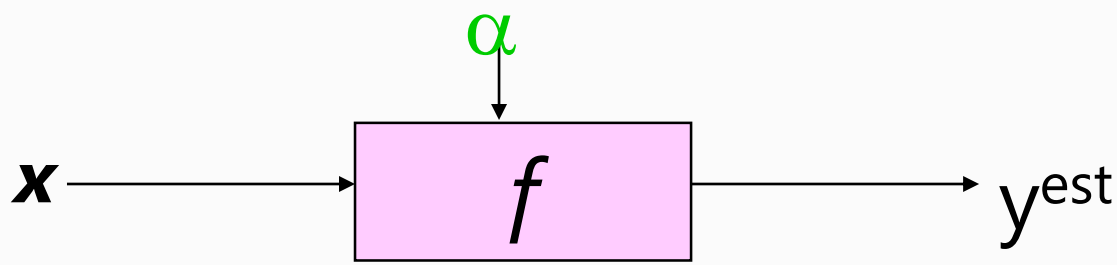
- denotes +1
- denotes -1



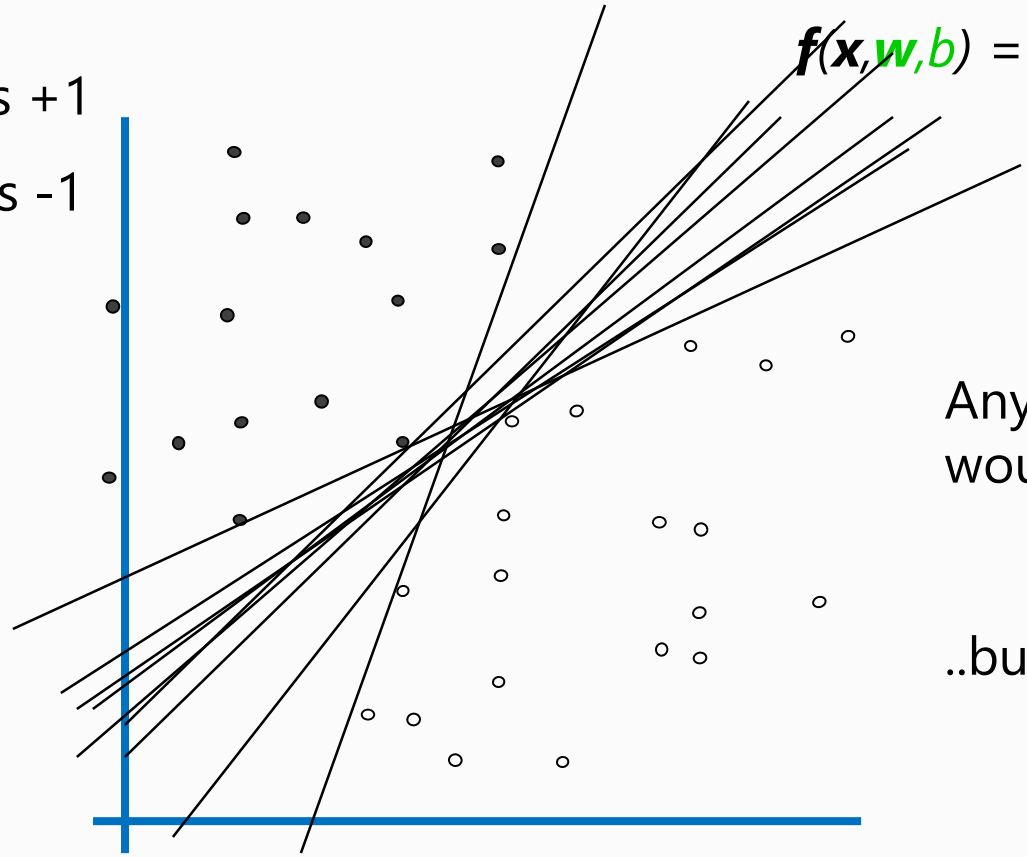
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

How would you classify this data?

Linear Classifiers



- denotes +1
- denotes -1

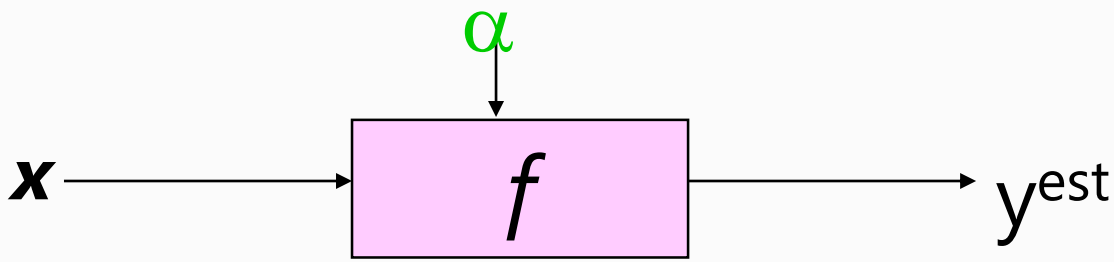


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} - b)$

Any of these would be fine..

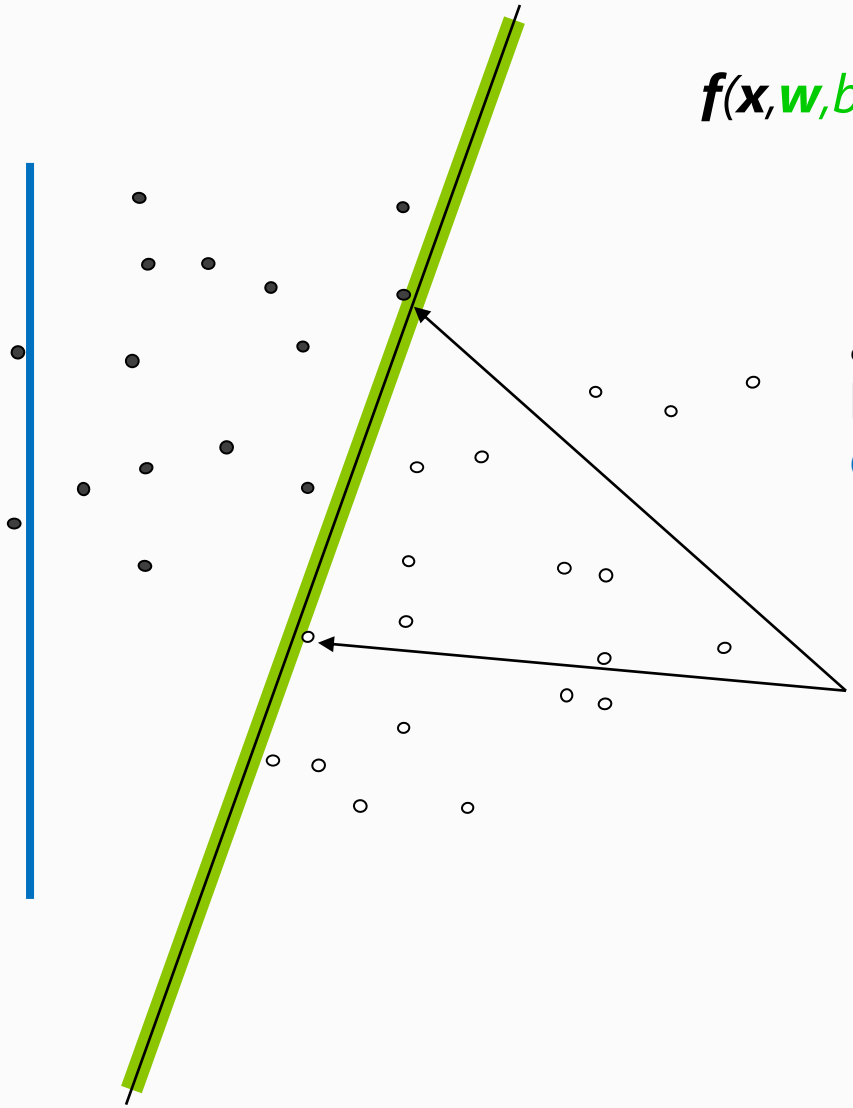
..but which is best?

Classifier Margin



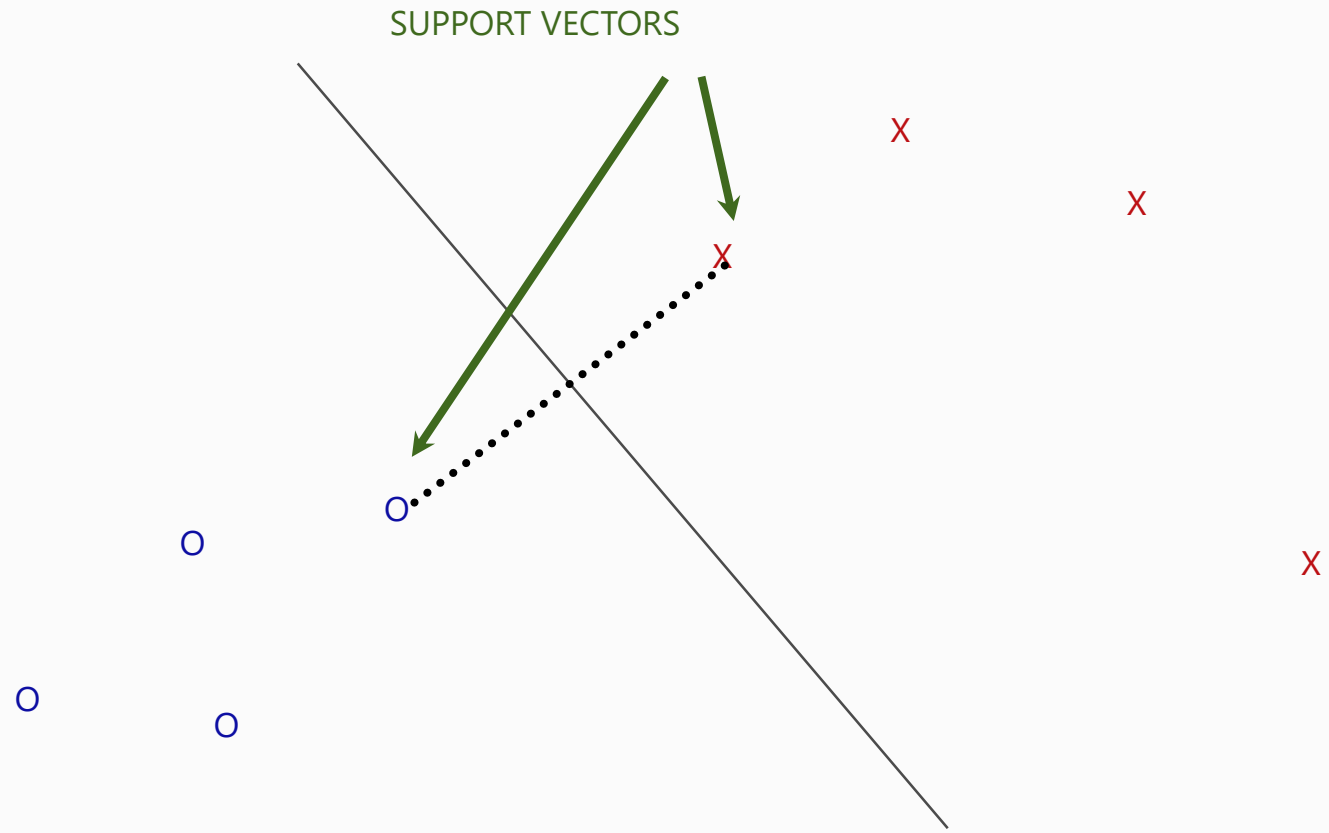
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} - b)$

- denotes +1
- denotes -1

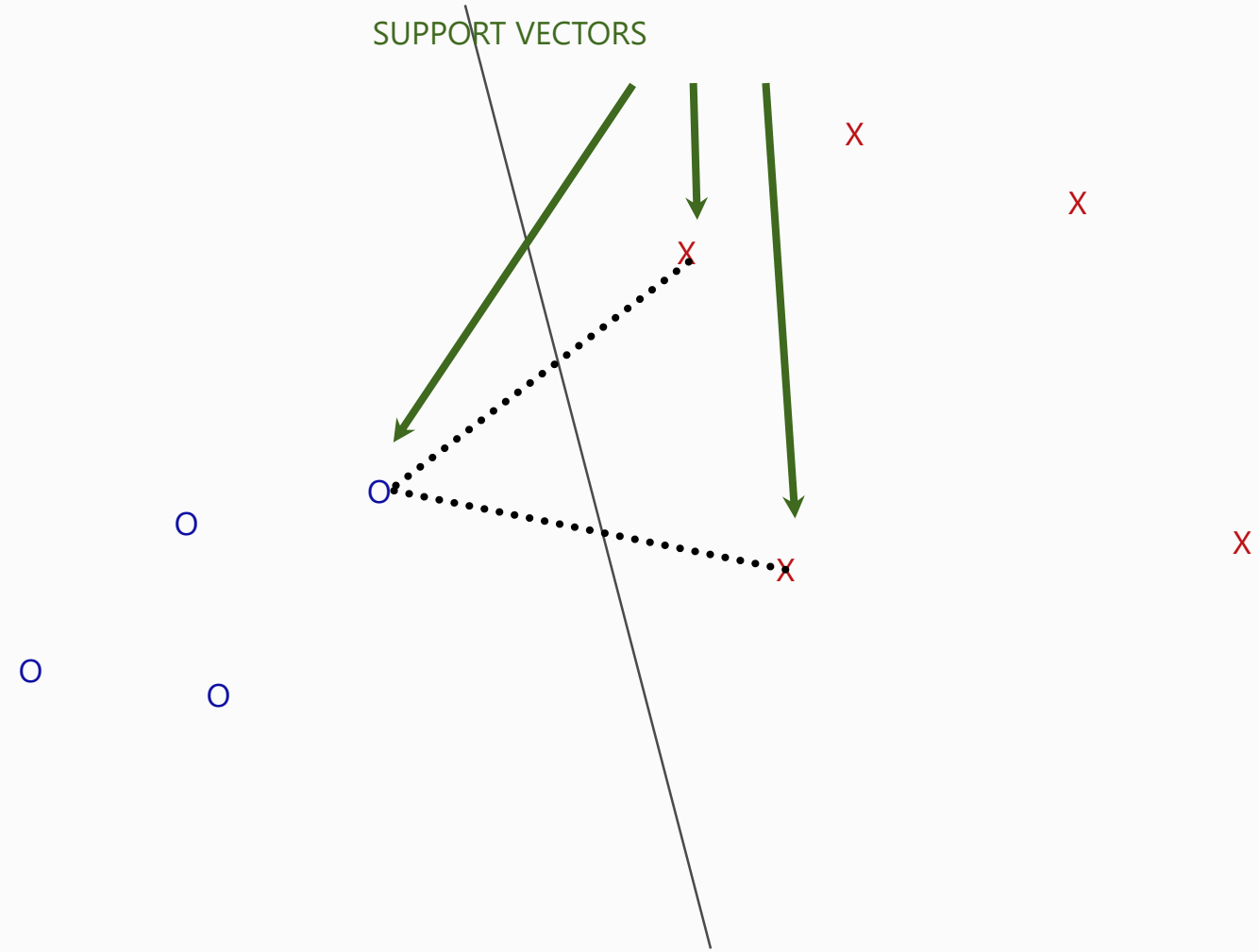


Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

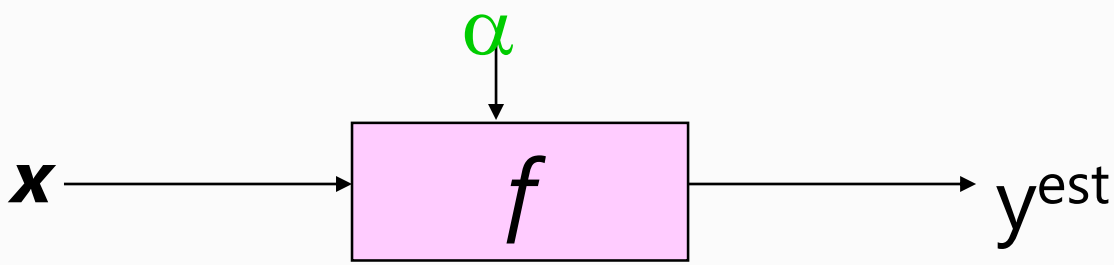
Geometric Intuition



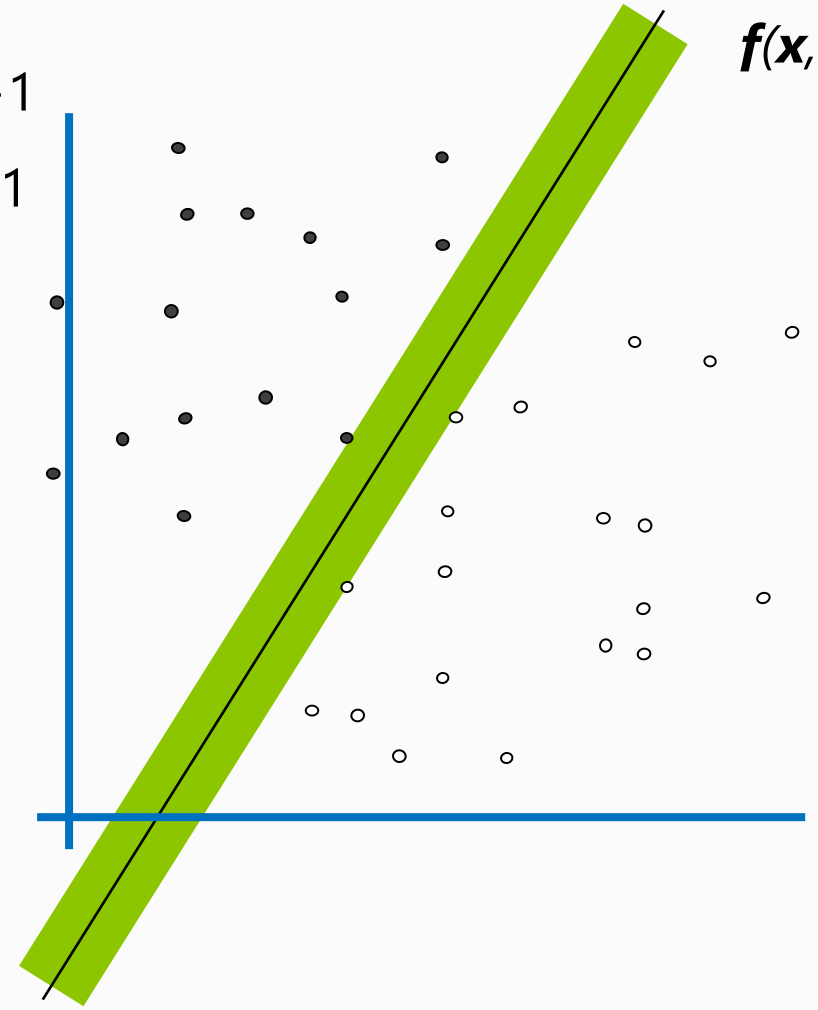
Geometric Intuition



Maximum Margin



- denotes +1
- denotes -1



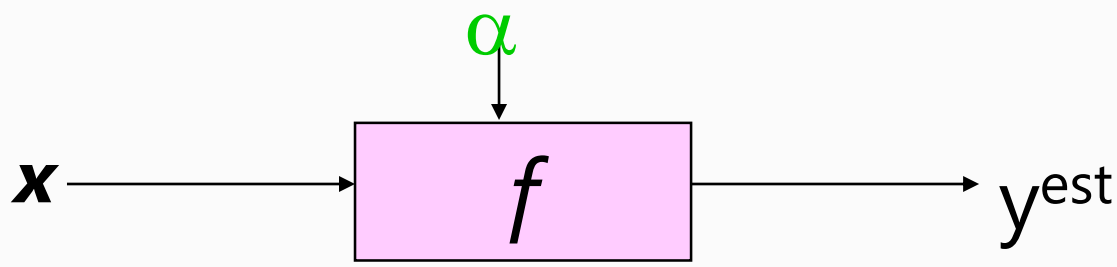
$f(x,w,b) = sign(w.x - b)$

The maximum margin linear classifier is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

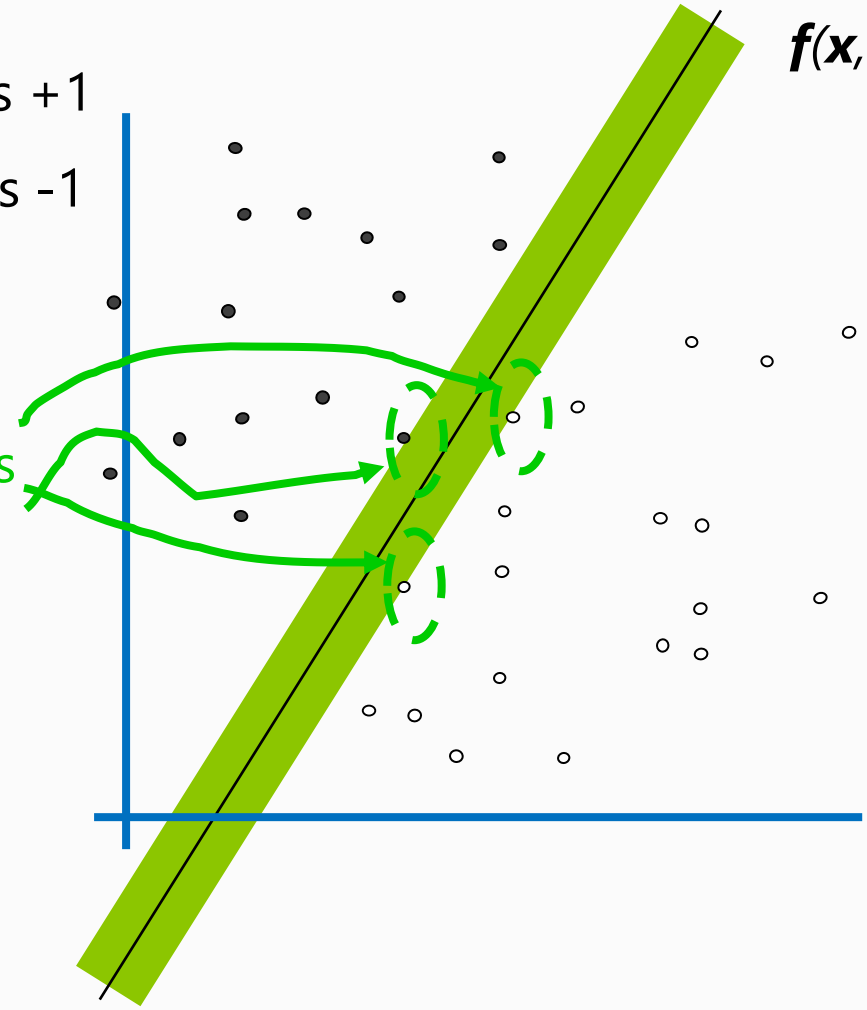
Linear SVM

Maximum Margin



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} + b)$

- denotes +1
- denotes -1



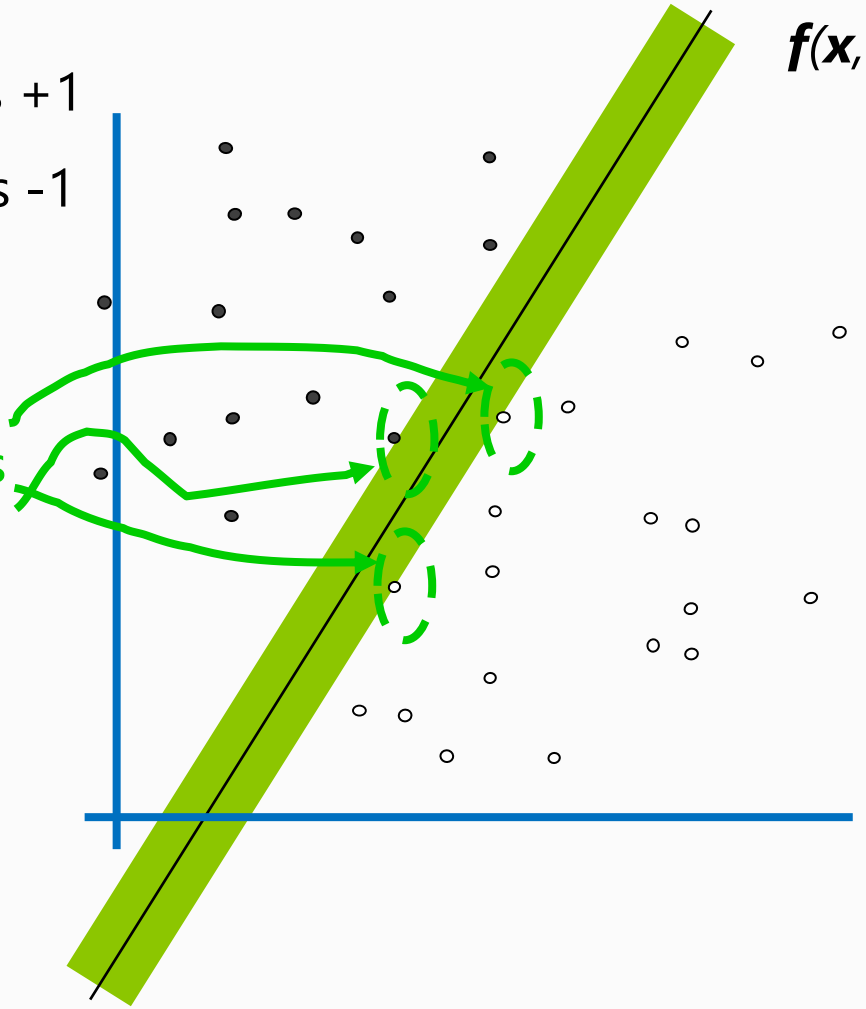
Support Vectors are those datapoints that the margin pushes up against

The maximum margin linear classifier is the linear classifier with the, um, maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Linear SVM

Why Maximum Margin?



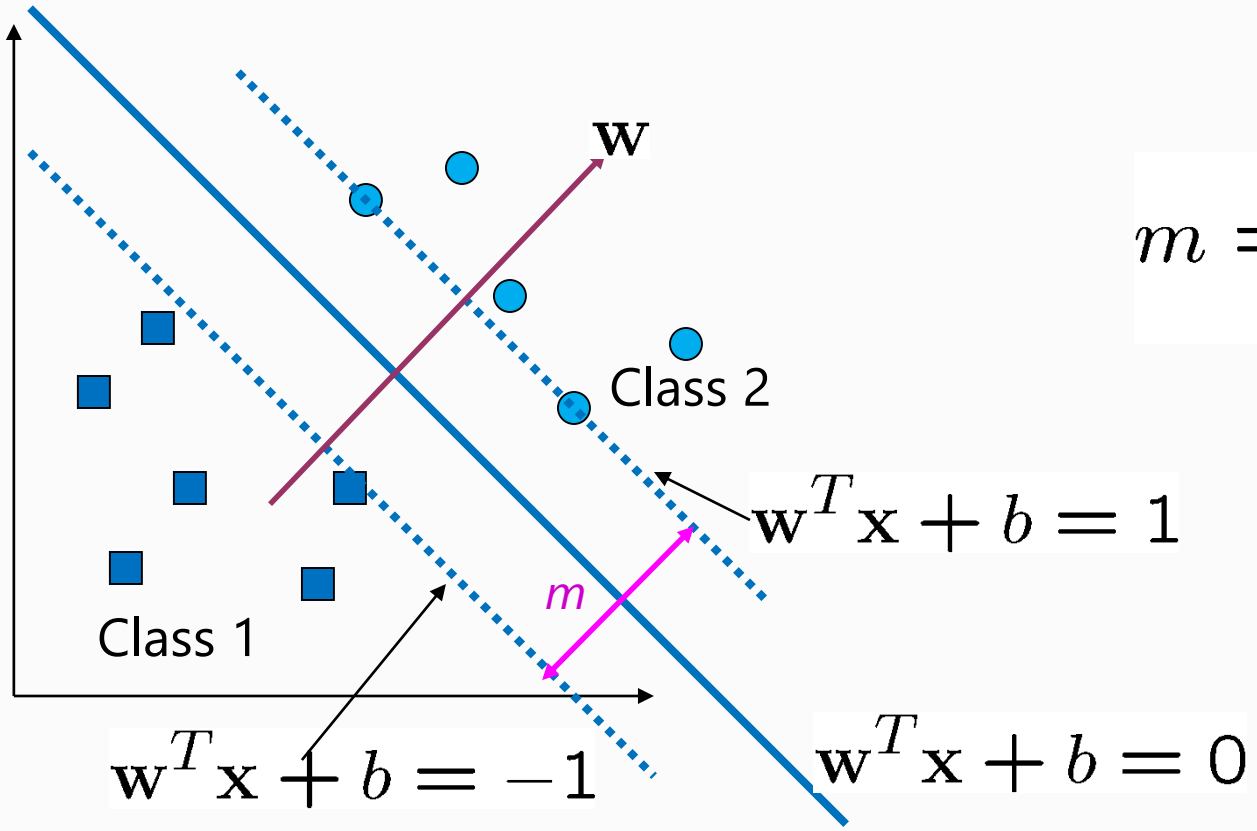
$$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}.\ \boldsymbol{x} - b)$$

1. Intuitively this feels safest.
2. If we've made a small error in the location of the boundary this gives us least chance of causing a misclassification.
3. LOOCV (leave one out cross validation) is easy since the model is immune to removal of any nonsupport-vector data points.
5. Empirically it works **very** well.

Large-Margin Decision Boundary

The decision boundary should be as far away from the data of both classes as possible

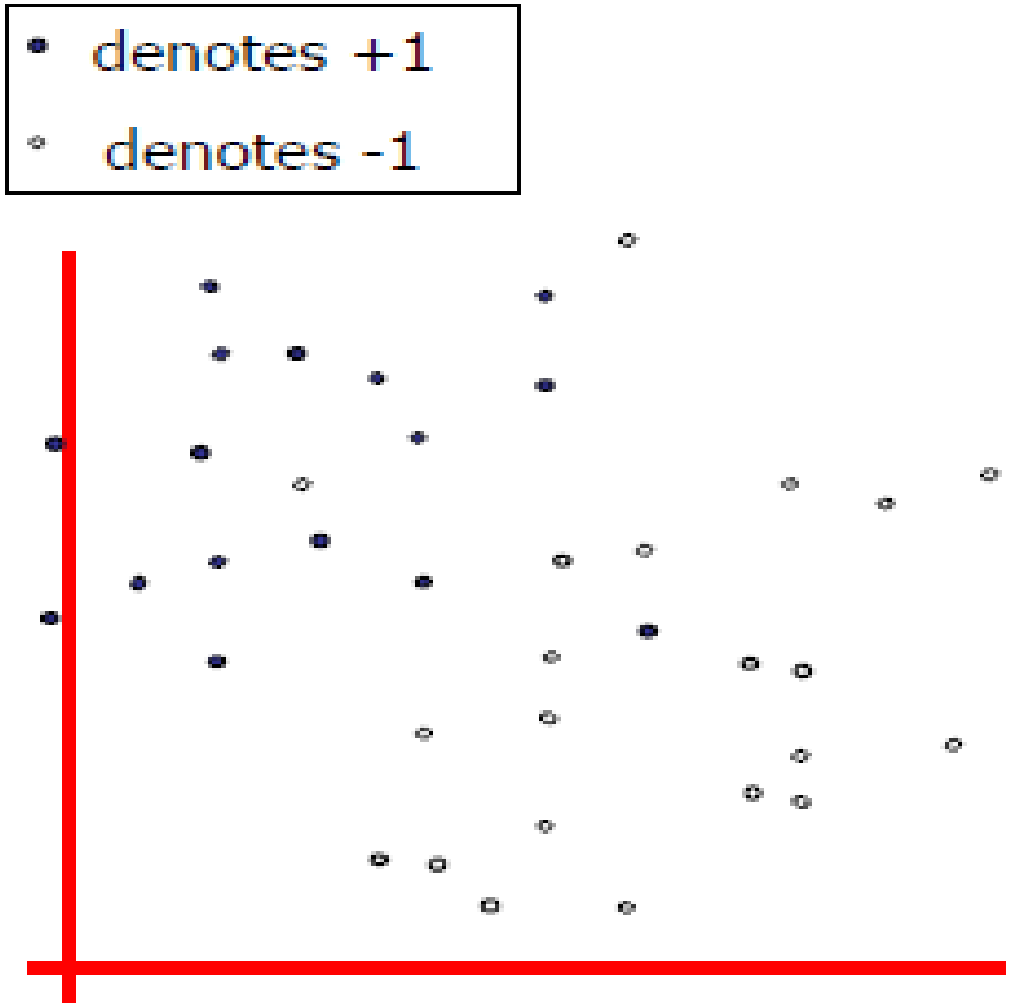
- We should maximize the margin, *m*
- Distance between the origin and the line $\mathbf{w}^t\mathbf{x}=-b$ is $b/||\mathbf{w}||$



$$m = \frac{2}{||\mathbf{w}||}$$

Uh-oh!

This is going to be a problem!



Slack Variables

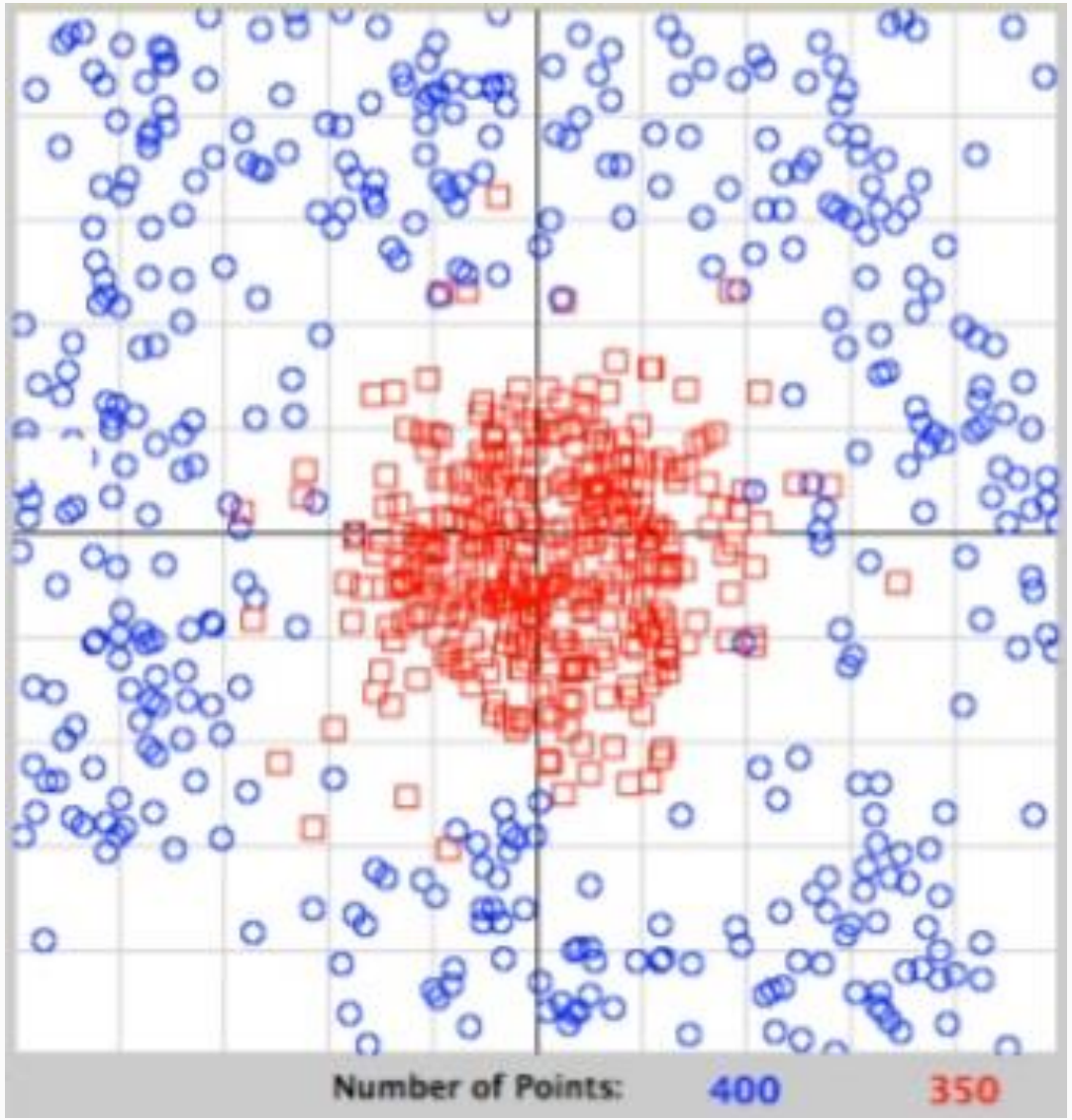
- SVMs are made robust by adding 'slack variables' that allow training error to be non-zero;
- One for each data point. Slack variable == 0 for correctly classified points

Noise

- Have assumed that the data is separable (in original or transformed space)
- Can apply SVMs to noisy data by introducing a "noise" parameter *C*
- C bounds the influence of any one training instance on the decision boundary
- Still a quadratic optimization problem
- Have to determine C by experimentation

The Kernel Trick

What if the data looks like that below?...

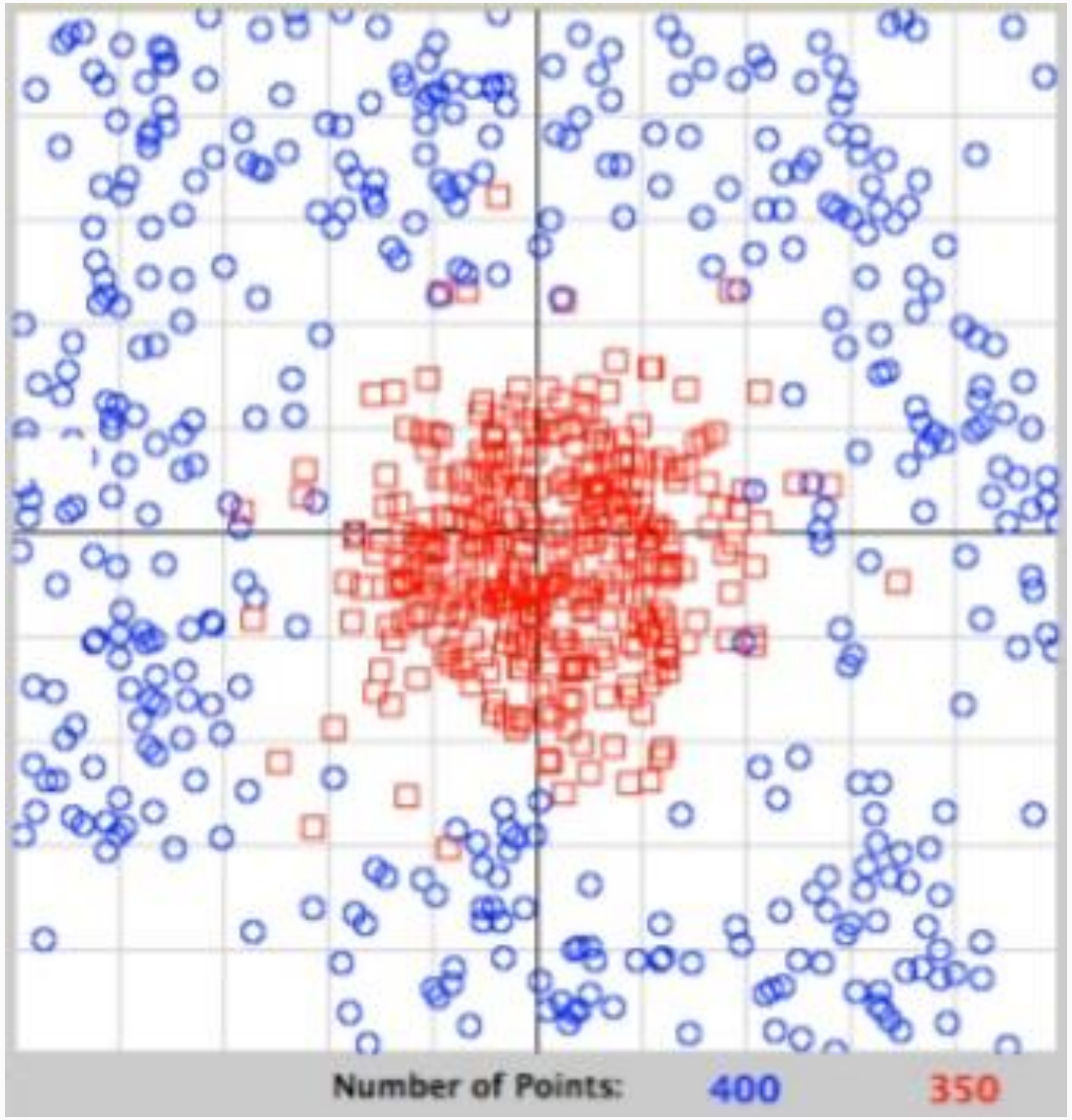


The simplest way to divide two groups is with a straight line, flat plane or an N-dimensional hyperplane. But what if the points are separated by a nonlinear region?

Rather than fitting nonlinear curves to the data, SVM handles this by using a *kernel function* to map the data into a different space where a hyperplane can be used to do the separation.

The Kernel Trick

What if the data looks like that below?...



The kernel trick allows you to use SVMs with non-linear separators

Different kernels

- Polynomial
- Gaussian
- Exponential
- …

The Kernel Trick

Everything you need to know in one slide…

Sometimes it improves your classifier if you map (i.e. transform) your data into a different feature space.

- Example 1: you have nonlinear data but want to use a linear classifier
- Example 2: your original data aren't numbers – for example they could be sequences

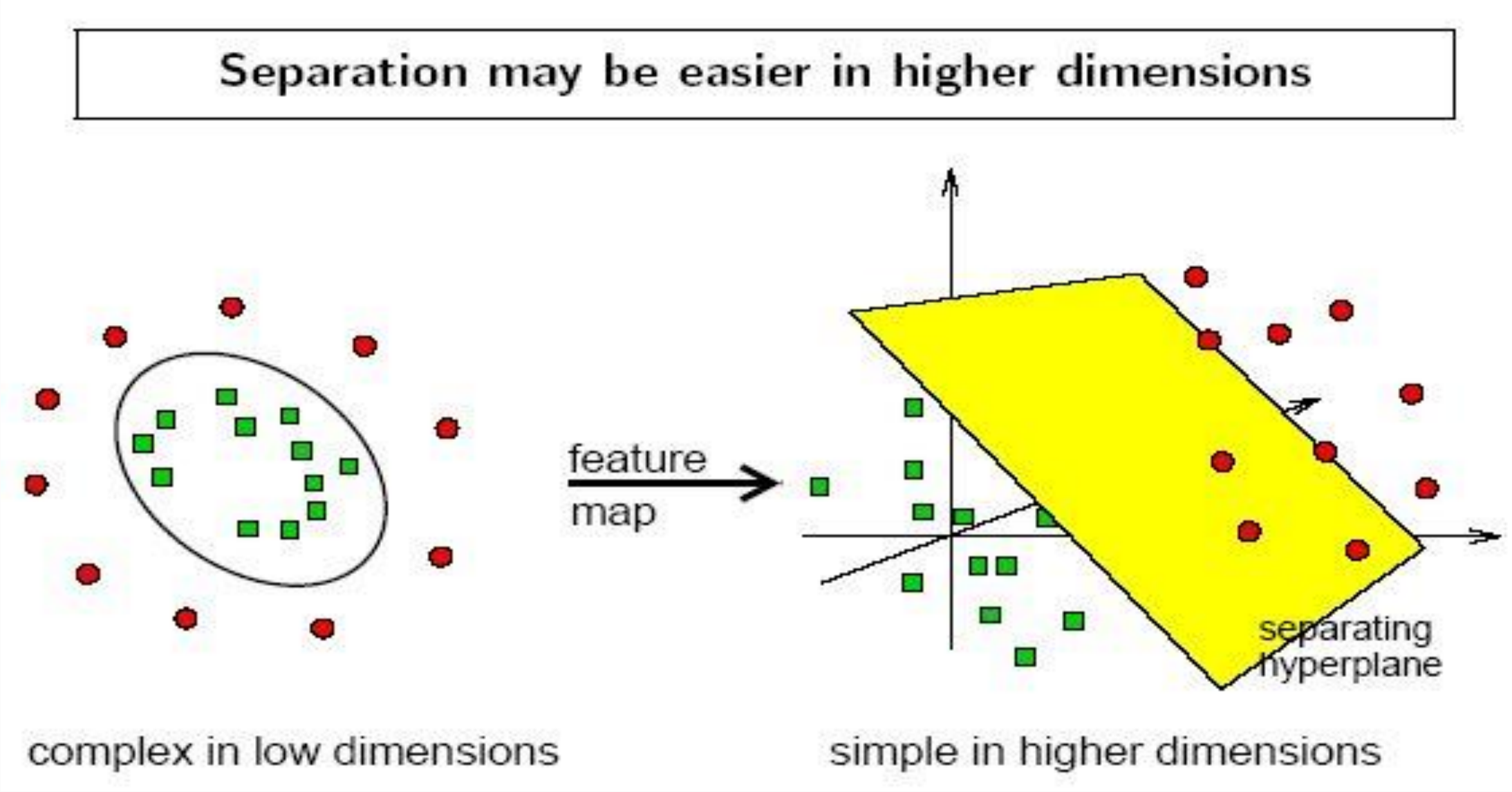
Kernel methods map your original data into a different space so that you can use linear classifiers.

This mapping can often substantially increases the number of features to consider.

- This can be problematic as your number of dimensions grows.

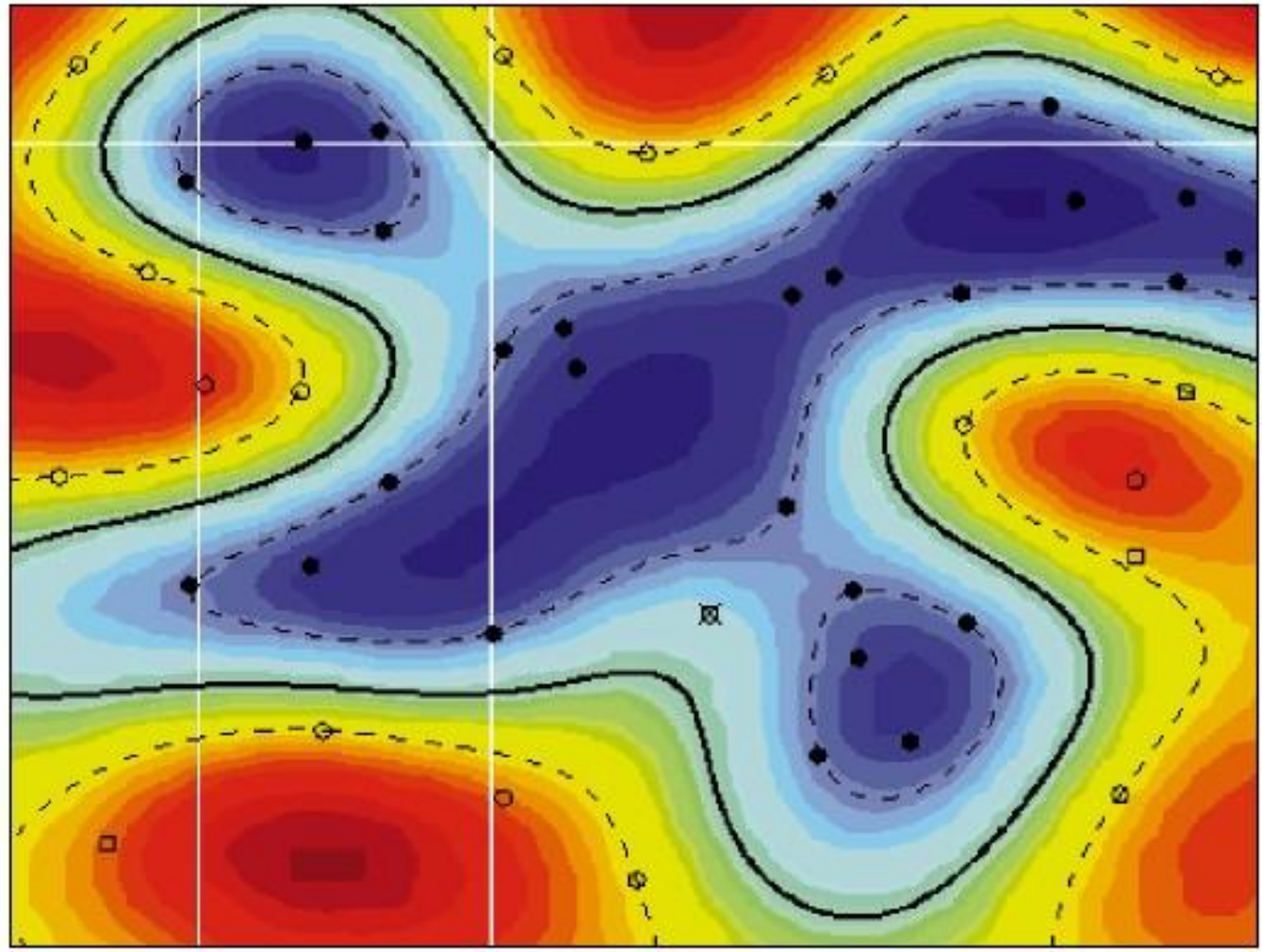
The "Kernel Trick" addresses this by putting a cap on the feature explosion so that the complexity of your classifier increases only linearly with the size of your original data.

Nonlinear Support Vector Machines



Nonlinear Support Vector Machines

The kernel function transforms the data into a higher dimensional space to make it possible to perform the separation.



Choosing the Kernel Function

Probably the most tricky part of using SVM

RBF is a good first option…

Depends on your data—try several.

- *Kernels have even been developed for nonnumeric data like sequences, structures, and trees/graphs.*

May help to use a combination of several kernels.

Don't touch your evaluation data while you're trying out different kernels and parameters.

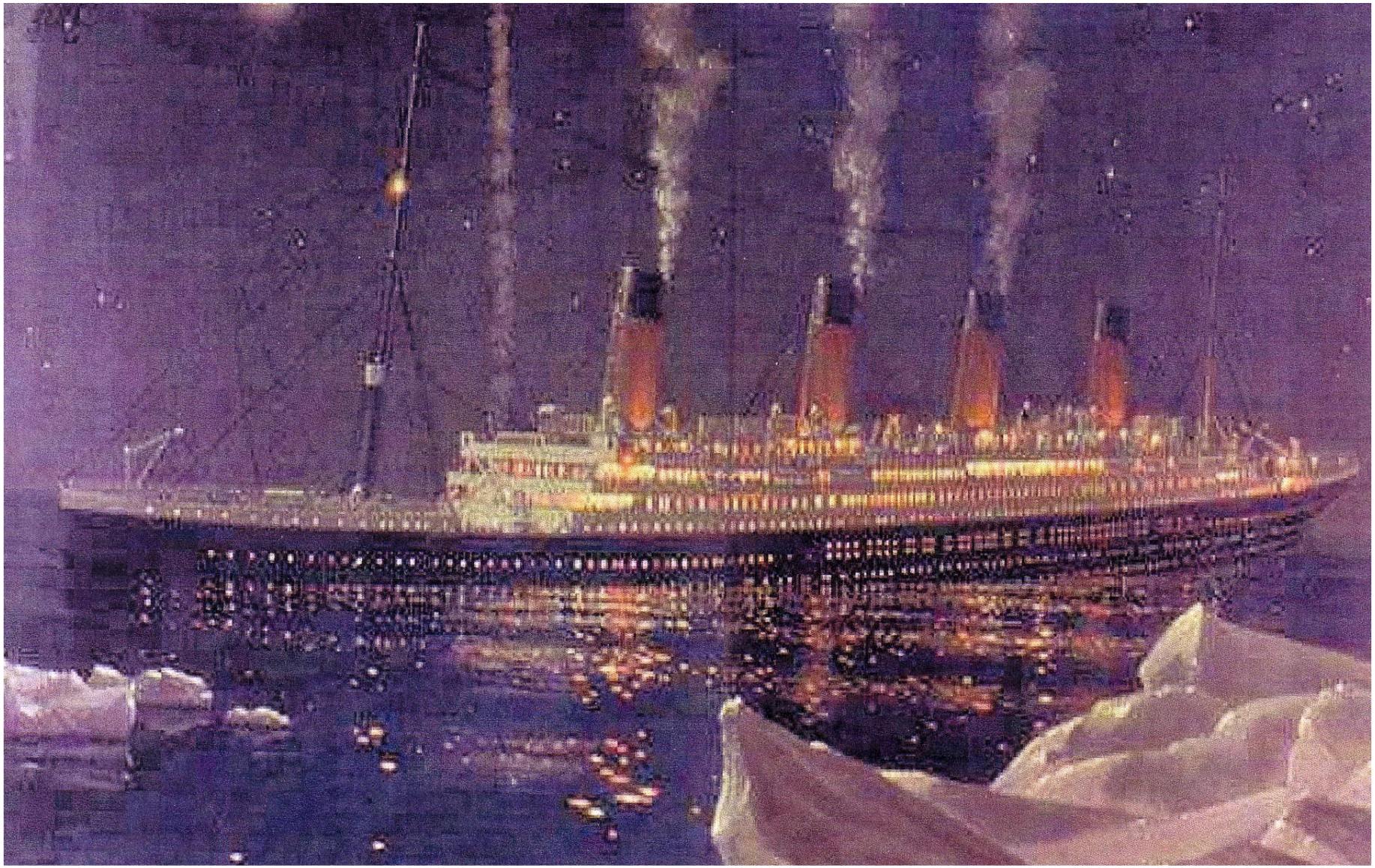
– Use cross-validation for this if you're short on data

Type of Kernel	Inner product kernel $K(\vec{x}, \vec{x}_i),\ i = 1, 2, \ldots, N$	Comments
Polynomial Kernel	$K(\vec{x}, \vec{x}_i) = \left(\vec{x}^T \vec{x}_i + \theta\right)^d$	Power p and threshold θ is specified a priori by the user
Gaussian Kernel	$K(\vec{x}, \vec{x}_i) = e^{-\frac{1}{2\sigma^2}\|\vec{x}-\vec{x}_i\|^2}$	Width σ^2 is specified a priori by the user
Sigmoid Kernel	$K(\vec{x}, \vec{x}_i) = tanh(\eta\, \vec{x}\, \vec{x}_i + \theta)$	Mercer's Theorem is satisfied only for some values of η and θ
Kernels for Sets	$K(\chi, \chi') = \sum_{i=1}^{N_\chi} \sum_{j=1}^{N_{\chi'}} k(x_i, x'_j)$	Where $k(x_i, x'_j)$ is a kernel on elements in the sets χ, χ'
Spectrum Kernel for strings	count number of substrings in common	It is a kernel, since it is a dot product between vectors of indicators of all the substrings.

Table 1: Summary of Inner-Product Kernels [Hay98]

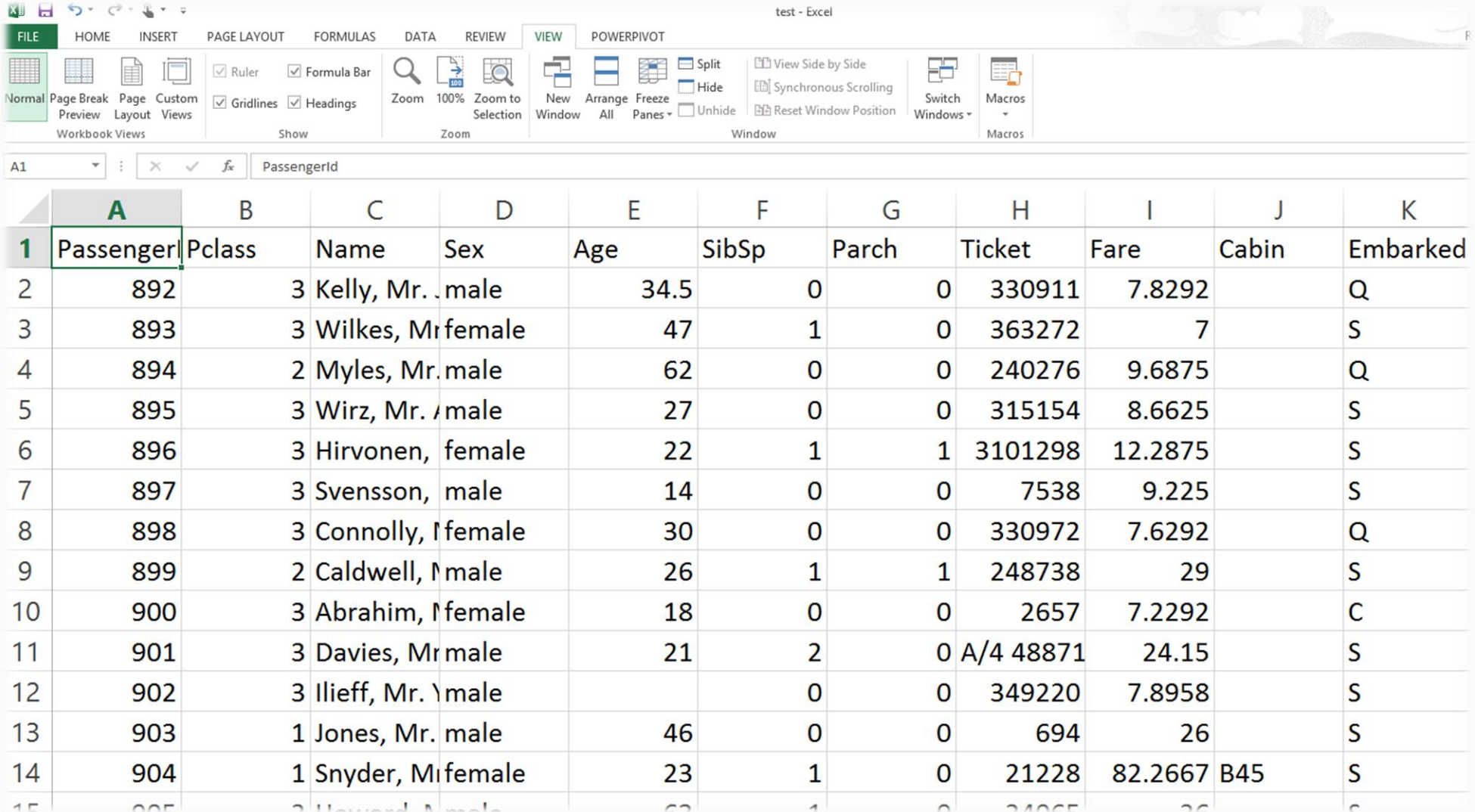
Complexity of the optimization problem remains only dependent on the dimensionality of the input space and not of the feature space!

The Tragedy of the Titanic



First Steps…

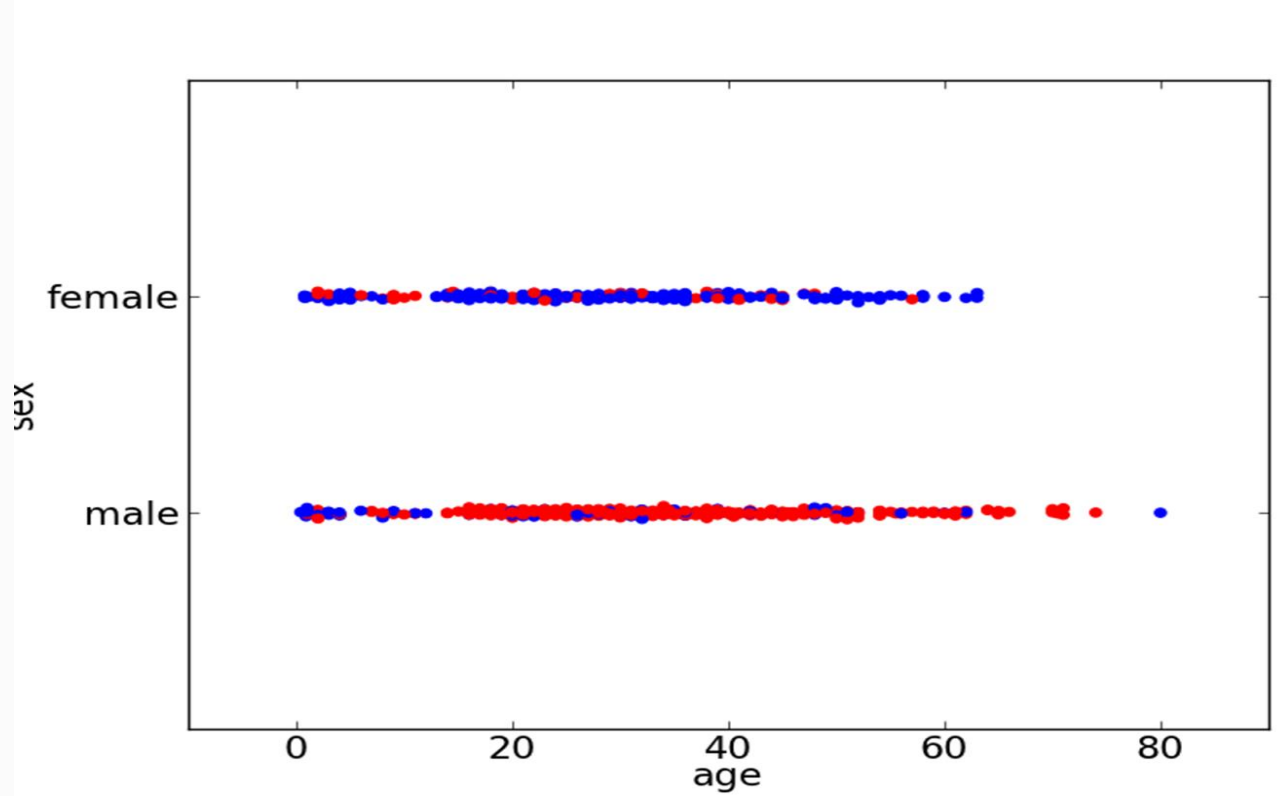
- Look at the data, exploratory data analysis: Excel, WEKA,…;



	A	B	C	D	E	F	G	H	I	J	K
1	PassengerI	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	892	3	Kelly, Mr. .	male	34.5	0	0	330911	7.8292		Q
3	893	3	Wilkes, Mr	female	47	1	0	363272	7		S
4	894	2	Myles, Mr.	male	62	0	0	240276	9.6875		Q
5	895	3	Wirz, Mr. /	male	27	0	0	315154	8.6625		S
6	896	3	Hirvonen,	female	22	1	1	3101298	12.2875		S
7	897	3	Svensson,	male	14	0	0	7538	9.225		S
8	898	3	Connolly, I	female	30	0	0	330972	7.6292		Q
9	899	2	Caldwell, I	male	26	1	1	248738	29		S
10	900	3	Abrahim, I	female	18	0	0	2657	7.2292		C
11	901	3	Davies, Mr	male	21	2	0	A/4 48871	24.15		S
12	902	3	Ilieff, Mr. \	male		0	0	349220	7.8958		S
13	903	1	Jones, Mr.	male	46	0	0	694	26		S
14	904	1	Snyder, Mi	female	23	1	0	21228	82.2667	B45	S

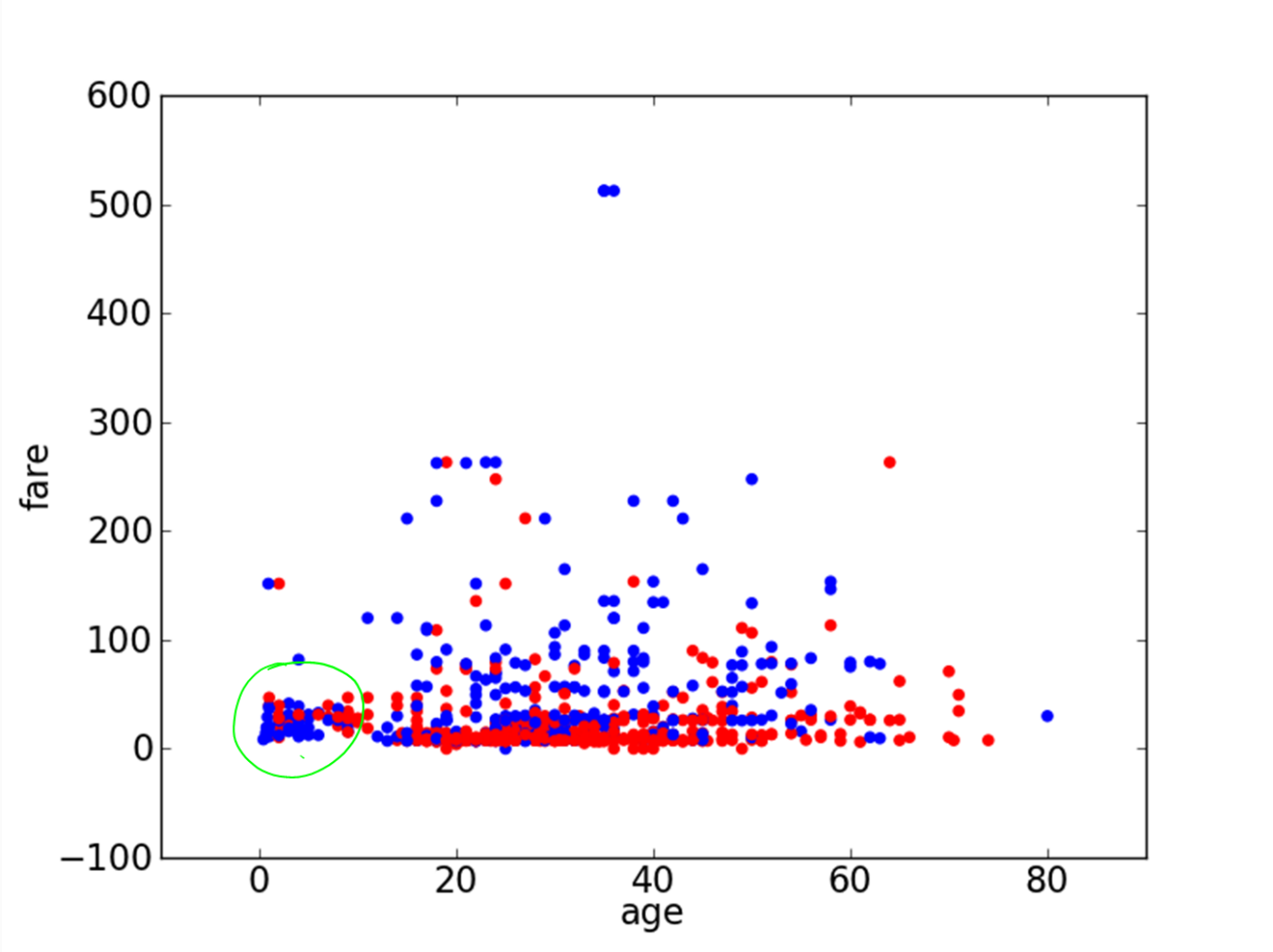
First Steps…

- Look at the data;
- Follow your intuition, *"women and children first…"*
  - *What percentage of Females survived, percentage of males survived?…*



First Steps…

- Other features good predictors?



A very naïve classifier

pclass	sex	age	sibsp	parch	fare	cabin	embarked
1	female	35	1	0	53.1	C123	S

Does the new data point x^* ***exactly*** match a previous point x_i?

If so, assign it to the same class as x_i

Otherwise, just guess.

This is the "rote" classifier

A minor improvement

pclass	sex	age	sibsp	parch	fare	cabin	embarked
1	female	35	1	0	53.1	C123	S

Does the new data point x^* match a set of previous points x_i on **some specific attribute**?

If so, take a vote to determine class.

Example: If most females survived, then assume every female survives

But there are lots of possible rules like this.
And an attribute can have more than two values.

If most people under 4 years old survive, then assume everyone under 4 survives
If most people with 1 sibling survive, then assume everyone with 1 sibling survives

How do we choose?

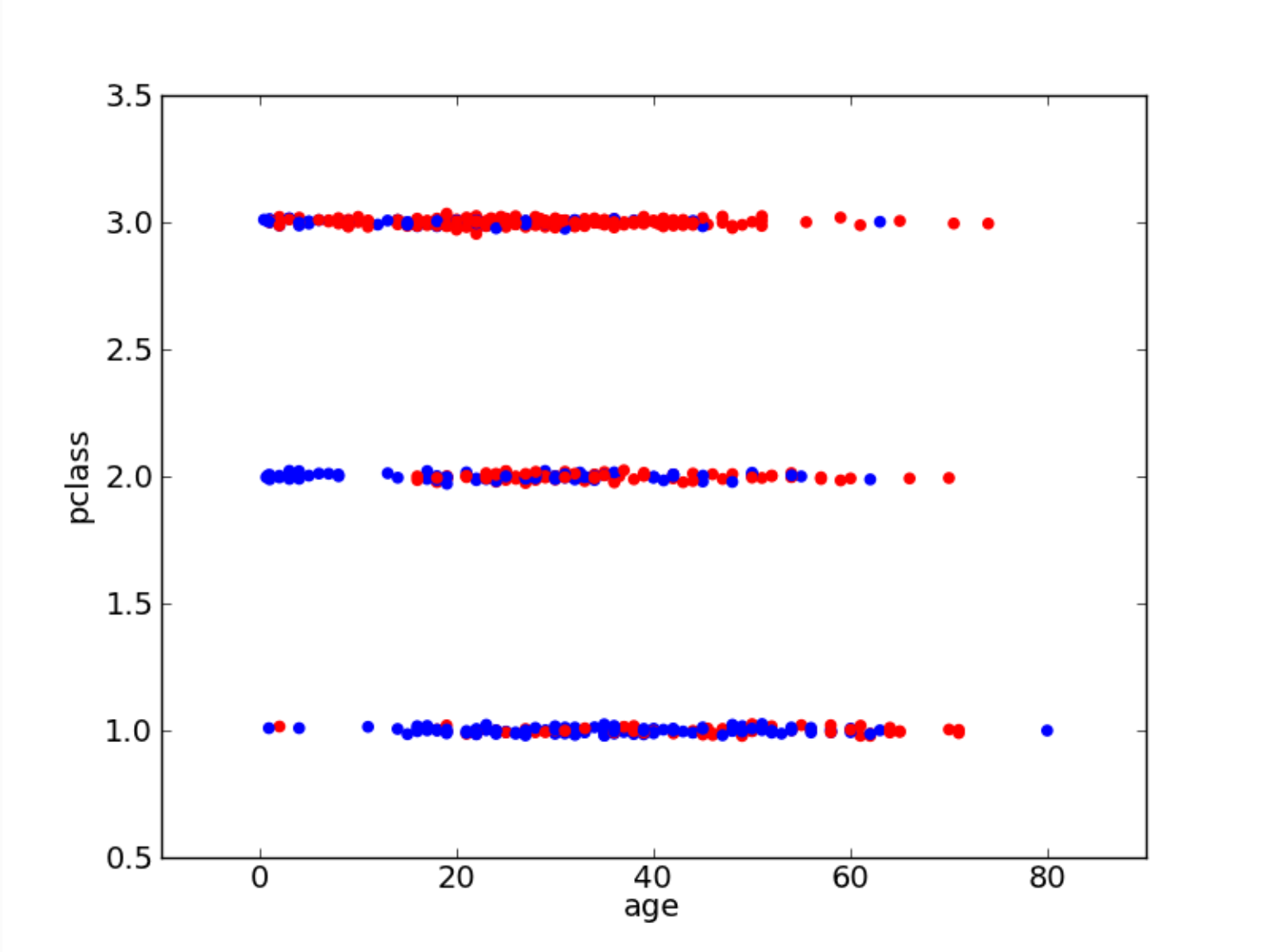
IF sex='female' THEN survive=yes
ELSE IF sex='male' THEN survive = no

```
confusion matrix

no   yes    <-- classified as
468  109  |   no
 81  233  |  yes
```

(468 + 233) / (468+109+81+233) = 79% correct (and 21% incorrect)

Not bad!


3.5
3.0
2.5
2.0
1.5
1.0
0.5
pclass
0
20
40
60
80
age

IF pclass=‘1’ THEN survive=yes
ELSE IF pclass=‘2’ THEN survive=yes
ELSE IF pclass=‘3’ THEN survive=no

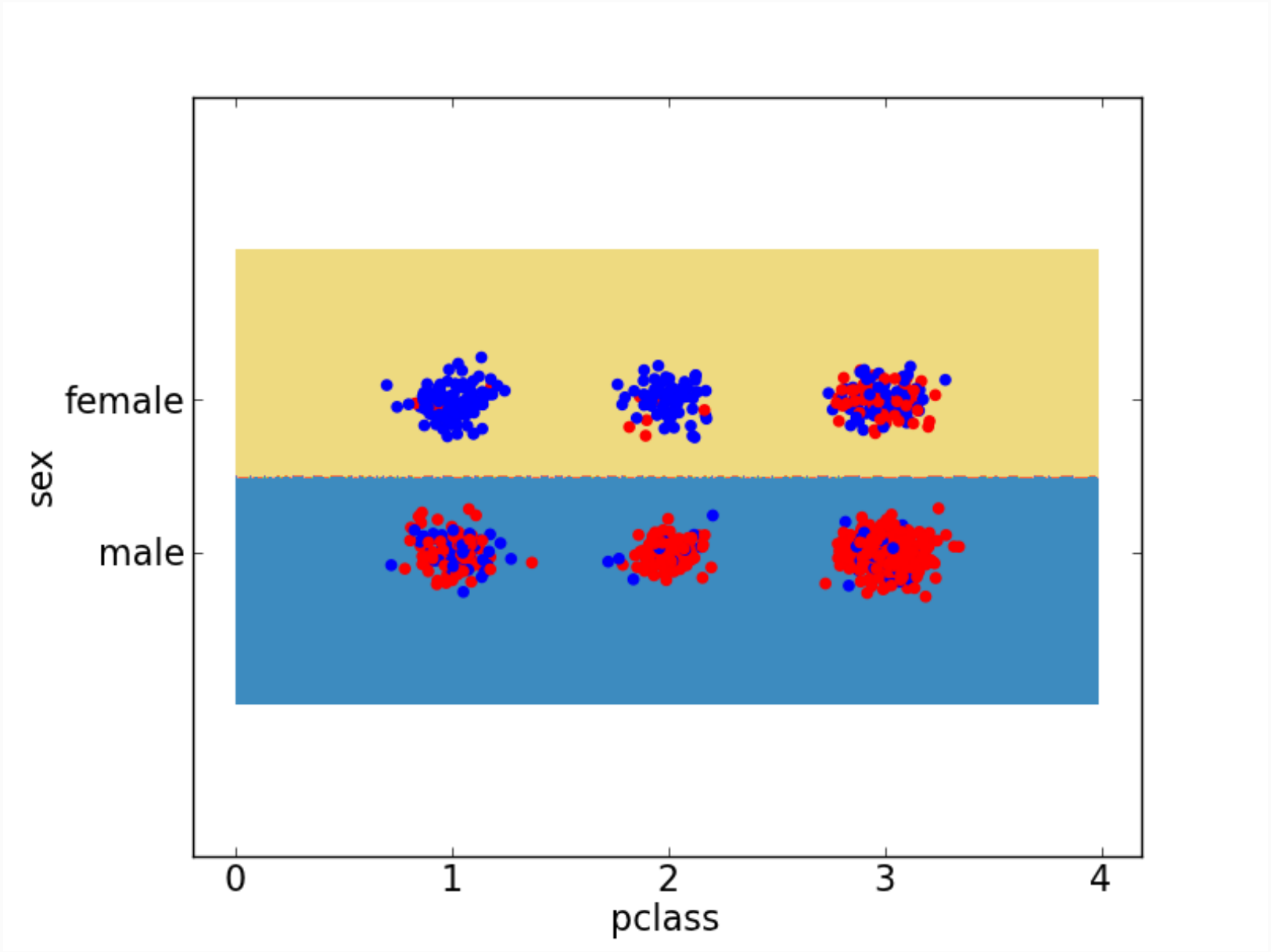
```
confusion matrix

no   yes    <-- classified as
372  119  |   no
177  223  |  yes
```

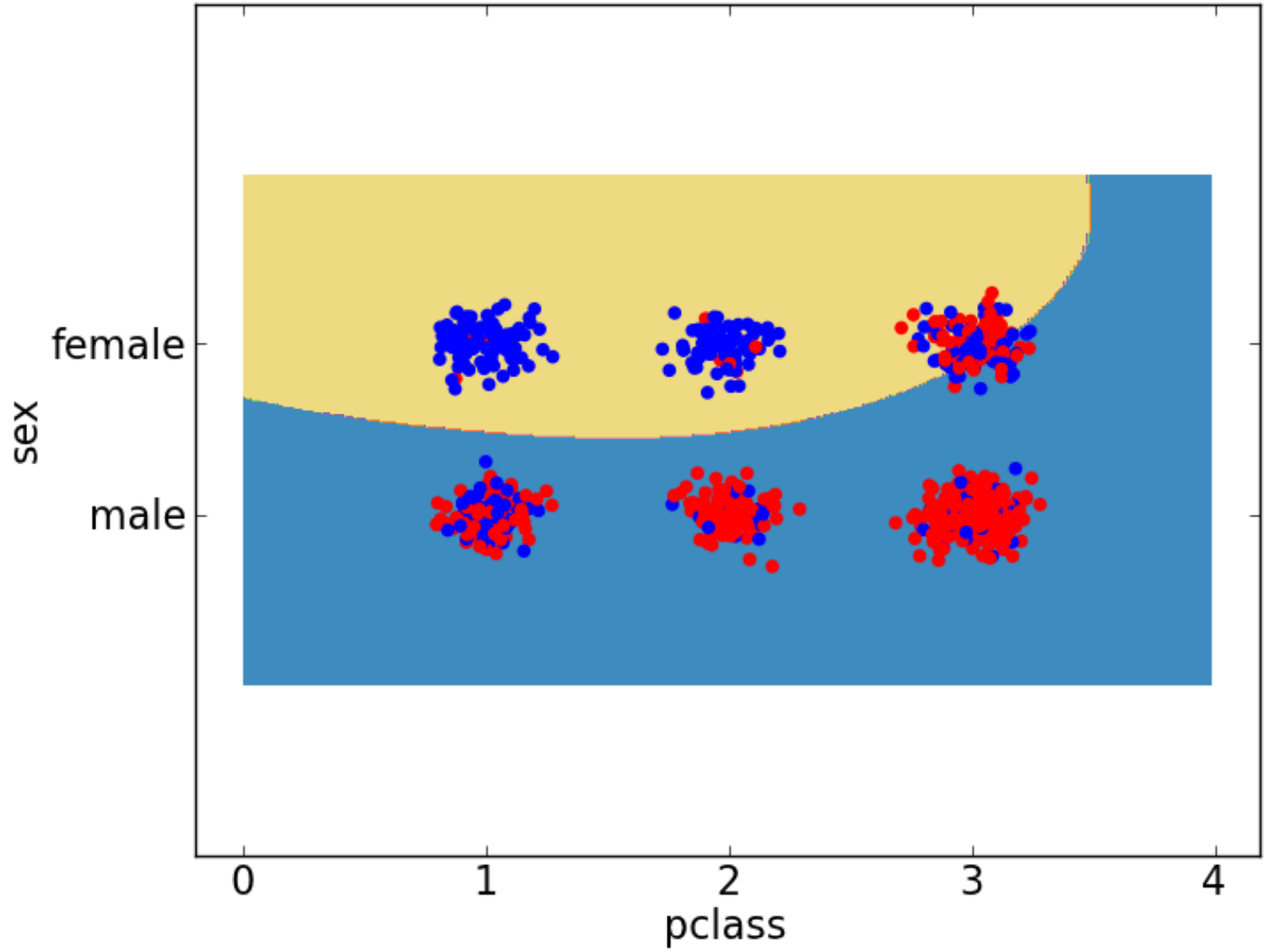
(372 + 223) / (372+119+223+177) = 67% correct (and 33% incorrect)

a little worse

Support Vector Machine Model, Titanic Data, Linear Kernel

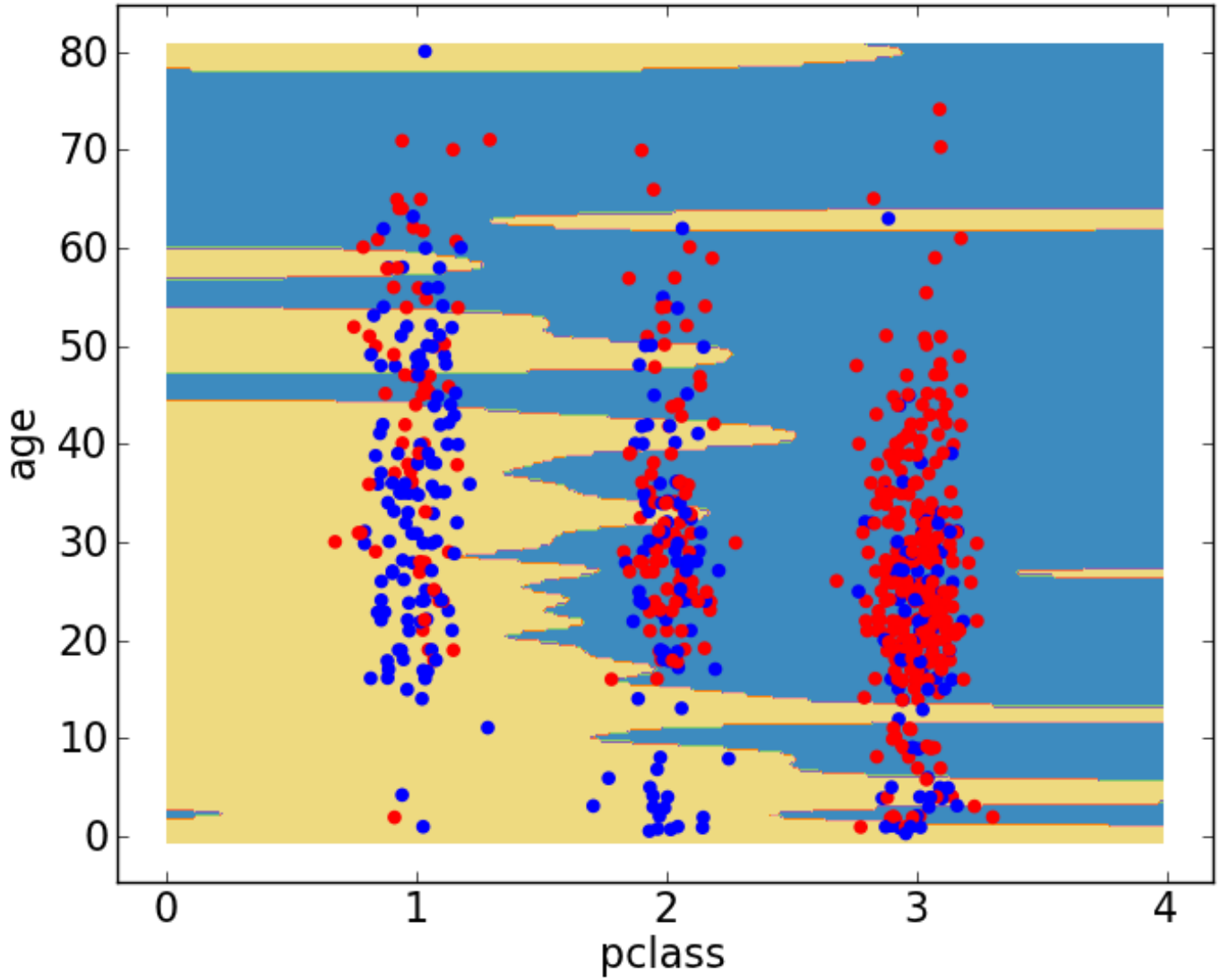


Support Vector Machine Model, RBF Kernel
Titanic Data



Support Vector Machine Model, RBF Kernel

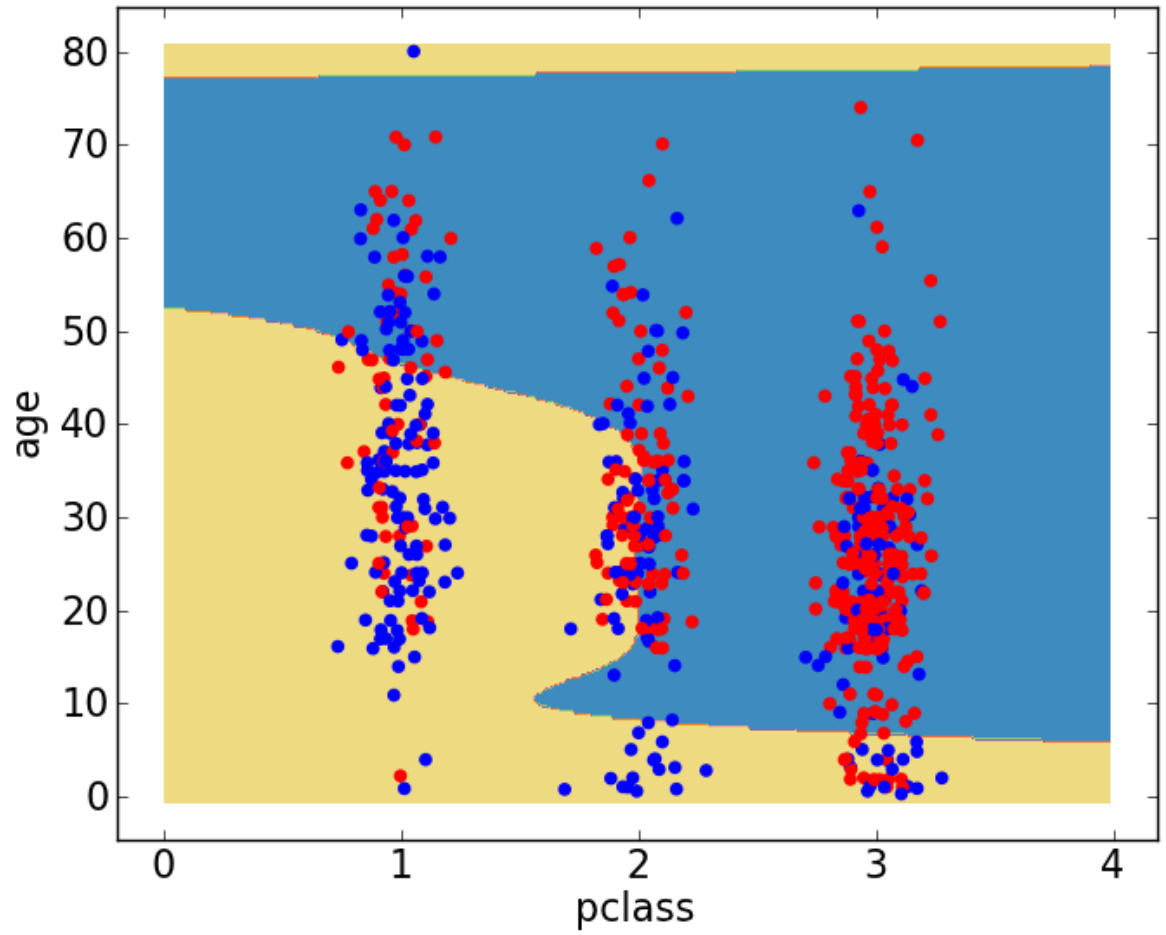
Titanic Data



overfitting?

Support Vector Machine Model, RBF Kernel
Titanic Data

A gamma, parameter that controls/balances model complexity against accuracy



Sparse Data

- SVM algorithms speed up dramatically if the data is sparse (i.e. many values are 0)
- Why? Because they compute lots and lots of dot products
- Sparse data compute dot products very efficiently
- Iterate only over nonzero values
- SVMs can process sparse datasets with 10,000s of attributes

Doing multi-class classification

- SVMs can only handle two-class outputs (i.e. a categorical output variable with arity 2).
- What can be done?
- Answer: with output arity N, learn N SVM's
  - SVM 1 learns "Output==1" vs "Output != 1"
  - SVM 2 learns "Output==2" vs "Output != 2"

    ….
  - SVM N learns "Output==N" vs "Output != N"
- Then to predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

What you need to know…

- First, try logistic regression. Easy, fast, stable. No 'tuning' parameters.
- Or try kNN classifier. It is simple ad fast!
- Equivalently, you can first try linear SVMs, but you need to tune 'C'
- If results are "good enough", stop
- Else, try SVMs with Gaussian kernels (RBF)

Need to tune bandwidth, C – by using validation data…

Summary: Steps for Classification

- SVMs require vector of real numbers
  - Categorical variables → numeric data {R,G,B} → {0,0,1},…{1,0,0}
  - Scaling to the range [-1, +1] or [0,1]
- Select the kernel function to use
  - RBF is a reasonable first choice, two parameters C and γ
  - Grid search to identify best values for parameters
  - Use v-fold cross validation to ensure good performance on test data
- Unseen data can be classified using support vectors

Conclusion

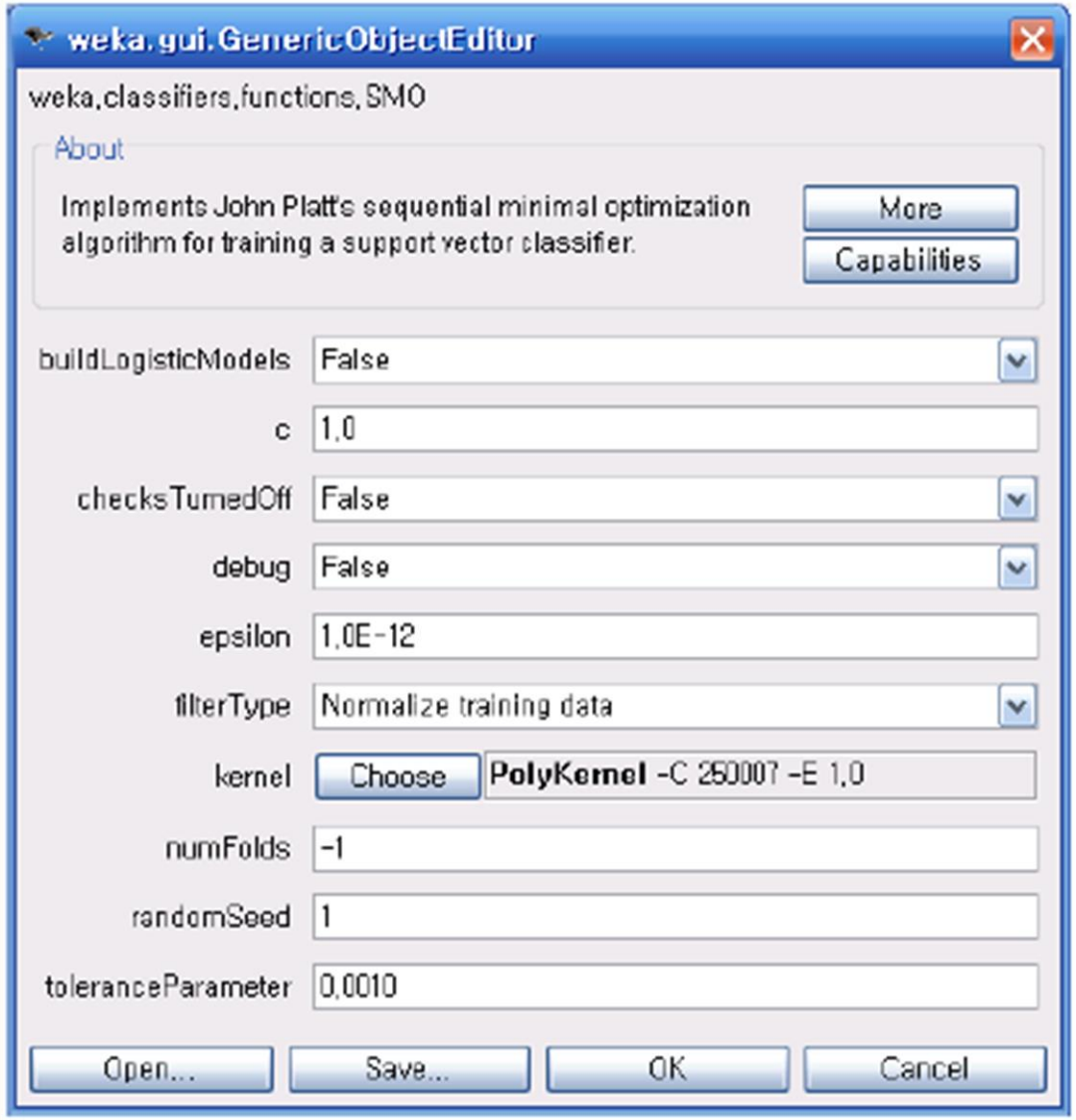
- SVMs balance between correctness and generalization
  - Decision boundaries
  - Margins
  - Support vector
- Two key concepts of SVM: maximize the margin and the kernel trick
- Many SVM implementations are available on the web for you to try on your data set!

Software

- A list of SVM implementation can be found at http://www.kernel-machines.org/software.html
- Some implementation (such as LIBSVM) can handle multi-class classification
- SVMLight is among one of the earliest implementation of SVM
- Several Matlab toolboxes for SVM are also available

In WEKA …

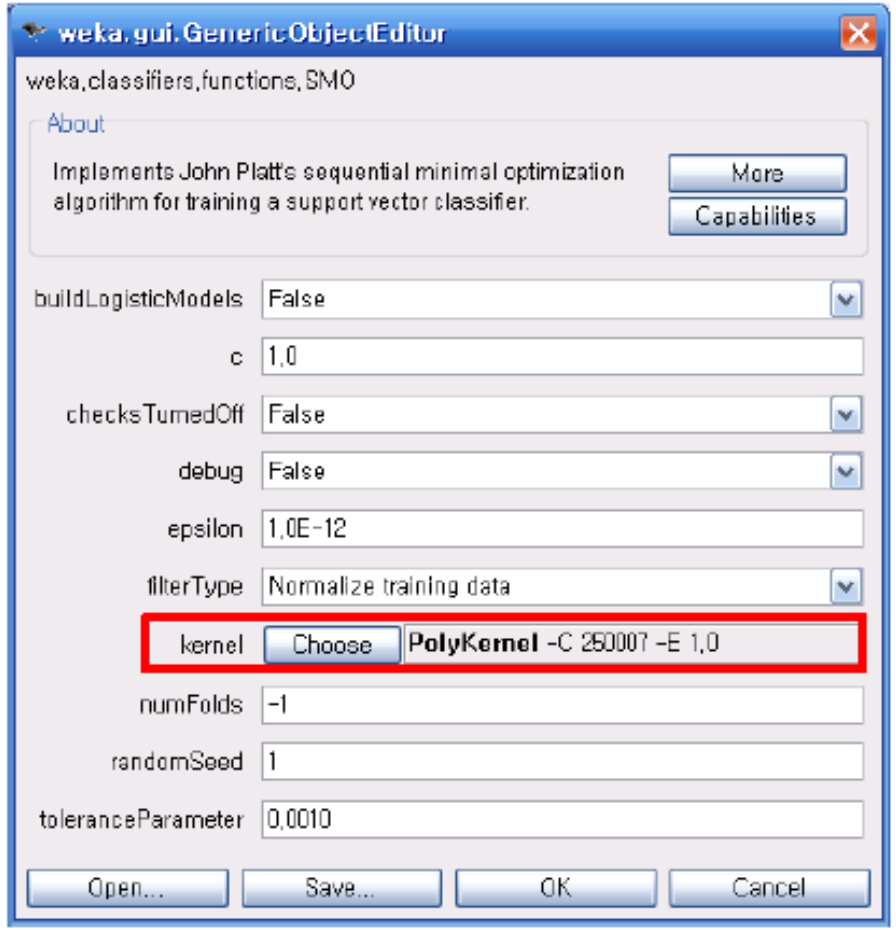
- Sequential minimal optimization (SMO) algorithm support vector *classification*
  - *weka.classifiers.functions.SMO*
- A Library for Support Vector Machines (libSVM)
  - *weka.classifiers.functions.libSVM*
- Sequential minimal optimization (SMO) algorithm support vector *regression*
  - *weka.classifiers.functions.SMOreg*



- buildLogisticModels: Whether to fit logistic models to the outputs (for proper probability estimates)
- numFolds: The number of folds for cross-validation used to generate training data for logistic models
- randomSeed: Random number seed for the cross-validation

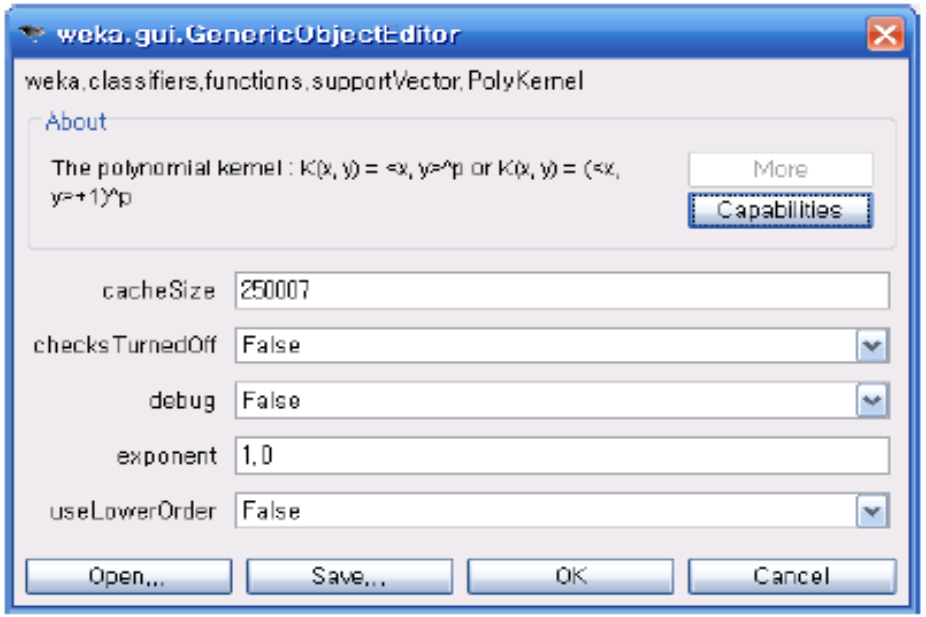
- c -- The complexity parameter C. It is the upper bound of alpha's
- filterType -- Determines how/if the data will be transformed.
- kernel -- The kernel to use.

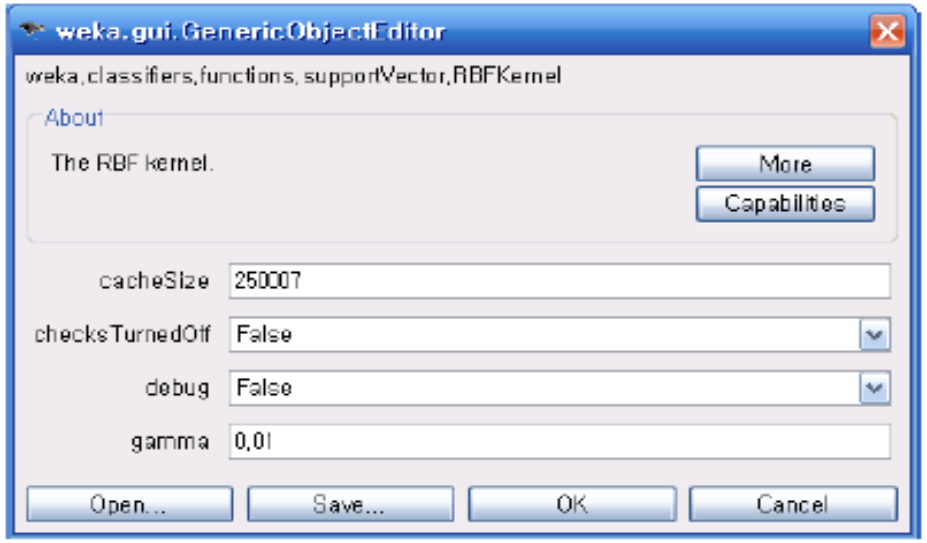
Parameter Setting Guide



- **Suggested**
  - buildLogisticModels: True
  - numFolds: 3 or 5
  - randomSeed: any value
  - C (complexity parameter, upper bound of alpha)
- **Your own choice**
  - filterType
  - Kernel and its subsequent parameters
  - Debug – if on, you can see intermediate results
- **Do not change!**
  - epsilon
  - checksTurnedOff
  - toleranceParameter

Parameter Setting Guide – Kernel





- **PolyKernel**
  - Try various exponents
    - Floating points are allowed
  - 1.0: linear kernel

- **RBFKernel**
  - Try various gammas
  - Gamma value corresponds to the inverse of the variance (width of the kernel)

How They Won It!

Lessons from data mining the past Kaggle contests…

Review at least 10 interviews, bounce around, do not go sequentially.

1) What model(s) did they use?
2) Insights;
3) Feature creation, selection;
4) Other observations.

1, Xavier Conort

- Uses pretty standard algorithms. Random Forests. Gradient Boosting Machines.
- Most of the effort is spent on feature engineering. Careful to discard features that lead to over-fit.
- Spends a lot of time exploring and visualizing the data to understand what brings value.
- Works in a team of 40+ data scientists in Singapore

2, Ben Hammer

- For a time series, took N trailing components as features.
- Used random forests
- Changing the error metric and optimizing for it directly
- Use data to decide how many trailing components to use

3, Tim Salimans

- Used Bayesian analysis
- Used a custom formula for posterior distribution. Change the error metric
- Look at intermediate data (viz, distribution). Change subsequent steps based on that.
- Very noisy, messy data. Bayesian seemed to help with that.
- Gradient-based optimization and random restarts (avoiding local minima)

4, James Petterson

- Used general boosting regression models
- Didn't look at the data much
- Used R
- Used a little bootstrapping to increase size/variety of data (24-hour moving window)
- Surprised that he did so well without more feature extraction or intermediate analysis

5 Minute Break…

Data Exploration

Quick Hands On…

Use Weka to build a predictive model over the bank data set, select the simple decision tree J4.8 with default parameters

- Use **Bank Data A** for one model;
- Use **Bank Data B** for the second model;

Q. Is there a difference in the model performance between the two?

Q. If there is a performance difference, can you identify the reason?

Why Data Preprocessing?

Data in the real world is dirty

- incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
- noisy: containing errors or outliers
- inconsistent: containing discrepancies in codes or names

No quality data, no quality modeling results!

- Quality decisions must be based on quality data
- Requires the collection and maintenance of gold standard data sets.

Data Understanding: Quantity

- Number of instances (records, objects)
  - *Rule of thumb: 5,000 or more desired*
  - If less, results are less reliable; use special methods (boosting, …)
- Number of attributes (fields)
  - *Rule of thumb: for each attribute, 10 or more instances*
  - If more fields, use feature reduction and selection
- Number of targets
  - *Rule of thumb: >100 for each class*
  - If very unbalanced, use stratified sampling or SMOTE

Major Tasks in Data Preprocessing

Data cleaning

- Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies

Data integration

- Integration of multiple databases, data cubes, or files

Data transformation

- Normalization and aggregation

Data reduction

- Obtains reduced representation in volume but produces the same or similar analytical results

Data discretization

- Data reduction but with particular importance, esp. for numerical data

Data Preparation

Data Transformation

Simple transformations can often have a large impact in performance

Example transformations (not all for performance improvement):

- Difference of two date attributes, distance between coordinate,…
- Ratio of two numeric (ratioscale) attributes, average for smoothing,….
- Concatenating the values of nominal attributes
- Encoding (probabilistic) cluster membership
- Adding noise to data (for robustness tests)
- Removing data randomly or selectively
- Obfuscating the data (for anonymity)

Intuition: add features that increase class discrimination (E, IG)…

Data Exploration Process

For each attribute:

Look at data summaries

- Identify potential problems and decide if an action needs to be taken

Visualize the distribution

- Identify potential problems (e.g., one dominant attribute value, even distribution, a relatively small number of unique values, distinct roughly equal to unique, outliers, etc.)
- Evaluate the potential usefulness of attributes

Feature **Selection**, *starts with you…*

Remove fields with no or little variability

- Examine the number of distinct field values
  - *Rule of thumb: remove a field where almost all values are the same*
- Irrelevant attributes in the input data will likely decrease the classification performance (supervised approaches)
- Your goal is to find the **smallest subset of attributes** leading to a higher classification accuracy than all attributes

Data Transformation

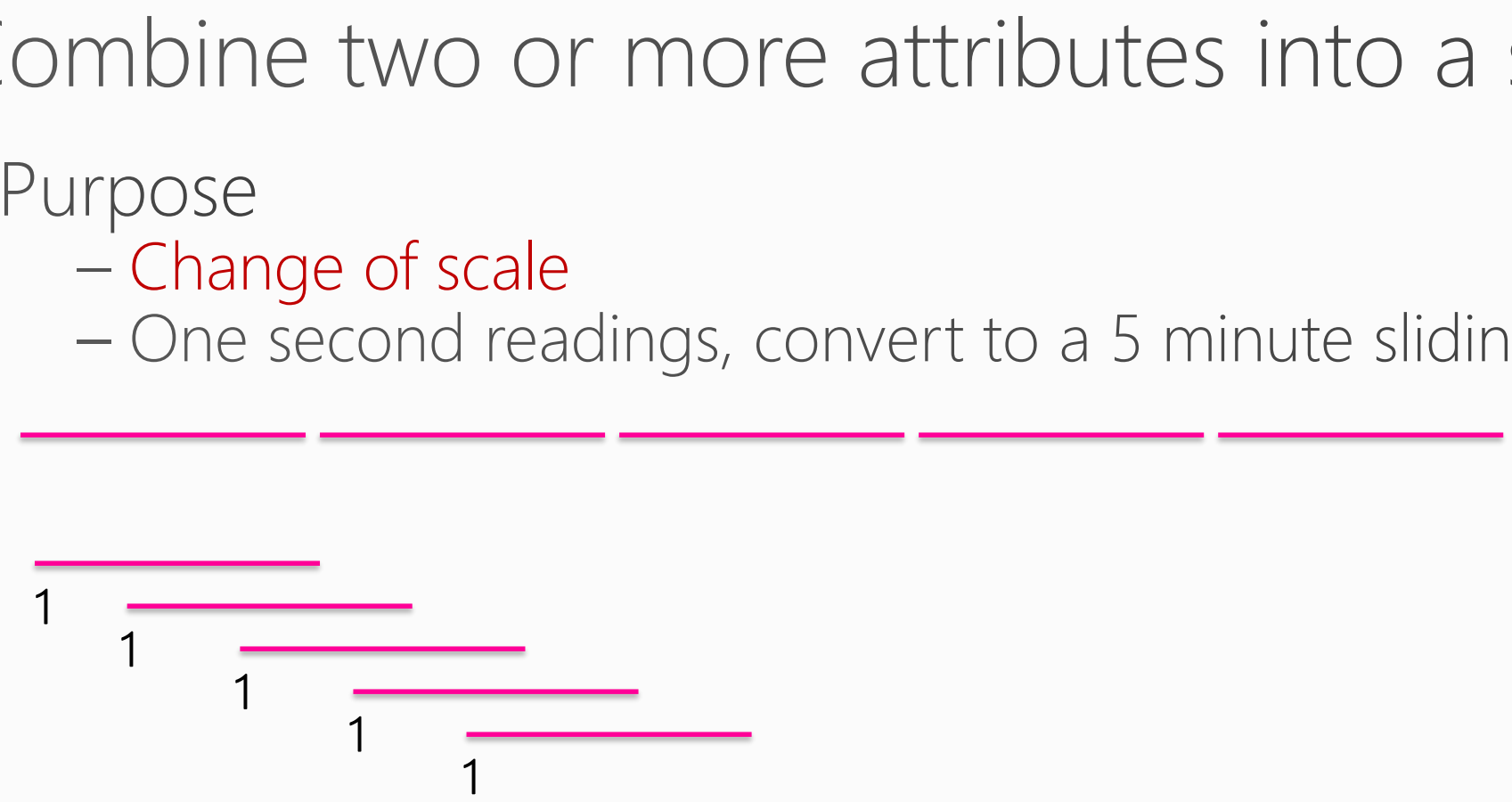
Why transform data?

- **Combine attributes**. For example, the ratio of two attributes might be more useful than keeping them separate
- **Normalizing data**. Having attributes on the same approximate scale helps many data mining algorithms (hence better models)
- **Simplifying data**. For example, working with discrete data is often more intuitive and helps the algorithms (hence better models)

Attribute Aggregation

Combine two or more attributes into a single attribute

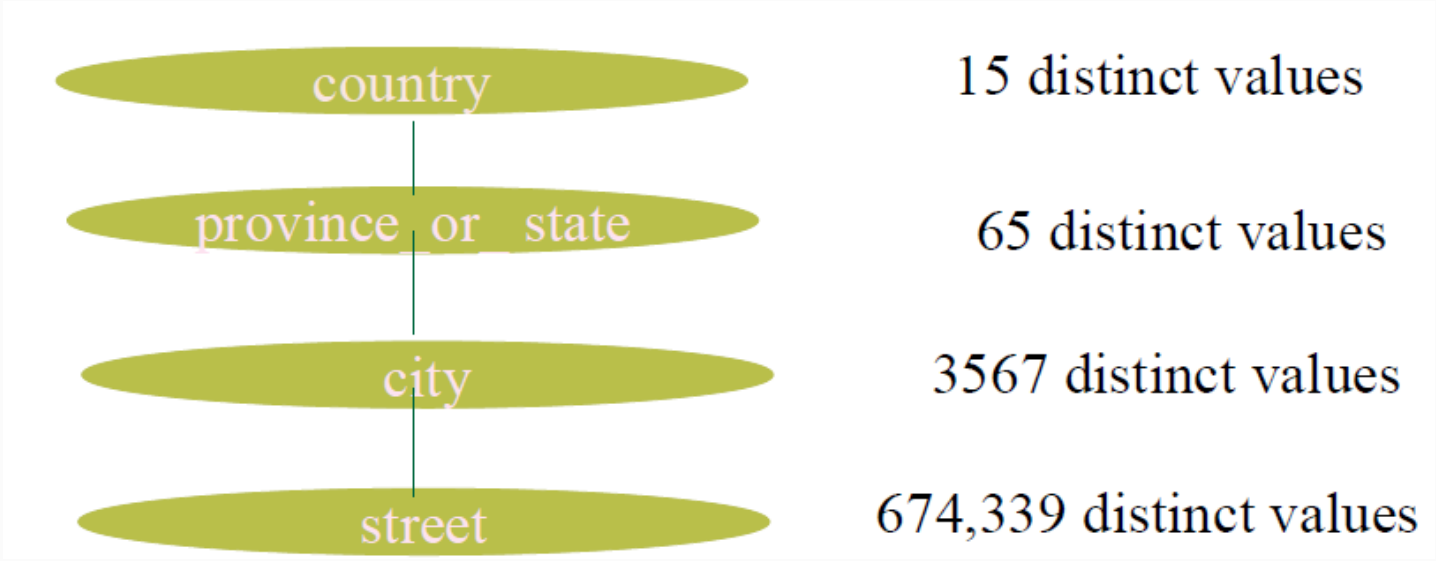
- Purpose
  - Change of scale
  - One second readings, convert to a 5 minute sliding/hopping window



Attribute Aggregation

Combine two or more attributes into a single attribute

- Purpose
  - Change of scale
  - One second readings, convert to a 5 minute sliding/hopping window
- Cities aggregated into regions, states, countries, etc
  - More "stable" data



Attribute Aggregation

Combine two or more attributes into a single attribute

- Purpose
  - Change of scale
  - One second readings, convert to a 5 minute sliding/hopping window
- Cities aggregated into regions, states, countries, etc
  - More "stable" data
- Ratio of two attributes carries more information that separate values
  - More "information" for prediction

Aggregated data tends to have less variability and potentially more information for better predictions…

Explore the Weka Filters

Weka has many filters helpful in preprocessing the data

- Attribute filters
  - Add, remove, or transform attributes
- Instance filters
  - Add, remove, or transform instances
- Process
  - Choose for drop-down menu
  - Edit parameters (if any)
  - Apply

Data Cleaning

- Missing values
  - Weka reports % of missing values
  - Can use filter called ReplaceMissingValues
- Noisy data
  - Due to uncertainty or errors
  - Weka reports unique values
  - Useful filters include
    - RemoveMisclassified
    - MergeTwoValues

Explore the Weka Filters

The data transformation filters in Weka include:

- Add – adds a new attribute to a data set
- AddExpression – adds a new attribute, value computed by expr
- MakeIndicator – filter that creates a new data set, replacing nominal variable with Boolean variable and values;
- NumericTransform – transforms numeric attributes given a transform method: abs, max, min, mean, mode;
- Normalize – by default [0…1], [-1…1], other specified ranges possible;
- Standardize – standardizes all numeric attributes in the data set to have zero means and unit variance;

Discretization

Discretization reduces number of values for a continuous attribute

- Why?
  - Some methods can only use nominal data
    - E.g., in Weka ID3 and Apriori algorithms
  - Helpful if data needs to be sorted frequently (e.g., when constructing a decision tree)

Unsupervised Discretization, Binning

Unsupervised - does not account for classes

- Equal-interval binning – the simplest unsupervised discretization method, determines the minimum and maximum values of the discretized attribute and then divides the range into the user-defined number of equal width discrete intervals.

- Equal-frequency binning – the unsupervised method, which divides the sorted values into k intervals so that each interval contains approximately the same number of training instances. Thus each interval contains n / k (possibly duplicated) adjacent values. k is a user predefined parameter.

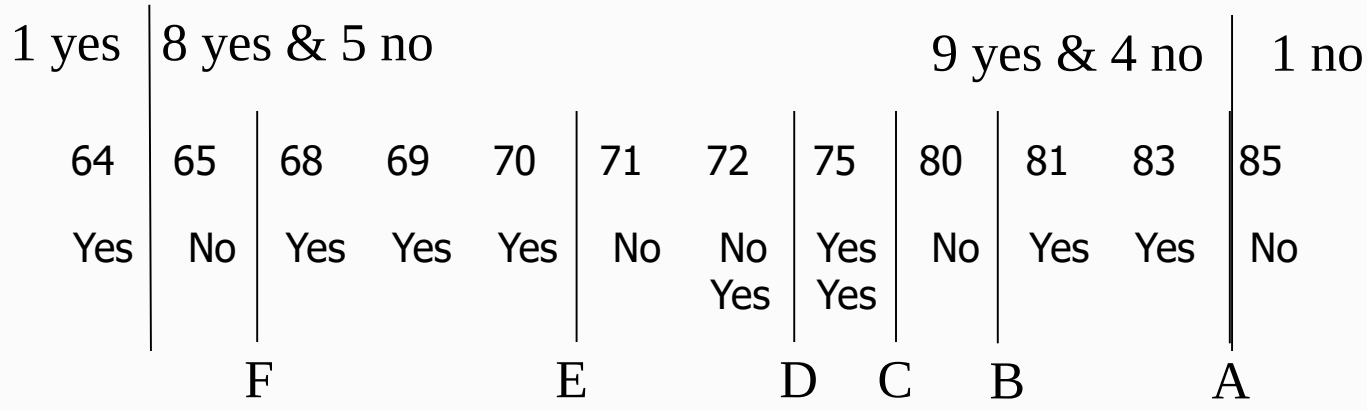
Binning, cont'd

Sort data in ascending order: 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

- Partition into equal-depth bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
- Smoothing by bin means:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
- Smoothing by bin boundaries:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

Supervised Discretization

- Take classification into account
- Use “entropy” to measure information gain
- Goal: Discretise into 'pure' intervals
- Usually no way to get completely pure intervals:



Explore the WEKA features, use in class next week

Discretize

- \Weka\diabetes
- Discretize "age" (equal bins vs equal frequency)

NumericToNominal

- \Weka\diabetes
- Discretize "age" (vs "Discretize" method)

NominalToBinary

- \UCI\autos
- Convert "num-of-doors"
- Convert "drive-wheels"

Data Reduction

- Another way is to reduce the size of the data before applying a learning algorithm (preprocessing)
- Some strategies
  - Sampling
  - Dimensionality reduction
  - Data compression
  - Numerosity reduction

Sampling Principle

Key principle for effective sampling

- Using a sample will work almost as well as using the entire data set, if the sample is representative
- A sample is representative if it has approximately the same property (of interest) as the original set of data

Sampling Techniques

- Different samples
  - Sample without replacement
  - Sample with replacement
  - Cluster sample
  - *Stratified sample*
- Complexity of sampling actually *sublinear*, that is, the complexity is O(*s*) where *s* is the number of samples and *s*<<*n*

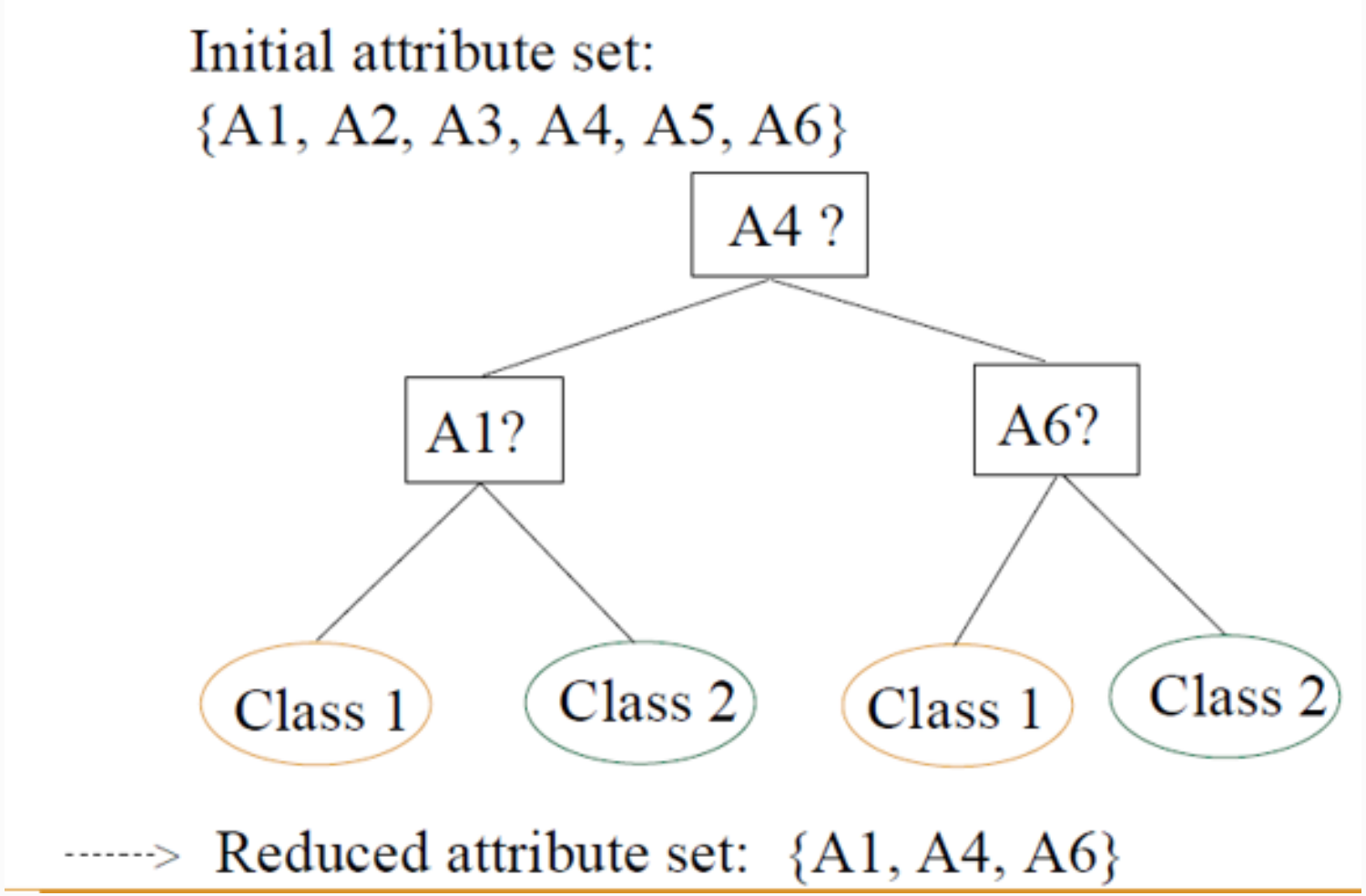
Dimensionality Reduction

- Remove irrelevant, weakly relevant, and redundant attributes
- Attribute selection
  - Many automated methods available
  - E.g., forward selection, backwards elimination, genetic algorithm search
- Often results in a much smaller learning problem
- Often little degeneration in predictive performance or even better performance

Dimensionality Reduction

One we have used in previous exercises?...

Decision Tree Induction...



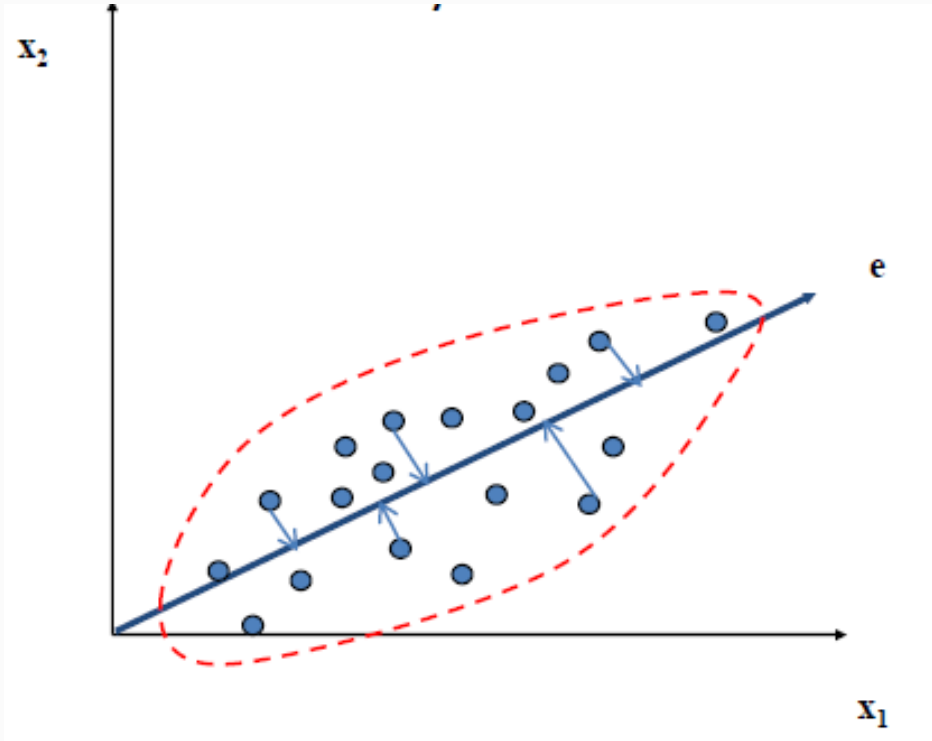
Data Compression

Transform the data into a smaller space

- *Principle Component Analysis*
- Canonical Correlation Analysis (CCA)
- Linear Discriminant Analysis (LDA)
- Independent Component Analysis (ICA)
- Manifold Learning

Principal Component Analysis (PCA)

- Find a projection that captures largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction.



Principal Component Analysis (Steps)

Given data from *n*-dimensions (*n* features), *find k ≤ n new features (principal components) that can best represent data*

- Normalize input data: each feature falls within the same range
- Compute *k principal components* (details omitted)
- Each input data is projected in the new *k*-dimensional space
- The new features (principal components) are sorted in order of decreasing "significance" or strength
- Eliminate *weak components / features* to reduce dimensionality.

→ *Works for numeric data only…*

PCA Exploration in WEKA, try it out…

\UCI\breast-w

- Accuracy with all features
- PrincipalComponents (data transformation)
- Visualize/save transformed data (first two features, last two features)
- Accuracy with all transformed features
- Accuracy with top 1 or 2 feature(s)

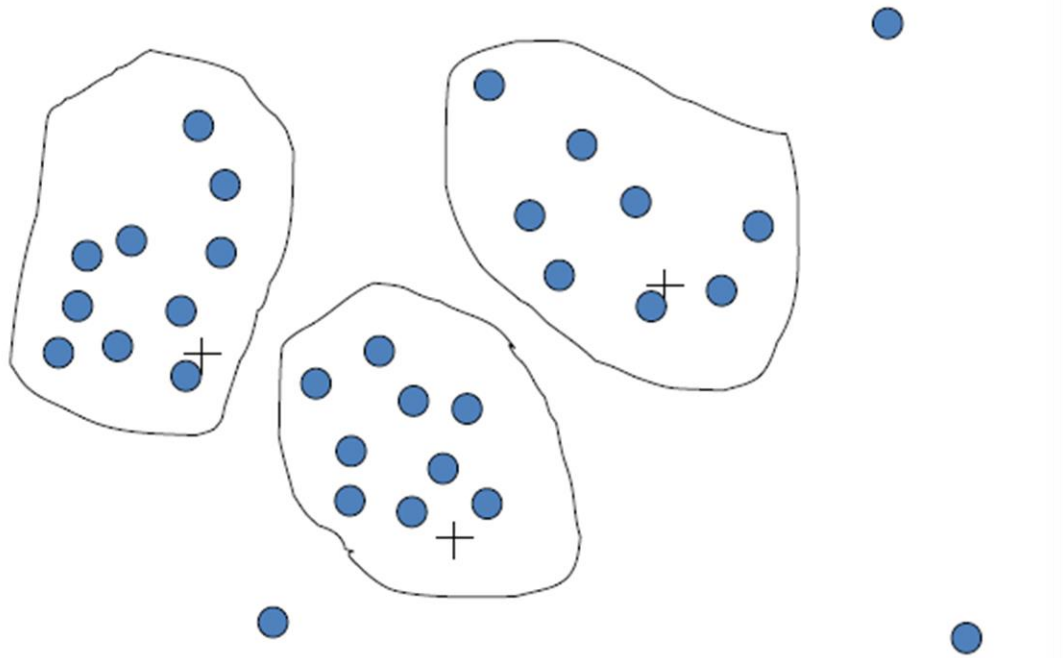
PrincipalComponents under Attribute Selection tab

Numerosity Reduction

- Clustering
  - Data objects (instance) that are in the same cluster can be treated as the same instance
  - Must use a scalable clustering algorithm
- Sampling
  - Randomly select a subset of the instances to be used

Clustering

- Partition data set into clusters, and one can **store cluster representation only**
- Can be very effective if data is clustered but not if data is "smeared"



Homework Lecture 7

Lecture 7 Homework – PART B

Due Date: Thursday August 27

Data Preparation Assignment

Consider the data collected by a hypothetical video store for 50 regular customers. This data consists of a table which, for each customer, records the following attributes: Gender, Income, Age, Rentals (total number of video rentals in the past year), Avg. per visit (average number of video rentals per visit during the past year), Incidentals (whether the customer tends to buy incidental items such as refreshments when renting a video), and Genre (the customer's preferred movie genre). This data is available as an Excel spreadsheet on our class website.

Perform each of the following data preparation tasks:

a. Use **smoothing by bin means** to smooth the values of the **Age** attribute. Use a **bin depth of 4**.
b. Use **min-max normalization** to transform the values of the **Income** attribute onto the range **[0.0-1.0]**.
c. Use **z-score normalization** to standardize the values of the **Rentals** attribute.
d. **Discretize** the (original) **Income** attribute based on the following categories: High = 60K+; Mid = 25K-59K; Low = less than $25K.
e. Convert the original data (not the results of parts a-d) into the **standard spreadsheet format** (note that this requires that you create, for every categorical attribute, additional attributes corresponding to values of that categorical attribute; numerical attributes in the original data remain unchanged).
f. Using the standardized data set (from part e), perform basic correlation analysis among the attributes. Discuss your results by indicating any strong correlations (positive or negative) among pairs of attributes. You need to construct a complete Correlation Matrix (Please read the brief document Basic Correlation Analysis (see course website) for more detail). Can you observe any "significant" patterns among groups of two or more variables? Explain.

Homework Lecture 7

SVM Assignment (optional)

For this exercise, we apply SVM with several different kernels and hyperparameter choices to the veh-prime.arff file. As a first step you will need to modify the .arff file so that the car, noncar classes are re-placed with 1 and -1 respectively. Import this new file into Weka and then select the SMO classifier found under classifiers/function. Use 10 fold cross-validation (this should come up as the default).

You can make kernel and hyperparameter choices by clicking on "SMO ....." appearing next to Choose. You will make 5 runs of the algorithm. Select PolyKernel with exponent options 1, 2 and 4. Then select RBFKernel with gamma set to 0.01 and 1.0. For each run record the number of correctly and incorrectly classified instances.

If some of the choices do not work well, explain why you think this is the case.

public class **SMO**
extends DistributionClassifier
implements OptionHandler
Implements John C. Platt's sequential minimal optimization algorithm for training a support vector classifier using polynomial kernels. Transforms output of SVM into probabilities by applying a standard sigmoid function that is not fitted to the data. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. For more information on the SMO algorithm, see

J. Platt (1998). *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. Advances in Kernel Methods - Support Vector Learning,

S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy (2001). *Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation, 13(3), pp 637-649, 2001.*

Note: for improved speed normalization should be turned off when operating on SparseInstances. Valid options are:

5 minute Break…

Attribute Selection

Reasons for Attribute Selection

Simpler model

- More transparent
- Easier to interpret

Faster model induction

- What about overall time?

Structural knowledge

- Knowing which attributes are important may be inherently important to the application

What about the accuracy?

Variable Ranking – Feature Selection

A simple method for feature selection using variable ranking is to select the k highest ranked features according to S.

This is usually not optimal

but often preferable to other, more complicated methods

Computationally efficient(!): only calculation and sorting of n scores

Feature Ranking

Steps

1. Rank all the individual features according to certain criteria (e.g., information gain, gain ratio, $\chi 2$)
2. Select / keep top *N* features

Properties

- Usually independent of the learning algorithm to be used
- Efficient (no search process)
- Hard to determine the threshold
- Unable to consider correlation between features

By convention a high score indicates a valuable feature.

Ranking Criteria – Correlation

Correlation Criteria:

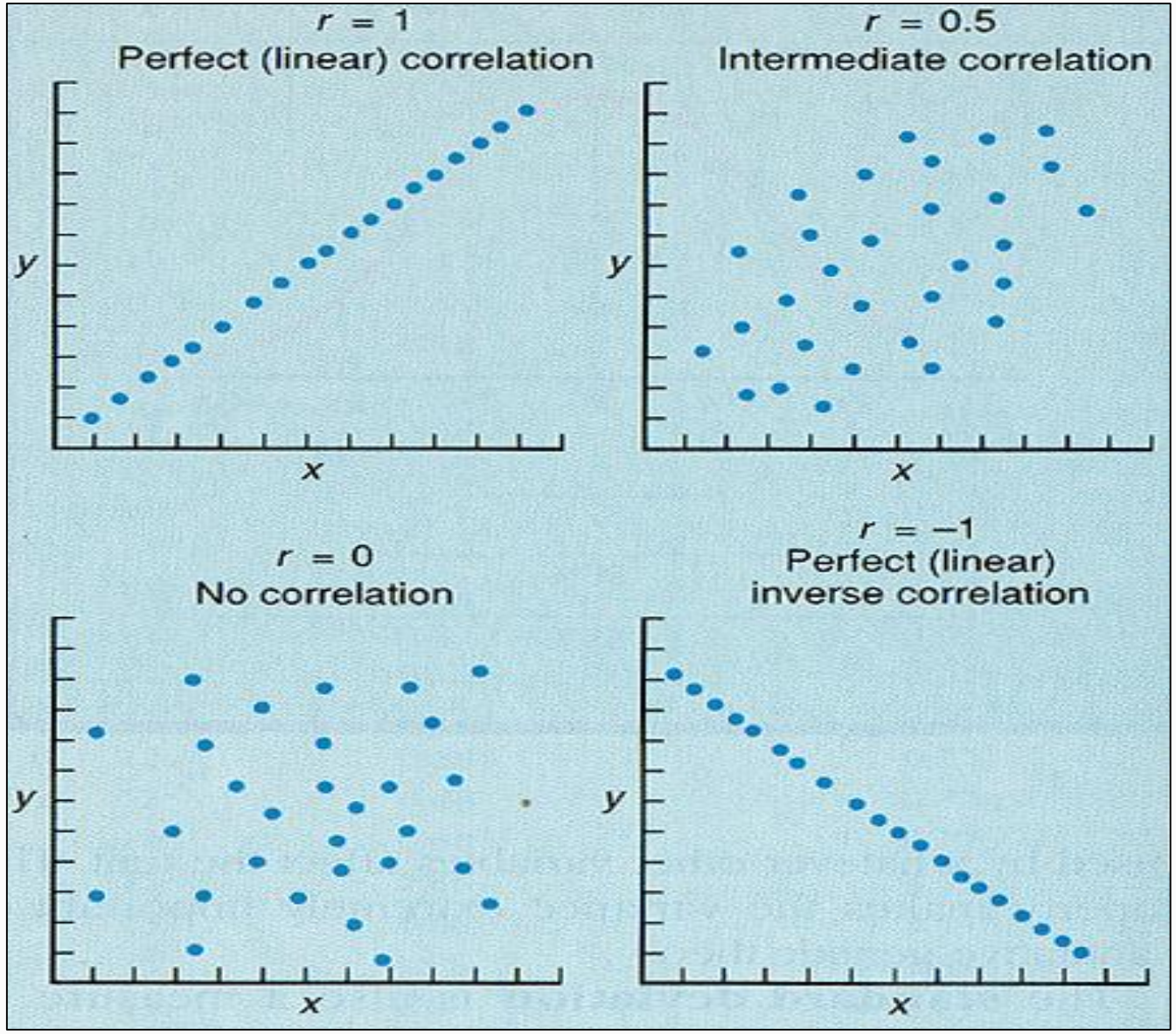
- Pearson correlation coefficient

- mostly $R(x_i,y)^2$ or $|R(x_i,y)|$ is used

- measure for the goodness of **linear** fit of x_i and y.
  (can only detect linear dependencies between variable and target.)

The higher the correlation between the feature and target, the higher the score!

Ranking Criteria – Correlation



Ranking Criteria – Correlation

Correlation Criteria:

- $R(X,Y) \in [-1,1]$
- mostly $R(x_i,y)^2$ or $|R(x_i,y)|$ is used
- measure for the goodness of **linear** fit of x_i and y.
  (can only detect linear dependencies between variable and target.)

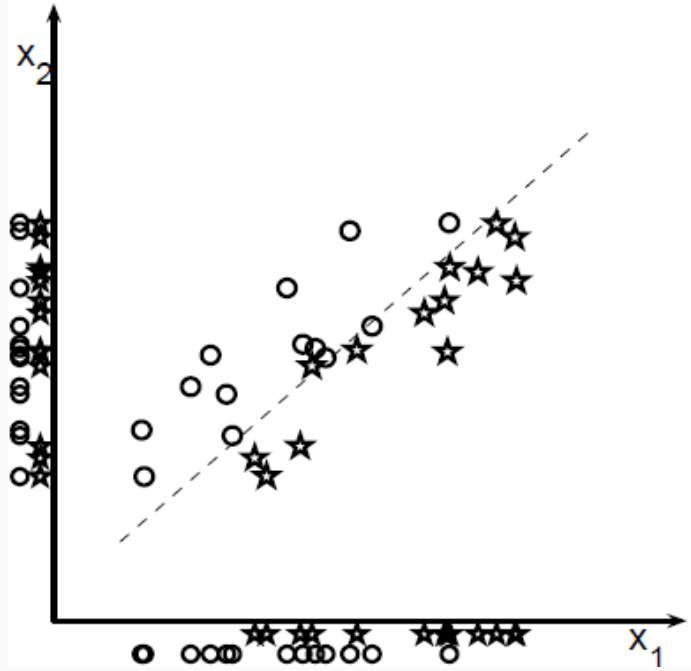
Ranking Criteria – Correlation

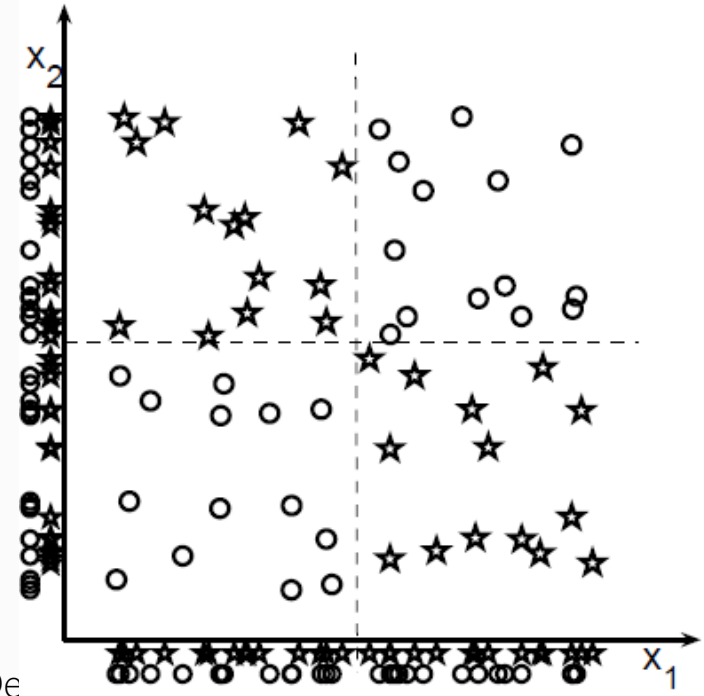
Questions:

- Can variables with small score be automatically discarded ?
- Can a useless variable (i.e. one with a small score) be useful together with others ?
- Can two variables that are useless by themselves can be useful together?)

Ranking Criteria – Correlation
Multivariate Dependencies

- Can variables with small score be discarded without further consideration?
  NO!
- Even variables with small score can improve class separability!
- Multivariate dependencies
  (a) Feature X2 is individually irrelevant to Y, but it becomes relevant in the context of feature X1;
  (b) Two individually irrelevant features become relevant when taken jointly





De

Ranking Criteria – Correlation

- Correlation between variables and target are not enough to assess relevance!
- Correlation / covariance between pairs of variables has to be considered too! (potentially very difficult to perform)
- **Diversity** of features

Feature Selection and Engineering

Filter Methods

Results in either

- Ranked list of attributes
  - Typical when each attribute is evaluated individually
  - Must select how many to keep
- A selected subset of attributes
  - Forward selection
  - Best first
  - Random search such as genetic algorithm

Filter Evaluation Examples

- Information Gain
- Gain ratio
- Relief

Optimal Correlation

- High correlation with class attribute
- Low correlation with other attributes

Feature Selection and Engineering

Wrapper Methods

Wrapper methods find features that work best with some particular learning algorithm:

- Best features for k-Means, SVM or a neural network may not be best features for decision trees;
- Can eliminate features learning algorithm "has trouble with";

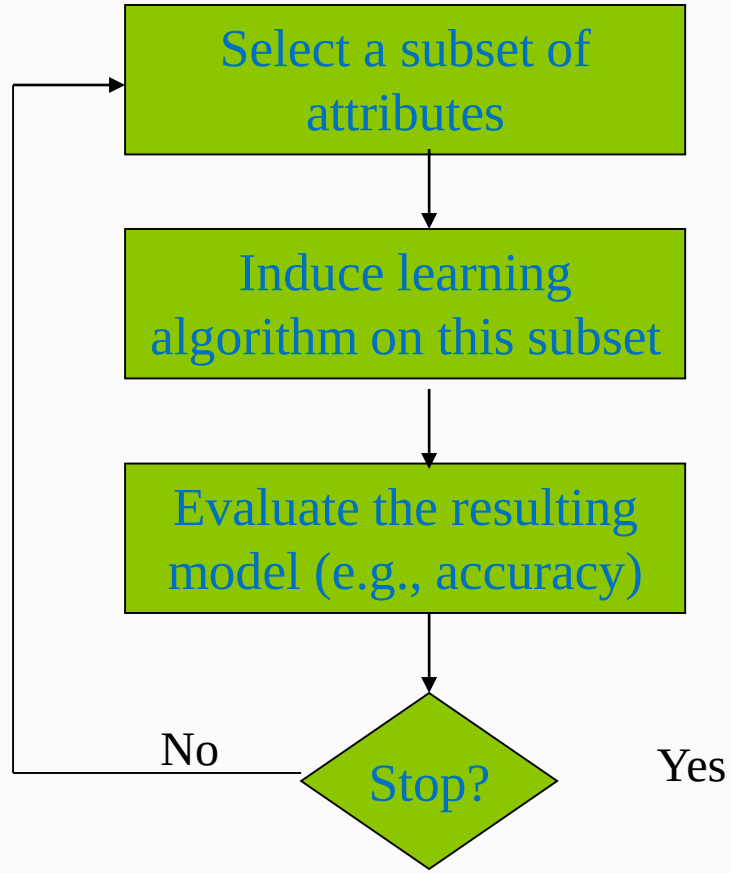
Forward stepwise selection, *forward reference…*

Backwards elimination, *backward reference…*

Bi-directional stepwise selection and elimination

Wrappers

- "Wrap around" the learning algorithm
- Must therefore always evaluate subsets
- Return the best subset of attributes
- Apply for each learning algorithm
- Use same search methods as before



Forward Feature Selection

Steps

1. First select the best single-feature (according to the learning algorithm)
2. Repeat (until some stop criterion is met):
   Select the next best feature, given the already picked features

Properties

- Usually learning algorithm dependent
- Feature correlation is considered
- More reliable
- Inefficient

Feature Selection and Engineering

Heuristics

SFS (Sequential Forward Selection)

- Begins with zero attributes
- Evaluates all features subsets w/ exactly 1 feature
- Selects the one with the best performance
- Adds to this subsets the feature that yields the best performance for subsets of next larger size
- If **EVAL()** is a heuristics measure, the feature selection algorithm acts as a filter, extracting features to be used by the main algorithm; If it is the actual accuracy, it acts as a wrapper around that algorithm

Feature Selection and Engineering

Heuristics

SFS (Sequential Forward Selection)

```
SS = 0
BestEval = 0
REPEAT
   BestF = None
   FOR each feature F in FS AND NOT in SS
      SS' = SS ∪ {F}
      IF Eval(SS') > BestEval THEN
                BestF = F; BestEval = Eval(SS')
   IF BestF <> None THEN SS = SS ∪ {BestF}
UNTIL BestF = None OR SS = FS
RETURN SS
```

Backward Feature Elimination

Steps

1. First build a model based on all the features
2. Repeat (until some criterion is met):
   Eliminate the feature that makes the least contribution.

Properties

- Usually learning algorithm dependent
- Feature correlation is considered
- More reliable
- Inefficient

Feature Selection and Engineering

Heuristics

SBS (Sequential Backward Selection)

```
SS =FS
BestEval = Eval(SS)
REPEAT
   WorstF = None
   FOR each feature in F in FS
      SS' = SS - {F}
      IF Eval(SS') >= BestEval THEN
               WorstF = F; BestEval = Eval(SS')
   IF WorstF <> None THEN SS = SS - {WorstF}
UNTIL WorstF = None OR SS = 0
RETURN SS
```

Weka, try it yourself (will cover in class next week)

Feature Ranking

- \Weka\weather
- ChiSquared, InfoGain, GainRatio

FFS & BFE

- \Weka\Diabetes
- ClassifierSubsetEval + GreedyStepwise

Summary, Heuristic Search in Feature Selection

- Given d features, there are 2d possible feature combinations
  - Exhaust search won't work
  - Heuristics have to be applied
- Typical heuristic feature selection methods:
  - Feature ranking
  - Forward feature selection
  - Backward feature elimination
  - Bidirectional search (selection + elimination)
  - Search based on evolution algorithm
  - ……

Summary

- In real world applications, data preprocessing usually occupies about 70% workload in a data mining task.
- Domain knowledge is usually required to do good data preprocessing.
- To improve a predictive performance of a model
  - –Improve learning algorithms (different algorithms, different parameters)
- Most data mining research focuses on here
    - –Improve data quality ---- data preprocessing
- Deserve more attention!

Filter vs Wrapper Model

Filter model

- –Separating feature selection from learning
- –Relying on general characteristics of data (information, etc.)
- –No bias toward any learning algorithm, fast
- –Feature ranking usually falls into here

Wrapper model

- –Relying on a predetermined learning algorithm
- –Using predictive accuracy as goodness measure
- –High accuracy, computationally expensive
- –FFS, BFE usually fall into here

Summary, Attribute Selection Methods

What is evaluated?

Evaluation Method

	Attributes	Subsets of attributes
Independent	Filters	Filters
Learning algorithm		Wrappers

Feature Selection and Engineering

Important points 1/2

- Feature selection can significantly increase the performance of a learning algorithm (both accuracy and computation time) – but it is not easy!

- Relevance <-> Optimality

- Correlation and Mutual information between single variables and the target are often used as Ranking-Criteria of variables.

Feature Selection and Engineering

Important points 2/2

- One cannot automatically discard variables with small scores – they may still be useful together with other variables.
- Filters – Wrappers - Embedded Methods
- How to search the space of all feature subsets ?
- How to asses performance of a learner that uses a particular feature subset ?

Course Project Discussion

Data Science

Deriving Knowledge from Data at Scale

That's all for tonight…

Valentine N. Fontama

August 20th, 2015