Lab for Media Search

NUS National University of Singapore

Social Media Computing

Lecture 4: Introduction to Information Retrieval and Classification

Lecturer: Aleksandr Farseev

E-mail: farseev@u.nus.edu

Slides: http://farseev.com/ainlfruct.html

At the beginning, we will talk about text a lot (text IR), but most off the techniques are applicable to all the other data modalities after feature extraction.

Purpose of this Lecture

- To introduce the background of text retrieval (IR) and classification (TC) methods
- Briefly introduce the machine learning framework and methods
- To highlight the differences between IR and TC
- Introduce evaluation measures and some TC results
- *Note: Many of the materials covered here are background knowledge for those who have gone thru IR and AI courses*

References:

IR:

- Salton (1988), Automatic Text Processing, Addison Wesley, Reading. Salton G (1972). Dynamic document processing. Comm of ACM, 17(7), 658-668

Classification:

- Yang Y & Pedersen JO (1997). A comparative study n feature selection in text categorization. Int'l Conference on Machine Learning (ICML), 412-420.
- Yang Y & Liu X (1999). A re-examination of text categorization methods. Proceedings of SIGIR'99, 42-49.
- **Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.**

Contents

- Free-Text Analysis and Retrieval
- Text Classification
- Classification Methods

Something from previous lecture…

What is Free Text?

- Unstructured sequence of text units with uncontrolled set of vocabulary, Example:

> To obtain more accuracy in search, additional information might be needed - such as the adjacency and frequency information. It may be useful to specify that two words must appear next to each other, and in proper word order
>
> Can be implemented by enhancing the inverted file with location information.

- Information must be analyzed and indexed for retrieval purposes
- Different from DBMS, which contains structured records:

Name: <s>	Sex: <s>	Age: <i>	NRIC: <s>	

Analysis of Free-Text -1

- Analyze document, *D*, to extract patterns to represent *D*:
- General problem:
  - To extract minimum set of (distinct) features to represent contents of document
  - To distinguish a particular document from the rest – **Retrieval**
  - To group common set of documents into the same category – **Classification**
- Commonly used text features
  - LIWC
  - Topics
  - N-Grams
  - Etc…

Analysis of Free-Text -2

- Most of the (large-scale) text analysis systems are term-based:
  - IR:
    - perform pattern matching,
    - no semantics,
    - general
  - Classification: similar
- We know that simple representation (single terms) **performs quite well**

Retrieval vs. Classification

- **Retrieval**: Given a query, find documents that best match the query
- **Classification**: Given a class, find documents that best fit the class

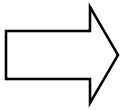
- What is the big DIFFERENCE between retrieval and classification requirements???

Analysis Example for IR

Free-text page:

To obtain more accuracy in search, additional information might be needed - such as the adjacency and frequency information. It may be useful to specify that two words must appear next to each other, and in proper word order.

Can be implemented by enhancing the inverted file with location information.



Text pattern extracted:

information x 3
Words
Word
Accuracy
Search
Adjacency
Frequency
Inverted
File
Location
implemented
….

Term Selection for IR

- Research suggests that (INTUITIVE !!):
  - high frequency terms are not discriminating
  - low to medium frequency terms are useful (enhance precision)
- A Practical Term Selection Scheme:
  - eliminate high frequency words
    (by means of a stop-list with 100-200 words)
  - use remaining terms for indexing
- One possible Stop Word list (more in the web)

also am an and are be because been

could did do does from

had hardly has have having he hence her here hereby herein hereof hereon hereto herewith him his however

if into it its me nor of on onto or our really

said she should so some such …… etc

Term Weighting for IR -1

- Precision ***(fraction of retrieved instances that are relevant)*** is better served by features that occur frequently in a small number of documents
- One such measure is the Inverse Doc Frequency (idf):

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

  - N - total # of doc in the collection
  - Denominator - # of doc where term t appears

- EXAMPLE: In a collection of 1000 documents:
  - ALPHA appears in 100 Doc, idf = 3.322
  - BETA appears in 500 Doc, idf = 1.000
  - GAMMA appears in 900 Doc, idf = 0.132

Term Weighting for IR -2

- General, idf helps in precision
- tf helps in recall ***(fraction of relevant instances that are retrieved)***

$$\mathrm{tf}(t,d) = 0.5 + \frac{0.5 \times \mathrm{f}(t,d)}{\max\{\mathrm{f}(t,d) : t \in d\}}$$

- Denominator - maximum raw frequency of any term in the document

- Combine both gives the famous tf.idf weighting scheme for a term k in document i as:

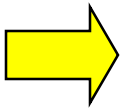
$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

Prev. Lesson: Term Normalization -1

Free-text page:

To obtain more accuracy in search, additional information might be needed - such as the adjacency and frequency information. It may be useful to specify that two words must appear adjacent to each other, and in proper word order.

Can be implemented by enhancing the inverted file with location information.



Text pattern extracted:

information x 3
words
word
accuracy
search
Adjacency
adjacent
frequency
inverted
file
location
implemented
….

to x 3
in x 2
the x 3
and x 2
is
more
might
that
such
as
two
by
….

Stop Word List

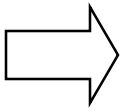
Prev. Lesson: Term Normalization -2

- What are the possible problems here?

Free-text page:

To obtain more accuracy in search, additional information might be needed - such as the adjacency and frequency information. It may be useful to specify that two words must appear adjacent to each other, and in proper word order.

Can be implemented by enhancing the inverted file with location information.



Text pattern extracted:

information x 3
Words
Word
Accuracy
Search
Adjacency
adjacent
Frequency
Inverted
File
Location
implemented
….

Prev. Lesson: Term Normalization -3

- Hence the NEXT PROBLEM:
  - Terms come in different grammatical variants
- Simplest way to tackle this problem is to perform stemming
  - to reduce the number of words/terms
  - to remove the variants in word forms, such as:
    RECOGNIZE, RECOGNISE,
    RECOGNIZED, RECOGNIZATION
  - hence it helps to identify similar words
- Most stemming algorithms:
  - only remove suffixes by operating on a dictionary of common word endings, such as -SES, -ATION, -ING etc.
  - might alter the meaning of a word after stemming

- DEMO: SMILE Stemmer (http://smile-stemmer.appspot.com/)

Putting All Together for IR

- Term selection and weighting for Docs:
  - Extract unique terms from documents
  - Remove stop words
  - Optionally:
    - *use thesaurus* – *to group low freq terms*
    - *form phrases* – *to combine high freq terms*
    - assign, say, tf.idf weights to stems/units
  - Normalize terms
- Do the same for query

Demo of Thesaurus: http://www.merriam-webster.com/

Similarity Measure

- Represent both query and document as weighted term vectors:
  - $\underline{Q} = (q_1, q_2, \ldots q_t)$
  - $\underline{D_i} = (d_{i1}, d_{i2}, \ldots d_{it})$
- A possible query-document similarity is:
  - $sim(\underline{Q},\underline{D_i}) = \sum(q_j \cdot d_{ij}), \quad j = 1,.. T$
- The similarity measure may be normalized:
  - $sim(\underline{Q},\underline{D_i}) = \sum (q_j \cdot d_{ij}) / |\underline{Q}| \cdot |\underline{D_i}|, \quad j = 1,..,T$

  → cosine similarity formula

A Retrieval Example

- Given:

$\underline{Q}$ = “information”, “retrieval”

$\underline{D_1}$ = “information retrieved by VS retrieval methods”
$\underline{D_2}$ = “information theory forms the basis for probabilistic methods”
$\underline{D_3}$ = “He retrieved his money from the safe”

- Document representation:

{info, retriev, method, theory, VS, form, basis, probabili, money, safe}

$\underline{Q}$ = {1, 1, 0, 0 …}

$\underline{D_1}$ = {1, 2, 1, 0, 1, …}
$\underline{D_2}$ = {1, 0, 1, 1, 0, 1, 1, 1, 0, 0}
$\underline{D_3}$ = {0, 1, 0, 0, 0, 0, 0, 0, 1, 1}

- The results:
  - Use the similarity formula: $sim(\underline{Q},\underline{D_i}) = \sum(q_j \cdot D_{ij})$
  - $sim(\underline{Q},\underline{D_1}) = 3$; $sim(\underline{Q},\underline{D_2}) = 1$; $sim(\underline{Q},\underline{D_3}) = 1$
  - Hence $\underline{D_1} >> \underline{D_2}$ and $\underline{D_3}$

Contents

- Free-Text Analysis and Retrieval
- Text Classification
- Classification Methods

Introduction to Text Classification

- Automatic assignment of pre-defined categories to free-text documents
- More formally:

  Given: *m* categories, & *n* documents, n >> m

  Task: to determine the probability that one or more categories is present in a document
- Applications: to automatically
  - assign subject codes to newswire stories
  - filter or categorize electronic emails (or spams) and on-line articles
  - pre-screen or catalog document in retrieval applications
- Many methods:
  - Many machine learning methods: kNN, Bayes probabilistic learning, decision tree, neural network, multi-variant regression analysis ..

Dimensionality Curse

- Features used
  - Most use single term, as in IR
  - Some incorporate relations between terms, eg. term co-occurrence statistics, context etc.
- Main problem: high-dimensionality of feature space
  - Typical systems deal with 10 of thousands of terms (or dimensions)
  - More training data is needed for most learning techniques
  - For example, for dimension D, typical Neural Network may need a minimum of $2D^2$ good samples for effective training

Feature Selection

- Aims:
  - Remove features that have little influence on categorization task
- Differences between IR and TC (Hypothesis)
  - IR favors rare features?
    - Retains all non-trivial terms
    - Use idf to select rare terms
  - TC needs common features in each category?
    - df is more important than idf..

Document Frequency

- Document Frequency $df(t_k)$
  - **$df(t_k)$: # of docs containing term t_k (at least once)**
  - IR gives high weights to terms with low $df(t_k)$, thru idf weight
  - What about TC method?
- Document Frequency Thresholding
  - TC prefers terms with high df
  - Assumptions: rare terms are non-informative in TC task
  - One approach is to retain terms t_k if $df(t_k) > \boldsymbol{\sigma}$
  - Training: needs training documents to determine $\boldsymbol{\sigma}_i$ for each category
- Advantages: simple, scale well, and performs well as it favors common terms
- Disadvantage: ad-hoc

Other Term Selection Methods

- Mutual Information
  - Measures the co-occurrence probabilities of t_k & category C_i
  - Favours rare terms
- Term Strength
  - Estimates term importance based on how commonly a term is likely to appear in “closely related” documents
  - A global measure – not category specific
- Principal Component Analysis (PCA)
  - PCA is widely used in multimedia applications, like face detection, to select features, but not popular in TC
  - Select top k components (or transformed features) with largest eigen-values
- Etc…

Contents

- Free-Text Analysis and Retrieval
- Text Classification
- Classification Methods

“Prediction is very difficult, especially about the future.”

Robert Storm Petersen (1882-1949)
Danish cartoonist, writer, animator, illustrator, painter and humorist

Classification Method

- Once the feature set has been selected, the next problem is the choice of classification method
  - In general, choice of classification method is not as critical as feature selection
  - Feature selection → need problem insights
  - Less so for classification method (Actually many would argue…)
- Many popular methods: kNN, Naïve Bayes, SVM, Decision Tree, Neural Networks ...
- Issues:
  - Parameter tuning
  - Problems of skewed category distribution

Classification Process

- Supervised Learning Approach
- Given: A set of training documents: $\underline{d}_i$, i=1, .., N with the category assignment:

$$y(\underline{d}_i, C_j) = \begin{cases} 1, & if\ \underline{d}_i \in C_j \\ 0, & otherwise \end{cases}$$

  - Different category might have different number of training documents
  - Each document i may be assigned to one or more categories
- Test document set
  - Each document may belong to one or more categories
- Unsupervised Learning: clustering
- Semi-supervised Learning: bootstrapping

Classification Process - **Training** Stage

1. Given the training document set: {$\underline{d}_i$, i=1, .., N} with:

$$y(\underline{d}_i, C_j) = \begin{cases} 1, & if\ \underline{d}_i \in C_j \\ 0, & otherwise \end{cases}$$

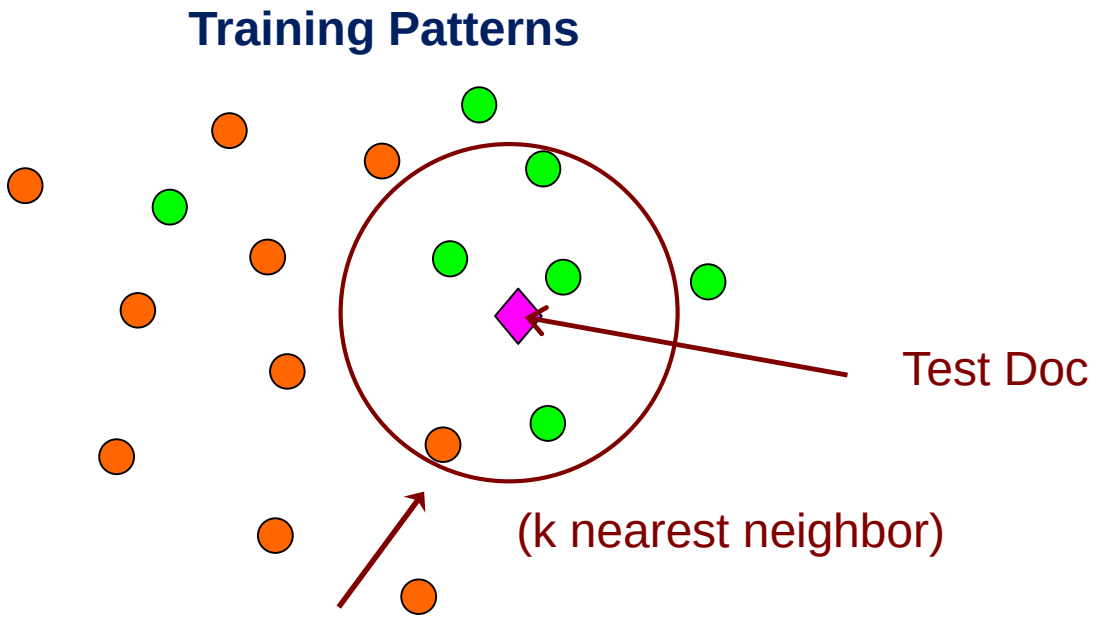
2. Pre-process all training documents:
   - extract all words + convert them to lower case
   - remove stop words
   - Stemming
   - apply a feature selection method to reduce dimension
3. Employ a Classifier to perform the training (or learning)
   - To learn a classifier
   - To derive other info such as the category-specific thresholds (each local classifier requires different category threshold)

Classification Process - **Testing** or Operation Stage

1. Given a test document x
2. Perform the pre-processing steps above on x
3. Retain only words in dictionary extracted in training stage to derive the test document x
4. Use the classifiers with category threshold to perform the classification

kNN Classifier

- How it works graphically (for two class problem):



Basic idea: Use the types and similarity measures of these k nearest neighbors as basis for classification

Hence: New doc belongs to green category since majority of its NNs is of type green

kNN Classifier -2
The Algorithm

- Given a test document $\underline{x}$
- Find k nearest neighbors among training documents, s.t.:
  - $\underline{d_i} \in kNN$, if $Sim(\underline{x}, \underline{d_i}) \geq SIM_k$
    where SIM_K is the top k^{th} Sim values
  - use Cosine Similarity measure for $Sim(\underline{x}, \underline{d_i})$
  - set k to i.e. to 45 (thru experimentation)

- The decision rule for kNN can be written as:

$$y(\underline{x}, \underline{c}_j) = \sum_{d_i \in kNN} \{Sim(\underline{x}, \underline{d}_i) * y(\underline{d}_i, c_j)\} - b_j$$

where b_j is the category-specific threshold

kNN Classifier -3
The Category-specific Threshold

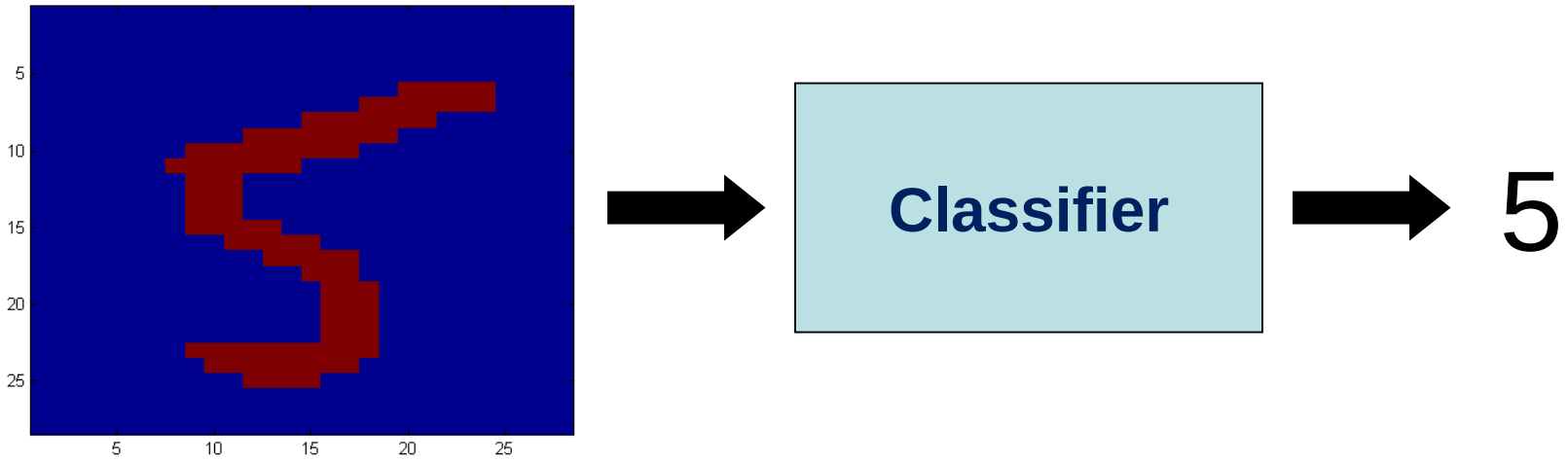
- Category specific threshold b_j is determined using a validation set of documents
  - Divide training documents into two sets
  - T-set (say, 80%) and V-set (the remaining 20%)
  - Train the classifier using T-set
  - Validate using V-set to adjust b_j such that the F_1 value of the overall categorization is maximized
- Category specific threshold b_j permits new document to be classified into multiple categories more effectively

- kNN is an online classifier:
  - does not carry out off-line learning
  - "remembers" every training sample

Bayes Classifier -1

Application – Digit recognition

- **Digit Recognition**



- **$X_1,\ldots,X_n \in \{0,1\}$ (Red vs. Blue pixels)**
- **$Y \in \{5,6\}$ (predict whether a digit is a 5 or a 6)**

Bayes Classifier -2

Main idea

- In class, we saw that a good strategy is to predict:

$$\arg\max_{Y} P(Y|X_1,\ldots,X_n)$$

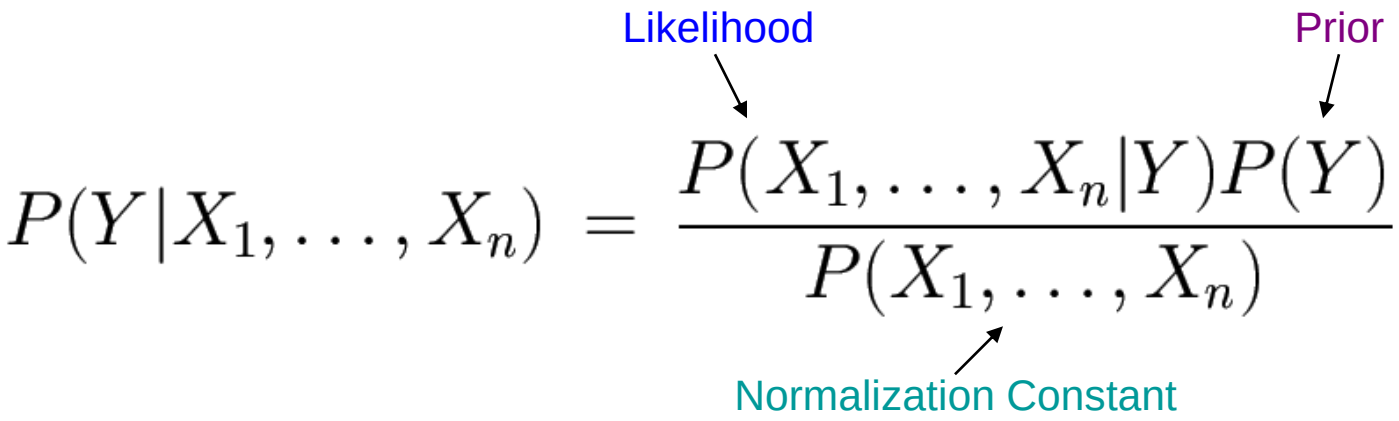
  - (for example: what is the probability that the image represents a 5 given its pixels?)

- So … How do we compute that?

Bayes Classifier -3

Main Rule

- Use Bayes Rule!



- Why did this help? Well, we think that we might be able to specify how features are "generated" by the class label.

Bayes Classifier -4

Example

- Let's expand this for our digit recognition task:

$$P(Y=5|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y=5)P(Y=5)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}$$

$$P(Y=6|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y=6)P(Y=6)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}$$

- To classify, we'll "simply" compute these two probabilities and predict based on which one is greater

Bayes Classifier -5
Model Parameters

- The problem with explicitly modeling $P(X_1,\ldots,X_n|Y)$ is that there are usually too many parameters:
  - We'll run out of space
  - We'll run out of time
  - And we'll need tons of training data (which is usually not available)

Why Naïve Bayes called Naive???

Naïve Bayes Classifier -1

Naïve Bayes Assumption

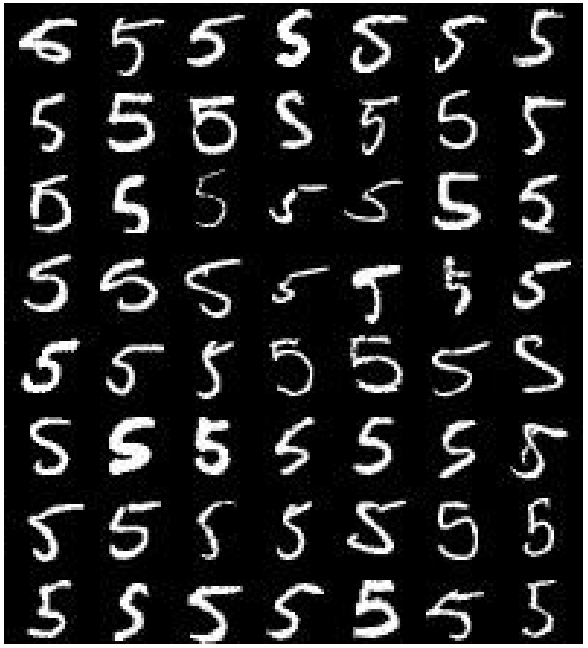
- The *Naïve Bayes Assumption*: Assume that all features are **independent given the class label Y**
- Equationally speaking:

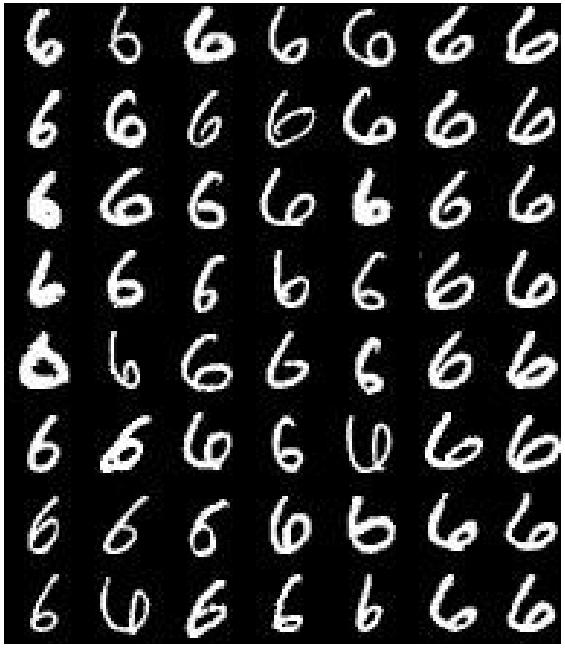
$$P(X_1, \ldots, X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

- # of parameters for modeling $P(X_1,\ldots,X_n|Y)$:
  - $2(2^n-1)$
- # of parameters for modeling $P(X_1|Y),\ldots,P(X_n|Y)$
  - $2n$

Naïve Bayes Classifier -2
Training

- Now that we've decided to use a Naïve Bayes classifier, we need to train it with some data:





Naïve Bayes Classifier -3
Training

- Training in Naïve Bayes is **easy**:
  - Estimate P(Y=v) as the fraction of records with Y=v

$$P(Y = v) = \frac{Count(Y = v)}{\#\ records}$$

  - Estimate $P(X_i=u|Y=v)$ as the fraction of records with Y=v for which $X_i=u$

$$P(X_i = u|Y = v) = \frac{Count(X_i = u \wedge Y = v)}{Count(Y = v)}$$

Naïve Bayes Classifier -4
Smoothing

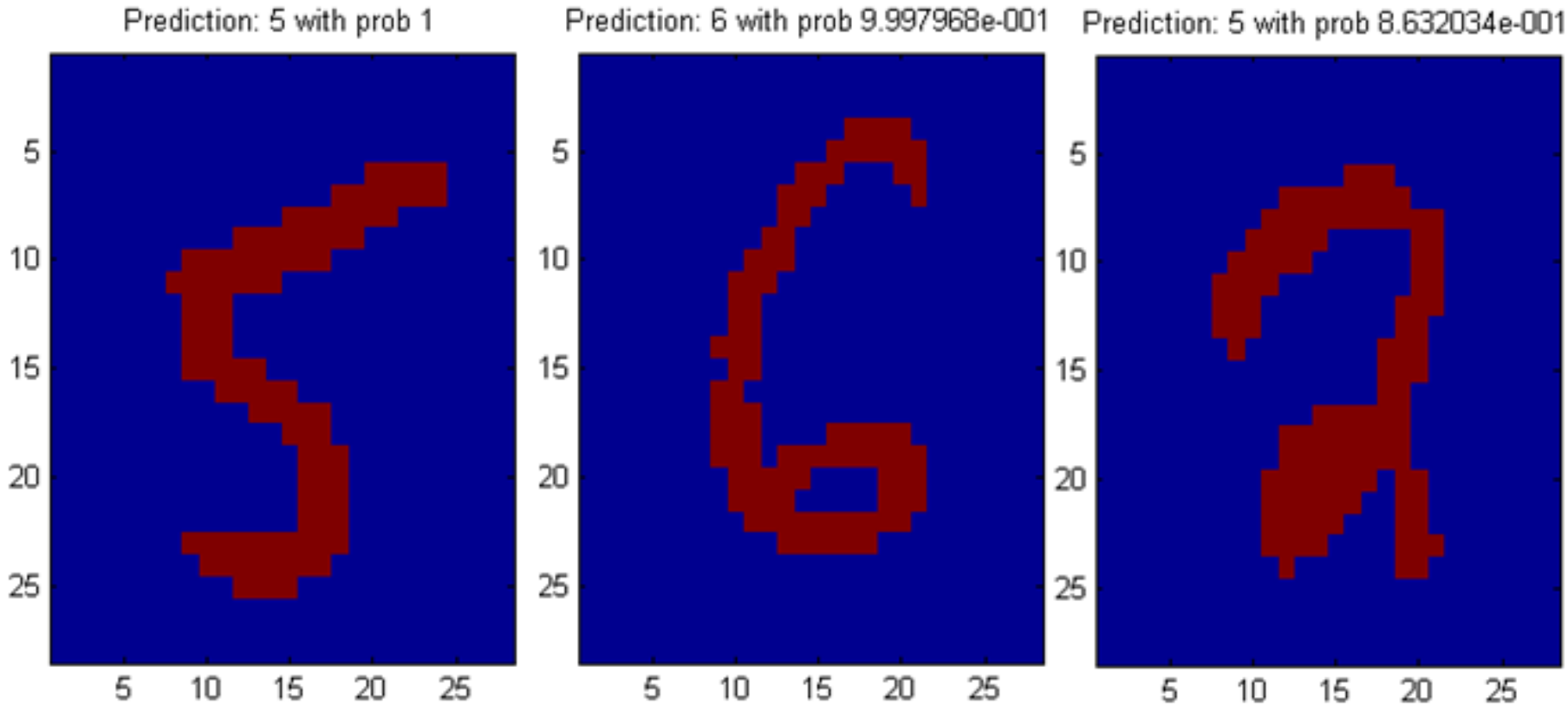
- In practice, some of these counts can be zero
- Fix this by adding “virtual” counts:

$$P(X_i = u | Y = v) = \frac{Count(X_i = u \wedge Y = v) + 1}{Count(Y = v) + 2}$$

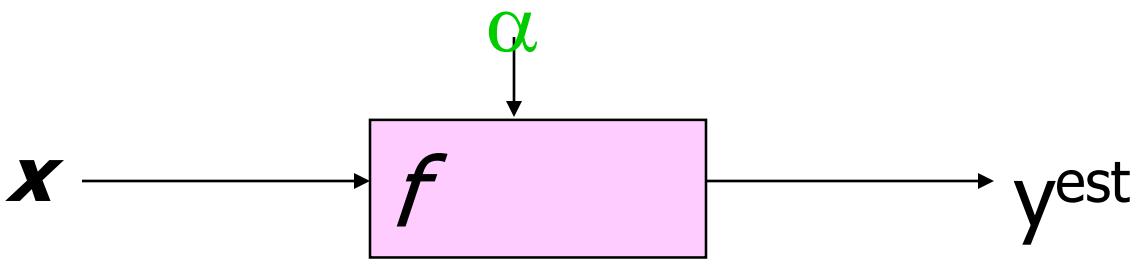
  - This is called *Smoothing*

Naïve Bayes Classifier -5

Classification

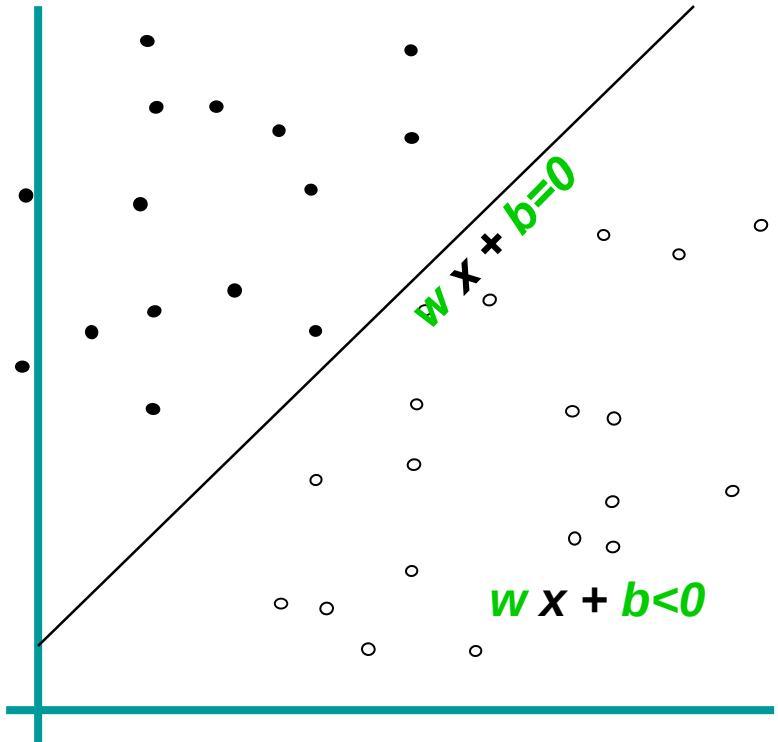


SVM Classifier -1



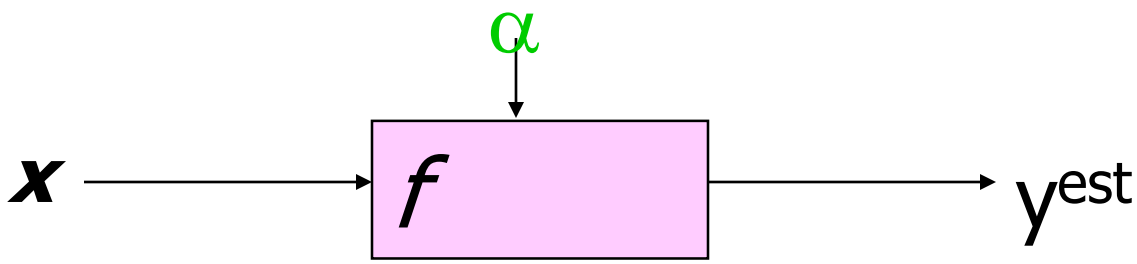
- denotes +1
- denotes -1

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



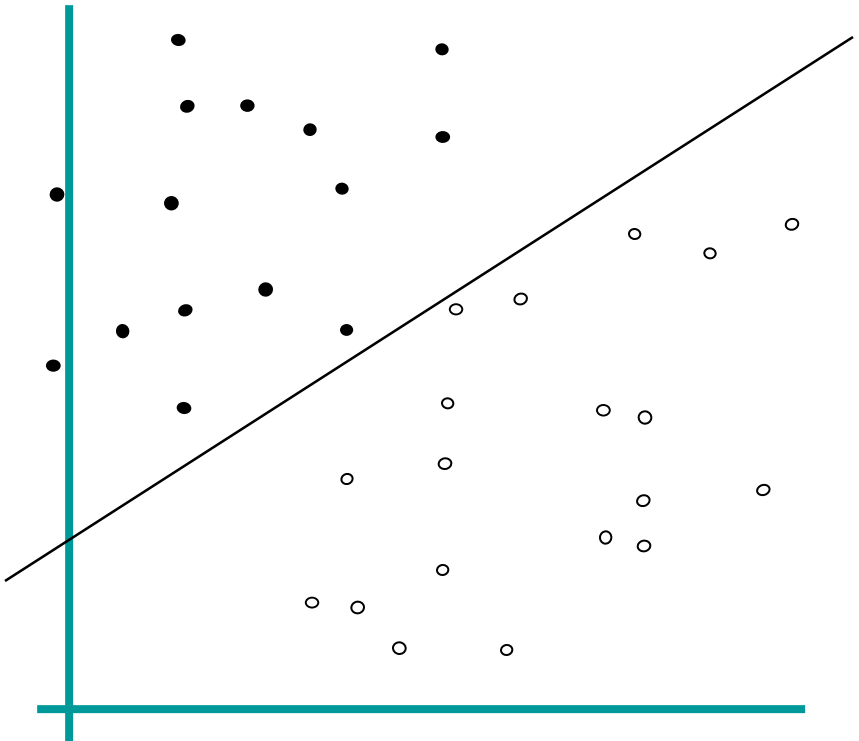
How would you classify this data?

SVM Classifier -2



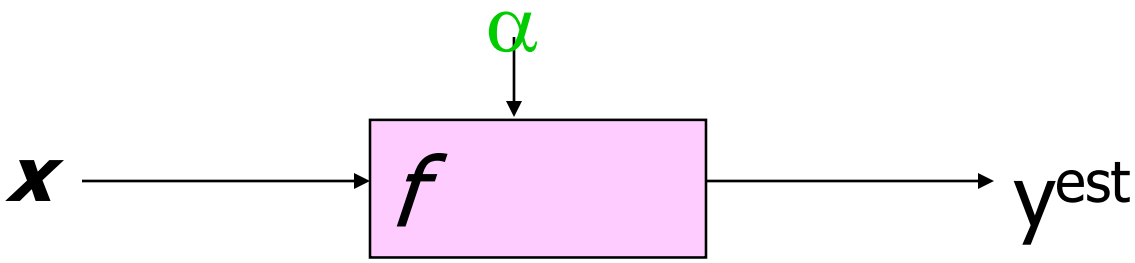
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

- denotes +1
- denotes -1



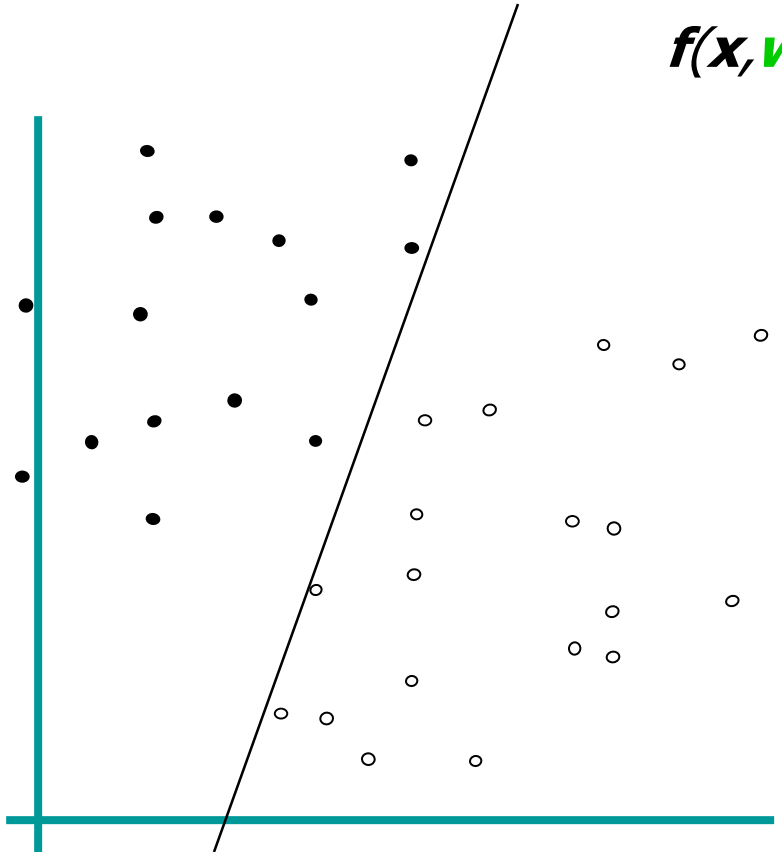
How would you classify this data?

SVM Classifier -2



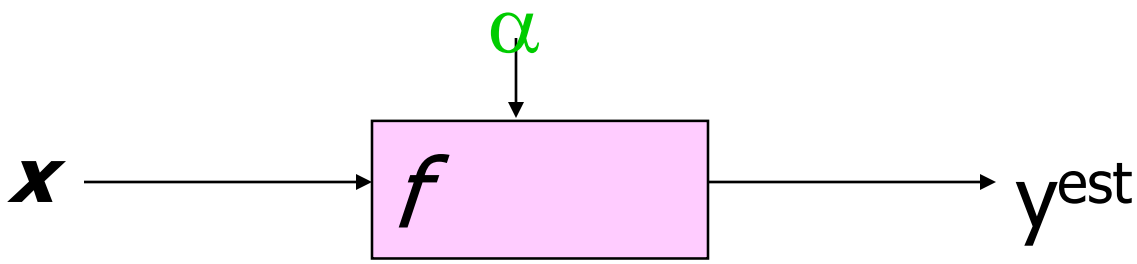
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

- denotes +1
- denotes -1



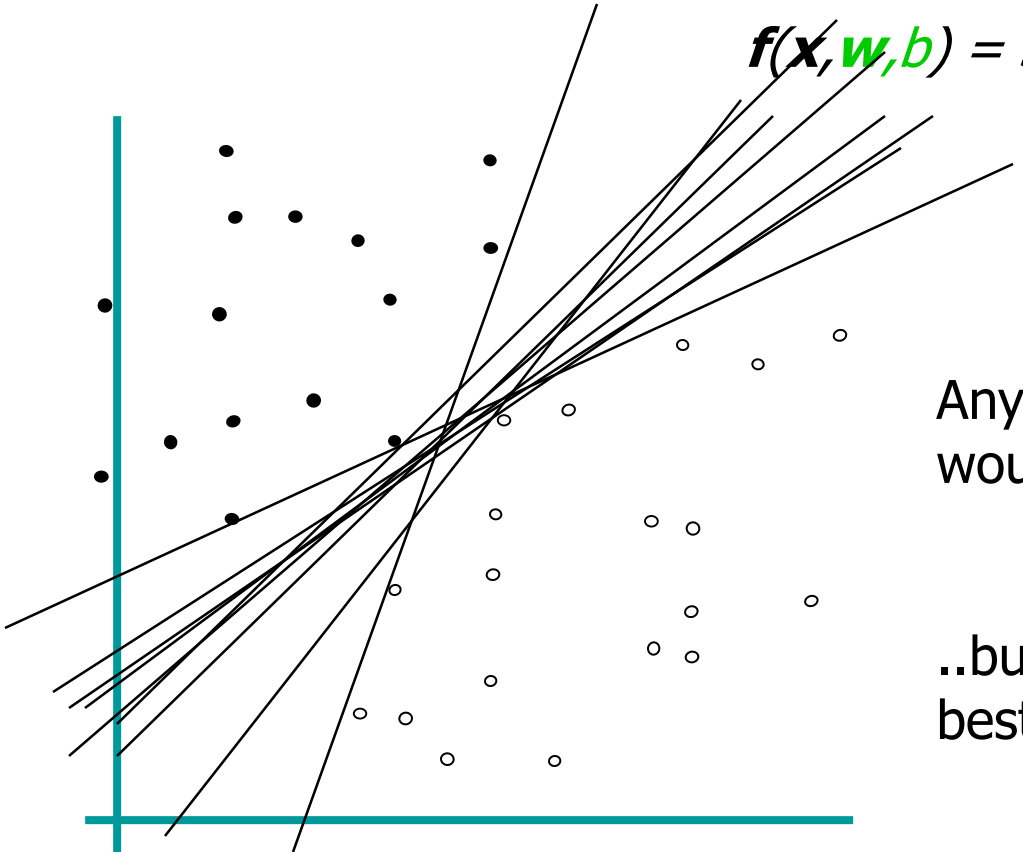
How would you classify this data?

SVM Classifier -3



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

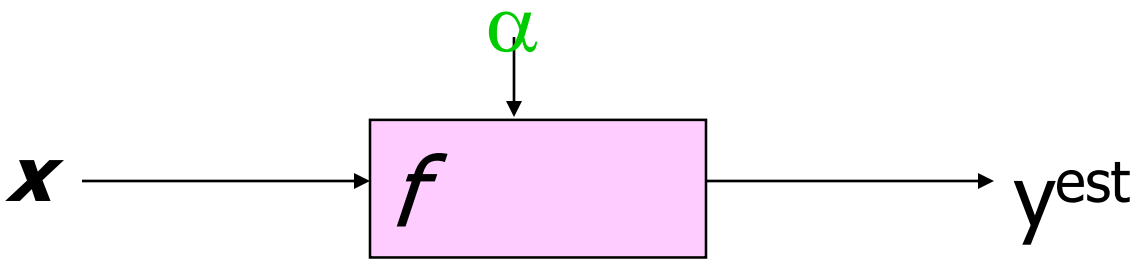
- denotes +1
- denotes -1



Any of these would be fine..

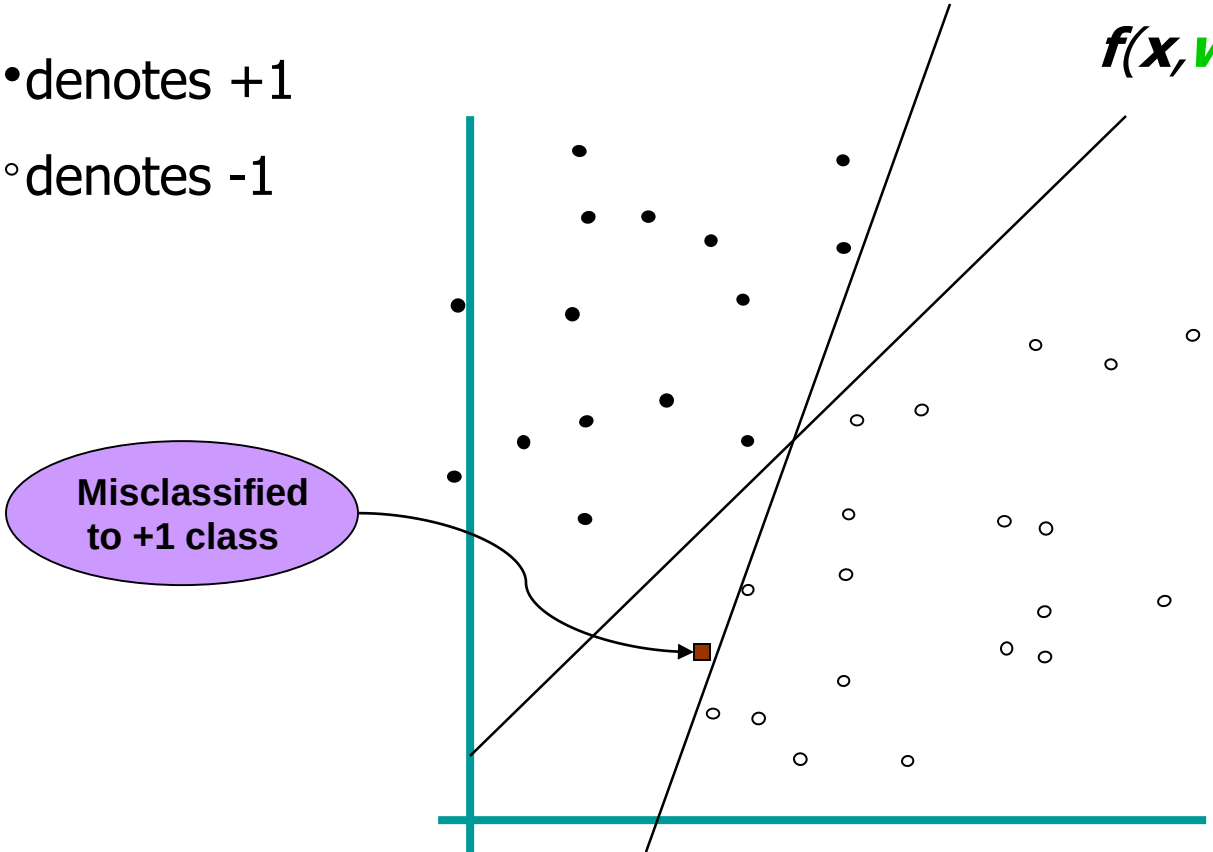
..but which is best?

SVM Classifier -3



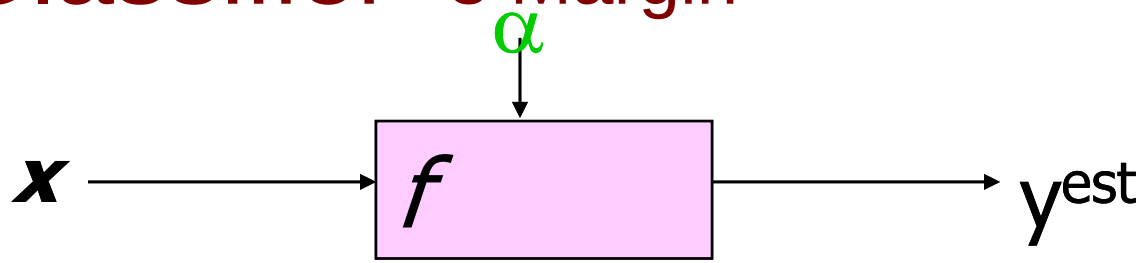
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

- denotes +1
- denotes -1



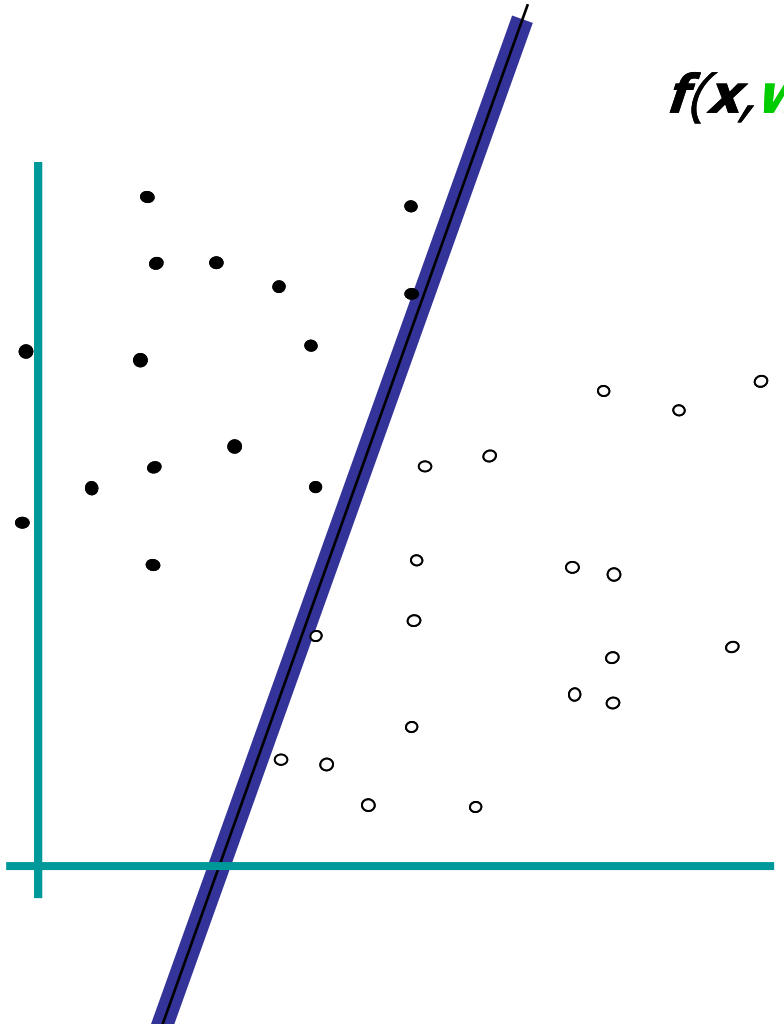
How would you classify this data?

SVM Classifier -3 Margin



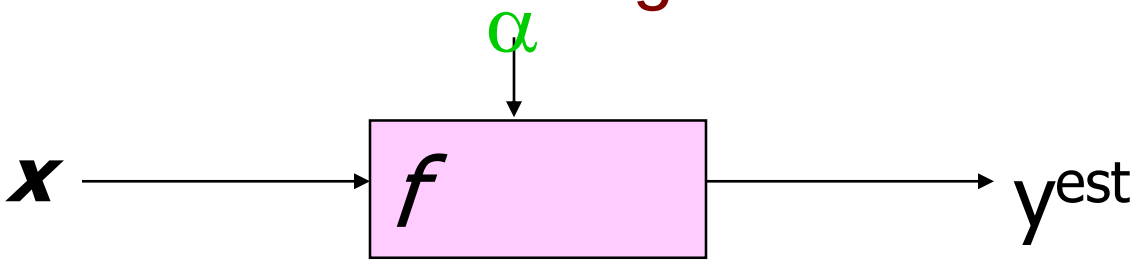
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

- 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.

SVM Classifier -3 Margin



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

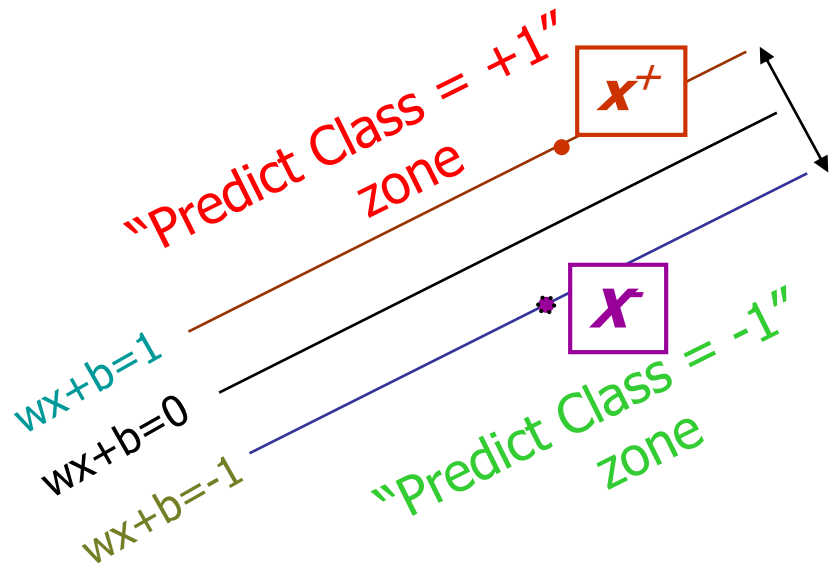
- denotes +1
- denotes -1

Support Vectors are those datapoints that the margin pushes up against

1. Maximizing the margin is good according to intuition and PAC theory
2. Implies that only support vectors are important; other training examples are ignorable.
3. Empirically it works very very well.

This is the simplest kind of SVM (Called a Linear SVM)

SVM Classifier -4 Mathematically



M=Margin Width

What we know:

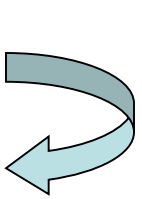
- $\boldsymbol{w} \cdot \boldsymbol{x}^+ + b = +1$
- $\boldsymbol{w} \cdot \boldsymbol{x}^- + b = -1$
- $\boldsymbol{w} \cdot (\boldsymbol{x}^+ - \boldsymbol{x}^-) = 2$

$$M = \frac{(x^+ - x^-) \cdot w}{|w|} = \frac{2}{|w|}$$

SVM Classifier -4 Mathematically

- Goal: **1) Correctly classify all training data**

$$wx_i + b \geq 1 \quad \text{if } y_i = +1$$
$$wx_i + b \leq 1 \quad \text{if } y_i = -1$$
$$y_i(wx_i + b) \geq 1 \quad \text{for all i}$$



2) Maximize the Margin

$$M = \frac{2}{|w|}$$

same as minimize

$$\frac{1}{2} w^t w$$

- **We can formulate a Quadratic Optimization Problem and solve for w and b**

- Minimize $\Phi(w) = \frac{1}{2} w^t w$

subject to $y_i(wx_i + b) \geq 1 \qquad \forall i$

SVM Classifier -5 Optimization

Find $\mathbf{w}$ and b such that
$\Phi(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w}$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i\,(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- **Need to optimize a *quadratic* function subject to *linear* constraints.**
- **Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.**
- **The solution involves constructing a *dual problem* where a *Lagrange multiplier* α_i is associated with every constraint in the primary problem:**

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \sum\alpha_i - \frac{1}{2}\sum\sum\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ is maximized and
(1) $\sum\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

SVM Classifier -5 Optimization

- The solution has the form:

$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w}^T \mathbf{x_k} \text{ for any } \mathbf{x_k} \text{ such that } \alpha_k \neq 0$$

- Each non-zero α_i indicates that corresponding $\mathbf{x_i}$ is a support vector.
- Then the classifying function will have the form:

$$f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x_i}^T \mathbf{x} + b$$

- Notice that it relies on an *inner product* between the test point $\mathbf{x}$ and the support vectors $\mathbf{x_i}$
- Also keep in mind that solving the optimization problem involved computing the inner products $\mathbf{x_i}^T \mathbf{x_j}$ between all pairs of training points.

SVM Classifier -6

- BASIC IDEA: Finding Optimal Hyperplane as decision for categorization
- Some observations:
  - Margin of separation ρ: twice the distance of nearest point to plane
  - Optimal hyperplane: one with largest margin of separation
  - Support vectors: $\underline{d}_i$'s that lies on margin
  - Note: decision surface is defined only by support vectors
- Characteristics of SVM:
  - Decision surface is defined only by data points on margin
  - Guaranteed to find global minimization

SVM Classifier -6

- Algorithm to solve linear case can be extended to solve linearly non-separable cases by introducing:
  - Soft margins hyperplanes, or
  - Mapping original data vectors to higher dimensional space
  - NOTE: Number of Support Vectors can be very large for non-linear cases -– leading to inefficiency
- Many SVM systems available publicly:
  - SVM^{LIGHT}: http://www-ai.cs.uni-ortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.htm
  - SVM^{TORCH}: http://www.idiap.ch/learning

Summary

- Information Retrieval and Classification are different:
  - We use frequent features for Classification
  - We use rare features for Information Retrieval
- Data pre-processing and feature selection are important
- There are many Classification approaches:
  - kNN simple technique that leverage on IR principles
  - Naïve Bayes (probabilistic approach that leverage on assumption about data features independency)
  - Support Vector Machines is linear or not linear (version with kernel) techniques that maximize the margin between class labels

Next Lesson

- **Source Fusion**