Web Content Mining: Classification

Outline

- Introduction to classification
  - Text classification as a special case
- Classification methods
  - kNN
  - Logistic regression classification
- Classification: model evaluation

Is this spam?

From: "" <takworlld@hotmail.com>

Subject: real estate is the only way... gem oalvgkay

Anyone can buy real estate with no money down

Stop paying rent TODAY !

There is no need to spend hundreds or even thousands for similar courses

I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.

Change your life NOW !

===

Click Below to order:

http://www.wholesaledaily.com/sales/nmd.htm

===

Text Classification Examples

Assign labels to each document or web-page:

- Labels are most often topics such as Yahoo-categories
  e.g., "*finance,*" "*sports,*" "*news>world>asia>business*"

- Labels may be genres
  e.g., "*editorials*" "*movie-reviews*" "*news*“

- Labels may be opinion
  e.g., “*like*”, “*hate*”, “*neutral*”

- Labels may be domain-specific binary
  e.g., "*interesting-to-me*" : "*not-interesting-to-me*”
  e.g., “*spam*” : “*not-spam*”
  e.g., “*contains adult language*” : “*doesn’t*”

Classification Methods (1)

- Manual classification
  - Used by Yahoo!, Looksmart, about.com, ODP, Medline
  - Very accurate when job is done by experts
  - Consistent when the problem size and team is small
  - Difficult and expensive to scale

Classification Methods (2)

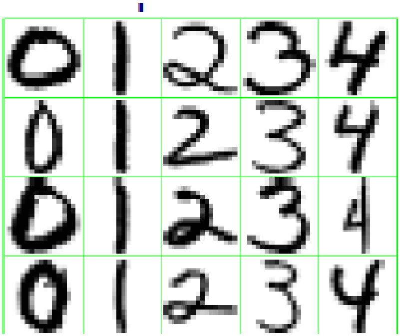
- Automatic document classification
  - Hand-coded rule-based systems
    - One technique used by CS dept's spam filter, Reuters, CIA, Verity, …
    - E.g., assign category if document contains a given boolean combination of words
    - Commercial systems have complex query languages (everything in IR query languages + accumulators)
    - Accuracy is often very high if a rule has been carefully refined over time by a subject expert
    - Building and maintaining these rules is expensive

Classification Methods (3)

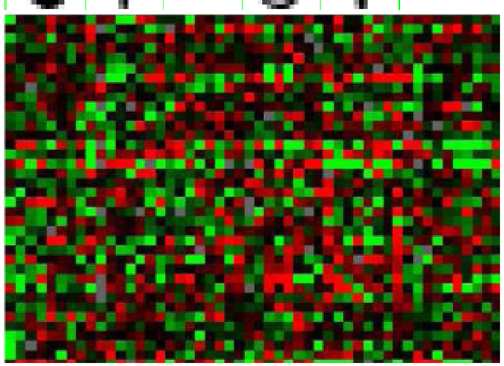
- Supervised learning of a document-label assignment function
  - Many systems partly rely on machine learning (Autonomy, MSN, Verity, Enkata, Yahoo!, …)
    - k-Nearest Neighbors (simple, powerful)
    - Naive Bayes (simple, common method)
    - Support-vector machines (new, more powerful)
    - … plus many other methods
    - No free lunch: requires hand-classified training data
    - But data can be built up (and refined) by amateurs
- Note that many commercial systems use a mixture of methods

Classification

- We are given a set of N observations $\{(\mathbf{x}_i, y_i)\}_{i=1..N}$
  - Issue: how to represent data: text, image, video…
- Need to map $x \in \mathcal{X}$ to a label $y \in \mathcal{Y}$
  - We want to know how to build classification functions ("classifiers").
- Examples:



digits recognition;
$\mathcal{Y} = \{0, \ldots, 9\}$



prediction from microarray data;
$\mathcal{Y} = \{\text{desease present/absent}\}$

Key Ingredients

Data

The data set D consists of N data points:

$D = \{x_1, x_2 \ldots, x_N\}$

Predictions（预测）

We are generally interested in predicting something based on the observed data set.

Given D what can we say about x_{N+1}?

Model

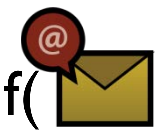
To make predictions, we need to make some assumptions. We can often express these assumptions in the form of a model, with some parameters（参数）

Given data D, we learn the model parameters , from which we can predict new data points.

Key Ingredients

$\min_f \text{Loss}(Y, f(X))$

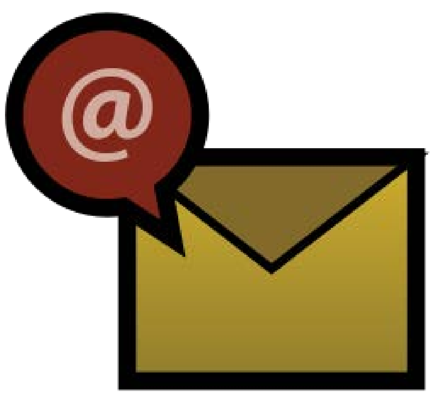
模型 → f(x)

预测： $y_{new}=f($  $)$

输入X

输出Y

数据 →



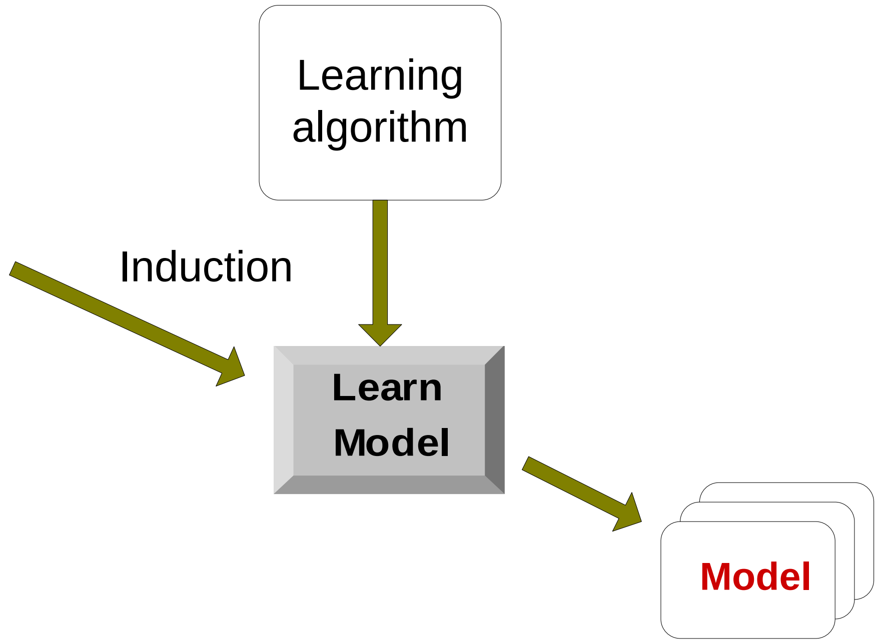
Spam or Not

General Approach for Building a Classification Model

Model Training

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

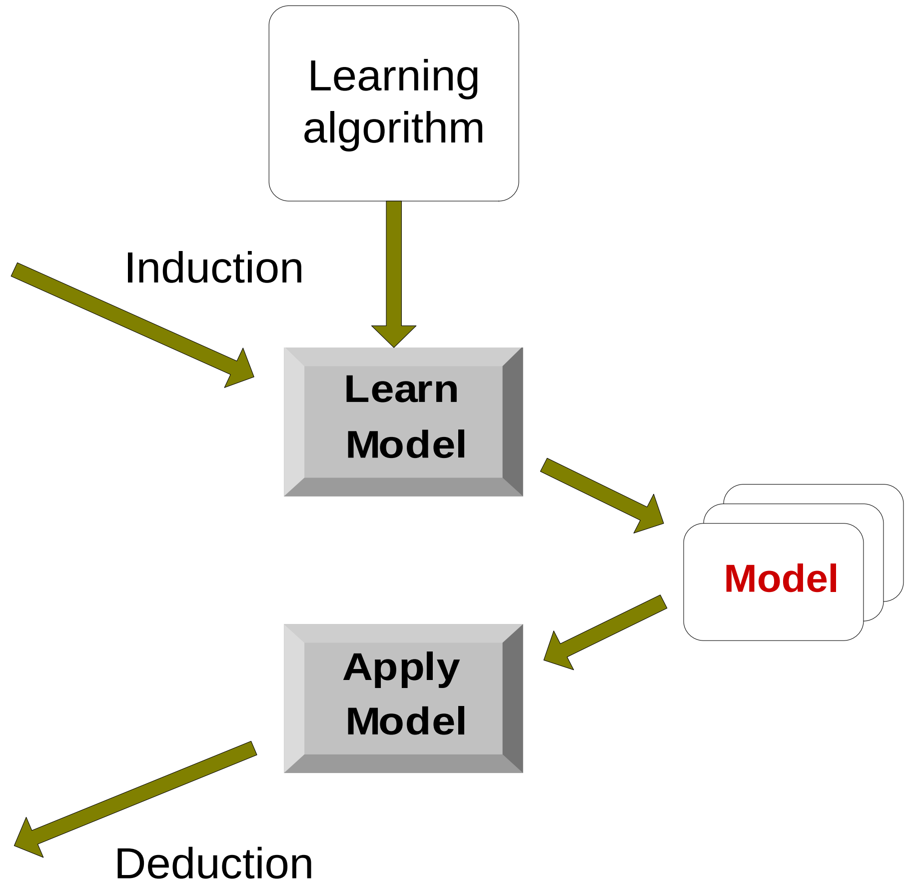


General Approach for Building a Classification Model

Model Testing

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Outline

- Introduction to classification
  - Text classification as a special case
- Classification methods
  - kNN
  - Logistic regression classification
- Classification: model evaluation

Vector Space Representation

- Each data example is a vector, one component for each attribute.

- High-dimensional vector space:
  - Attributes are axes
  - Data examples are vectors in this space

Vector Space Representation

Web document classification

Each document is a vector, one component for each term (=word).

	Doc 1	Doc 2	Doc 3	…
Word 1	3	0	0	…
Word 2	0	8	1	…
Word 3	12	1	10	…
…	0	1	3	…
…	0	0	0	…

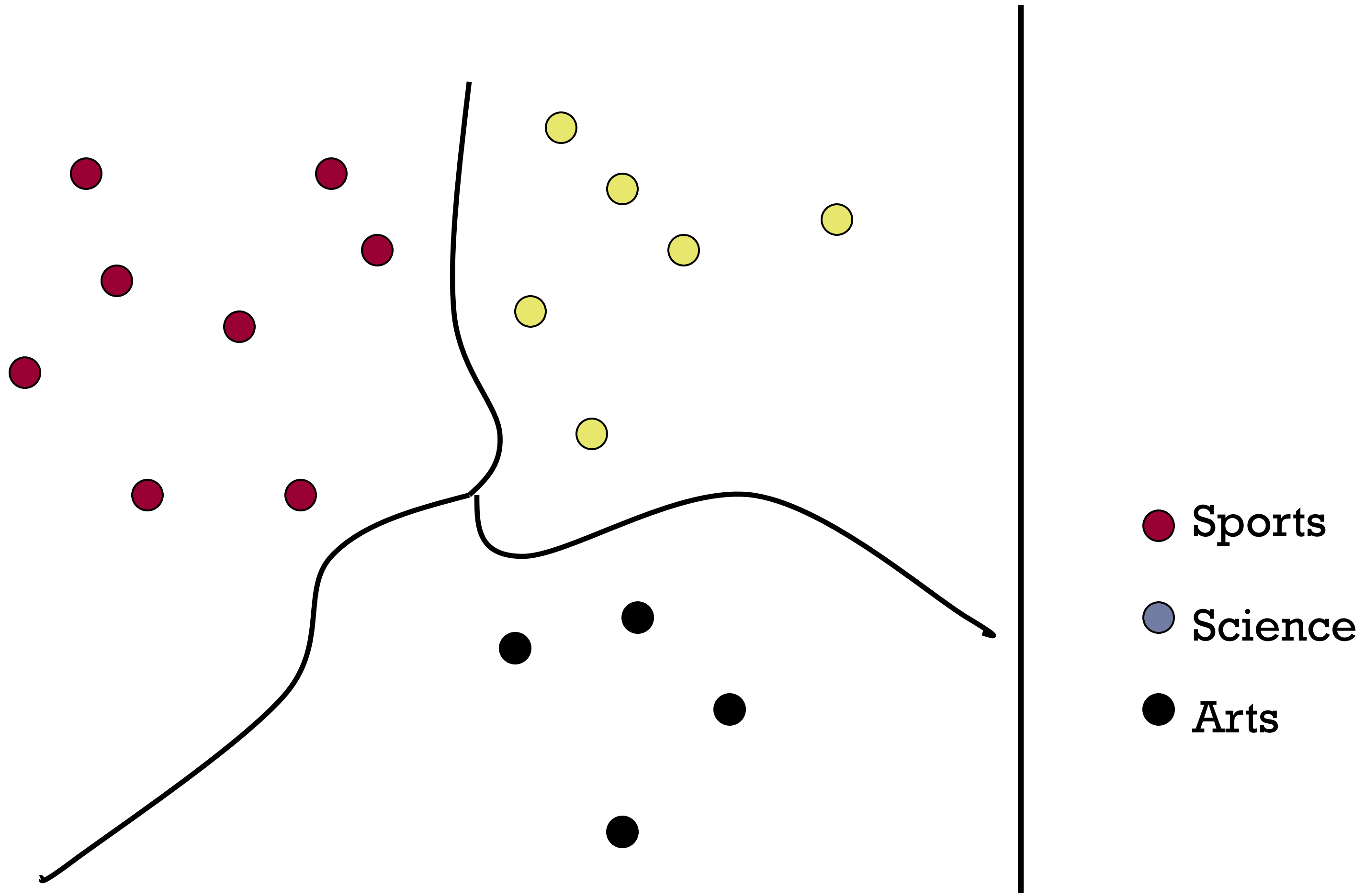
High-dimensional vector space:

- Terms are axes, 10,000+ dimensions, or even 100,000+
- Docs are vectors in this space

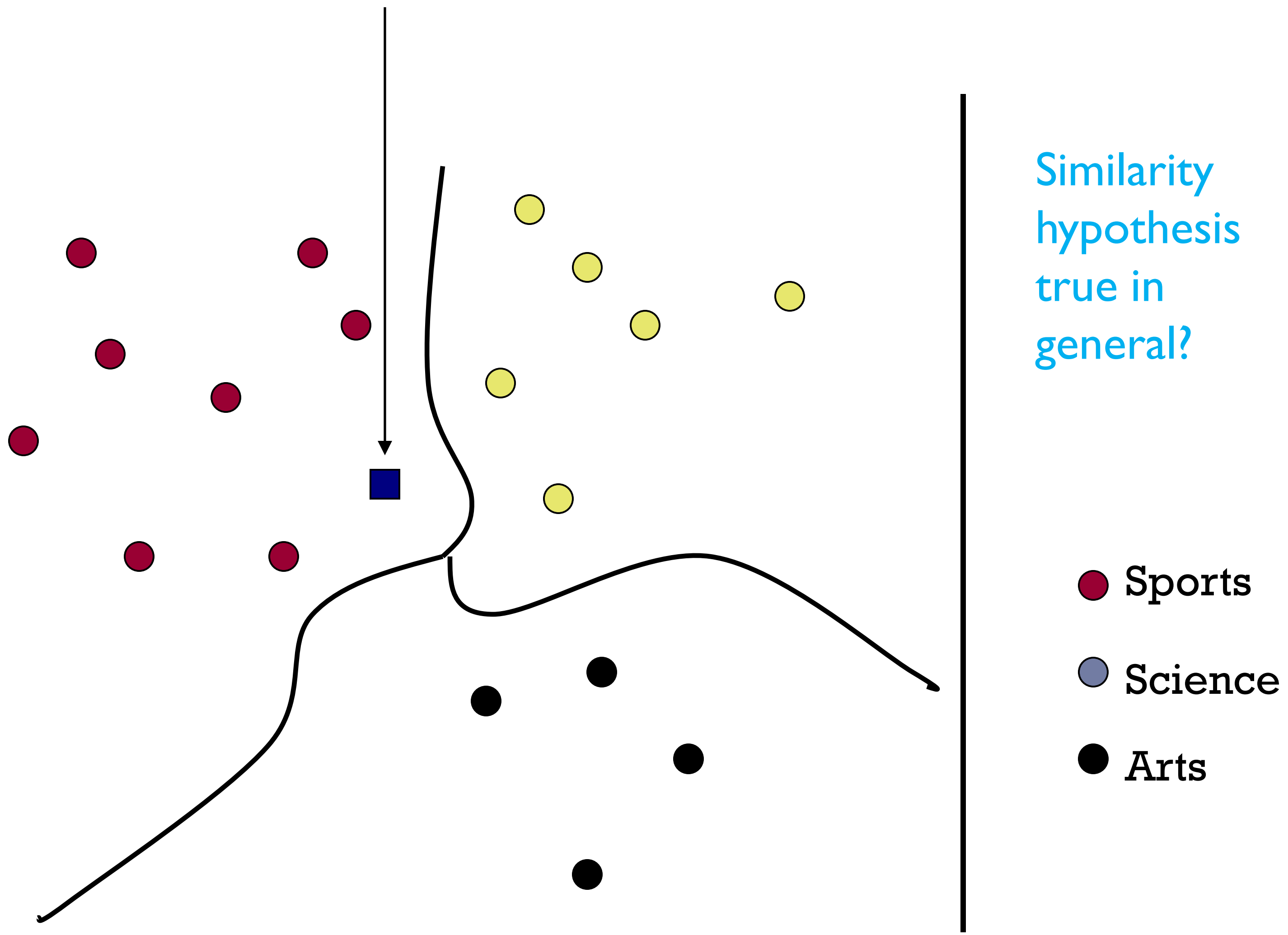
Classification Using Vector Spaces

- Each training doc a point (vector) labeled by its topic (= class)
- Hypothesis: docs of the same class form a contiguous region of space
- We define surfaces to delineate classes in space

Classes in a Vector Space

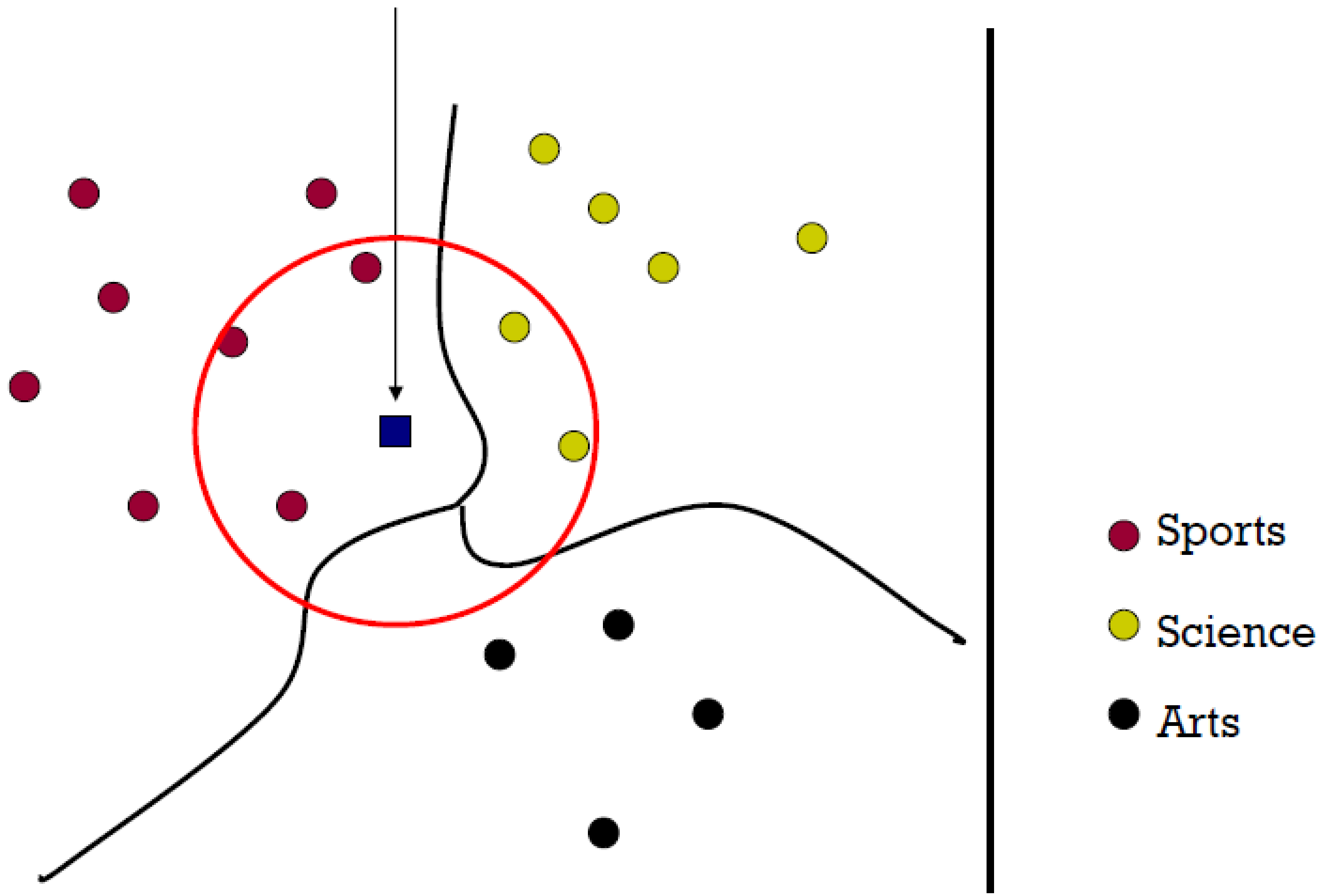


Classes in a Vector Space



Similarity hypothesis true in general?

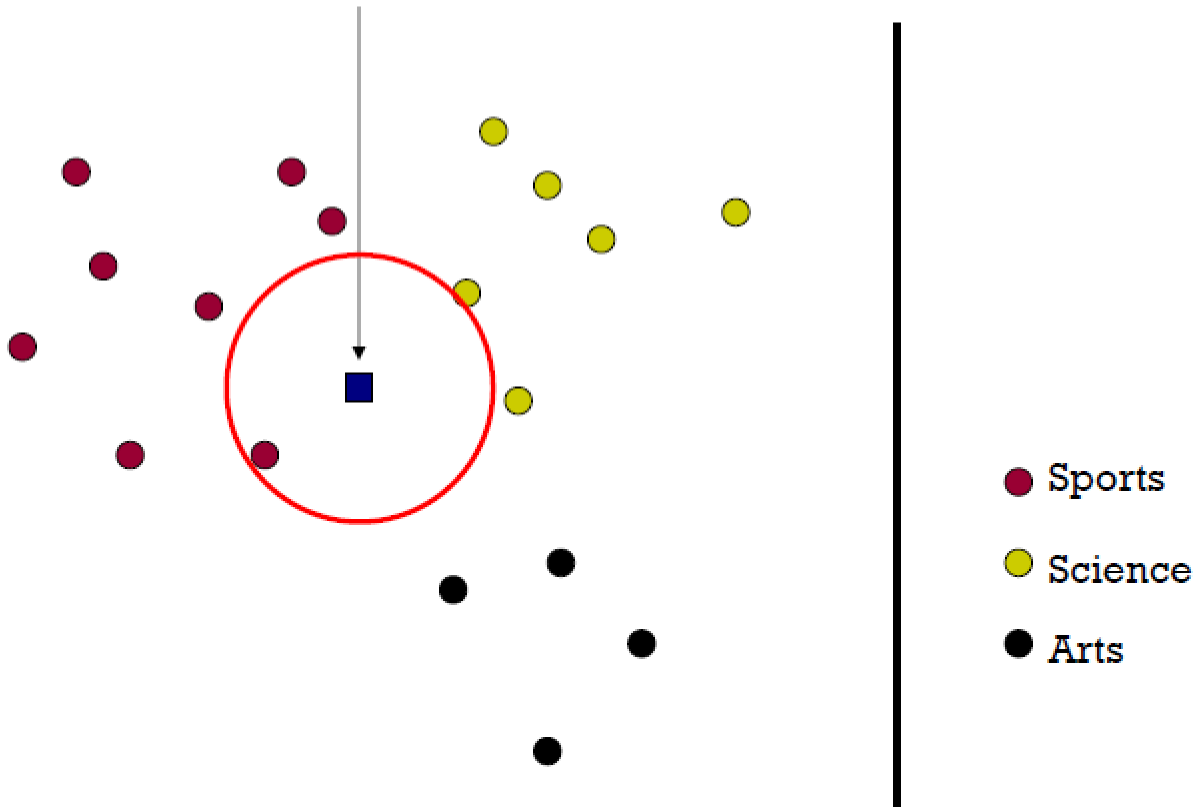
K-Nearest Neighbor (kNN) classifier



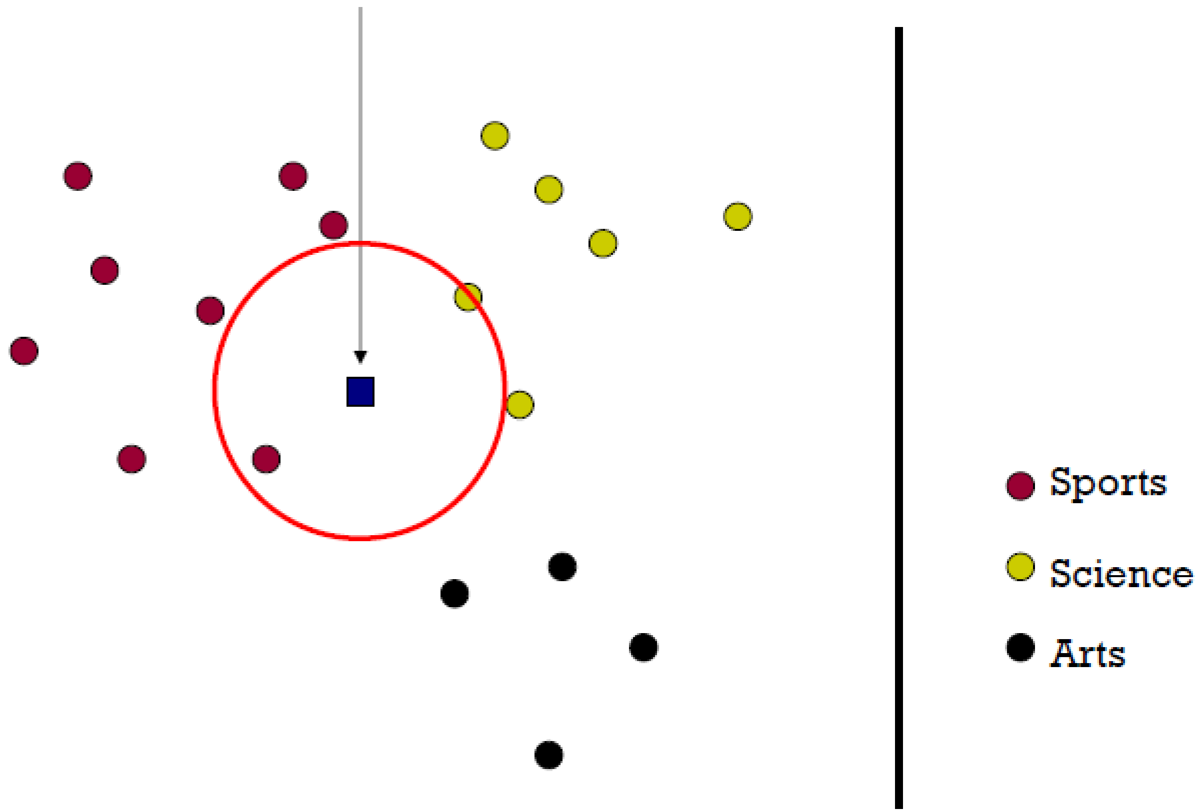
Key ingredients of kNN

- A distance metric（距离度量）
- How many nearby neighbors to look at?
- How to relate to the local points?

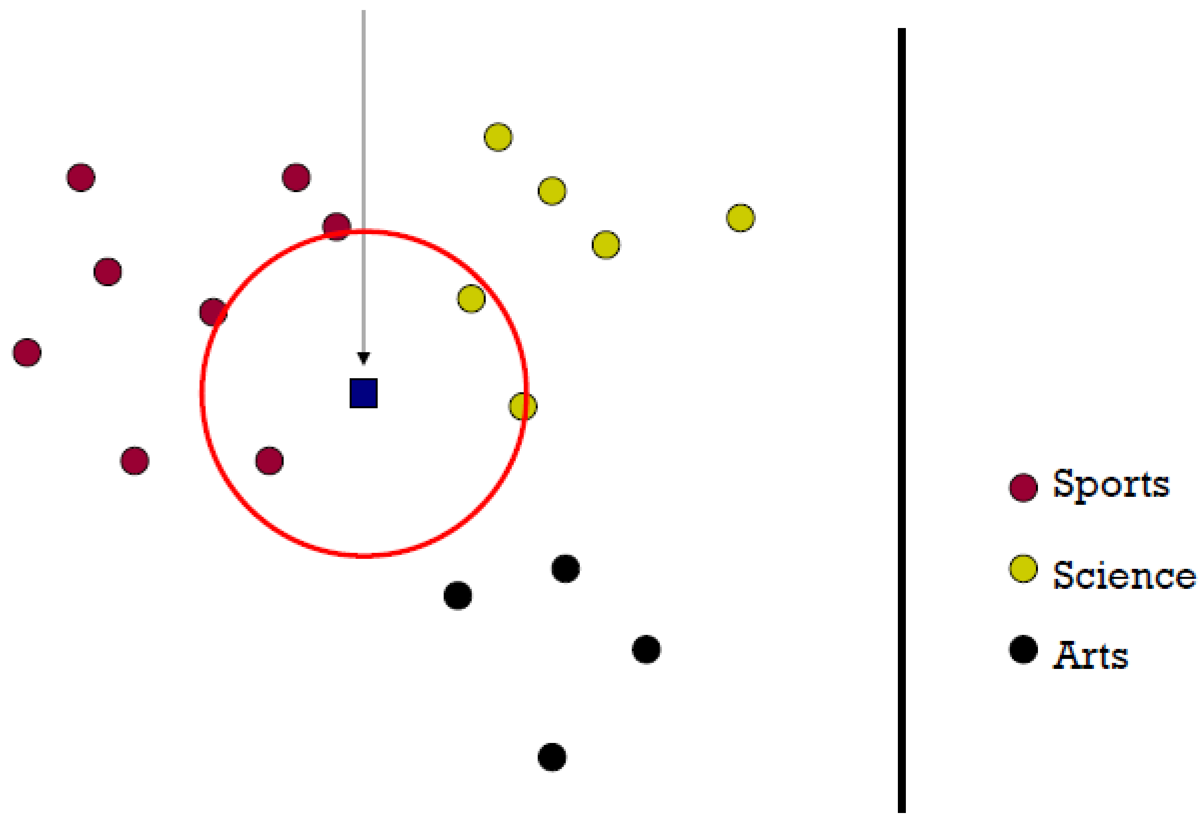
1-Nearest Neighbor (kNN) classifier



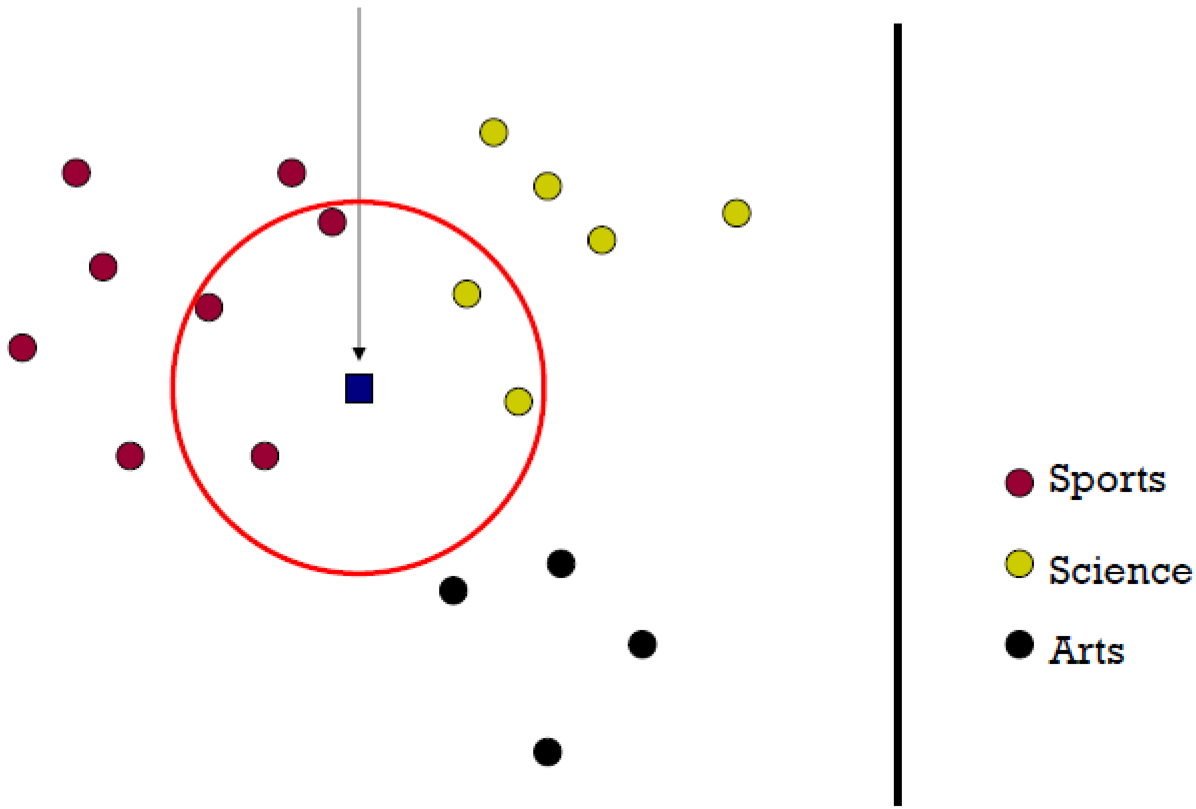
2-Nearest Neighbor (kNN) classifier



3-Nearest Neighbor (kNN) classifier



5-Nearest Neighbor (kNN) classifier



Nearest-Neighbor Learning Algorithm

Learning is just storing the representations of the training examples in *D.*

Testing instance *x:*

- Compute similarity between *x and all examples in D.*
- Assign *x the category of the majority of the k most similar examples in D.*

Also called:

- Case-based learning（基于实例的学习）
- Memory-based learning
- Lazy learning

k Nearest-Neighbor

- Using only the closest example to determine the categorization is subject to errors due to:
  - A single atypical example.
  - Noise (i.e. error) in the category label of a single training example.
- More robust alternative is to find the *k* most-similar examples and return the majority category of these *k* examples.
- Value of *k* is typically odd to avoid ties; 3 and 5 are most common.

Distance/Similarity Metrics

- Nearest neighbor method depends on a similarity (or distance) metric.
- Simplest for continuous *m*-dimensional instance space is *Euclidian distance*.
- Simplest for *m*-dimensional binary instance space is *Hamming distance* (number of feature values that differ).
- For text, cosine similarity of tf.idf weighted vectors is typically most effective.

Euclidean Distance Metric

$$D(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_i \sigma_i^2 (\mathbf{x}_i - \mathbf{x}'_i)^2}$$

or equivalently

$$D(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^\top \Sigma (\mathbf{x} - \mathbf{x}')}$$

Other metrics

- L_1 norm: $|x-x'|$
- L_∞ norm: $\max |x-x'|$ (elementwise …)
- Mahalanobis（马氏距离）: where Σ is full, and symmetric
- Angle
- Hamming distance, Manhattan distance
- …

Common Properties of a Distance Metric

- Distances, such as the Euclidean distance, have some well known properties.
  1. $d(p, q) \geq 0$ for all p and q and $d(p, q) = 0$ only if $p = q$. (Positive definiteness)
  2. $d(p, q) = d(q, p)$ for all p and q. (Symmetry)
  3. $d(p, r) \leq d(p, q) + d(q, r)$ for all points p, q, and r. (Triangle Inequality)

  where $d(p, q)$ is the distance (dissimilarity) between points (data objects), p and q.

- A distance that satisfies these properties is a metric

Case Study: kNN for Web Classification

Dataset

- 20 News Groups (20 classes)
- Download :(http://people.csail.mit.edu/jrennie/20Newsgroups/)
- 61,118 words, 18,774 documents
- Class labels descriptions

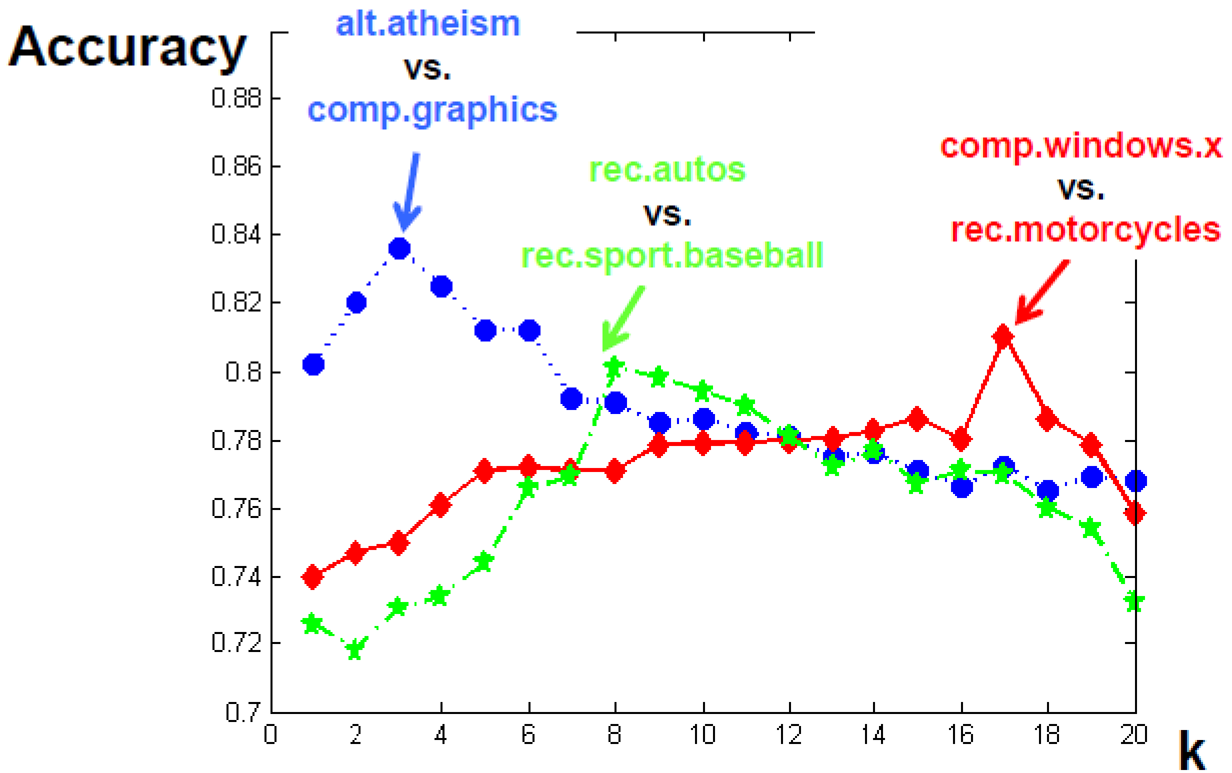
comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

Experimental Setup

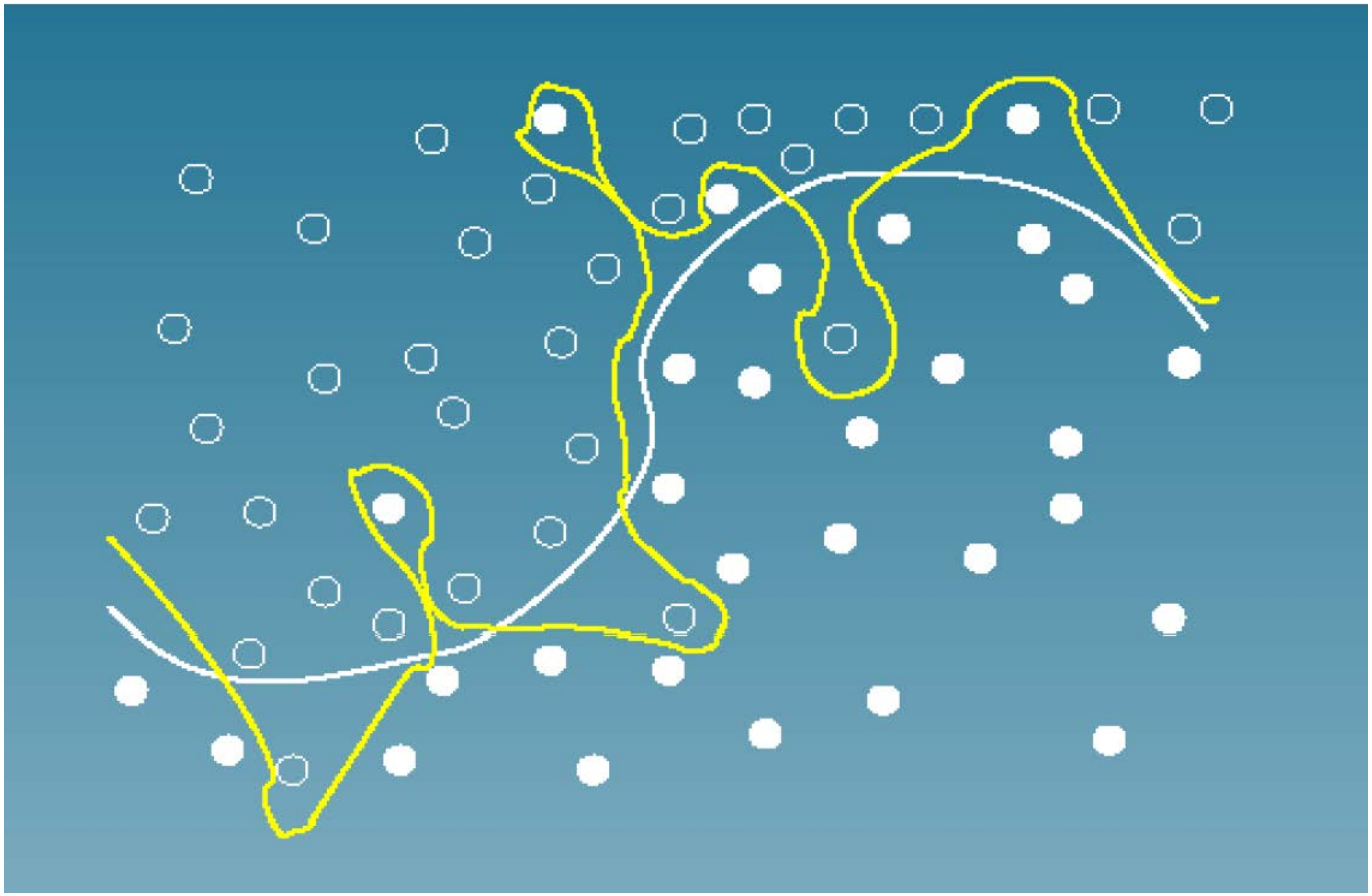
- Training/Test Sets:
  - 50%-50% randomly split.
  - 10 runs
  - report average results
- Evaluation Criteria:

$$Accuracy = \frac{\sum_{i \in \text{test set}} I(predict_i = truelabel_i)}{\text{\#of test samples}}$$

Results: Binary Classes



Is kNN ideal? …



Curse of Dimensionality in kNN

- The k-nearest neighbor approach is not immune to the curse of dimensionality

- However…
  - kNN can be able to work with high dimensionality unless a large number of attributes/features/dimensions are independent

Homework 1: kNN with Inverted Index

- Naively finding nearest neighbors requires a linear search through $|D|$ documents in collection

- But if cosine of tf.idf vectors is the similarity metric then determining *k* nearest neighbors is the same as determining the *k* best retrievals using the test document as a query to a database of training documents.

- Use standard vector space inverted index methods to find the *k* nearest neighbors.

Comments on kNN

Instance-based learning: kNN – a Nonparametric（无参数的）classifier

A nonparametric method does not rely on any assumption concerning the structure of the underlying density function.

Very little "learning" is involved in these methods

Sample size

- The more the better
- Need efficient search algorithm for NN

Good news:

Simple and powerful methods; Flexible and easy to apply to many problems.

Bad news:

High memory requirements

Very dependant on the scale factor for a specific problem.

Outline

- Introduction to classification
  - Text classification as a special case
- Classification methods
  - kNN
  - Logistic regression classification
- Classification: model evaluation

Logistic Regression Classification

- Consider binary classification:
  - $y = 0, 1$
  - Each example represented by a feature vector $\mathbf{x}$
- Intuition: map $\mathbf{x}$ to a real number → $\mathbf{w}^\top \mathbf{x}$
  - Very positive $\mathbf{w}^\top \mathbf{x}$ means $\mathbf{x}$ is likely in the positive class $(y = 1)$
  - Very negative $\mathbf{w}^\top \mathbf{x}$ means $\mathbf{x}$ is likely in the negative class $(y = 0)$
- Probability interpretation: $\mathbf{w}^\top \mathbf{x} \to p(y|\mathbf{x})$
- Squash the range of $\mathbf{w}^\top \mathbf{x} \in (-\infty, +\infty)$ down to $[0, 1]$

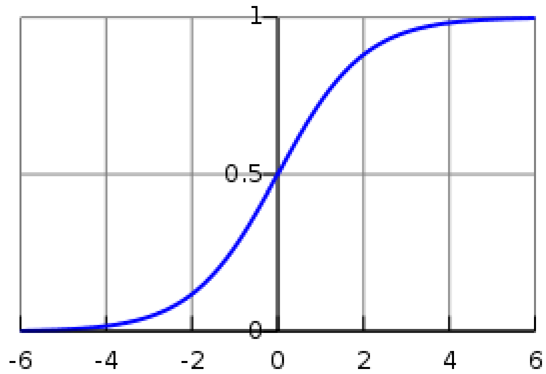
Logistic Regression Classification

- Probability interpretation: $\mathbf{w}^\top \mathbf{x} \to p(y|\mathbf{x})$

Conditional Probability: relevant in classification

$\sigma(z) = \frac{1}{1+e^{-z}}$ Logistic function / sigmoid function

$z \to +\infty, \sigma(z) \to 1; z \to -\infty, \sigma(z) \to 0$



$$p(y=1|\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x}) = \frac{1}{1+exp(-\mathbf{w}^\top \mathbf{x})} = \frac{exp(\mathbf{w}^\top \mathbf{x})}{1+exp(\mathbf{w}^\top \mathbf{x})}$$

$$p(y=0|\mathbf{x}) = 1 - p(y=1|\mathbf{x}) = \frac{1}{1+exp(\mathbf{w}^\top \mathbf{x})}$$

Logistic Regression: Log Odds

- 一个事件的几率(odds)：
  - 该事件发生的概率与不发生的概率的比值，p/(1-p)
  - log odds / logit function: log[p/(1-p)]

- Log odds for logistic regression:

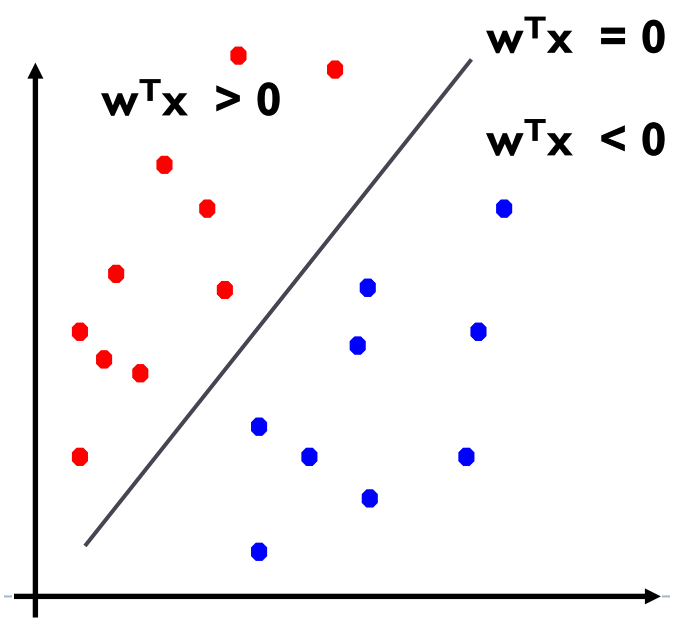
$$\log \frac{p(y=1|\mathbf{x})}{1-p(y=1|\mathbf{x})} = \mathbf{w}^\top \mathbf{x}$$

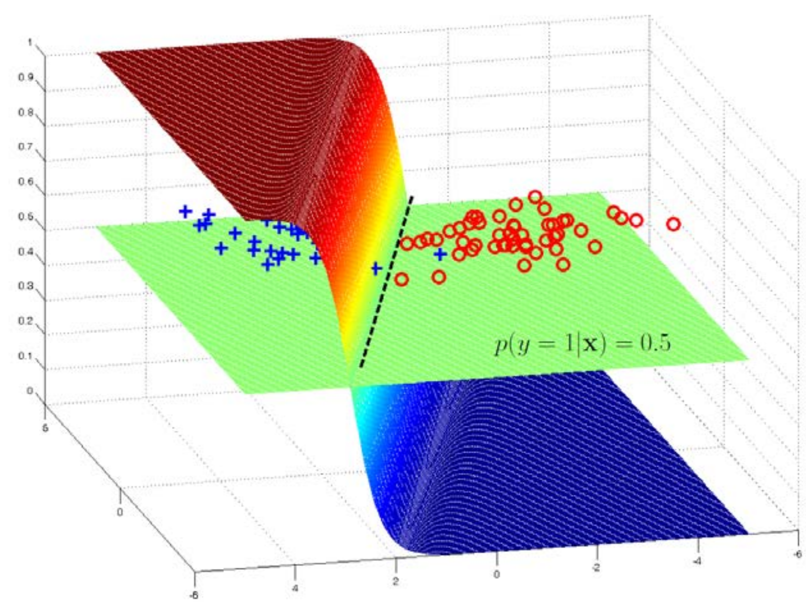
Logistic Regression: Decision Boundary

If $p(y=1|\mathbf{x}) \geq 0.5$, predict $y = 1$

If $p(y=1|\mathbf{x}) < 0.5$, predict $y = 0$

- Decision boundary: $p(y=1|\mathbf{x}) = 0.5 \Leftrightarrow \mathbf{w}^\top \mathbf{x} = 0$
- linear logistic model → a linear decision boundary





Likelihood under the Logistic Model

- Logistic regression: observe labels, measure their probability under the model

$$
\begin{aligned}
p(y_i|\mathbf{x}_i;\mathbf{w}) &= \begin{cases} \sigma(\mathbf{w}^\top\mathbf{x}_i) & \text{if} \quad y_i = 1, \\ 1-\sigma(\mathbf{w}^\top\mathbf{x}_i) & \text{if} \quad y_i = 0 \end{cases} \\
&= \sigma(\mathbf{w}^\top\mathbf{x}_i)^{y_i}(1-\sigma(\mathbf{w}^\top\mathbf{x}_i))^{1-y_i}
\end{aligned}
$$

- The conditional log-likelihood of w:

$$
\begin{aligned}
\ell(\mathbf{w}) &= \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i;\mathbf{w}) \\
&= \sum_{i=1}^{N} y_i \log \sigma(\mathbf{w}^\top\mathbf{x}_i) + (1-y_i)\log\left(1-\sigma(\mathbf{w}^\top\mathbf{x}_i)\right)
\end{aligned}
$$

Training the Logistic Model

- Training (i.e., finding the parameter w) can be done by maximizing the conditional log likelihood of training data

$\{(\mathbf{x}_i, y_i)\}_{i=1:N}$

$$\max_{\mathbf{w}} \ell(\mathbf{w}) = \max_{\mathbf{w}} \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i; \mathbf{w})$$

or

$$\begin{aligned} \min_{\mathbf{w}} J(\mathbf{w}) &= \min_{\mathbf{w}} -\ell(\mathbf{w}) \\ &= \min_{\mathbf{w}} -\left[\sum_{i=1}^{N} y_i \log \sigma(\mathbf{w}^\top \mathbf{x}_i) + (1-y_i) \log\left(1-\sigma(\mathbf{w}^\top \mathbf{x}_i)\right)\right] \end{aligned}$$

Gradient Descent

- Want $\min_{\mathbf{w}} J(\mathbf{w})$

Repeat {

$$\mathrm{w}_j := \mathrm{w}_j - \alpha \frac{\partial}{\partial \mathrm{w}_j} J(\mathbf{w})$$

} (simultaneously update all w_j)

Homework 2: Derivative of the Logistic

- A useful fact

$$\frac{\partial}{\partial z}\sigma(z) = \frac{\partial}{\partial z}\frac{1}{1+e^{-z}} = \underbrace{-\left(\frac{1}{1+e^{-z}}\right)^2}_{\partial\sigma/\partial(1+e^{-z})} \times \underbrace{-e^{-z}}_{\partial(1+e^{-z})/\partial z}$$

$$= \sigma^2(z)\left(\frac{1-\sigma(z)}{\sigma(z)}\right) = \sigma(z)(1-\sigma(z)).$$

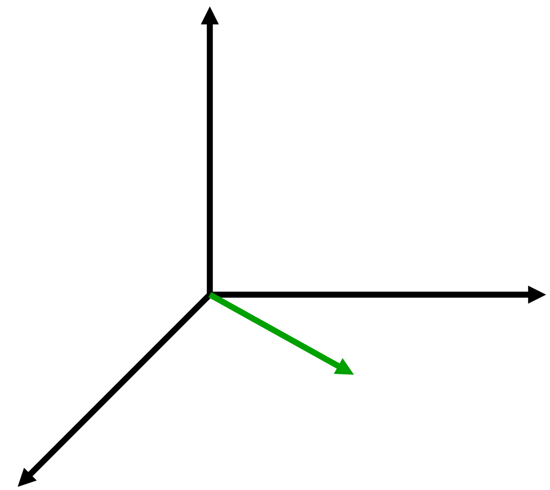
- Compute $\frac{\partial}{\partial \mathrm{w}_j} J(\mathbf{w})$

Comments on Logistic Regression

- Parametric learning model
- Linear classification
- Discriminative model: estimate conditional likelihood $p(y|x)$ directly

High Dimensional Data

- Pictures like the one at right are absolutely misleading!
- Documents are zero along almost all axes
- Most document pairs are very far apart (i.e., not strictly orthogonal, but only share very common words and a few scattered others)
- In classification terms: virtually all document sets are separable, for most any classification
- This is part of why linear classifiers are quite successful in this domain



Aside: Author identification

- Federalist papers
  - 77 short essays written in 1787-1788 by Hamilton, Jay and Madison to persuade NY to ratify the US Constitution; published under a pseudonym
  - The authorships of 12 papers was in dispute
  - In 1964 Mosteller and Wallace* solved the problem
  - They identified 70 *function* words as good candidates for authorship analysis
  - Using statistical inference they concluded the author was Madison

*Mosteller, Frederick and Wallace, David L. 1964. Inference and Disputed Authorship: The Federalist.

Function words for Author Identification

1	*a*	15	*do*	29	*is*	43	*or*	57	*this*
2	*all*	16	*down*	30	*it*	44	*our*	58	*to*
3	*also*	17	*even*	31	*its*	45	*shall*	59	*up*
4	*an*	18	*every*	32	*may*	46	*should*	60	*upon*
5	*and*	19	*for*	33	*more*	47	*so*	61	*was*
6	*any*	20	*from*	34	*must*	48	*some*	62	*were*
7	*are*	21	*had*	35	*my*	49	*such*	63	*what*
8	*as*	22	*has*	36	*no*	50	*than*	64	*when*
9	*at*	23	*have*	37	*not*	51	*that*	65	*which*
10	*be*	24	*her*	38	*now*	52	*the*	66	*who*
11	*been*	25	*his*	39	*of*	53	*their*	67	*will*
12	*but*	26	*if*	40	*on*	54	*then*	68	*with*
13	*by*	27	*in*	41	*one*	55	*there*	69	*would*
14	*can*	28	*into*	42	*only*	56	*things*	70	*your*

Table 1: Function Words and Their Code Numbers

Function words for Author Identification

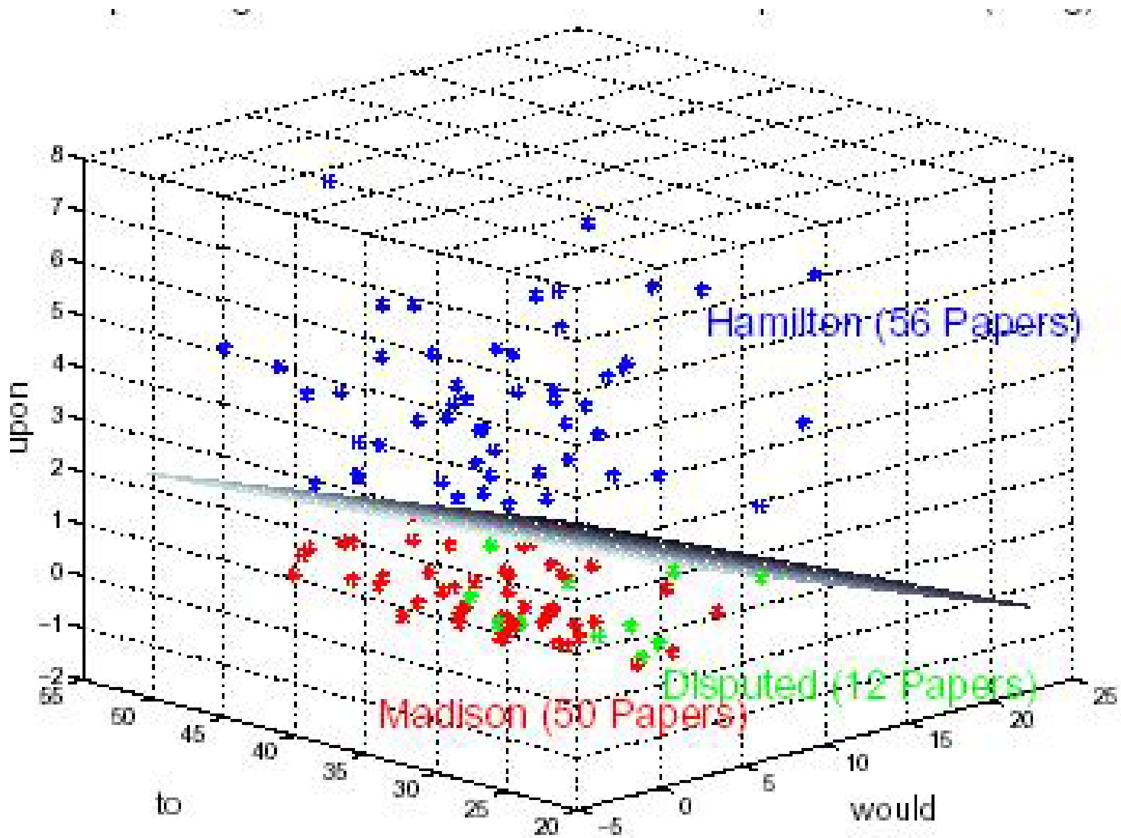


Figure 1: Obtained Hyperplane in 3 dimensions

Outline

- Introduction to classification
  - Text classification as a special case

- Classification methods
  - kNN
  - Logistic regression classification

- Classification: model evaluation

Classification Errors

- Training errors (apparent errors) — 训练误差
  - Errors committed on the training set
- Test errors — 测试误差
  - Errors committed on the test set

$$Accuracy = \frac{\sum_{i \in \text{test set}} I(predict_i = truelabel_i)}{\#\text{of test samples}}$$

- Generalization errors — 泛化误差
  - Expected error of a model over random selection of records from same distribution（未知记录上的期望误差）

Using Validation Set（确认集）

- Divide <u>training</u> data into two parts:
  - Training set:
    - use for model building
  - Validation set:
    - use for estimating generalization error
    - Note: validation set is not the same as test set
- Drawback:
  - Less data available for training

Summary

- Classification in vector space
  - k nearest neighbor
  - Logistic regression classification

- Model evaluation
  - Training errors
  - Test errors
  - Generalization errors