Text Categorization using Naïve Bayes

Mausam

(based on slides of Dan Weld, Prabhakar Raghavan, Hinrich Schutze, Guillaume Obozinski, David D. Lewis)

Categorization

- Given:
  - A **description of an instance**, $x \in X$, where X is the *instance language* or *instance space*.
  - A **fixed set of categories**: $C = \{c_1, c_2, \ldots c_n\}$
- Determine:
  - The **category of *x***: $c(x) \in C$, where $c(x)$ is a categorization function whose domain is X and whose range is C.

Sample Category Learning Problem
County *vs.* Country?

article | discussion | edit this page | history

• Ten things you didn't know about images on Wikipedia •

King County, Washington

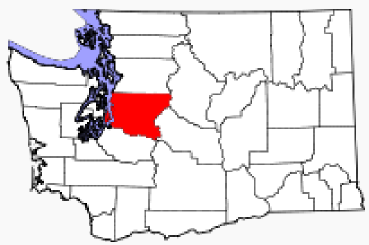
From Wikipedia, the free encyclopedia

Coordinates: 47.47, -121.84

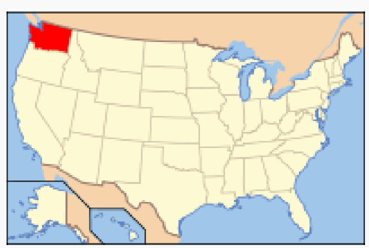
"King County" redirects here. For other uses, see King County (disambiguation).

King County, Washington

Map



Location in the state of Washington



Washington's location in the USA

Statistics	
Founded	December 22, 1852
Seat	Seattle

King County is located in the U.S. state of Washington. The population in the 2000 census was 1,737,034 and in 2006 was an estimated 1,835,300. By population, King is the largest county in Washington, and the 12th largest in the United States. As of 2006, the county had a population comparable to that of the state of Nebraska.

The county seat is Seattle, which is the state's largest city. About two-thirds of the county's population lives in the city's suburbs. King County ranks among the 100 highest-income counties in the United States.

Contents [show]

History [edit]

The county was formed out of territory within Thurston County on December 22, 1852, by the Oregon Territory legislature, and was named after Alabama resident William Rufus King, vice president under president Franklin Pierce. Seattle was made the county seat on January 11, 1853.[1][2]

King County originally extended to the Olympic Peninsula. According to historian Bill Speidel,

article | discussion | edit this page | history

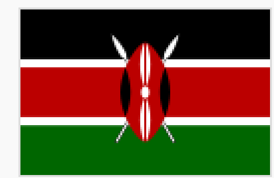
• Ten things you didn't know about W

Kenya

From Wikipedia, the free encyclopedia

This article needs additional references or sources for ve
Please help improve this article by adding reliable references. Unverifiable m
be challenged and removed.

Jamhuri ya
Republic o



Flag

Motto
"Harambee"
"Let us all pull

Anthe
Ee Mungu Ng
"Oh God of All

The **Republic of Kenya** is a country in Eastern Africa. It is bordered by Ethiopia to the north, Somalia to the northeast, Tanzania to the south, Uganda to the west, and Sudan to the northwest, with the Indian Ocean running along the southeast border.

Contents [show]

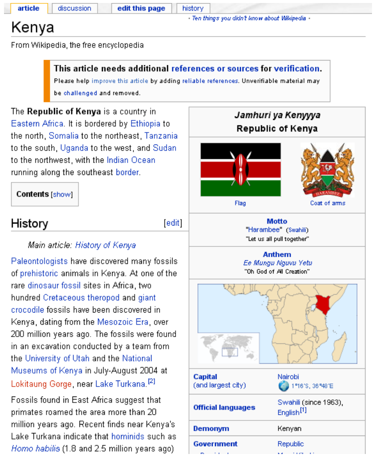
History [edit]

Main article: History of Kenya

Paleontologists have discovered many fossils of prehistoric animals in Kenya. At one of the rare dinosaur fossil sites in Africa, two hundred Cretaceous theropod and giant crocodile fossils have been discovered in Kenya, dating from the Mesozoic Era, over 200 million years ago. The fossils were found in an excavation conducted by a team from the University of Utah and the National Museums of Kenya in July-August 2004 at

Example: County *vs.* Country?

- Given:
  - A description of an instance, $x \in X$, where X is the *instance language* or *instance space*.
  - A fixed set of categories: $C = \{c_1, c_2, \ldots c_n\}$
- Determine:
  - The category of x: $c(x) \in C$, where $c(x)$ is a categorization function whose domain is X and whose range is C.



Text Categorization

- Assigning documents to a fixed set of categories, *e.g.*
- Web pages
  - Yahoo-like classification
- What else?
- Email messages
  - Spam filtering
  - Prioritizing
  - Folderizing
- News articles
  - Personalized newspaper
- Web Ranking
  - Is page related to selling something?

Procedural Classification

- Approach:
  - Write a procedure to determine a document's class
  - E.g., Spam?

Learning for Text Categorization

- Hard to construct text categorization functions.
- Learning Algorithms:
  - **Bayesian (naïve)**
  - Neural network
  - Relevance Feedback (Rocchio)
  - Rule based (C4.5, Ripper, Slipper)
  - Nearest Neighbor (case based)
  - Support Vector Machines (SVM)

Learning for Categorization

- A ***training example*** is an instance $x \in X$, paired with its correct category $c(x)$: $<x, c(x)>$ for an unknown categorization function, c.
- Given a set of training examples, D.

{< 

, county>, < 

, country>,…

- Find a hypothesized categorization function, $h(x)$, such that: $\forall < x, c(x) > \in D : h(x) = c(x)$

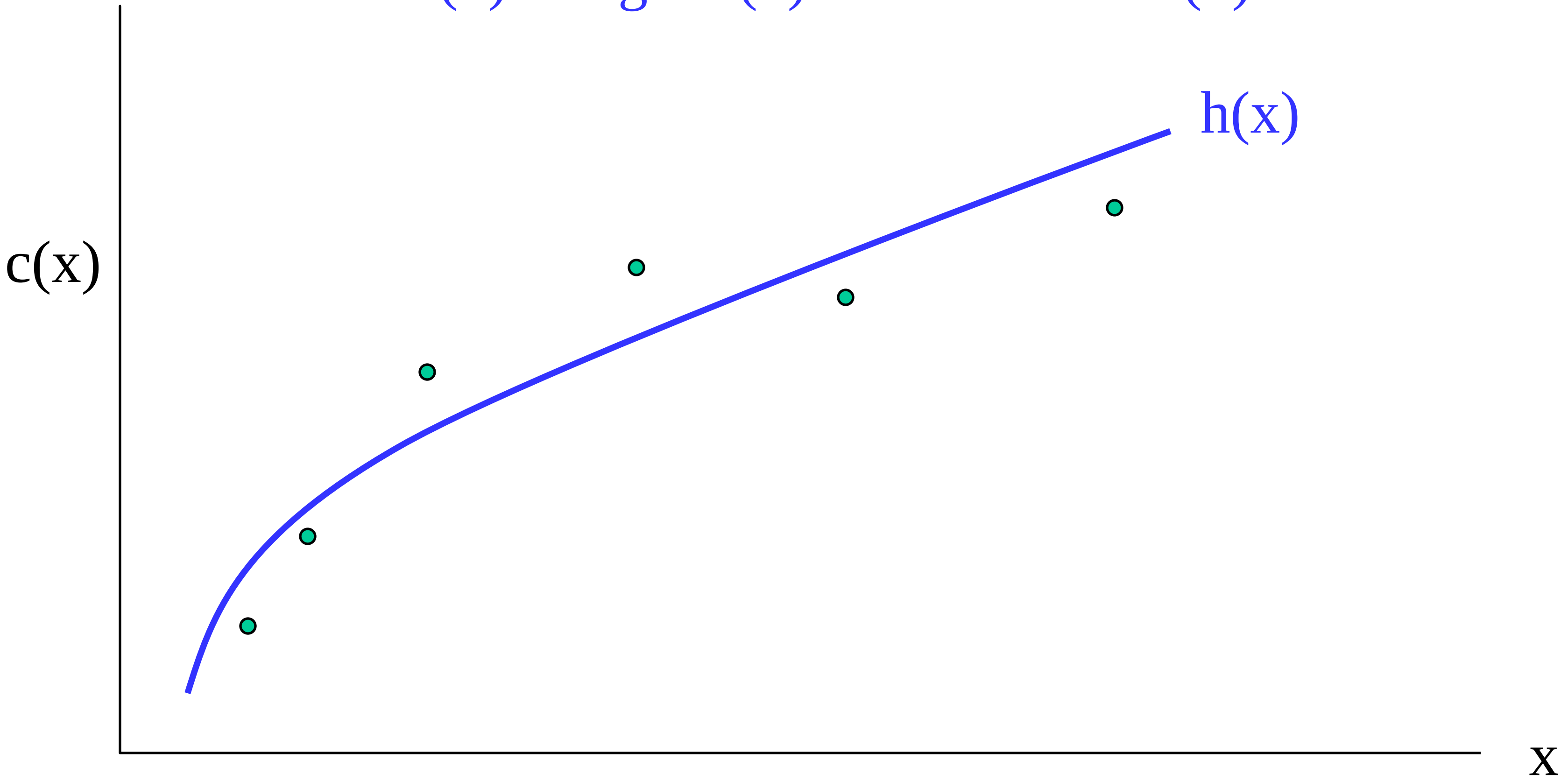
Consistency

Function Approximation

May not be any perfect fit

Classification ~ discrete functions

$h(x) = \text{nigeria}(x) \wedge \text{wire-transfer}(x)$



General Learning Issues

- Many hypotheses consistent with the training data.
- **Bias**
  - Any criteria other than consistency with the training data that is used to select a hypothesis.
- Classification accuracy
  - % of instances classified correctly
  - (Measured on independent test data.)
- Training time
  - Efficiency of training algorithm
- Testing time
  - Efficiency of subsequent classification

Why is Learning Possible?

Experience alone never justifies any conclusion about any unseen instance.

Learning occurs when

PREJUDICE meets DATA!

Bias

- The nice word for prejudice is “bias”.

- What kind of hypotheses will you ***consider***?
  - What is allowable ***range*** of functions you use when approximating?
- What kind of hypotheses do you ***prefer***?

Generalization

- Hypotheses must ***generalize*** to correctly classify instances not in the training data.

- Simply memorizing training examples is a consistent hypothesis ***that does not generalize***.

Bayesian Methods

- Learning and classification methods based on probability theory.
  - Bayes theorem plays a critical role in probabilistic learning and classification.
  - Uses *prior* probability of each category given no information about an item.
- Categorization produces a ***posterior*** probability distribution over the possible categories given a description of an item.

Bayesian Categorization

- Let set of categories be $\{c_1, c_2, \ldots c_n\}$
- Let E be description of an instance.
- Determine category of E by determining for each c_i

$$P(c_i \mid E) = \frac{P(c_i)P(E \mid c_i)}{P(E)}$$

- P(E) can be ignored since is factor ∀ categories

$$P(c_i \mid E) \sim P(c_i)P(E \mid c_i)$$

Bayesian Categorization

$$P(c_i \mid E) \sim P(c_i)P(E \mid c_i)$$

- Need to know:
  - Priors: P(c_i)
  - Conditionals: P(E | c_i)

Problem!

- P(c_i) are easily estimated from data.
  - If n_i of the examples in D are in c_i,then P(c_i) = n_i / |D|
- Assume instance is a conjunction of binary features:

$$E = e_1 \wedge e_2 \wedge \cdots \wedge e_m$$

- Too many possible instances (exponential in m) to estimate all P(E | c_i)

Naïve Bayesian Motivation

- Problem: Too many possible instances
  (exponential in *m*)
  to estimate all P(E | c_i)

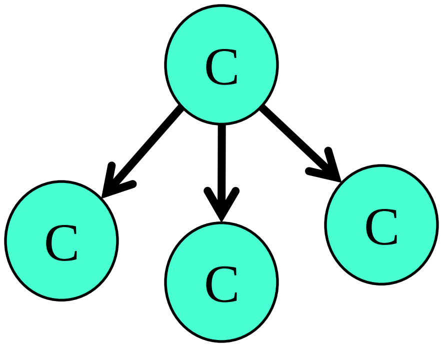
- If we assume features of an instance are independent given the category (c_i) (*conditionally independent*).

$$P(E \mid c_i) = P(e_1 \wedge e_2 \wedge \cdots \wedge e_m \mid c_i) = \prod_{j=1}^{m} P(e_j \mid c_i)$$

- Therefore, we then only need to know P(e_j | c_i) for each feature and category.

Naïve Bayes Example

- C = {allergy, cold, well}
- e_1 = sneeze; e_2 = cough; e_3 = fever
- E = {sneeze, cough, $\neg$fever}



Prob	Well	Cold	Allergy
$P(c_i)$	0.9	0.05	0.05
$P(\text{sneeze}\|c_i)$	0.1	0.9	0.9
$P(\text{cough}\|c_i)$	0.1	0.8	0.7
$P(\text{fever}\|c_i)$	0.01	0.7	0.4

Naïve Bayes Example (cont.)

Probability	Well	Cold	Allergy
$P(c_i)$	0.9	0.05	0.05
$P(\text{sneeze} \mid c_i)$	0.1	0.9	0.9
$P(\text{cough} \mid c_i)$	0.1	0.8	0.7
$P(\text{fever} \mid c_i)$	0.01	0.7	0.4

$E=\{\text{sneeze}, \text{cough}, \neg\text{fever}\}$

$P(\text{well} \mid E) = (0.9)(0.1)(0.1)(0.99)/P(E)=0.0089/P(E)$

$P(\text{cold} \mid E) = (0.05)(0.9)(0.8)(0.3)/P(E)=0.01/P(E)$

$P(\text{allergy} \mid E) = (0.05)(0.9)(0.7)(0.6)/P(E)=0.019/P(E)$

Most probable category: allergy

$P(E) = 0.089 + 0.01 + 0.019 = 0.0379$

$P(\text{well} \mid E) = 0.23$

$P(\text{cold} \mid E) = 0.26$

$P(\text{allergy} \mid E) = 0.50$

Learning Probabilities

- Normally, probabilities are estimated based on observed frequencies in the training data.
- If D contains n_i examples in category c_i, and n_{ij} of these n_i examples contains feature e_j, then:

$$P(e_j \mid c_i) = \frac{n_{ij}}{n_i}$$

- However, estimating such probabilities from small training sets is error-prone.
- If due only to chance, a rare feature, e_k, is always false in the training data, $\forall c_i$:P($e_k \mid c_i$) = 0.
- If e_k then occurs in a test example, E, the result is that $\forall c_i$: P($E \mid c_i$) = 0 and $\forall c_i$: P($c_i \mid E$) = 0

Smoothing

- To account for estimation from small samples, probability estimates are adjusted or *smoothed*.
- Laplace smoothing using an *m*-estimate assumes that each feature is given a prior probability, *p*, that is assumed to have been previously observed in a "virtual" sample of size *m*.

$$P(e_j \mid c_i) = \frac{n_{ij} + mp}{n_i + m} \qquad = (n_{ij} + 1) / (n_i + 2)$$

- For binary features, *p* is simply assumed to be 0.5.

Naïve Bayes for Text

- Modeled as generating a bag of words for a document in a given category by repeatedly sampling with replacement from a vocabulary $V = \{w_1, w_2, \ldots w_m\}$ based on the probabilities $P(w_j \mid c_i)$.
- Smooth probability estimates with Laplace *m*-estimates assuming a uniform distribution over all words ($p = 1/|V|$) and $m = |V|$
  - Equivalent to a virtual sample of seeing each word in each category exactly once.

Text Naïve Bayes Algorithm (Train)

Let V be the vocabulary of all words in the documents in D

For each category $c_i \in C$

Let D_i be the subset of documents in D in category c_i

$\mathrm{P}(c_i) = |D_i| \,/\, |D|$

Let T_i be the concatenation of all the documents in D_i

Let n_i be the total number of word occurrences in T_i

For each word $w_j \in V$

Let n_{ij} be the number of occurrences of w_j in T_i

Let $\mathrm{P}(w_i \mid c_i) = (n_{ij} + 1) \,/\, (n_i + |V|)$

Text Naïve Bayes Algorithm (Test)

Given a test document X

Let n be the number of word occurrences in X

Return the category:

$$\underset{c_i \in C}{\operatorname{argmax}} \; P(c_i) \prod_{i=1}^{n} P(a_i \mid c_i)$$

where a_i is the word occurring the ith position in X

Naïve Bayes Time Complexity

- Training Time: $O(|D|L_d + |C||V|))$ where L_d is the average length of a document in D.
  - Assumes V and all D_i , n_i, and n_{ij} pre-computed in $O(|D|L_d)$ time during one pass through all of the data.
  - Generally just $O(|D|L_d)$ since usually $|C||V| < |D|L_d$
- Test Time: $O(|C|\ L_t)$ where L_t is the average length of a test document.
- Very efficient overall, linearly proportional to the time needed to just read in all the data.

Easy to Implement

- But…
- If you do… it probably won't work…

Probabilities: Important Detail!

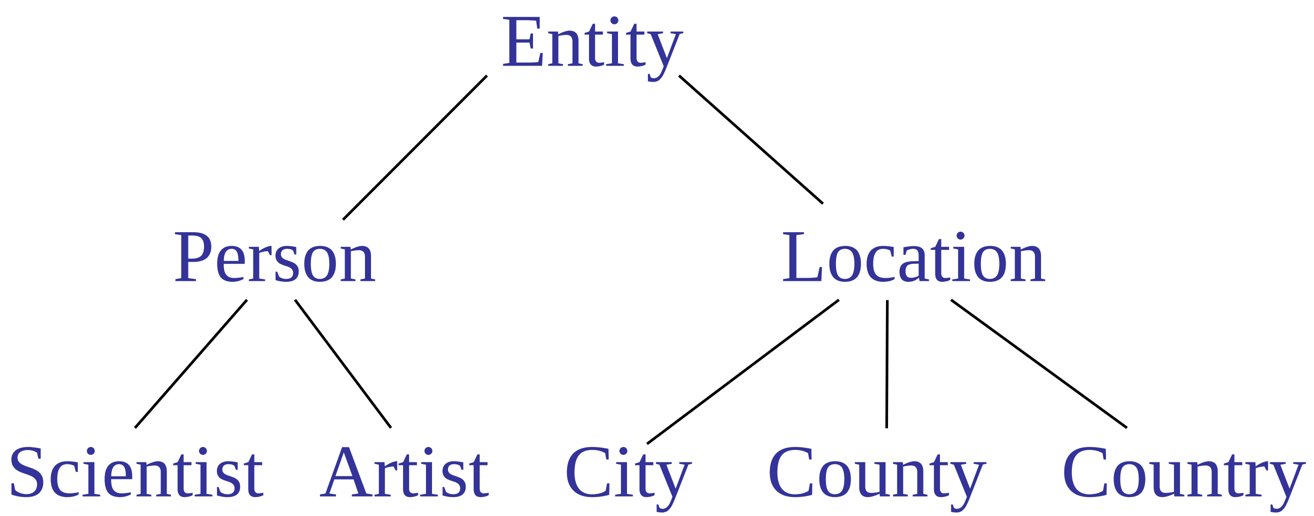
- We are multiplying lots of small numbers
  Danger of underflow!
  - $0.5^{57} = 7\text{ E }-18$
- Solution? Use logs and add!
  - $p_1 * p_2 = e^{\log(p1)+\log(p2)}$
  - Always keep in log form

Multi-Class Categorization

- Pick the category with max probability
- One-vs-all (OVA) Create many 1 vs other classifiers
  - Classes = City, County, Country
  - Classifier 1 = {City} {County, Country}
  - Classifier 2 = {County} {City, Country}
  - Classifier 3 = {Country} {City, County}
- **All-vs-all (AVA)** For each pair of classes build a classifier
  - {City vs. County}, {City vs Country}, {County vs. Country}

Multi-Class Categorization

- Pick the category with max probability
- Create many OVA/AVA classifiers
- Use a hierarchical approach (wherever hierarchy available)



Advantages

- Simple to implement
  - No numerical optimization, matrix algebra, etc
- Efficient to train and use
  - Easy to update with new data
  - Fast to apply
- Binary/multi-class
- Independence allows parameters to be estimated on different datasets
- Comparitively good effectiveness with small training sets

Disadvantages

- Independence assumption wrong
  - Absurd estimates of class probabilities
    - Output probabilities close to 0 or 1
  - Thresholds must be tuned; not set analytically

- Generative model
  - Generally lower effectiveness than discriminative techniques
  - Improving parameter estimates can hurt classification effectiveness

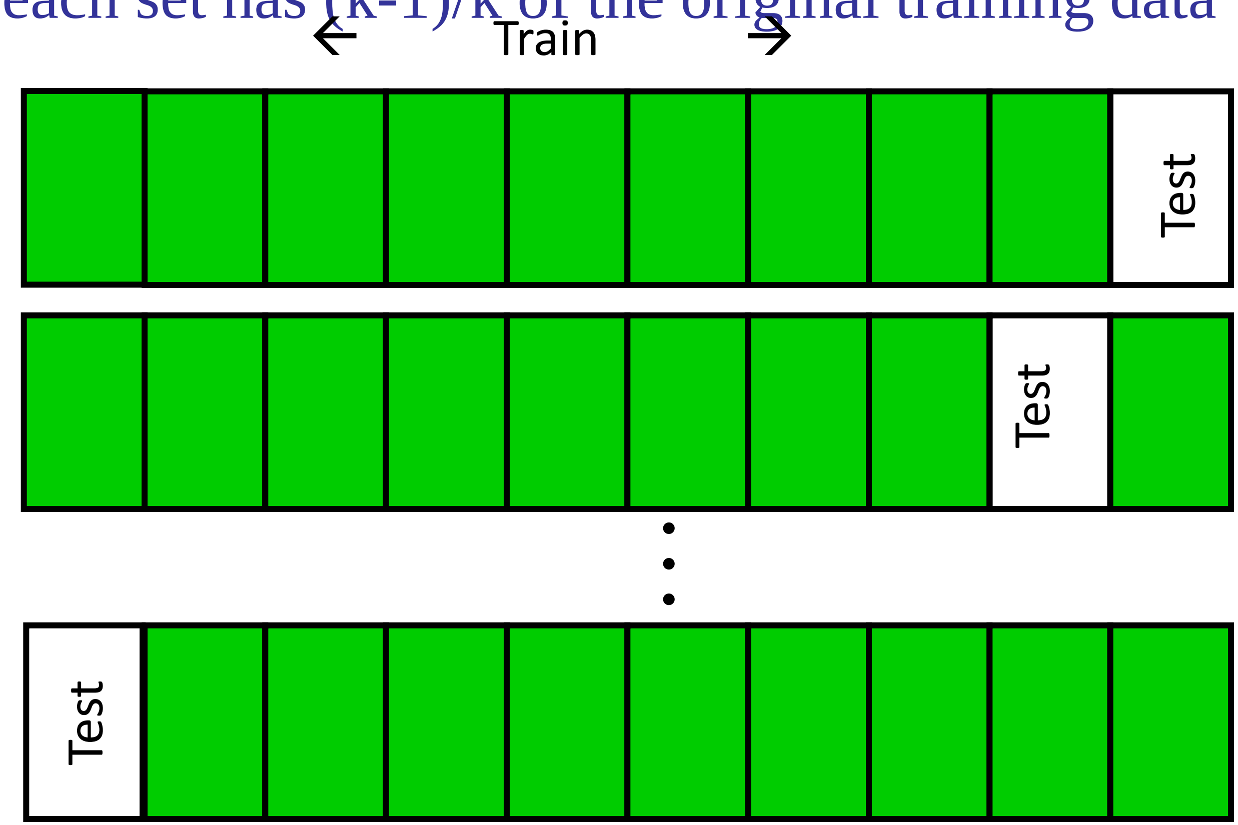
Experimental Evaluation

Question: How do we estimate the performance of classifier on unseen data?

- Can’t just at accuracy on training data – this will yield an over optimistic estimate of performance
- Solution: Cross-validation
- Note: this is sometimes called estimating how well the classifier will generalize

Evaluation: Cross Validation

- Partition examples into *k* disjoint sets
- Now create *k* training sets
  - Each set is union of all equiv classes ***except one***
  - So each set has (k-1)/k of the original training data



Cross-Validation (2)

- Leave-one-out
  - Use if < 100 examples (rough estimate)
  - Hold out one example, train on remaining examples
- 10-fold
  - If have 100-1000's of examples
- M of N fold
  - Repeat M times
  - Divide data into N folds, do N fold cross-validation

Evaluation Metrics

- Accuracy: no. of questions correctly answered
- Precision (for one label): accuracy when classification = label
- Recall (for one label): measures how many instances of a label were missed.
- F-measure (for one label): harmonic mean of precision & recall.
- Area under Precision-recall curve (for one label): vary parameter to show different points on p-r curve; take the area

Precision & Recall

Two class situation

	Predicted	
Actual	**"P"**	**"N"**
P	**TP**	**FN**
N	**FP**	**TN**

Precision = TP/(TP+FP)
Recall = TP/(TP+FN)
F-measure = 2pr/(p+r)

A typical precision-recall curve