Machine Learning

A large and fascinating field;
there's much more than what you'll see in this class!

What should we try to learn, if we want to…

- make computer systems more efficient or secure?
- make money in the stock market?
  - avoid losing money to fraud or scams?
- do science or medicine?
- win at games?
- make more entertaining games?
- improve user interfaces?
  - even brain-computer interfaces …
- make traditional applications more useful?
  - word processors, drawing programs, email, web search, photo organizers, …

What should we try to learn, if we want to…

- make c ?
- make r
  - avoid
- do scie
- win at g
- make r
- improv
  - even
- make t
  - word
    photo earch,

This stuff has got to be an important part of the future …

… beats trying to program all the special cases directly

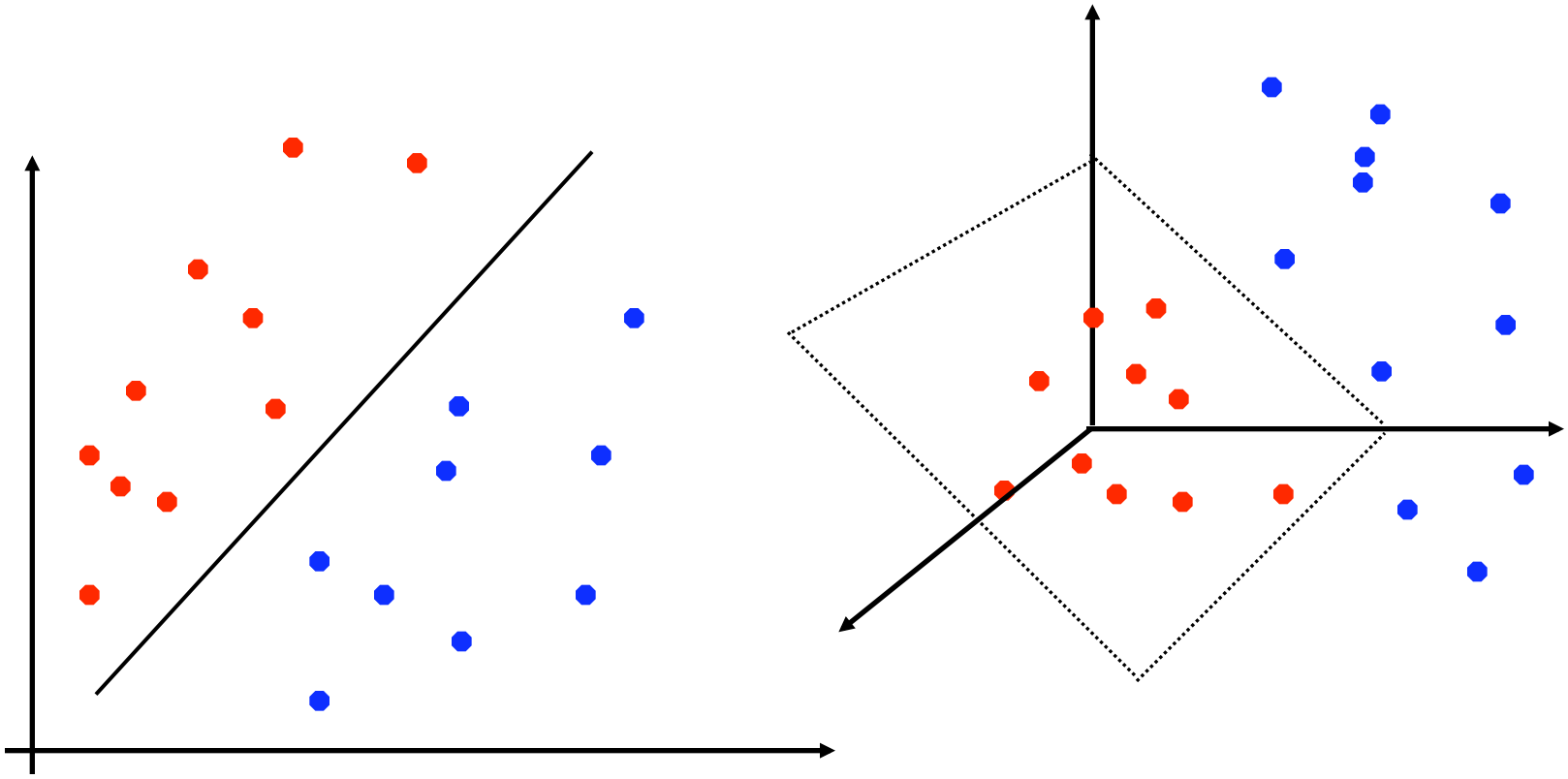
… and there are "intelligent" behaviors you can't imagine programming directly. (Most of the stuff now in your brain wasn't programmed in advance, either!)

The simplest problem: Supervised binary classification of vectors

- Training set:
  $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$
  where $x_1, x_2, \ldots$ are in R^n
  and $y_1, y_2, \ldots$ are in {0,1} or {-,+} or {-1,1}

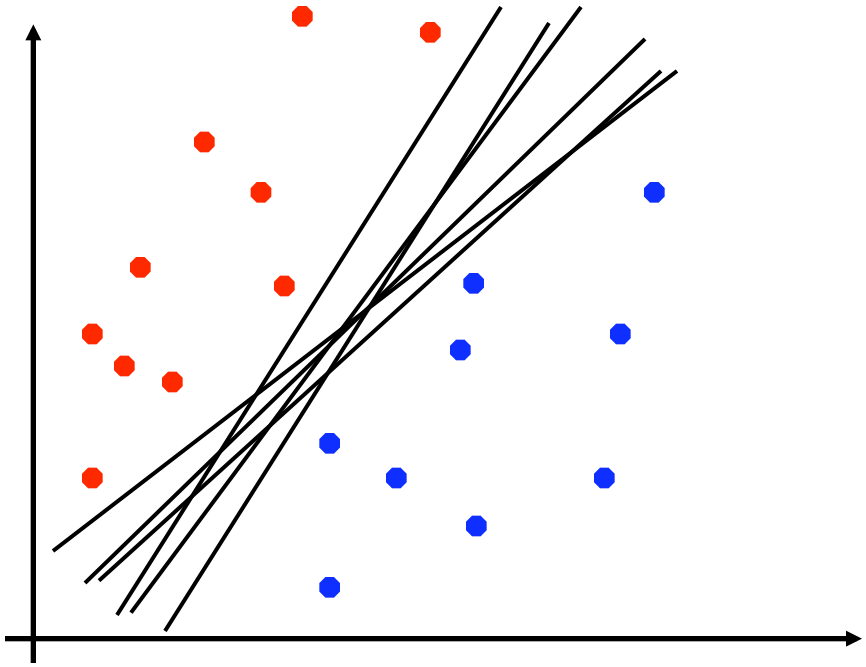
- Test set:
  $(x_{n+1}, ?), (x_{n+2}, ?), \ldots (x_{n+m}, ?)$
  where these x's were probably **not** seen in training

Linear Separators



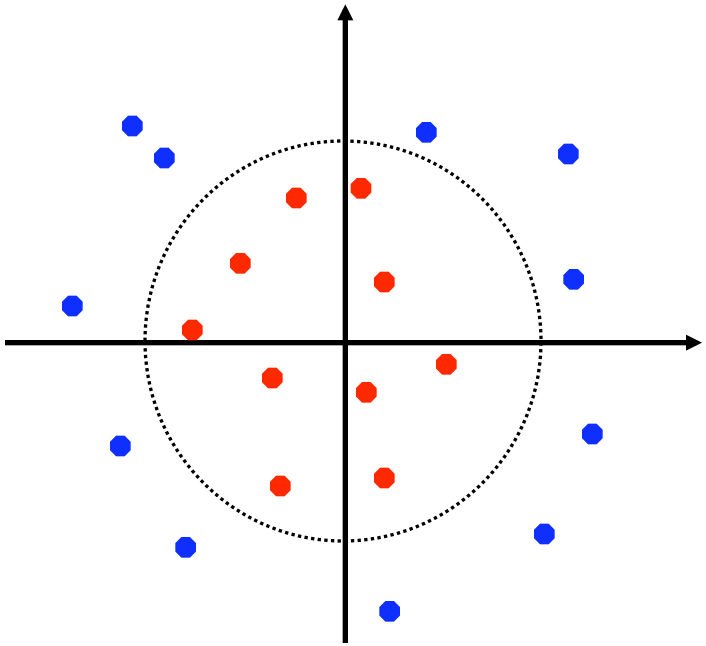
slide thanks to Ray Mooney

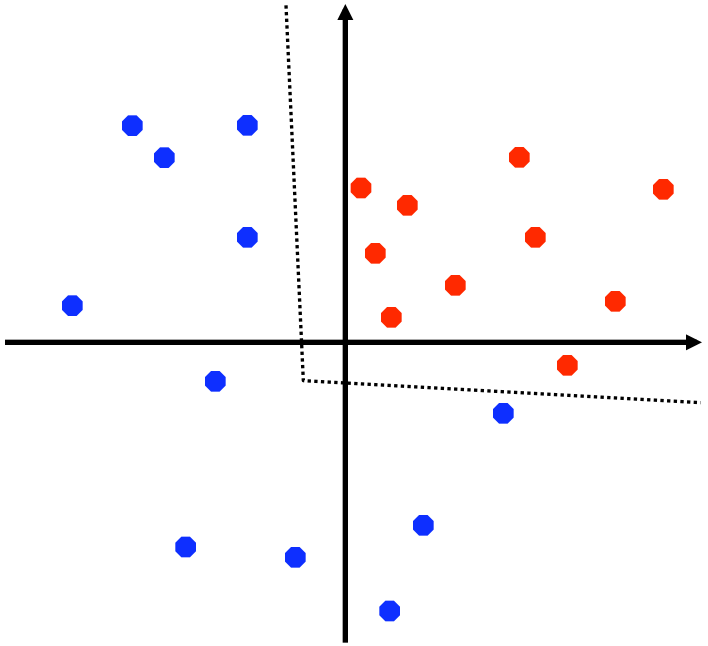
Linear Separators



slide thanks to Ray Mooney

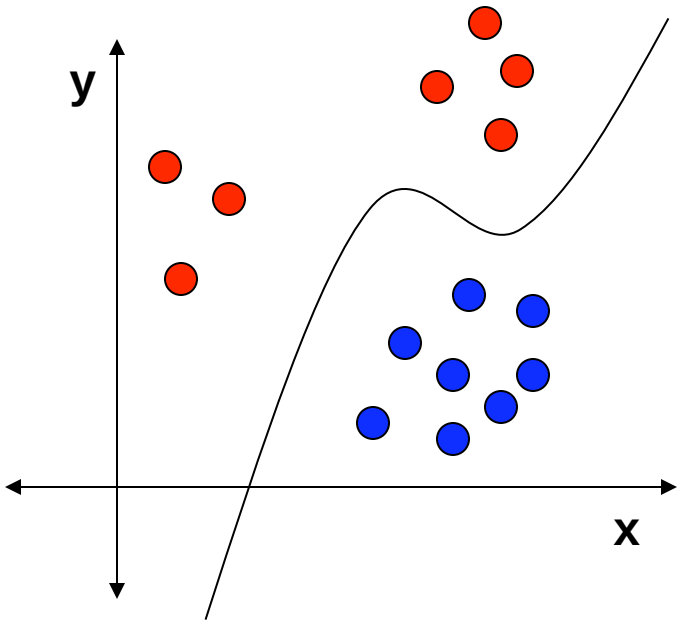
Nonlinear Separators





slide thanks to Ray Mooney (modified)

Nonlinear Separators



Note: A more complex function requires more data to generate an accurate model (sample complexity)

Encoding and decoding for learning

- Binary classification of vectors … but how do we treat "real" learning problems in this framework?

- We need to encode each input example as a vector in R^n:

  feature extraction

Features for recognizing a chair?












Features for recognizing childhood autism?

(from DSM IV, the Diagnostic and Statistical Manual)

A. A total of six (or more) items from (1), (2), and (3), with at least two from (1), and one each from (2) and (3):

- (1) Qualitative impairment in social interaction, as manifested by at least two of the following:
  - marked impairment in the use of multiple nonverbal behaviors such as eye-to-eye gaze, facial expression, body postures, and gestures to regulate social interaction.
  - failure to develop peer relationships appropriate to developmental level
  - a lack of spontaneous seeking to share enjoyment, interests, or achievements with other people (e.g., by a lack of showing, bringing, or pointing out objects of interest)
  - lack of social or emotional reciprocity
- (2) Qualitative impairments in communication as manifested by at least one of the following: …

Features for recognizing childhood autism?

(from DSM IV, the Diagnostic and Statistical Manual)

B. Delays or abnormal functioning in at least one of the following areas, with onset prior to age 3 years:

(1) social interaction

(2) language as used in social communication, or

(3) symbolic or imaginative play.

C. The disturbance is not better accounted for by Rett's disorder or childhood disintegrative disorder.

Features for recognizing a prime number?

- (2,+) (3,+) (4,-) (5,+) (6,-) (7,+) (8,-) (9,-) (10,-) (11,+) (12,-) (13,+) (14,-) (15,-) …

- Ouch!
- But what kinds of features might you try if you didn't know anything about primality?
- How well would they work?
  - False positives vs. false negatives?
  - Expected performance vs. worst-case

Features for recognizing masculine vs. feminine words in French?

- le fromage (cheese)
- le monument (monument)
- le sentiment (feeling)
- le couteau (knife)
- le téléphone (telephone)
- le microscope (microscope)
- le romantisme (romanticism)

la salade (salad, lettuce)
la fourchette (fork)
la télévision (television)
la culture (culture)
la situation (situation)
la société (society)
la différence (difference)
la philosophie (philosophy)

Features for recognizing when the user who's typing isn't the usual user?

- (And how do you train this?)

Measuring performance

- Simplest: Classification error (fraction of wrong answers)
- Better: Loss functions – different penalties for false positives vs. false negatives
- If the learner gives a confidence or probability along with each of its answers, give extra credit for being confidently right but extra penalty for being confidently wrong
  - What's the formula?
  - Correct answer is $y_i \in \{-1, +1\}$
  - System predicts $z_i \in [-1, +1]$ (perhaps fractional)
  - Score is $\Sigma_i\ y_i * z_i$

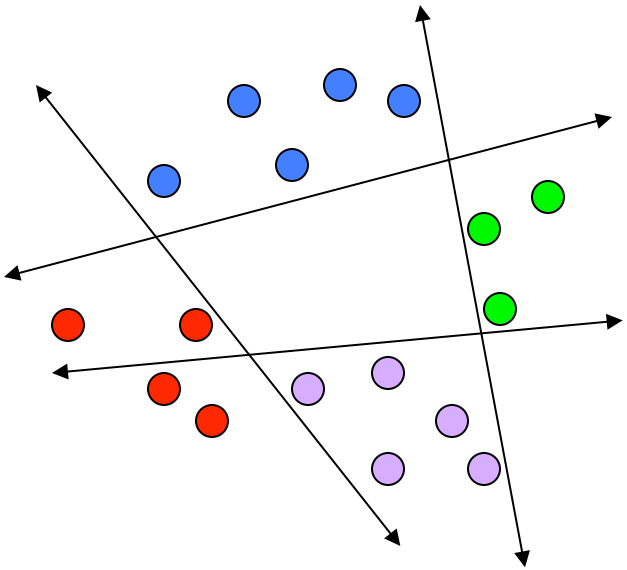
Encoding and decoding for learning

- Binary classification of vectors … but how do we treat "real" learning problems in this framework?

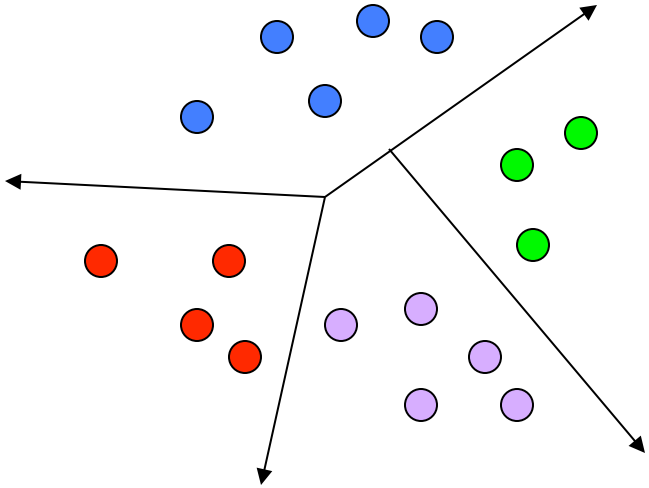
- If the output is to be binary, we need to encode each input example as a vector in R^n:
  feature extraction

- If the output is to be more complicated, we may need to obtain it as a sequence of binary decisions, each on a different feature vector

Multiclass Classification



Many binary classifiers
("one versus all")



One multiway classifier

Regression: predict a number, not a class

- Don't just predict whether stock will go up or down in the present circumstance – predict by how much!

- Better, predict probabilities that it will go up and down by different amounts

Inference: Predict a whole pattern

- Predict a whole object
  (in the sense of object-oriented programming)
- Output is a vector, or a tree, or something
  - Why useful?
- Or, return many possible trees with a different probability on each one
- Some fancy machine learning methods can handle this directly … but how would you do a simple encoding?

Defining Learning Problems

- ML algorithms are mathematical formalisms and problems must be modeled accordingly
- Feature Space – space used to describe each instance; often $\mathbf{R}^d$, $\{0,1\}^d$, etc.
- Output Space – space of possible output labels
- Hypothesis Space – space of functions that can be selected by the machine learning algorithm (depends on the algorithm)

Context Sensitive Spelling

Did anybody (else) want **too** sleep for **to** more hours this morning?

- Output Space
  - Could use the entire vocabulary; *Y*={a,aback,...,zucchini}
  - Could also use a confusion set; *Y*={to, too, two}
- Model as (single label) multiclass classification

$$\Re^d \rightarrow \{to, too, two\}$$

- Hypothesis space is provided by your learner
- Need to define the feature space

Sentence Representation

S = I would like a **piece** of cake too!

- Define a set of features
  - Features are relations that hold in the sentence.
- Two components to defining features
  - Describe relations in the sentence: text, text ordering, properties of the text (information sources)
  - Define functions based upon these relations (more on this later)

Sentence Representation

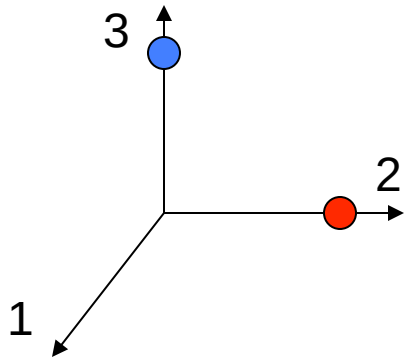
S_1 = I would like a **piece** of cake too!

S_2 = This is not the way to achieve **peace** in Iraq.

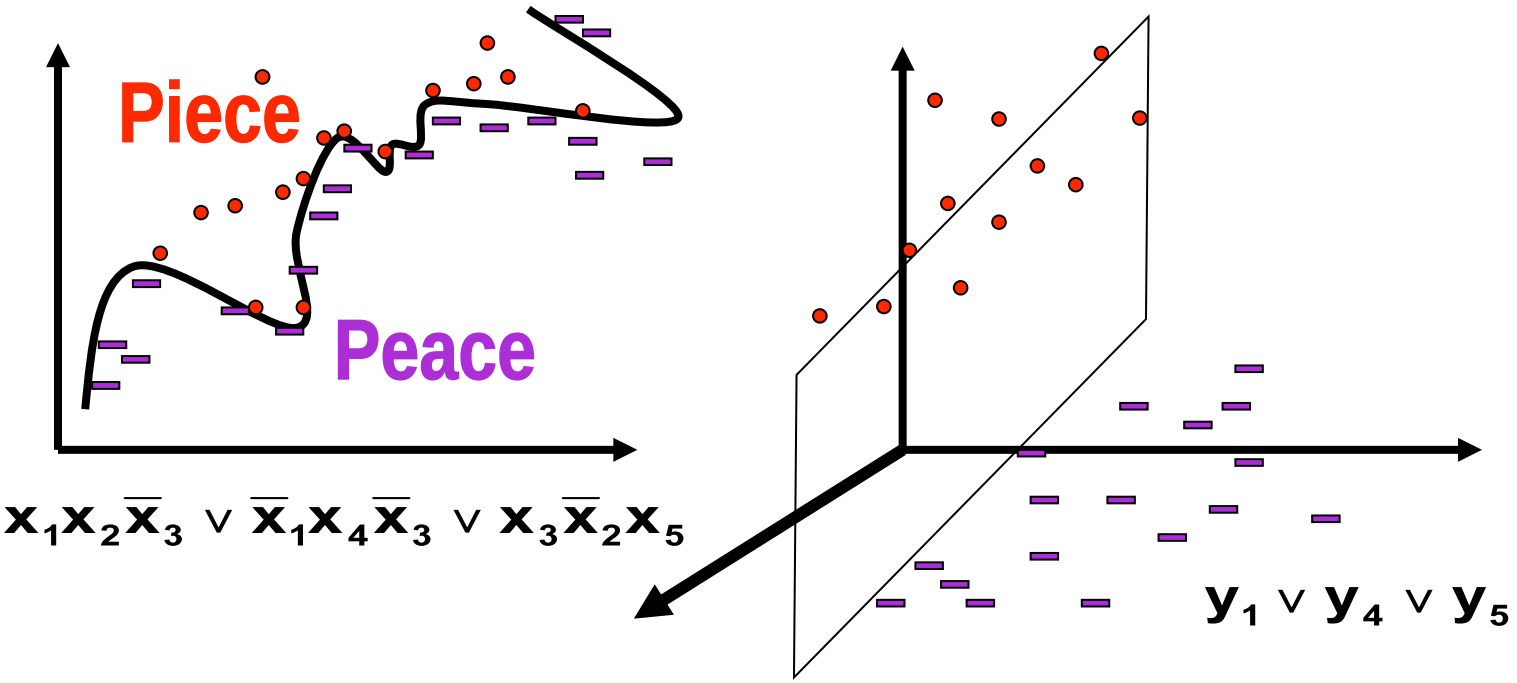
- Examples of (simple) features
  1. Does 'ever' appear within a window of 3 words?
  2. Does 'cake' appear within a window of 3 words?
  3. Is the preceding word a verb?

S_1 = 0, 1, 0

S_2 = 0, 0, 1



Embedding



- Requires some knowledge engineering
- Makes the discriminant function simpler (and learnable)

Sparse Representation

- Between basic and complex features, the dimensionality will be very high
  - Most features will not be active in a given example
- Represent vectors with a list of active indices

$S_1 = 1, 0, 1, 0, 0, 0, 1, 0, 0, 1$ becomes $S_1 = 1, 3, 7, 10$

$S_2 = 0, 0, 0, 1, 0, 0, 1, 0, 0, 0$ becomes $S_2 = 4, 7$

Types of Sparsity

- Sparse Function Space
  - High dimensional data where target function depends on a few features (many irrelevant features)
- Sparse Example Space
  - High dimensional data where only a few features are active in each example
- In NLP, we typically have both types of sparsity.

Training paradigms

- Supervised?
- Unsupervised?
- Partly supervised?
- Incomplete?
- Active learning, online learning
- Reinforcement learning

Training and test sets

- How this relates to the midterm
  - Want you to do well – proves I'm a good teacher (merit pay?)
  - So I want to teach to the test …
    - heck, just show you the test in advance!
  - Or equivalently, test exactly what I taught …
    - what was the title of slide 29?
  - How should JHU prevent this?
    - what would the title of slide 29 ½ have been?

- Development sets
  - the market newsletter scam
  - so, what if we have an army of robotic professors?
    - some professor's class will do well just by luck! she wins!
    - JHU should only be able to send one prof to the professorial Olympics
      - Olympic trials are like a development set

Overfitting and underfitting

- Overfitting: Model the training data all too well (autistic savants?). Do really well if we test on the training data, but poorly if we test on new data.
- Underfitting: Try too hard to generalize. Ignore relevant distinctions – try to find a simple linear separator when the data are actually more complicated than that.
- How does this relate to the # of parameters to learn?
- Lord Kelvin: "And with 3 parameters, I can fit an elephant …"

"Feature Engineering" Workshop in 2005

CALL FOR PAPERS

Feature Engineering for Machine Learning in Natural Language Processing

Workshop at the Annual Meeting of the Association of Computational Linguistics (ACL 2005)

http://research.microsoft.com/~ringger/FeatureEngineeringWorkshop/

Submission Deadline: April 20, 2005

Ann Arbor, Michigan
June 29, 2005

"Feature Engineering" Workshop in 2005

As experience with machine learning for solving natural language processing tasks accumulates in the field, practitioners are finding that feature engineering is as critical as the choice of machine learning algorithm, if not more so.

Feature design, feature selection, and feature impact (through ablation studies and the like) significantly affect the performance of systems and deserve greater attention.

In the wake of the shift away from knowledge engineering and of the successes of data-driven and statistical methods, researchers in the field are likely to make further progress by incorporating additional, sometimes familiar, sources of knowledge as features.

Although some experience in the area of feature engineering is to be found in the theoretical machine learning community, the particular demands of natural language processing leave much to be discovered.

"Feature Engineering" Workshop in 2005

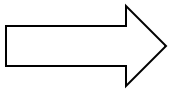
Topics may include, but are not necessarily limited to:

- Novel methods for discovering or inducing features, such as mining the web for closed classes, useful for indicator features.
- Comparative studies of different feature selection algorithms for NLP tasks.
- Interactive tools that help researchers to identify ambiguous cases that could be disambiguated by the addition of features.
- Error analysis of various aspects of feature induction, selection, representation.
- Issues with representation, e.g., strategies for handling hierarchical representations, including decomposing to atomic features or by employing statistical relational learning.
- Techniques used in fields outside NLP that prove useful in NLP.
- The impact of feature selection and feature design on such practical considerations as training time, experimental design, domain independence, and evaluation.
- Analysis of feature engineering and its interaction with specific machine learning methods commonly used in NLP.
- Combining classifiers that employ diverse types of features.
- Studies of methods for defining a feature set, for example by iteratively expanding a base feature set.
- Issues with representing and combining real-valued and categorical features for NLP tasks.

A Machine Learning System

Preprocessing Text

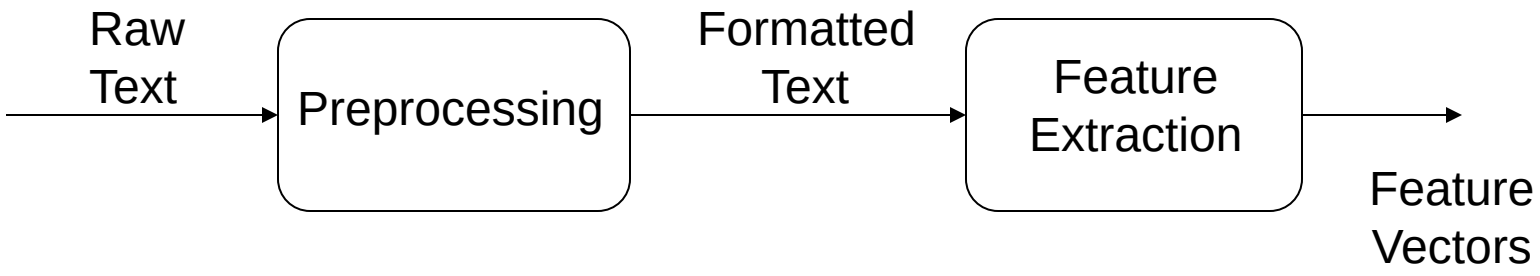
They recently recovered a small piece of a live Elvis concert recording.
He was singing gospel songs, including “Peace in the Valley.”



0	0	0	They
0	0	1	recently
0	0	2	recovered
0	0	3	a
0	0	4	small
piece	0	5	piece
0	0	6	of
:			
0	1	6	including
0	1	7	QUOTE
peace	1	8	Peace
0	1	9	in
0	1	10	the
0	1	11	Valley
0	1	12	.
0	1	13	QUOTE

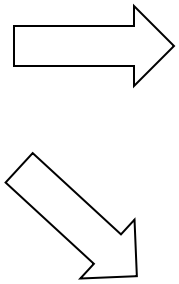
- Sentence splitting, Word Splitting, etc.
- Put data in a form usable for feature extraction

A Machine Learning System



Feature Extraction

0	0	0	They
0	0	1	recently
0	0	2	recovered
0	0	3	a
0	0	4	small
piece	0	5	piece
0	0	6	of
:			
0	1	6	including
0	1	7	QUOTE
peace	1	8	Peace
0	1	9	in
0	1	10	the
0	1	11	Valley
0	1	12	.
0	1	13	QUOTE



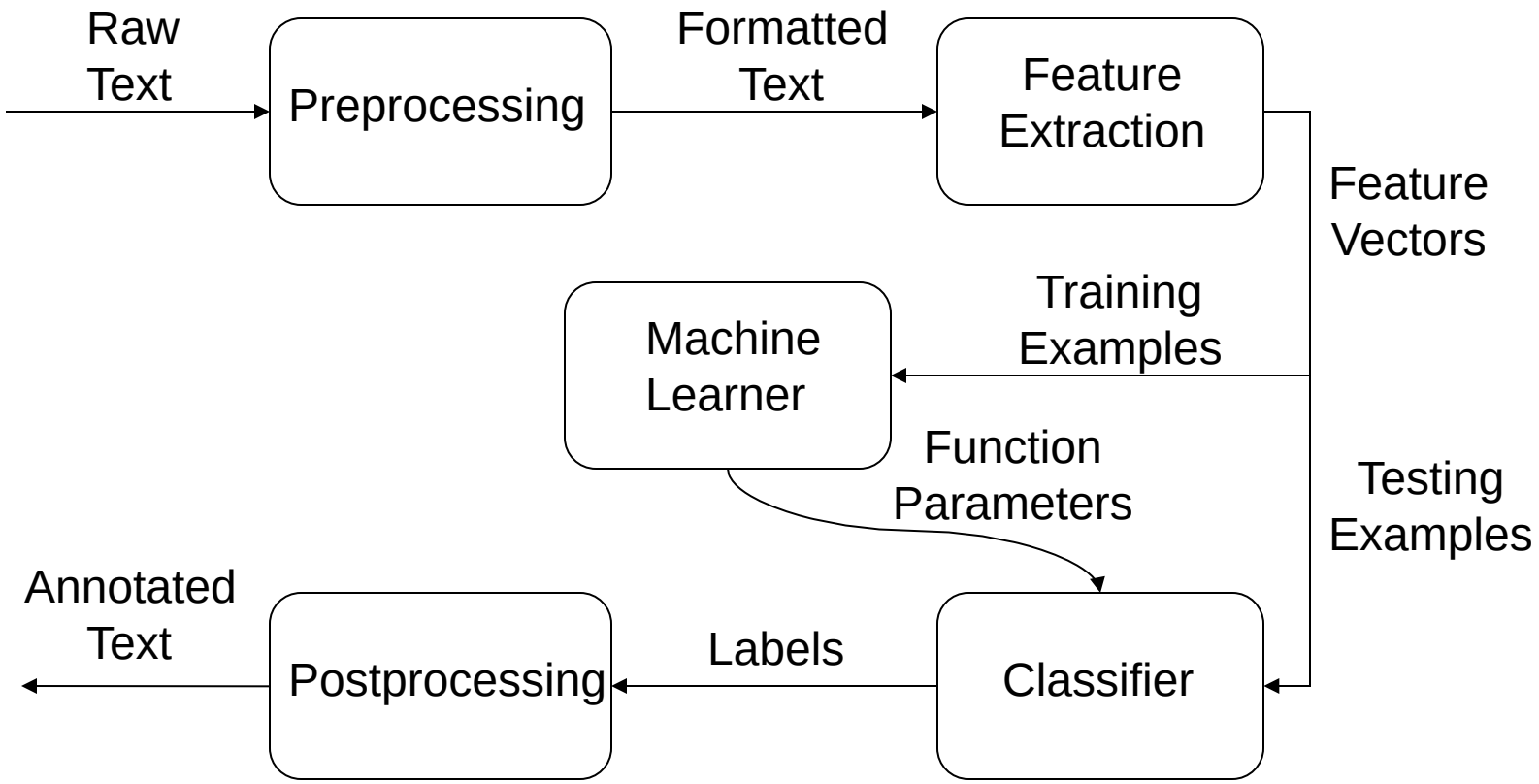
0, 1001, 1013, 1134, 1175, 1206

1, 1021, 1055, 1085, 1182, 1252

Lexicon File

- Converts formatted text into feature vectors
- Lexicon file contains feature descriptions

A Machine Learning System



A Machine Learning System