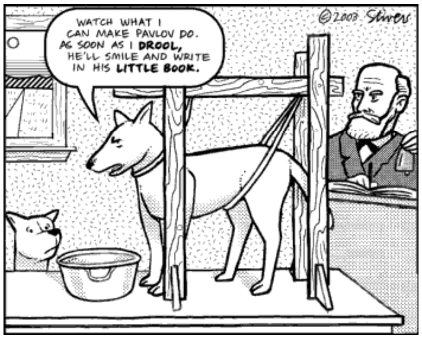
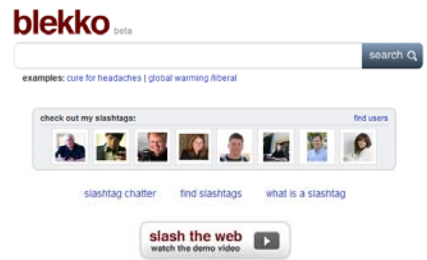
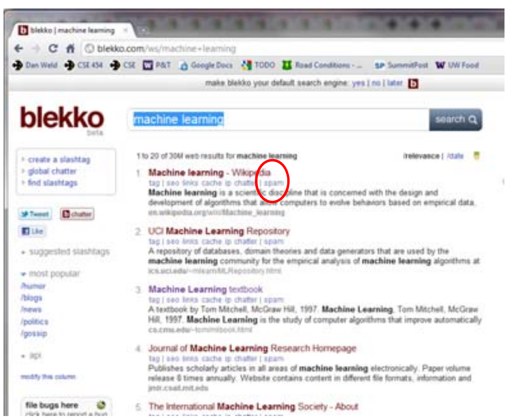
Machine Learning

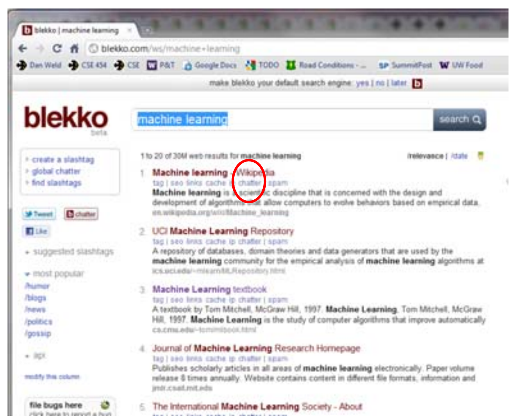


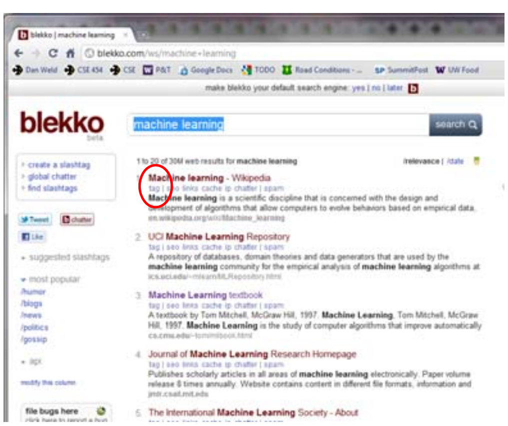
CSE 454

Search Engine News

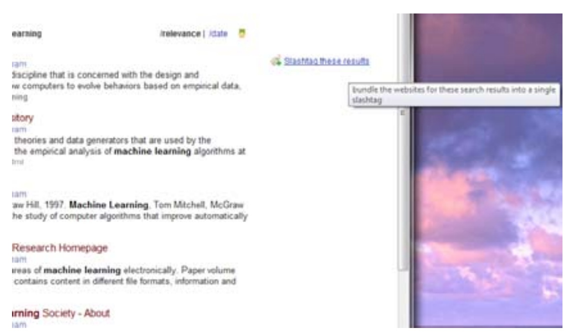


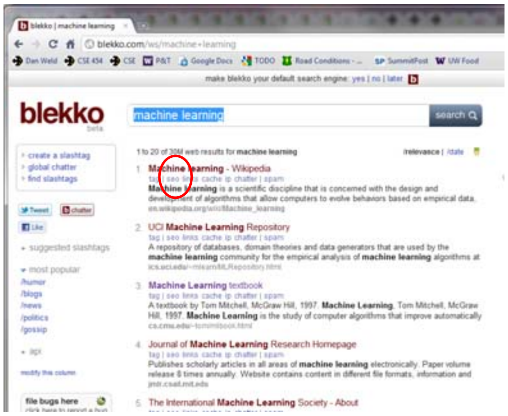


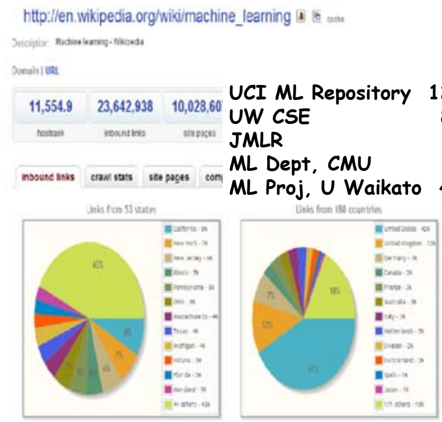




Slashtags



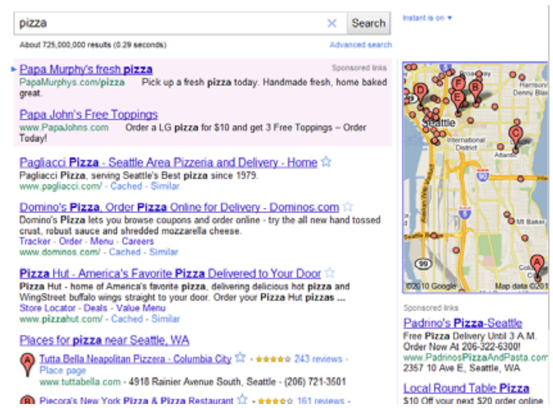


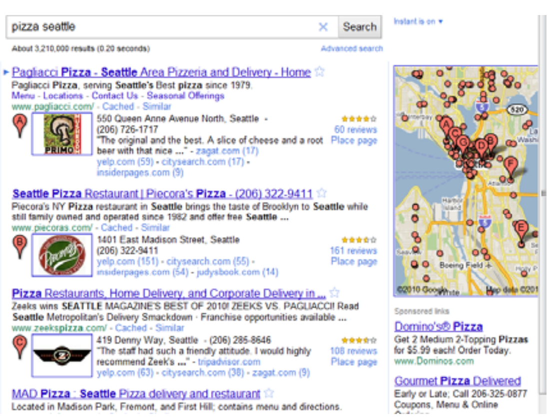


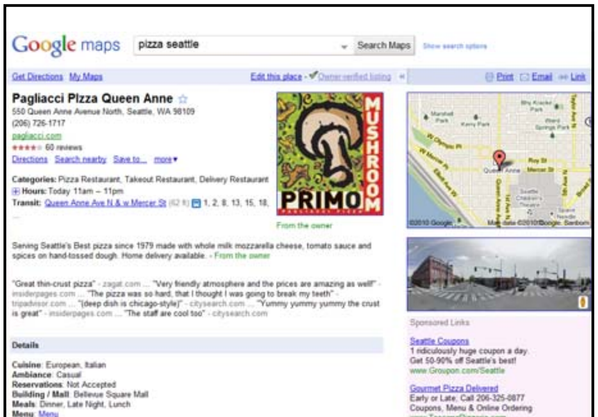
UCI ML Repository	1201
UW CSE	879
JMLR	148
ML Dept, CMU	32
ML Proj, U Waikato	458

Google Local Search

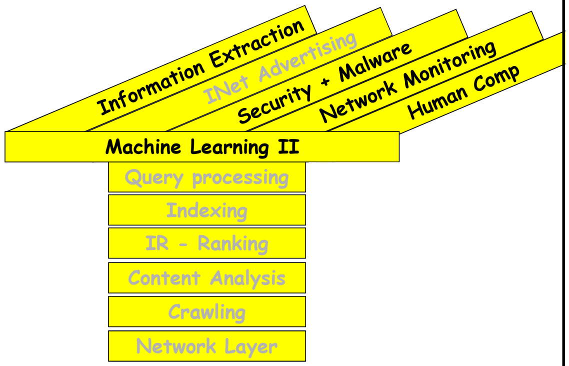
- Place search
  - Finding local businesss w/ maps, reviews, etc
- Boost
  - Ads for local businesses
- 20% searches are related to location







Class Overview



Next few classes

- Machine Learning
- Malware (Arvind Krishnamurthy)
- Information Extraction
- " " Continued
- NLP Basics: Parsing & POS Tagging
- Internet-Enabled Human Computation

Today's Outline

- Brief supervised learning review
- Evaluation
- Overfitting
- Ensembles
  Learners: The more the merrier
- Co-Training
  (Semi) Supervised learning with few labeled training ex

Sample Category Learning Problem

- Instance language: <size, color, shape>
  size ∈ {small, medium, large}
  color ∈ {red, blue, green}
  shape ∈ {square, circle, triangle}
- *C* = {**positive**, negative}
- *D*:

Example	Size	Color	Shape	Category
1	**small**	**red**	**circle**	**positive**
2	**large**	**red**	**circle**	**positive**
3	small	red	triangle	negative
4	large	blue	circle	negative

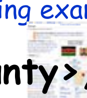
Example: County *vs.* Country?

- **Given:**
  - **A description of an instance, $x \in X$, where X is the *instance language* or *instance space*.**
  - **A fixed set of categor... $C = \{c_1, c_2, \ldots c_n\}$**
- **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.**

Bag of words representation

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>, <  , c

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

Consistency

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***.

19

Why is Learning Possible?

Experience alone never justifies any conclusion about any unseen instance.

Learning occurs when
PREJUDICE meets DATA!

Learning a "Frobnitz

© Daniel S. Weld

20

Bias

- Which hypotheses ***will you consider***?
- Which hypotheses do you ***prefer***?

© Daniel S. Weld

21

Some Typical Biases

Occam's razor
"It is needless to do more when less will suffice"
– William of Occam,
died 1349 of the Black plague

MDL – Minimum description length

Concepts can be approximated by
... **conjunctions** of predicates
... by **linear** functions
... by **short** decision trees

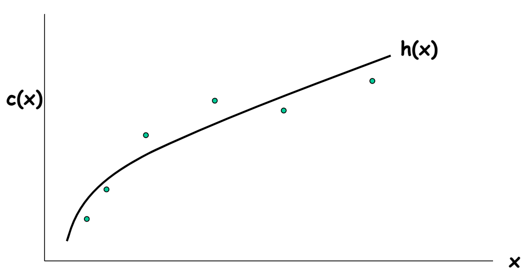
Frobnitz?

© Daniel S. Weld

22

ML = Function Approximation

May not be any perfect fit
Classification ~ discrete functions



23

Supervised Learning

- **Inductive learning** or "**Prediction**":
  Given examples of a function *(X, F(X))*
  Predict function *F(X)* for new examples *X*
- Classification
  F(X) = Discrete
- Regression
  F(X) = Continuous
- Probability estimation
  F(X) = Probability*(X)*:

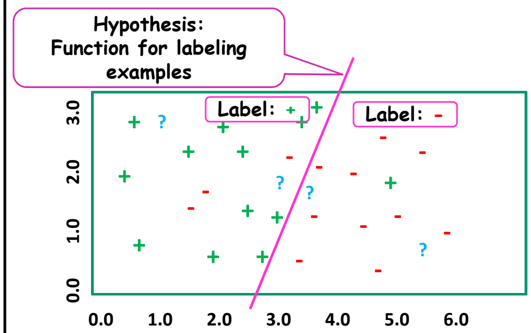
© Daniel S. Weld

24

Types of Learning

- **Supervised (inductive) learning**
  Training data includes desired outputs
- **Semi-supervised learning**
  Training data includes a ***few*** desired outputs
- **Unsupervised learning**
  Training data ***doesn't*** include desired outputs
- **Reinforcement learning**
  Rewards from sequence of actions

Classifier



Today's Outline

- Brief supervised learning review
- Evaluation
- Overfitting
- Ensembles
  Learners: The more the merrier
- Co-Training
  (Semi) Supervised learning with few labeled training ex

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

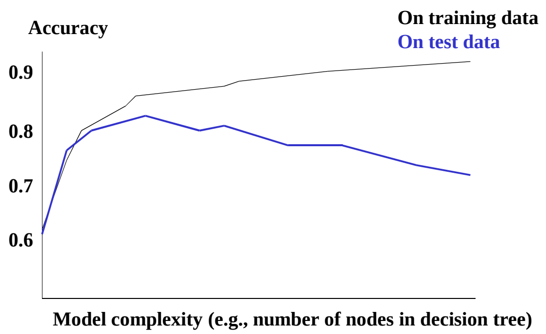
Today's Outline

- Brief supervised learning review
- Evaluation
- Overfitting
- Ensembles
  - Learners: The more the merrier
- Co-Training
  - (Semi) Supervised learning with few labeled training ex
- Clustering
  - No training examples

Overfitting Definition

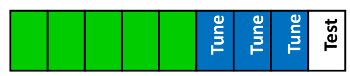
- Hypothesis H is *overfit* when ∃ H' and
  - H has ***smaller*** error on training examples, but
  - H has ***bigger*** error on test examples
- Causes of overfitting
  - Noisy data, or
  - Training set is too small
  - Large number of features
- Big problem in machine learning
- One solution: Validation set

Overfitting



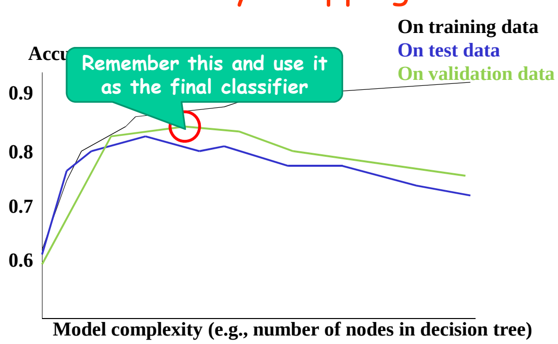
Validation/Tuning Set

- Split data into train and validation set

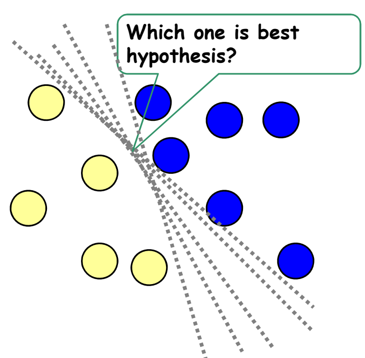


- Score each model on the tuning set, use it to pick the 'best' model

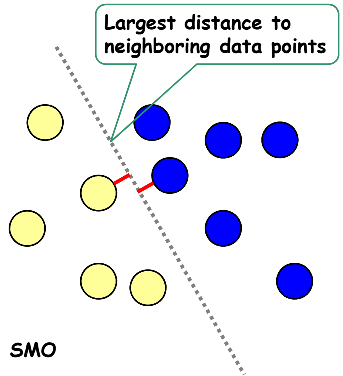
Early Stopping



Support Vector Machines



Support Vector Machines



SVMs in Weka: SMO

Construct Better Features

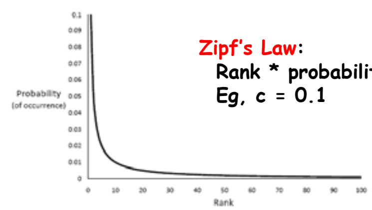
- Key to machine learning is having good features
- In industrial data mining, large effort devoted to constructing appropriate features
- Ideas??

Possible Feature Ideas

- Look at capitalization (may indicated a proper noun)
- Look for commonly occurring sequences
  - E.g. New York, New York City
  - Limit to 2-3 consecutive words
  - Keep all that meet minimum threshold (e.g. occur at least 5 or 10 times in corpus)

Properties of Text

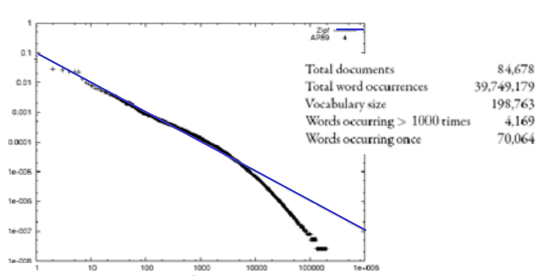
- Word frequencies - skewed distribution
- \`The' and \`of' account for 10% of all words
- Six most common words account for 40%



Zipf's Law:
Rank * probability = c
Eg, c = 0.1

From [Croft, Metzler & Strohman 2010]

Associate Press Corpus \`AP89'



Total documents	84,678
Total word occurrences	39,749,179
Vocabulary size	198,763
Words occurring > 1000 times	4,169
Words occurring once	70,064

From [Croft, Metzler & Strohman 2010]

Middle Ground

- Very common words → bad features
- Language-based **stop list**:
  words that bear little meaning
  20-500 words
  http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words
- Subject-dependent stop lists
- Very rare words ***also*** bad features
  Drop words appearing less than k times / corpus

Stop lists

- Language-based stop list:
  - words that bear little meaning
  - 20-500 words
  - http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words
- Subject-dependent stop lists

Stemming

- Are there different index terms?
  - retrieve, retrieving, retrieval, retrieved, retrieves…
- Stemming algorithm:
  - (retrieve, retrieving, retrieval, retrieved, retrieves) ⇨ retriev
  - Strips prefixes of suffixes (-s, -ed, -ly, -ness)
  - Morphological stemming

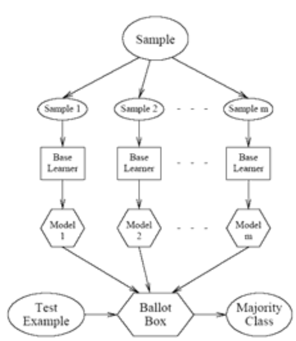
Today's Outline

- Brief supervised learning review
- Evaluation
- Overfitting
- Ensembles
  - Learners: The more the merrier
- Co-Training
  - (Semi) Supervised learning with few labeled training ex

Ensembles of Classifiers

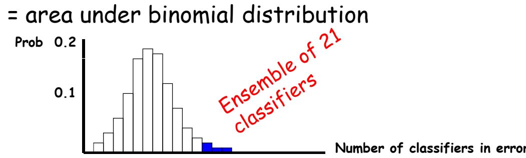
- Traditional approach: Use one classifier
- Alternative approach: Use lots of classifiers
- Approaches:
  - Cross-validated committees
  - Bagging
  - Boosting
  - Stacking

Voting



Ensembles of Classifiers

- Assume
  - Errors are independent (suppose 30% error)
  - Majority vote
- Probability that majority is wrong…
  - = area under binomial distribution

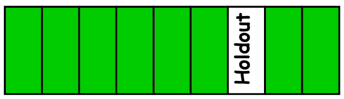


- If individual area is 0.3
- **Area under curve for ≥11 wrong is 0.026**
- Order of magnitude improvement!

Constructing Ensembles

Cross-validated committees

- Partition examples into *k* disjoint equiv classes
- 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
- Now train a classifier on each set



Ensemble Construction II

Bagging

- Generate k sets of training examples
- For each set
  - Draw m examples randomly (with replacement)
  - From the original set of m examples
- Each training set corresponds to
  - 63.2% of original (+ duplicates)
- Now train classifier on each set
- Intuition: Sampling helps algorithm become more robust to noise/outliers in the data

Ensemble Creation III

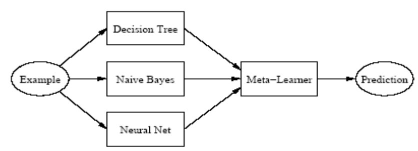
Boosting

- Maintain prob distribution over set of training ex
- Create k sets of training data iteratively:
- On iteration *i*
  - Draw m examples randomly (like bagging)
  - But use probability distribution to bias selection
  - Train classifier number *i* on this training set
  - Test partial ensemble (of *i* classifiers) on all training exs
  - Modify distribution: increase P of each error ex
- Create harder and harder learning problems...
- "Bagging with ***optimized*** choice of examples"

Ensemble Creation IV

Stacking

- Train several base learners
- Next train meta-learner
  - Learns when base learners are right / wrong
  - Now meta learner arbitrates



Train using cross validated committees
- Meta-L inputs = base learner predictions
- Training examples = 'test set' from cross validation

Today's Outline

- Brief supervised learning review
- Evaluation
- Overfitting
- Ensembles
  - Learners: The more the merrier
- Co-Training
  - (Semi) Supervised learning with few labeled training ex

Types of Learning

- **Supervised (inductive) learning**
  - Training data includes desired outputs
- **Semi-supervised learning**
  - Training data includes a ***few*** desired outputs
- **Unsupervised learning**
  - Training data ***doesn't*** include desired outputs
- **Reinforcement learning**
  - Rewards from sequence of actions

Co-Training Motivation

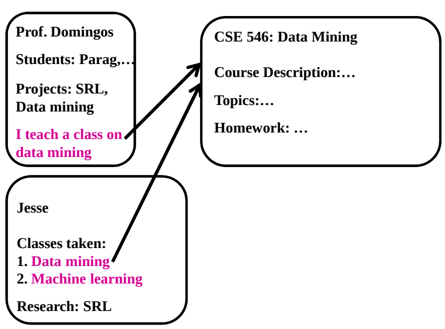
- Learning methods need labeled data
  - Lots of <x, f(x)> pairs
  - Hard to get… (who wants to label data?)
- But unlabeled data is usually plentiful…
  - Could we use this instead??????
- Semi-supervised learning

Co-training

Suppose

- Have *little* labeled data + *lots* of unlabeled
- Each instance has two parts:
  - x = [x1, x2]
  - x1, x2 conditionally independent given f(x)
- Each half can be used to classify instance
  - $\exists$f1, f2 such that f1(x1) ~ f2(x2) ~ f(x)
- Both f1, f2 are learnable
  - f1 $\in$ H1, f2 $\in$ H2, $\exists$ learning algorithms A1, A2

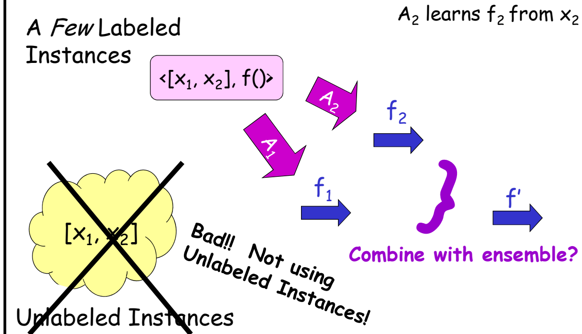
Co-training Example



Without Co-training

$f_1(x_1) \sim f_2(x_2) \sim f(x)$

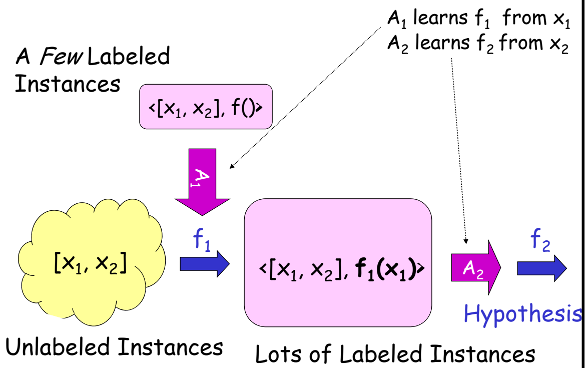
A_1 learns f_1 from x_1
A_2 learns f_2 from x_2



Co-training

$f_1(x_1) \sim f_2(x_2) \sim f(x)$

A_1 learns f_1 from x_1
A_2 learns f_2 from x_2

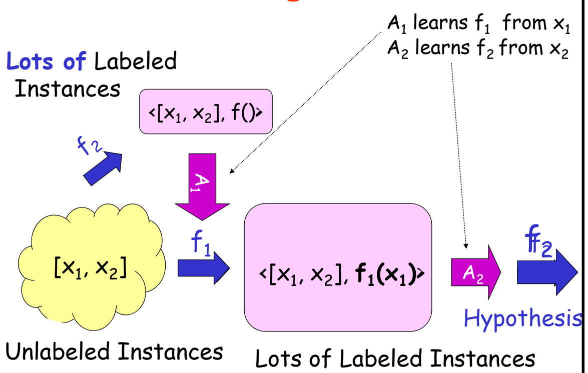


Observations

- Can apply A_1 to generate as much training data as one wants
  - If x_1 is conditionally independent of x_2 / f(x), then the error in the labels produced by A_1
  - ***will look like random noise to A_2 !!!***
- Thus ***no limit*** to quality of the hypothesis A_2 can make

Co-training

$f_1(x_1) \sim f_2(x_2) \sim f(x)$



It really works!

- Learning to classify web pages as course pages
  - x1 = bag of words on a page
  - x2 = bag of words from all anchors pointing to a page
- Naïve Bayes classifiers
  - 12 labeled pages
  - 1039 unlabeled

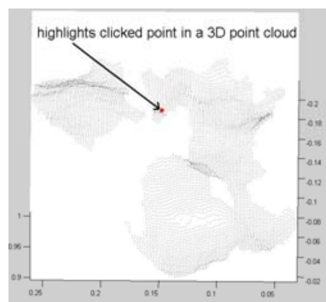
	Page-based classifier	Hyperlink-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

Table 2: Error rate in percent for classifying web pages as course home pages. The top row shows errors when training on only the labeled examples. Bottom row shows errors when co-training, using both labeled and unlabeled examples.

Types of Learning

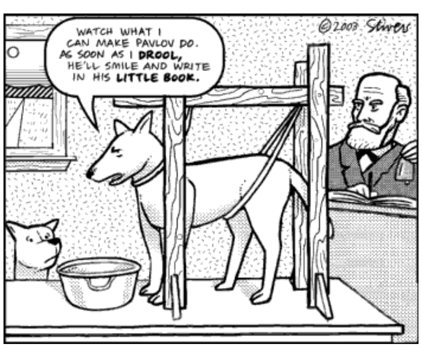
- **Supervised (inductive) learning**
  - Training data includes desired outputs
- **Semi-supervised learning**
  - Training data includes a ***few*** desired outputs
- **Unsupervised learning**
  - Training data ***doesn't*** include desired outputs
- **Reinforcement learning**
  - Rewards from sequence of actions

Learning with Hidden Labels



- Expectation Maximization Algorithm

Reinforcement Learning



Pieter Abeel / Andrew Ng