Ensemble Classifiers

Mausam

(based on slides of Dan Weld)

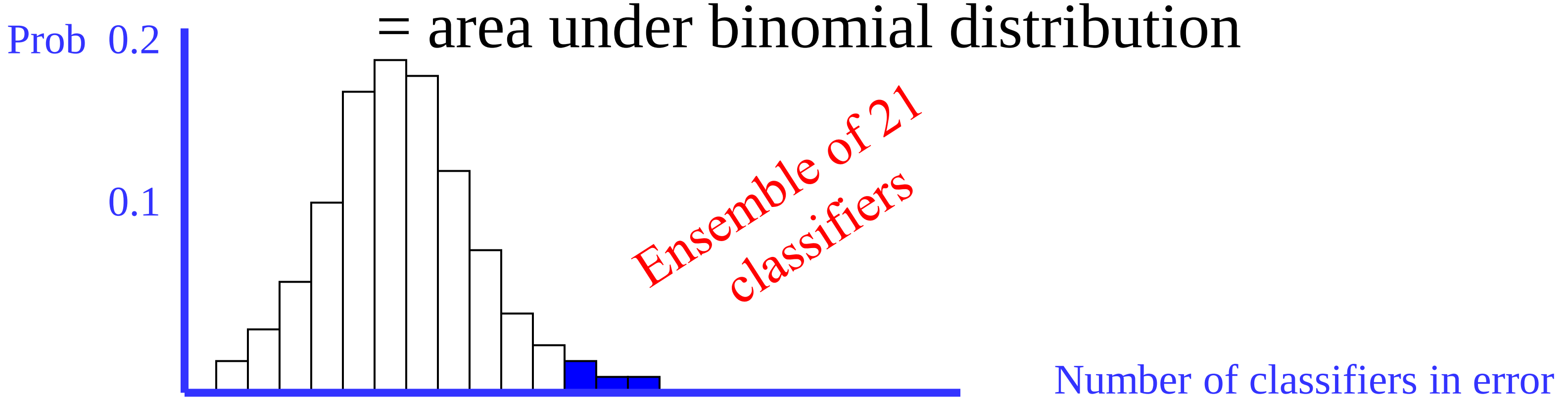
Ensembles of Classifiers

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

Ensembles of Classifiers

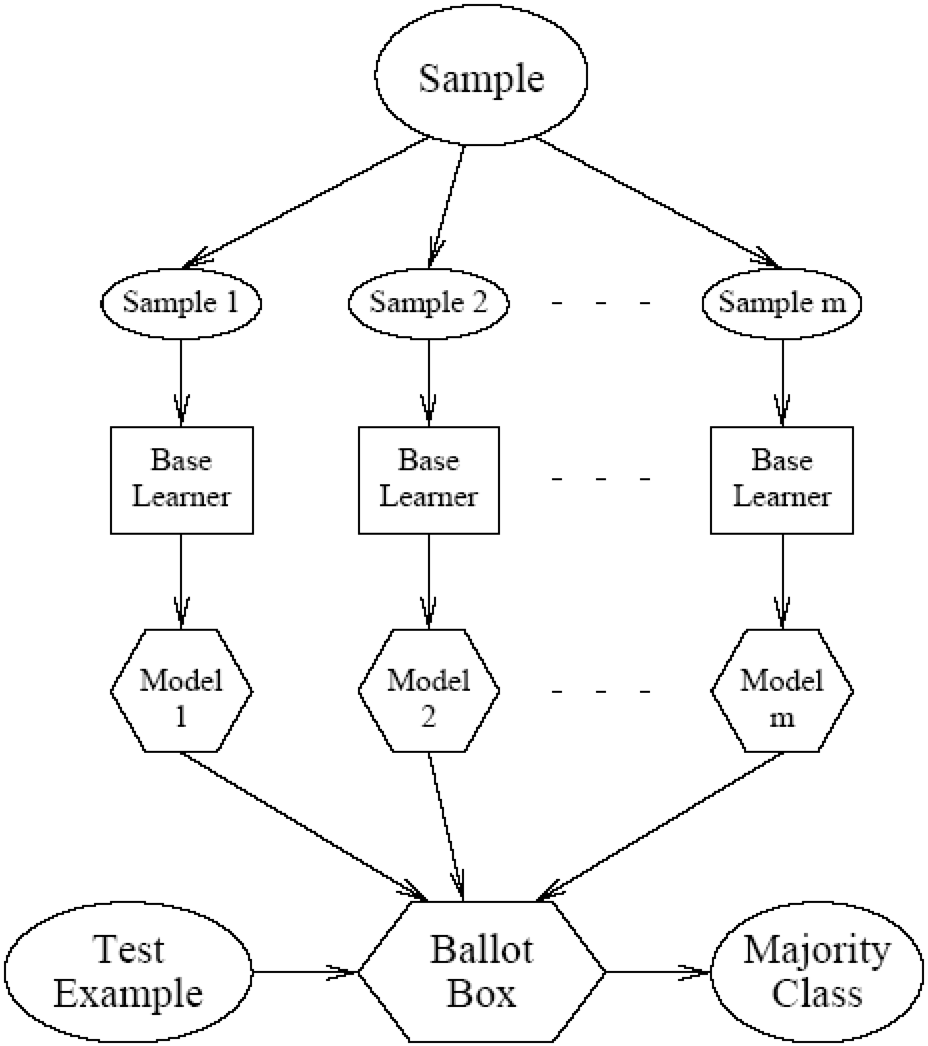
- Assume
  - Errors are independent (suppose 30% error)
  - Majority vote
- Probability that majority is wrong…

= area under binomial distribution



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

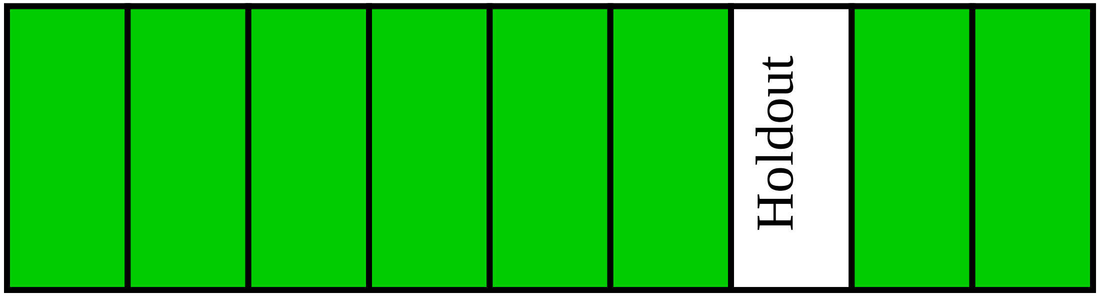
Voting


Sample
Sample 1
Sample 2
- - -
Sample m
Base Learner
Base Learner
- - -
Base Learner
Model 1
Model 2
- - -
Model m
Test Example
Ballot Box
Majority Class

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

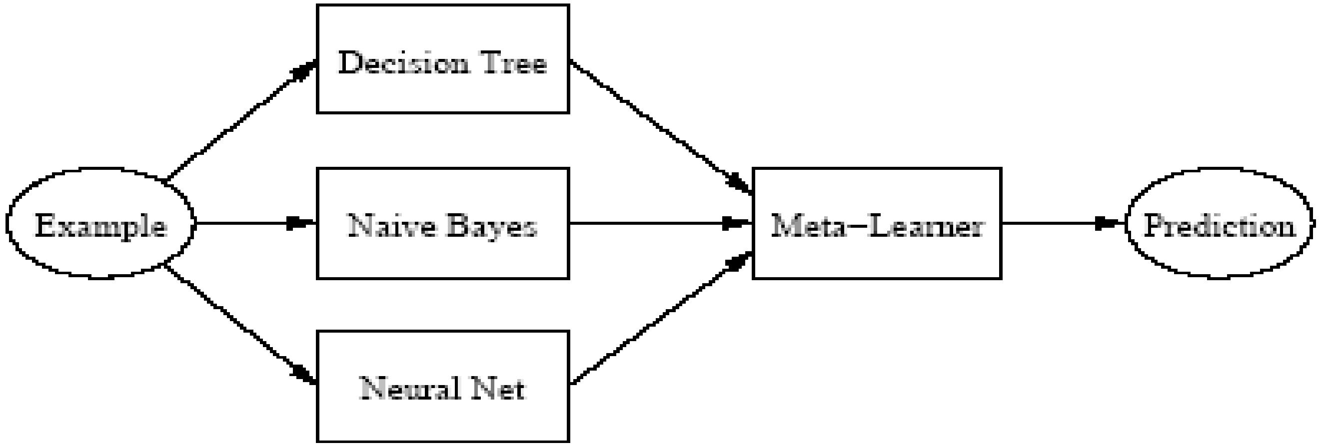
Boosting

- Maintain prob distribution over set of training examples
- 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 example

- 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

Why do ensembles work?

- Statistical
  - Search through hypothesis space
  - average: reduces risk of wrong classifier
- Computational
  - Intractable to get best hypothesis
- Representational
  - Increases the representable hypotheses

Example: Random Forests

- Create k decision trees
- For each decision tree
  - Pick training data as in bagging
  - Randomly sample f features in the data
  - Construct best tree based only on these features
- Voting for final prediction
- Advantages
  - Efficient, highly accurate, thousands of vars

Semi-Supervised Learning

Mausam

(based on slides of Dan Weld,

Oren Etzioni, Tom Mitchell)

Semi-supervised learning 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

Training Data Size

- Machine Translation and speech recognition are quite successful. Why?
- Plenty of labeled data
  - European parliament proceedings
  - Closed-caption broadcasts
- In MT, we have phrase tables
  - Blue bicycle → bicicleta azul
- Side note: this is also a key win for price prediction for Farecast and Zillow.

NLP Challenges

- Document classification
- Named-entity recognition (person, place, or organization?)
- Part-of speech tagging (verb, noun, or adjective?)
- Limited amount of labeled data.
- Labeling is expensive and slow.

Statistical learning methods require LOTS of training data

Can we use all that unlabeled text?

Document Classification: Bag of Words Approach

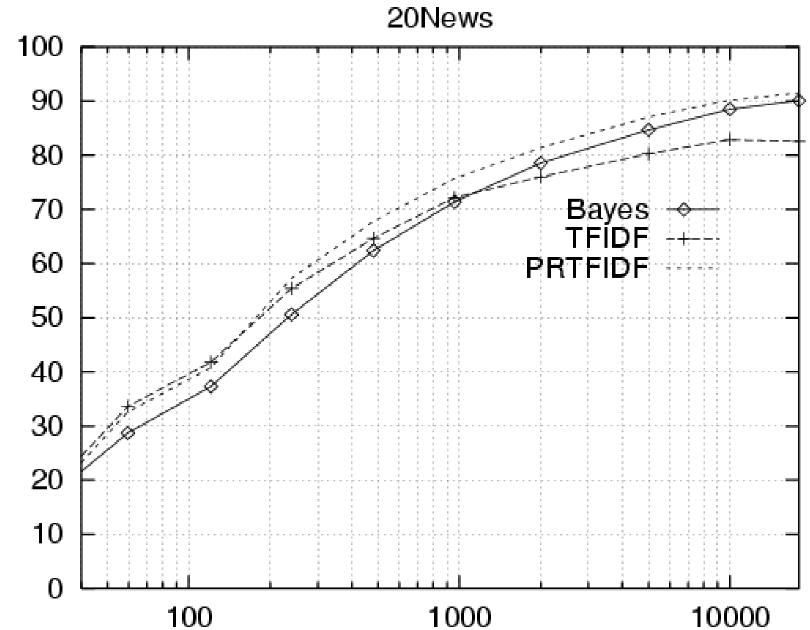
aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

Twenty NewsGroups

Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

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

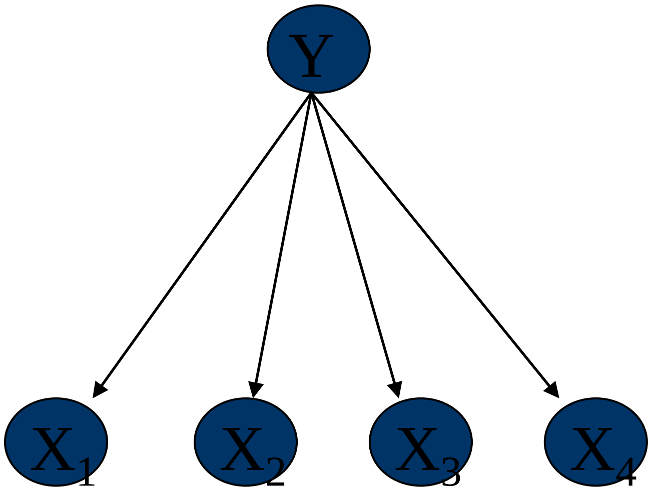
Naive Bayes: 89% classification accuracy



Accuracy vs. # training examples

What if we have labels missing?

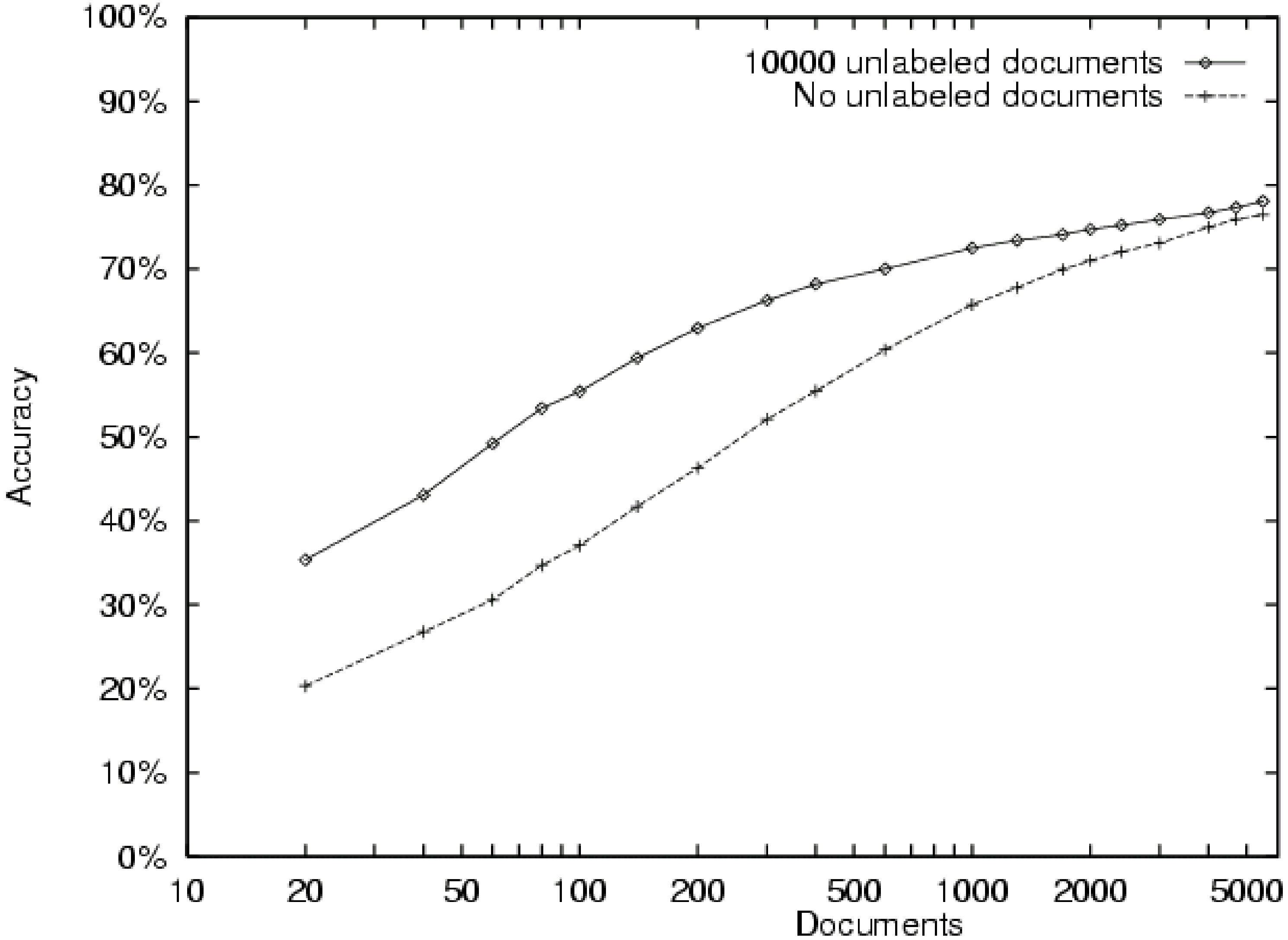
Learn P(Y|X)



Y	X1	X2	X3	X4
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
?	0	1	1	0
?	0	1	0	1

EM Algorithm

20 Newsgroups



Unsupervised Learning: Clustering

- K-means clustering algorithm:
- Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
- Assign each object to the group that has the closest centroid.
- When all objects have been assigned, recalculate the positions of the K centroids.
- Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Co-training

- Have ***little*** labeled data + ***lots*** of unlabeled

- Each instance has two parts:

  $x = [x1, x2]$

  x1, x2 conditionally independent given f(x)

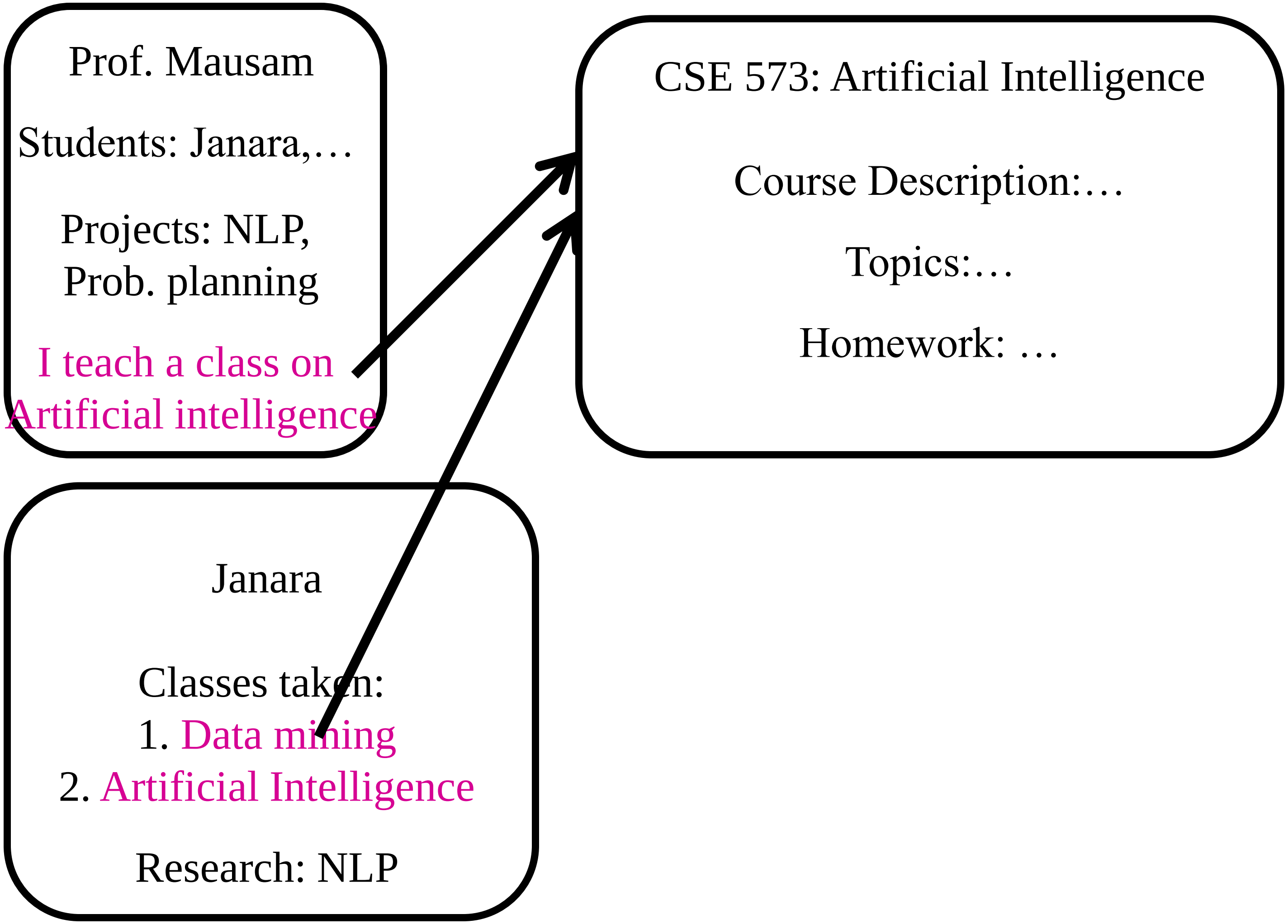
- Each half can be used to classify instance

  $\exists f1, f2$ such that $f1(x1) \sim f2(x2) \sim f(x)$

- Both f1, f2 are learnable

  $f1 \in H1, \quad f2 \in H2, \quad \exists$ learning algorithms A1, A2

Co-training Example



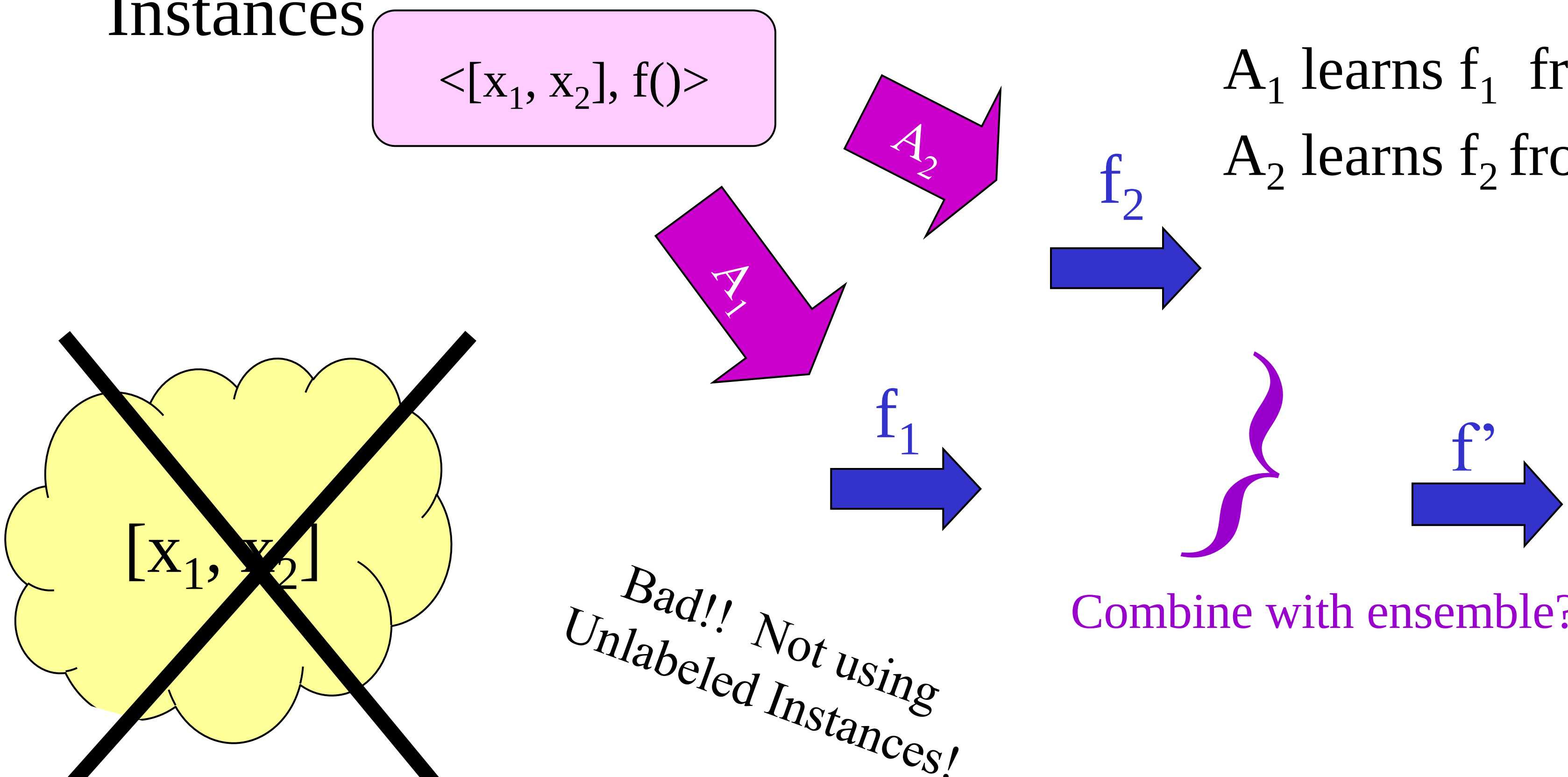
Without Co-training

A *Few* Labeled Instances

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

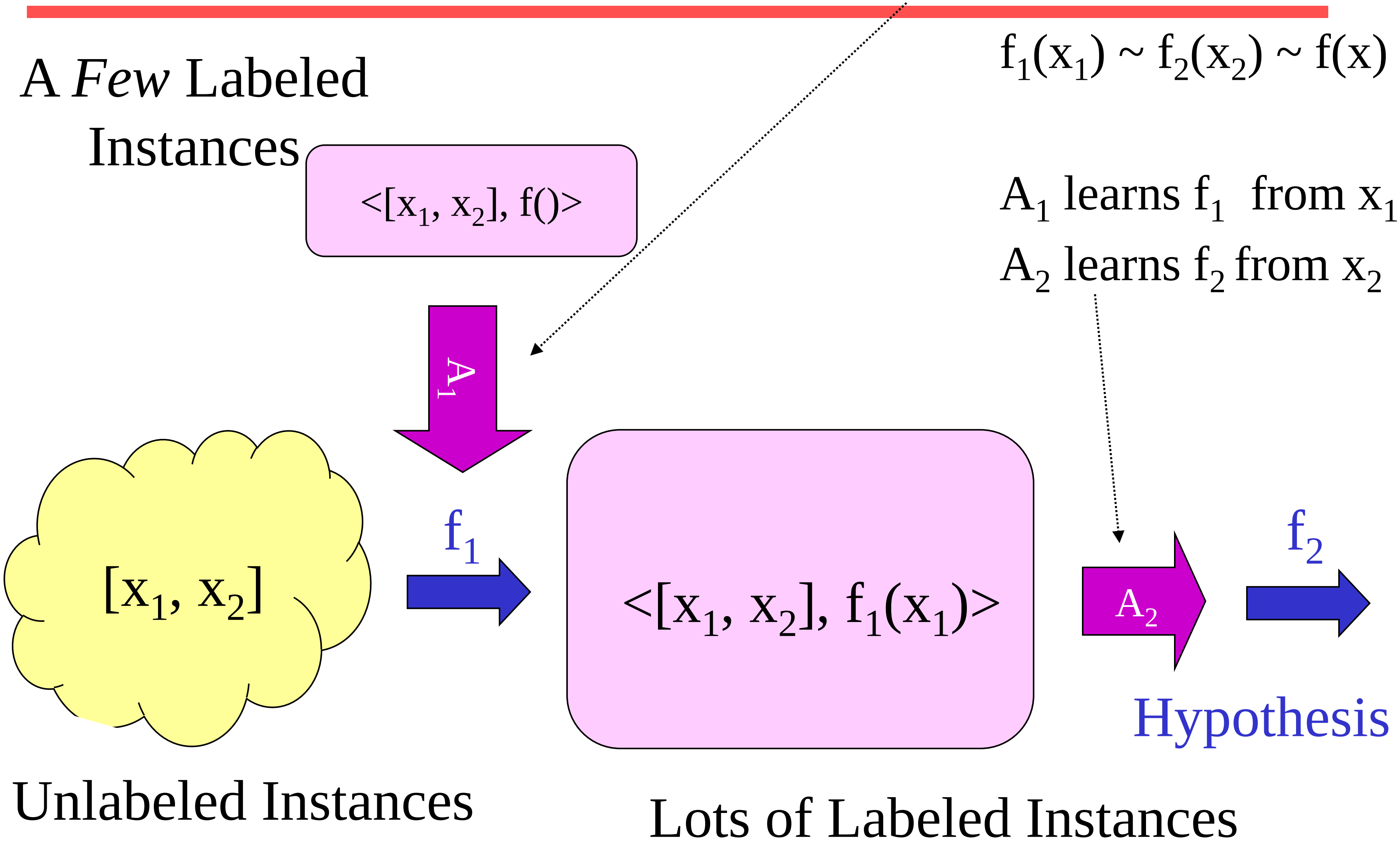
A_1 learns f_1 from x_1

A_2 learns f_2 from x_2



~~Unlabeled Instances~~

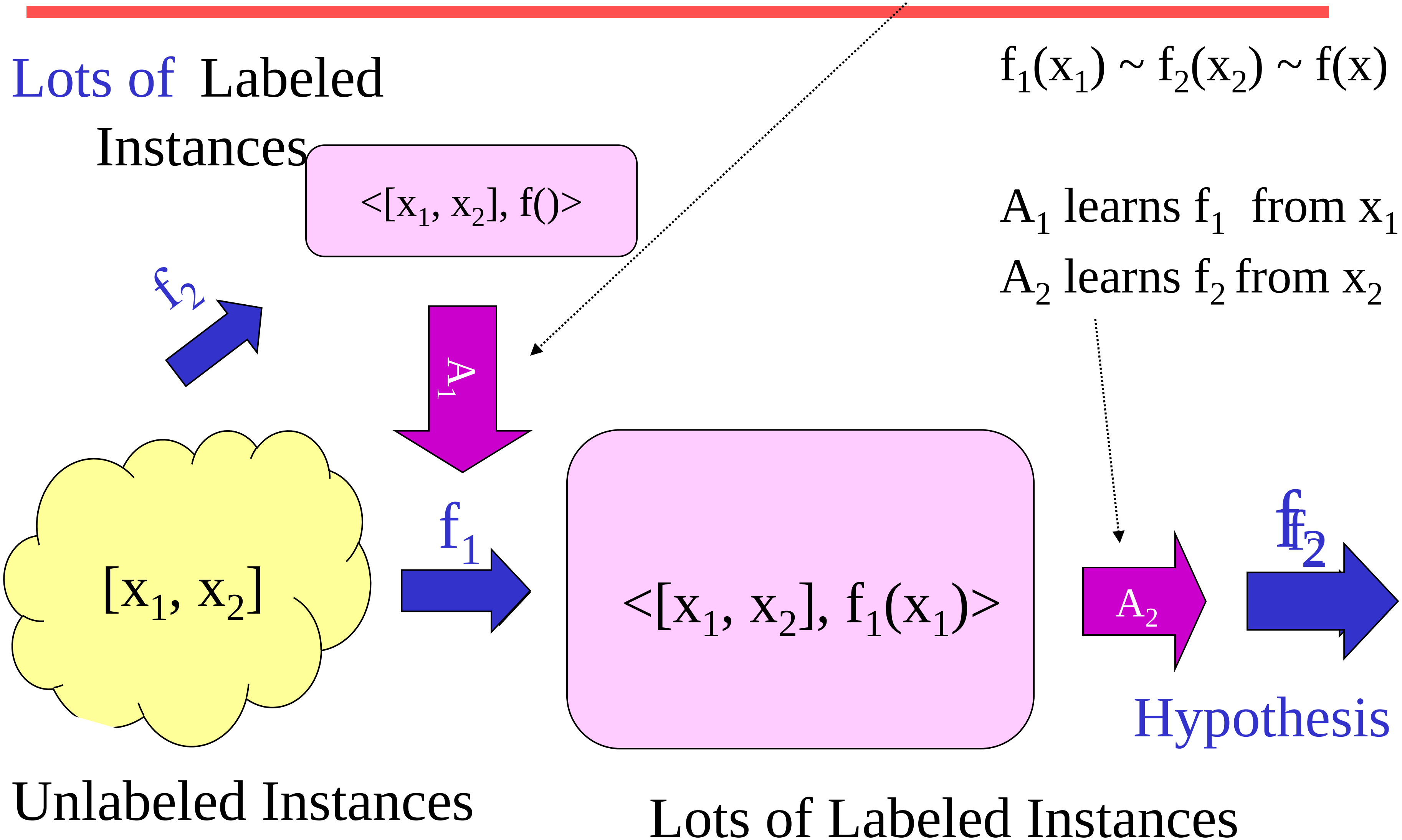
Co-training


$f_1(x_1) \sim f_2(x_2) \sim f(x)$
A *Few* Labeled Instances
$<[x_1, x_2], f()>$
A_1 learns f_1 from x_1
A_2 learns f_2 from x_2
A_1
$[x_1, x_2]$
f_1
$<[x_1, x_2], f_1(x_1)>$
A_2
f_2
Hypothesis
Unlabeled Instances
Lots of Labeled Instances

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


Lots of Labeled Instances
$f_1(x_1) \sim f_2(x_2) \sim f(x)$
$<[x_1, x_2], f()>$
A_1 learns f_1 from x_1
A_2 learns f_2 from x_2
f_2
A_1
f_1
$[x_1, x_2]$
$<[x_1, x_2], f_1(x_1)>$
A_2
f_2
Hypothesis
Unlabeled Instances
Lots of Labeled Instances

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.

Machine Learning Summary

- Right Bias is very important
- Avoid overfitting
  - reduce model complexity (Occam's razor)
  - feature selection
  - regularization
  - termination/tuning based on dev set
- Bias-variance tradeoff
- Feature Engineering
- Ensemble classifiers
- More data superior than complex models
  - even if unlabeled