MACHINE LEARNING BASICS

David Kauchak
CS159 Fall 2014

Admin

Assignment 6
- How'd it go?
- Which option/extension did you pick?

MT lab

Assignment 7
- Out on Thursday
- Due 10/21 (next Friday)

Quiz #3 next Tuesday

Final project

1. Your project should relate to something involving NLP
2. Your project must include a solid experimental evaluation
3. Your project should be in a pair or group of three. If you'd like to do it solo or in a group of four, please come talk to me.

Final project

date	time	description
11/18	in-class	Project proposal presentation
11/20	11:59pm	Project proposal write-up
12/2	2:45pm	Status report
12/10	5pm	Paper draft
12/16	2pm	Final paper, code and presentation

Read the final project handout ASAP!

Start forming groups and thinking about what you want to do

Final project ideas

pick a text classification task
- evaluate different machine learning methods
- implement a machine learning method
- analyze different feature categories

n-gram language modeling
- implement and compare other smoothing techniques
- implement alternative models

parsing
- lexicalized PCFG (with smoothing)
- n-best list generation
- parse output reranking
- implement another parsing approach and compare
- parsing non-traditional domains (e.g. twitter)

EM
- try and implement IBM model 2
- word-level translation models

Final project ideas

- spelling correction
- part of speech tagger
- text chunker
- dialogue generation
- pronoun resolution
- compare word similarity measures (more than the ones we looked at)
- word sense disambiguation
- machine translation
- information retrieval
- information extraction
- question answering
- summarization
- speech recognition

EM

Anybody notice anything at Thursday's colloquium (related to this class)?

The mind-reading game

How good are you at guessing random numbers?

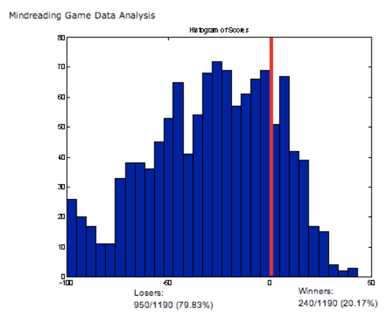
```
Repeat 100 times:
  Computer guesses whether you'll type 0/1
  You type 0 or 1
```

http://seed.ucsd.edu/~mindreader/

[written by Y. Freund and R. Schapire]

The mind-reading game

The computer is right much more than half the time…



The mind-reading game

The computer is right much more than half the time…

Strategy: computer predicts next keystroke based on the last few (maintains weights on different patterns)

There are patterns everywhere… even in "randomness"!

Machine Learning is…

Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data.

Wikipedia
The Free Encyclopedia

Machine Learning is…

Machine learning is programming computers to optimize a performance criterion using example data or past experience.

-- Ethem Alpaydin

The goal of machine learning is to develop methods that can automatically detect patterns in data, and then to use the uncovered patterns to predict future data or other outcomes of interest.

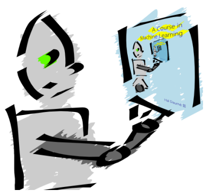
-- Kevin P. Murphy

The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions.

-- Christopher M. Bishop

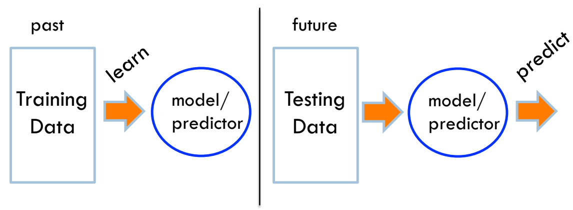
Machine Learning is…

Machine learning is about predicting the future based on the past.
-- Hal Daume III



Machine Learning is…

Machine learning is about predicting the future based on the past.
-- Hal Daume III



Why machine learning?

Lot's of data

Hand-written rules just don't do it

Performance is much better than what people can do

Why not just study machine learning?
- Domain knowledge/expertise is still very important
- What types of features to use
- What models are important

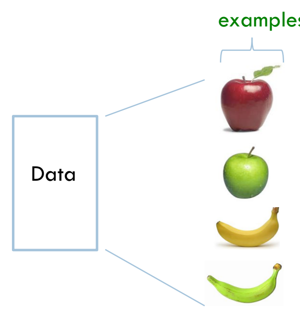
Why machine learning?

Be able to laugh at these signs

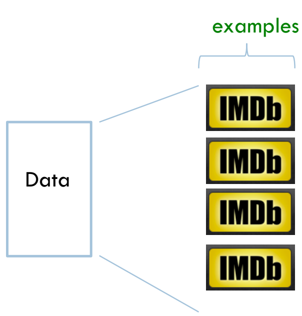
Machine learning problems

What high-level machine learning problems have you seen or heard of before?

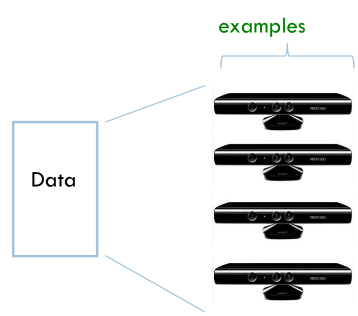
Data

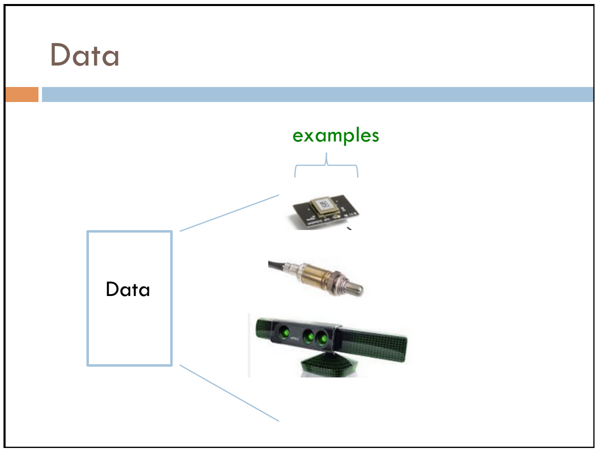


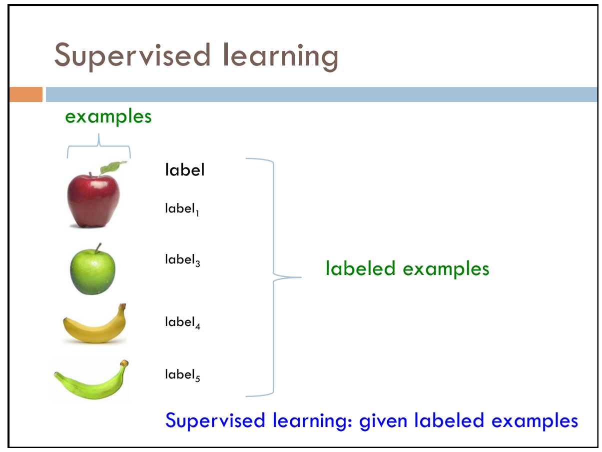
Data

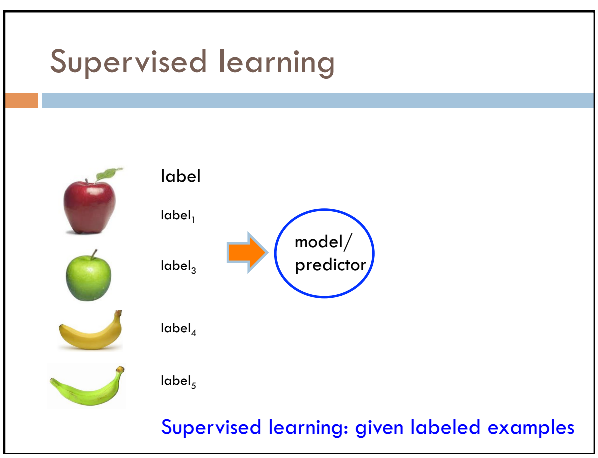


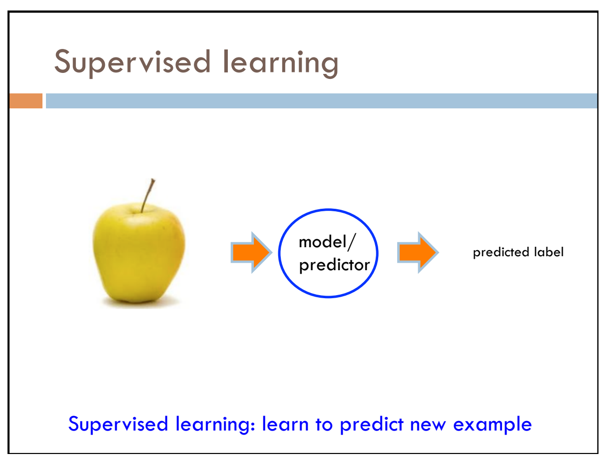
Data




Data
examples
Data


Supervised learning
examples
label
label$_1$
label$_3$
label$_4$
label$_5$
labeled examples
Supervised learning: given labeled examples


Supervised learning
label
label$_1$
label$_3$
label$_4$
label$_5$
model/ predictor
Supervised learning: given labeled examples


Supervised learning
model/ predictor
predicted label
Supervised learning: learn to predict new example

Supervised learning: classification

label

apple

apple

banana

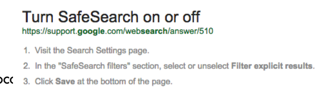
banana

Classification: a finite set of labels

Supervised learning: given labeled examples

NLP classification applications

Document classification
- spam
- sentiment analysis
- topic classification



Turn SafeSearch on or off

https://support.google.com/websearch/answer/510

1. Visit the Search Settings page.
2. In the "SafeSearch filters" section, select or unselect **Filter explicit results**.
3. Click **Save** at the bottom of the page.

Does linguistics phenomena X occ

Digit recognition

Grammatically correct or not?

Word sense disambiguation

Any question you can pose as to have a discrete set of labels/answers!

Supervised learning: regression

label

-4.5

10.1

3.2

4.3

Regression: label is real-valued

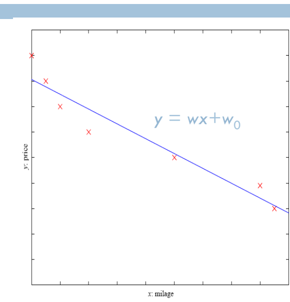
Supervised learning: given labeled examples

Regression Example

Price of a used car

x : car attributes
(e.g. mileage)

y : price



Regression applications

How many clicks will a particular website, ad, etc. get?

Predict the readability level of a document

Predict pause between spoken sentences?

Economics/Finance: predict the value of a stock

Car/plane navigation: angle of the steering wheel, acceleration, …

Temporal trends: weather over time

…

Supervised learning: ranking

label


1


4

Ranking: label is a ranking


2


3

Supervised learning: given labeled examples

NLP Ranking Applications

reranking N-best output lists (e.g. parsing, machine translation, …)

User preference, e.g. Netflix "My List" -- movie queue ranking

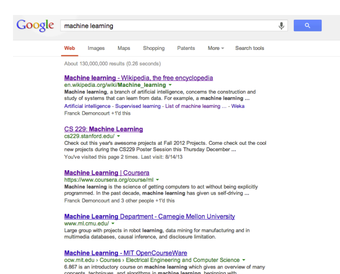
iTunes

flight search (search in general)

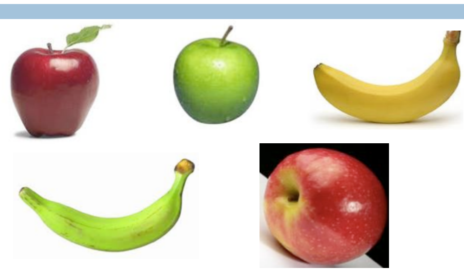
…

Ranking example

Given a query and a set of web pages, rank them according to relevance



Unsupervised learning



Unsupervised learning: given data, i.e. examples, but no labels

Unsupervised learning applications

learn clusters/groups without any label

- cluster documents
- cluster words (synonyms, parts of speech, …)

compression

bioinformatics: learn motifs

…

Reinforcement learning

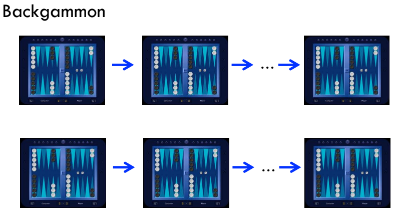
left, right, straight, left, left, left, straight	GOOD
left, straight, straight, left, right, straight, straight	BAD

left, right, straight, left, left, left, straight	18.5
left, straight, straight, left, right, straight, straight	-3

Given a *sequence* of examples/states and a *reward* after completing that sequence, learn to predict the action to take in for an individual example/state

Reinforcement learning example

Backgammon



WIN!

LOSE!

Given sequences of moves and whether or not the player won at the end, learn to make good moves

Reinforcement learning example

http://www.youtube.com/watch?v=VCdxqn0fcnE

Other learning variations

What data is available:

- Supervised, unsupervised, reinforcement learning
- semi-supervised, active learning, …

How are we getting the data:

- online vs. offline learning

Type of model:

- generative vs. discriminative
- parametric vs. non-parametric

Text classification

label

spam

not spam

not spam

For this class, I'm mostly going to focus on classification

I'll use text classification as a running example

Representing examples

examples

What is an example?
How is it represented?

Features

examples

features

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

$f_1, f_2, f_3, \ldots, f_n$

How our algorithms actually "view" the data

Features are the questions we can ask about the examples

Features

examples

features

red, round, leaf, 3oz, …

green, round, no leaf, 4oz, …

yellow, curved, no leaf, 4oz, …

green, curved, no leaf, 5oz, …

How our algorithms actually "view" the data

Features are the questions we can ask about the examples

Text: raw data

Raw data

Features?

Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, "banana, banana, banana"

(1, 1, 1, 0, 0, 1, 0, 0, …)

banana clinton said california across tv wrong capital

Occurrence of words (unigrams)

Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, “banana, banana, banana”

(4, 1, 1, 0, 0, 1, 0, 0, …)

banana clinton said california across tv wrong capital

Frequency of word occurrence (unigram frequency)

Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, “banana, banana, banana”

(1, 1, 1, 0, 0, 1, 0, 0, …)

banana repeatedly clinton said said banana california schools across the tv banana wrong way capital city

Occurrence of bigrams

Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, “banana, banana, banana”

(1, 1, 1, 0, 0, 1, 0, 0, …)

banana repeatedly clinton said said banana california schools across the tv banana wrong way capital city

Other features?

Lots of other features

POS: occurrence, counts, sequence

Constituents

Whether ‘V1agra’ occurred 15 times

Whether ‘banana’ occurred more times than ‘apple’

If the document has a number in it

…

Features are very important, but we’re going to focus on the model

Classification revisited

examples	label
red, round, leaf, 3oz, …	apple
green, round, no leaf, 4oz, …	apple
yellow, curved, no leaf, 4oz, …	banana
green, curved, no leaf, 5oz, …	banana

learn → model/ classifier

During learning/training/induction, learn a model of what distinguishes apples and bananas *based on the features*

Classification revisited

red, round, no leaf, 4oz, … → model/ classifier → predict → Apple or banana?

The model can then classify a new example *based on the features*

Classification revisited

red, round, no leaf, 4oz, … → model/ classifier → predict → Apple

Why?

The model can then classify a new example *based on the features*

Classification revisited

Training data

examples	label
red, round, leaf, 3oz, …	apple
green, round, no leaf, 4oz, …	apple
yellow, curved, no leaf, 4oz, …	banana
green, curved, no leaf, 5oz, …	banana

Test set

red, round, no leaf, 4oz, … ?

Classification revisited

Training data

examples	label
red, round, leaf, 3oz, …	apple
green, round, no leaf, 4oz, …	apple
yellow, curved, no leaf, 4oz, …	banana
green, curved, no leaf, 5oz, …	banana

Test set

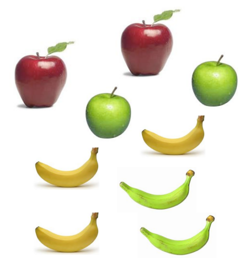
red, round, no leaf, 4oz, … ?

Learning is about ***generalizing*** from the training data

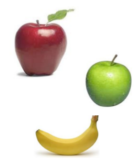
What does this assume about the training and test set?

Past predicts future

Training data

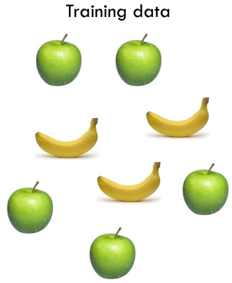


Test set

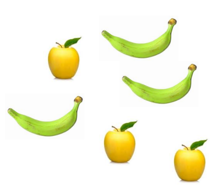


Past predicts future

Training data



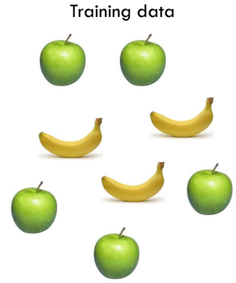
Test set



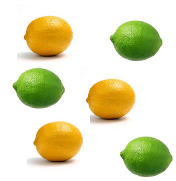
Not always the case, but we'll often assume it is!

Past predicts future

Training data



Test set



Not always the case, but we'll often assume it is!

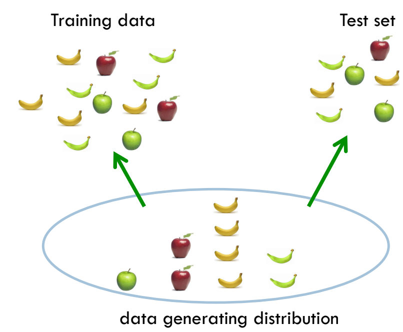
More technically…

We are going to use the *probabilistic model* of learning

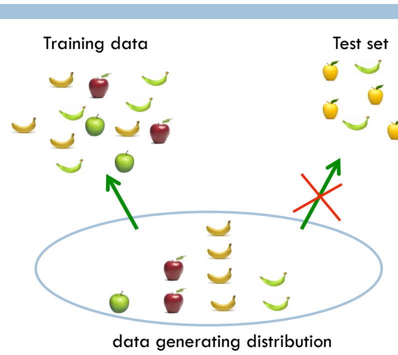
There is some probability distribution over example/label pairs called the *data generating distribution*

Both the training data **and** the test set are generated based on this distribution

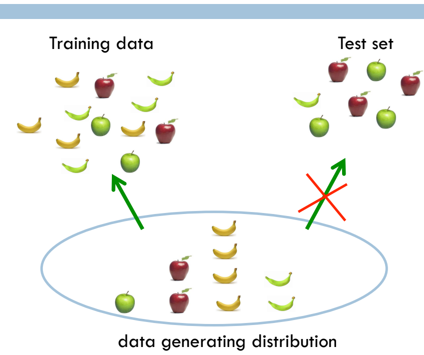
data generating distribution



data generating distribution



data generating distribution



Probabilistic Modeling

training data

train

probabilistic model

Model the data with a probabilistic model

specifically, learn p(*features, label*)

p(*features, label*) tells us how likely these features and this example are

An example: classifying fruit

Training data

examples	label
red, round, leaf, 3oz, …	apple
green, round, no leaf, 4oz, …	apple
yellow, curved, no leaf, 4oz, …	banana
green, curved, no leaf, 5oz, …	banana

train

probabilistic model: p(*features, label*)

Probabilistic models

Probabilistic models define a *probability distribution* over features and labels:

yellow, curved, no leaf, 6oz, banana → probabilistic model: p(*features, label*) → 0.004

Probabilistic model vs. classifier

Probabilistic model:

yellow, curved, no leaf, 6oz, banana → probabilistic model: p(*features, label*) → 0.004

Classifier:

yellow, curved, no leaf, 6oz → probabilistic model: p(*features, label*) → banana

Probabilistic models: classification

Probabilistic models define a *probability distribution* over features and labels:

yellow, curved, no leaf, 6oz, banana → probabilistic model: p(*features, label*) → 0.004

Given an unlabeled example: yellow, curved, no leaf, 6oz predict the label

How do we use a probabilistic model for classification/prediction?

Probabilistic models

Probabilistic models define a *probability distribution* over features and labels:

yellow, curved, no leaf, 6oz, banana → probabilistic model: p(*features, label*) → **0.004**

yellow, curved, no leaf, 6oz, apple → probabilistic model: p(*features, label*) → 0.00002

For each label, ask for the probability under the model
Pick the label with the highest probability

Probabilistic model vs. classifier

Probabilistic model:

yellow, curved, no leaf, 6oz, banana → probabilistic model: p(*features, label*) → 0.004

Classifier:

yellow, curved, no leaf, 6oz → probabilistic model: p(*features, label*) → banana

Why probabilistic models?

Probabilistic models

Probabilities are nice to work with

- range between 0 and 1
- can combine them in a well understood way
- lots of mathematical background/theory

Provide a strong, well-founded groundwork

- Allow us to make clear decisions about things like smoothing
- Tend to be much less "heuristic"
- Models have very clear meanings

Probabilistic models: big questions

1. Which model do we use, i.e. how do we calculate p(*feature, label*)?
2. How do train the model, i.e. how to we we estimate the probabilities for the model?
3. How do we deal with overfitting (i.e. smoothing)?

Basic steps for probabilistic modeling

Step 1: pick a model

Step 2: figure out how to estimate the probabilities for the model

Step 3 (optional): deal with overfitting

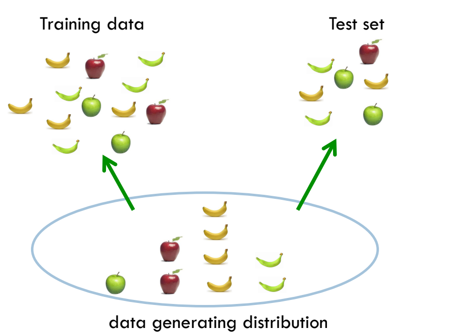
Probabilistic models

Which model do we use, i.e. how do we calculate p(*feature, label*)?

How do train the model, i.e. how to we we estimate the probabilities for the model?

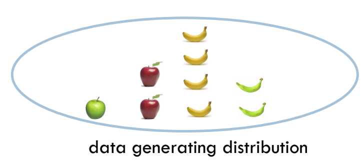
How do we deal with overfitting?

What was the data generating distribution?



Step 1: picking a model

What we're really trying to do is model the data generating distribution, that is how likely the feature/label combinations are



Some maths

$$p(features, label) = p(x_1, x_2, \ldots, x_m, y)$$

$$= p(y)p(x_1, x_2, \ldots, x_m \mid y)$$

What rule?

Some maths

$$p(features, label) = p(x_1, x_2, \ldots, x_m, y)$$

$$= p(y)p(x_1, x_2, \ldots, x_m \mid y)$$

$$= p(y)p(x_1 \mid y)p(x_2, \ldots, x_m \mid y, x_1)$$

$$= p(y)p(x_1 \mid y)p(x_2 \mid y, x_1)p(x_3, \ldots, x_m \mid y, x_1, x_2)$$

$$= p(y)\prod_{j=1}^{m} p(x_i \mid y, x_1, \ldots, x_{i-1})$$

Step 1: pick a model

$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_i \mid y, x_1, \ldots, x_{i-1})$$

So, far we have made NO assumptions about the data

$$p(x_m \mid y, x_1, x_2, \ldots, x_{m-1})$$

How many entries would the probability distribution table have if we tried to represent all possible values and we had 7000 binary features?

Full distribution tables

x_1	x_2	x_3	…	y	p( )
0	0	0	…	0	*
0	0	0	…	1	*
1	0	0	…	0	*
1	0	0	…	1	*
0	1	0	…	0	*
0	1	0	…	1	*
			…		

All possible combination of features!

Table size: 2^{7000} = ?

2^{7000}

1621696755662202026466665085478377095191112430363743256235982084151527023162702352987080237879
4460004651996019099530984538652557892546513204107022110253564658647431585227076599373340842842
7224200122818782600729310826170431944842663920777841250999968601694360066600112098175792966787
8196255237700655294757256678055809293844627218640216108862600816097132874749204352087401101862
6908423275017246052311293955235059054544214554772509509096507889478094683592939574112569473438
6191215296848474344406741204174020887540371869421701550220735398381224299258743537536161041593
4359455766656170179090417259702533652666268202180849389281269970952857089069637557541434487608
8248369941993802415197514510125127043829087280919538476302857811854024099958895964192277601255
3604911562403499947144160905730842429313962119953679373012944795600248333570738998392029910322
3465980389530690429801740098017325210691307971242016963397230218353007589784519525848553710885
8195631737000743805167411189134617501484521767984296782842287373127422122022517597535994839257
0298779077063553347902449354353866605125910795672914312162977887848185522928196541766009803989
9799168140474938421574351580260381151068286406789730483829220346042775765507377656754750702714
4662263487685709621261074762705203049488907208978593689047063428548531668665657327174660658185
6090664849508012761754614572161769555751992117507514067775104496728590822558547771447242334900
7640263217608921135525612411945387026802990440018385850576719369689759366121356888838680023840
9325673807775018914703049621509969838539752071549396339237202875920415172949370790977853625108
3200928396048072379548870695466216880446521124930762900919907177423550391351174415329737479300
8995583051888413533479846411368000499940373724560035428811232632821866113106455077289922996946
9156018580839820741704606832124388152026099584696588161375826382921029547343888832163627122302
9212297953848683554835357106034077891774170263636562027269554375177807413134551018100094688094
0781122057380335371124632958916237089580476224595091825301636909236240671411644331656159828058
37207834398885623908920284409025538 29376

Any problems with this?

Full distribution tables

x_1	x_2	x_3	…	y	p( )
0	0	0	…	0	*
0	0	0	…	1	*
1	0	0	…	0	*
1	0	0	…	1	*
0	1	0	…	0	*
0	1	0	…	1	*
			…		

- Storing a table of that size is impossible!
- How are we supposed to learn/estimate each entry in the table?

Step 1: pick a model

$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_i \mid y, x_1, \ldots, x_{i-1})$$

So, far we have made NO assumptions about the data

Model selection involves making assumptions about the data

We've done this before, n-gram language model, parsing, etc.

These assumptions allow us to represent the data more compactly and to estimate the parameters of the model

Naïve Bayes assumption

$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_i \mid y, x_1, \ldots, x_{i-1})$$

$$p(x_i \mid y, x_1, x_2, \ldots, x_{i-1}) = p(x_i \mid y)$$

What does this assume?

Naïve Bayes assumption

$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_i \mid y, x_1, \ldots, x_{i-1})$$

$$p(x_i \mid y, x_1, x_2, \ldots, x_{i-1}) = p(x_i \mid y)$$

Assumes feature i is independent of the the other features *given the label*

Is this true for text, say, with unigram features?

Naïve Bayes assumption

$$p(x_i \mid y, x_1, x_2, \ldots, x_{i-1}) = p(x_i \mid y)$$

For most applications, this is not true!

For example, the fact that "San" occurs will probably make it *more likely* that "Francisco" occurs

However, this is often a reasonable approximation:

$$p(x_i \mid y, x_1, x_2, \ldots, x_{i-1}) \approx p(x_i \mid y)$$