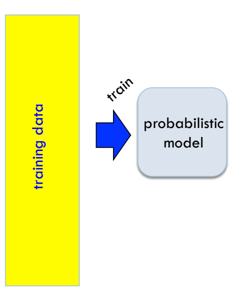
PROBABILISTIC MODELS

David Kauchak
CS451 – Fall 2013

Admin

Assignment 6
- L2 normalization constant should be 2 (not 1)
- Just a handful of changes to the Perceptron code!

Probabilistic Modeling



Model the data with a probabilistic model

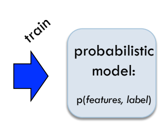
specifically, learn p(*features, label*)

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



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
- an aside: to get the benefit of probabilistic output you can sometimes calibrate the confidence output of a non-probabilistic classifier

Provide a strong, well-founded groundwork
- Allow us to make clear decisions about things like regularization
- Tend to be much less "heuristic" than the models we've seen
- Different models have very clear meanings

Probabilistic models: big questions

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?

Same problems we've been dealing with so far

Probabilistic models	ML in general
Which model do we use, i.e. how do we calculate p(*feature, label*)?	Which model do we use (decision tree, linear model, non-parametric)
How do train the model, i.e. how to we we estimate the probabilities for the model?	How do train the model?
How do we deal with overfitting?	How do we deal with overfitting?

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?

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

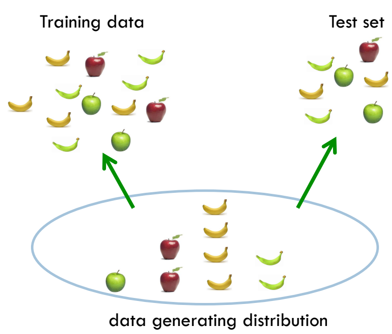
Probabilistic models

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?

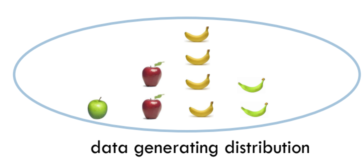
How do we deal with overfitting?

What was the data generating distribution?



Step 1: picking a model

What we're really trying to do is model is 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 (e.g. for the wine data set)?

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	*
			…		

Wine problem:

- all possible combination of features
- ~7000 binary features
- Sample space 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
3720783439888562390892028440902553829376

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 did this before, e.g. assume the data is linearly separable

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

An aside: independence

Two variables are independent if one has nothing whatever to do with the other

For two independent variables, knowing the value of one does not change the probability distribution of the other variable (or the probability of any individual event)

- the result of the toss of a coin is independent of a roll of a dice
- price of tea in England is independent of the whether or not you pass AI

independent or dependent?

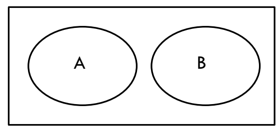
Catching a cold and having cat-allergy

Miles per gallon and driving habits

Height and longevity of life

Independent variables

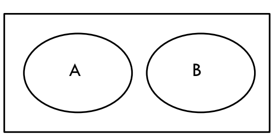
How does independence affect our probability equations/properties?



If A and B are independent (written …)

- P(A,B) = ?
- P(A|B) = ?
- P(B|A) = ?

Independent variables



If A and B are independent (written …)

- P(A,B) = P(A)P(B)
- P(A|B) = P(A)
- P(B|A) = P(B)

How does independence help us?

Independent variables

If A and B are independent

- P(A,B) = P(A)P(B)
- P(A|B) = P(A)
- P(B|A) = P(B)

Reduces the storage requirement for the distributions

Reduces the complexity of the distribution

Reduces the number of probabilities we need to estimate

Conditional Independence

Dependent events can become independent given certain other events

Examples,

- height and length of life
- "correlation" studies
  - size of your lawn and length of life

If A, B are conditionally independent of C

- P(A,B|C) = P(A|C)P(B|C)
- P(A|B,C) = P(A|C)
- P(B|A,C) = P(B|C)
- but P(A,B) ≠ P(A)P(B)

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*

For the wine problem?

Naïve Bayes assumption

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

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

Assumes the probability of a word occurring in a review is independent of the other words *given the label*

For example, the probability of "pinot" occurring is independent of whether or not "wine" occurs given that the review is about "chardonnay"

Is this assumption true?

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 "pinot" occurs will probably make it *more likely* that "noir" occurs (or take a compound phrase like "San Francisco")

However, this is often a reasonable approximation:

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

Naïve Bayes model

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

$$= p(y)\prod_{j=1}^{m} p(x_i \mid y)$$ naïve bayes assumption

$p(x_i \mid y)$ is the probability of a particular feature value given the label

How do we model this?
- for binary features
- for discrete features, i.e. counts
- for real valued features

p(x | y)

Binary features:

$$p(x_i \mid y) = \begin{cases} \theta_i & if\ x_i = 1 \\ 1-\theta_i & otherwise \end{cases}$$ biased coin toss!

Other features:

Could use lookup table for each value, but doesn't generalize well

Better, model as a distribution:
- gaussian (i.e. normal) distribution
- poisson distribution
- multinomial distribution (more on this later)
- …

Basic steps for probabilistic modeling

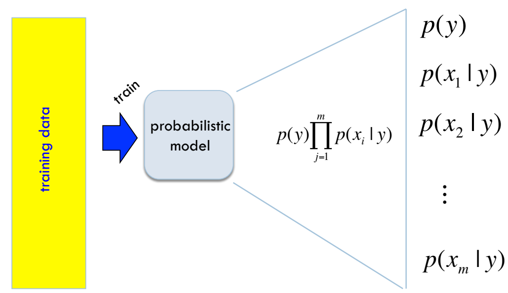
	Probabilistic models
Step 1: pick a model	Which model do we use, i.e. how do we calculate p(*feature, label*)?
Step 2: figure out how to estimate the probabilities for the model	How do train the model, i.e. how to we we estimate the probabilities for the model?
Step 3 (optional): deal with overfitting	How do we deal with overfitting?

Obtaining probabilities

We've talked a lot about probabilities, but not where they come from

- How do we calculate $p(x_i | y)$ from training data?
- What is the probability of surviving the titanic?
- What is that any review is about Pinot Noir?
- What is the probability that a particular review is about Pinot Noir?

Obtaining probabilities



Estimating probabilities

What is the probability of a pinot noir review?

We don't know!

We can ***estimate*** that based on data, though:

$$\frac{\text{number of review labeled pinot noir}}{\text{total number of reviews}}$$

This is called the maximum likelihood estimation. Why?

Maximum Likelihood Estimation (MLE)

Maximum likelihood estimation picks the values for the model parameters that maximize the likelihood of the training data

You flip a coin 100 times. 60 times you get heads.

What is the MLE for heads?

p(head) = 0.60

Maximum Likelihood Estimation (MLE)

Maximum likelihood estimation picks the values for the model parameters that maximize the likelihood of the training data

You flip a coin 100 times. 60 times you get heads.

What is the likelihood of the data under this model (each coin flip is a data point)?

MLE example

You flip a coin 100 times. 60 times you get heads.

MLE for heads: p(head) = 0.60

What is the likelihood of the data under this model (each coin flip is a data point)?

$$likelihood(data) = \prod_i p(x_i)$$

$\log(0.60^{60} * 0.40^{40}) = -67.3$

MLE example

Can we do any better?

$$likelihood(data) = \prod_i p(x_i)$$

p(heads) = 0.5

$\log(0.50^{60} * 0.50^{40}) = -69.3$

p(heads) = 0.7

- $\log(0.70^{60} * 0.30^{40}) = -69.5$