Caltech

Machine Learning & Data Mining

CS/CNS/EE 155

Lecture 10:

Generative Models, Naïve Bayes

Announcements

- Kaggle submissions end 1:59pm Thursday
- HW5 will be released by the end of this week
  - Last year people found it tough, so start early and write good code

Where We Are

- Part 1: Discriminative Models
  - Basics: Bias/Variance, Over/Underfitting
  - Linear models: perceptron, SVM, logistic regression
  - Regularization
  - Decision Trees
  - Model Aggregation: bagging and boosting
  - Deep Learning
- Part 2: Generative and Unsupervised Models **<- We Are Here**
  - Probabilistic generative models
  - Deep generative models
  - Dimensionality Reduction
  - Matrix Factorization, Embeddings, etc.

Note: "Generative" and "Unsupervised" are not synonyms, but generative models are often unsupervised and vice versa.

What's a Generative Model?

Consider a dataset $S = \{(x_i, y_i)\}_{i=1}^{N}$

- Discriminative models learn $P(y \mid x)$
  - Well defined task, but intelligence can't be built on discrimination alone
- Generative models learn $P(x, y)$
  - Less well-defined task, but models can do much more than just discriminate
  - Currently the focus of a significant percent of high-profile ML research

What can Generative Models Do?

Example model:

y	x	P(x,y)
SPAM	Help!	0.15
NOT	Help!	0.1
SPAM	Homework	0.05
NOT	Homework	0.45
SPAM	Winner!	0.2
NOT	Winner!	0.05

Our model specifies these.
Don't worry yet about the model architecture or how we trained it.

Full joint distribution; sums to 1

What can Generative Models Do?

Example model:

y	x	P(x,y)
SPAM	Help!	0.15
NOT	Help!	0.1
SPAM	Homework	0.05
NOT	Homework	0.45
SPAM	Winner!	0.2
NOT	Winner!	0.05

<u>Discriminate</u>

We can compute

$$P(y \mid x) = \frac{P(x,y)}{P(x)} \qquad P(x) = \sum_y P(y,x)$$

So know, e.g., P(y=SPAM|x=Help!):

P(y=SPAM, x=Help!) = 0.15

P(x=Help!) = 0.25

=> P(y=SPAM|x=Help!) = 0.6

What can Generative Models Do?

Example model:

y	x	P(x,y)
SPAM	Help!	0.15
NOT	Help!	0.1
SPAM	Homework	0.05
NOT	Homework	0.45
SPAM	Winner!	0.2
NOT	Winner!	0.05

Summarize and Predict

E.g. the marginal distribution of y is

$$P(y) = \sum_x P(y, x)$$

So we know P(y=SPAM) = 0.4 before we've even seen x.

You can't get that from a discriminative model.

E.g. Variational Autoencoders

Can discover structure in, e.g., the MNIST dataset using just the unlabeled images:

This is a “walk” through the structure this generative model discovered.

http://blog.fastforwardlabs.com/2016/08/22/under-the-hood-of-the-variational-autoencoder-in.html

E.g. Latent Dirichlet Allocation

(Disclaimer: this was the top result of Google Search "LDA example")

Can identify topics from reading unlabeled text corpus, e.g. Presidential Campaign, Wildlife:

We understand that you have been discussed as a possible choice for the **Vice Presidency**.

As **people** who **support** the democratic process and care about protecting our **wildlife** for future generations, we want **you** to know that we don't believe **people** in our states would vote for **you** for any office if they knew your record on these issues.

It is troubling that **you** are **now** working to deny more than 50,000 Alaskans a vote on **aerial** killing of **wolves** and **bears** with legislation now **being** considered in the Alaska legislature.

http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/

What can Generative Models Do?

Example model:

y	x	P(x,y)
SPAM	Help!	0.15
NOT	Help!	0.1
SPAM	Homework	0.05
NOT	Homework	0.45
SPAM	Winner!	0.2
NOT	Winner!	0.05

Generate

We can generate examples straight from the data distribution:

```
>>>random.uniform(0,1)
0.404...
>>>return (NOT, Homework)
```

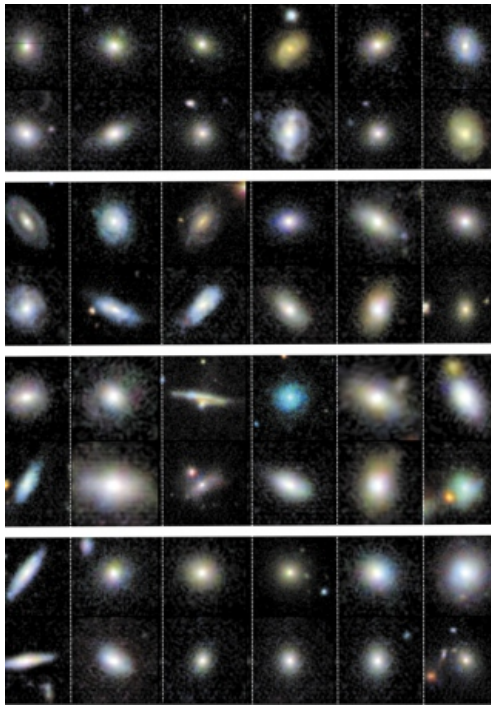
Or we can generate conditionally, since going from label to data is as easy as from data to label:

$$P(x \mid y) = \frac{P(x,y)}{P(y)} \qquad P(y) = \sum_x P(y,x)$$

So know, e.g., P(x=Help!|y=SPAM), P(x=Homework|y=SPAM), and P(x=Winner!|y=SPAM) and can sample from that conditional distribution.

E.g. Generative Adversarial Networks

Learn from unlabeled telescope imagery,
generate new images,
now astronomers have more data



Learn from labeled images,
condition on “Volcano” and generate new images, because why not?

http://www.nature.com/news/astronomers-explore-uses-for-ai-generated-images-1.21398?WT.mc_id=FBK_NatureNews

E.g. Style Transfer

Condition on a style and a content image, generate new images:

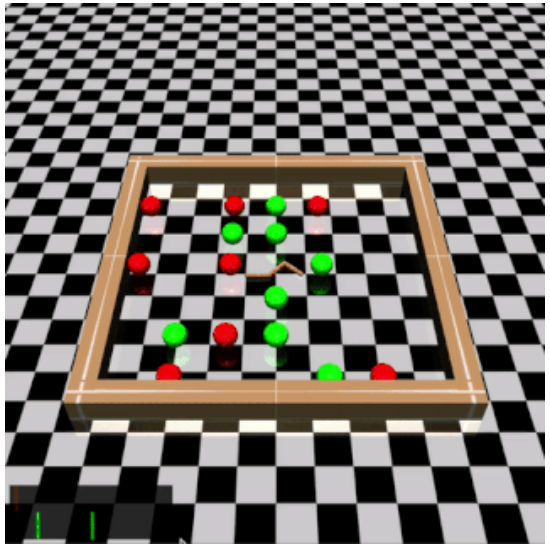
http://gitxiv.com/posts/jG46ukGod8R7Rdtud/a-neural-algorithm-of-artistic-style

What can Generative Models Do?

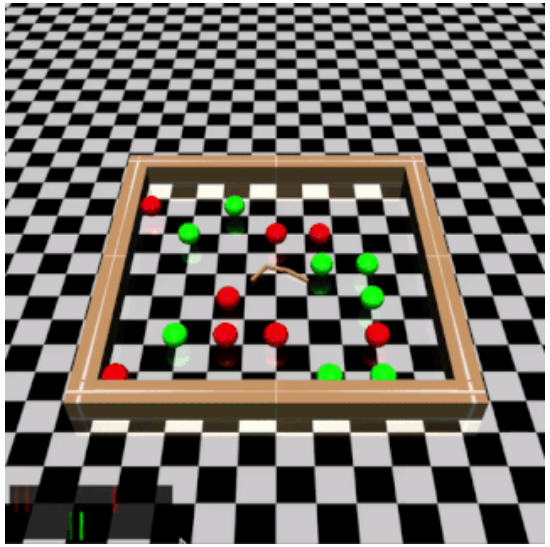
Handle Missing Values and Model Uncertainty

(Previous example was too simple to illustrate this.)

By conditioning on known values, generative models can handle missing values probabilistically. Further, many generative models are structured to model uncertainty about their missing value estimates.



Naïve exploration



Exploration with a generative uncertainty model

https://openai.com/blog/generative-models/

What's the Catch?

- Discriminative models make (much) better predictions
  - Discriminative models are directly optimized to predict
  - Generative models make predictions by combining multiple estimated values

$$P(y \mid x) = \frac{P(x,y)}{P(x)} \qquad P(x) = \sum_{y} P(y,x)$$

What's the Catch?

- Generative modeling is an underspecified task
  - Goal of discriminative models is clear: improve accuracy
  - Goal of generative models is less clear: the quality of a model is task-dependent and somewhat subjective.

What's the Catch?

- Generative models are more complicated
  - In our example, the generative model had 6 parameters; a discriminative model would have needed 3 to do its job
  - Imagine what happens when feature space gets large, e.g. 784 continuous-valued pixels!

$$P(y \mid x) = \frac{P(x, y)}{P(x)}$$

What if there are so many different x that P(x) underflows?

  - Requires very clever model design and assumptions about data

Example

Suppose we have a binary y label and binary x labels.

What's wrong with the *Probability Table* approach (arguably the most naïve generative model)?

y	x^1=Winner!	x^2=Homework	P(x,y)
SPAM	1	1	0.01
NOT	1	1	0.01
SPAM	0	1	0.03
NOT	0	1	0.35
SPAM	1	0	0.25
NOT	1	0	0.05
SPAM	0	0	0.2
NOT	0	0	0.1

Example

Suppose we have a binary y label and binary x labels.

What's wrong with the *Probability Table* approach (arguably the most naïve generative model)?

y	x^1=Winner!	x^2=Homework	P(x,y)
SPAM	1	1	0.01
NOT	1	1	0.01
SPAM	0	1	0.03
NOT	0	1	0.35
SPAM	1	0	0.25
NOT	1	0	0.05

Model Complexity is Exponential w.r.t. the length of x!

We need a better model…

Naïve Bayes

Reminder: Conditional Independence

x^1 and x^2 are *conditionally independent given y* iff

$$P(x^1, x^2 \mid y) = P(x^1 \mid y) P(x^2 \mid y)$$

or, equivalently

$$P(x^1 \mid x^2, y) = P(x^1 \mid y)$$

and

$$P(x^2 \mid x^1, y) = P(x^2 \mid y)$$

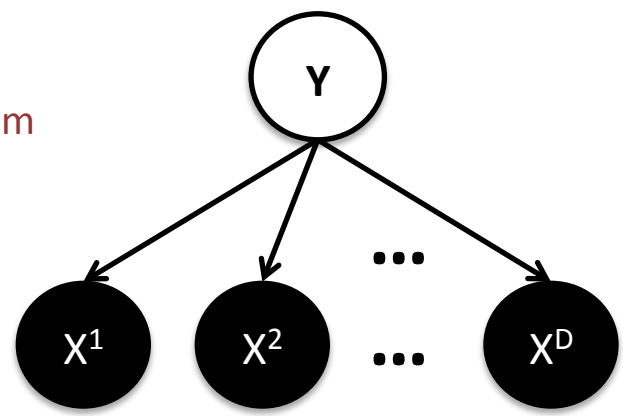
Reminder: Conditional Independence

If D random variables x^d are all conditionally independent given y, then

$$P(x^1,\ldots,x^D,y) = P(x^1,\ldots,x^D \mid y)P(y) = P(y)\prod_{d=1}^{D} P(x^d \mid y)$$

We can express this graphically:

Roughly speaking, an edge from A to B means B depends on A. Absence of edge means conditional independence.



But graphical models are a world unto themselves, and we won't get into them too deeply…

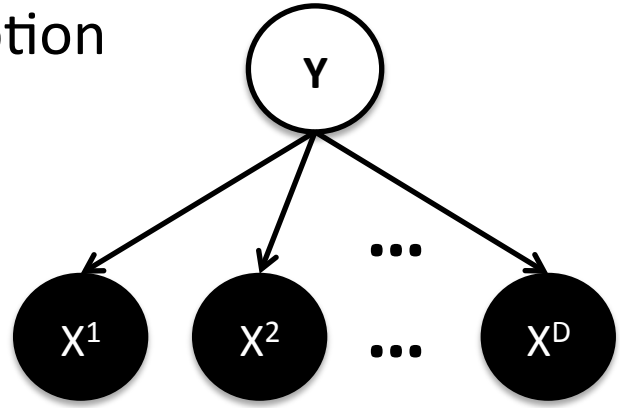
"Bayesian Network", a type of Graphical Model Diagram

The Naïve Bayes Model

- A generative model, but used in the context of prediction.
- For a label y and D features x^d, Naïve Bayes posits that all the features are conditionally independent given y.

$$P(x^1,\ldots,x^D,y) = P(x^1,\ldots,x^D \mid y)P(y) = P(y)\prod_{d=1}^{D} P(x^d \mid y)$$

- This is a strong (perhaps naïve) assumption about our data
- But we only have to keep track of P(y) and $P(x^d|y)$!



Why is Naïve Bayes Convenient?

- Compact representation
- Easy to compute any quantity
  - $P(y|x)$, $P(x^d|y)$, …
- Easy to estimate model components
  - $P(y)$, $P(x^d|y)$
- Easy to sample
- Easy to deal with missing values

Example Model (Discrete)

- Each x^d binary (-1 or +1)
  - E.g., presence (+1) or absence (-1) of word

P(x|y)

	x^1=Homework	x^2=Winner!
y=SPAM	$P(x^1=1\|y)=0.2$	$P(x^2=1\|y)=0.5$
y=NOT	$P(x^1=1\|y)=0.6$	$P(x^2=1\|y)=0.1$

Model Complexity is Linear w.r.t. the length of x!

P(y)

	P(y)
y=SPAM	0.7
y=NOT	0.3

Making Predictions

$$P(y \mid x) = \frac{P(x,y)}{P(x)}$$

$$= \frac{P(x \mid y)P(y)}{P(x)}$$

Bayes Rule (hence the name)

$$= \frac{P(y)}{P(x)} \prod_d P(x^d \mid y)$$

$$\propto P(y) \prod_d P(x^d \mid y)$$

Model components we keep track of.

	x^1=Homework	x^2=Winner!
y=SPAM	$P(x^1=1 \mid y)=0.2$	$P(x^2=1 \mid y)=0.5$
y=NOT	$P(x^1=1 \mid y)=0.6$	$P(x^2=1 \mid y)=0.1$

	P(y)
y=SPAM	0.7
y=NOT	0.3

Example Prediction

	x^1=Homework	x^2=Winner!
y=SPAM	$P(x^1=1\|y)=0.2$	$P(x^2=1\|y)=0.5$
y=NOT	$P(x^1=1\|y)=0.6$	$P(x^2=1\|y)=0.1$

	P(y)
y=SPAM	0.7
y=NOT	0.3

$$P(y = NOT \mid x^1 = 1, x^2 = -1)$$

$$= \frac{P(y = NOT)P(x^1 = 1 \mid y = NOT)P(x^2 = -1 \mid y = NOT)}{P(x^1 = 1, x^2 = -1)}$$

$$= \frac{P(y = NOT)P(x^1 = 1 \mid y = NOT)P(x^2 = -1 \mid y = NOT)}{\sum_y P(y)P(x^1 = 1, x^2 = -1 \mid y)}$$

$$= \frac{0.3 * 0.6 * (1 - 0.1)}{0.3 * 0.6 * (1 - 0.1) + 0.7 * 0.2 * (1 - 0.5)} = 0.698...$$

One Empirical Comparison

MODEL	1ST	2ND	3RD	4TH	5TH	6TH	7TH	8TH	9TH	10TH
BST-DT	0.580	0.228	0.160	0.023	0.009	0.000	0.000	0.000	0.000	0.000
RF	0.390	0.525	0.084	0.001	0.000	0.000	0.000	0.000	0.000	0.000
BAG-DT	0.030	0.232	0.571	0.150	0.017	0.000	0.000	0.000	0.000	0.000
SVM	0.000	0.008	0.148	0.574	0.240	0.029	0.001	0.000	0.000	0.000
ANN	0.000	0.007	0.035	0.230	0.606	0.122	0.000	0.000	0.000	0.000
KNN	0.000	0.000	0.000	0.009	0.114	0.592	0.245	0.038	0.002	0.000
BST-STMP	0.000	0.000	0.002	0.013	0.014	0.257	0.710	0.004	0.000	0.000
DT	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.616	0.291	0.089
LOGREG	0.000	0.000	0.000	0.000	0.000	0.000	0.040	0.312	0.423	0.225
NB	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.030	0.284	0.686

- Columns denote how frequently each model places 1st, 2nd, 3rd, etc.
- Only generative model (Naïve Bayes) is in last place

"An Empirical Comparison of Supervised Learning Algorithms"
Caruana, Niculescu-Mizil, ICML 2006

Example Prediction #2

- What if we want to compute $P(x^1 \mid x^{2:D}, y)$?

- It's an explicitly defined model component:

$$P(x^1 \mid x^{2:D}, y) = P(x^1 \mid y)$$

Example Prediction #3

- What if we want to compute $P(x^1 \mid x^{2:D})$?

$$P(x^1 \mid x^{2:D}) = \frac{P(x^1, x^{2:D})}{P(x^{2:D})} = \frac{\sum_y P(y) P(x^1, x^{2:D} \mid y)}{\sum_y P(y) P(x^{2:D} \mid y)}$$

"Marginalizing out the y"

$$= \frac{\sum_y P(y) \prod_{d=1}^{D} P(x^d \mid y)}{\sum_y P(y) \prod_{d=2}^{D} P(x^d \mid y)}$$

Why is the numerator smaller than the denominator?

Marginalization in Matrix Form

For heaven's sake, don't use a for loop

O

	x^1=Homework	x^2=Winner!
y=SPAM	$P(x^1=1\|y)=0.2$	$P(x^2=1\|y)=0.5$
y=NOT	$P(x^1=1\|y)=0.6$	$P(x^2=1\|y)=0.1$

P

	P(y)
y=SPAM	0.7
y=NOT	0.3

- Compute $P(x^d=1)$:

$$P(x^d = 1) = \left[O^T P\right]_d$$

d-th row

$$P(x^d = 1) = \sum_y P(x^d = 1 \mid y) P(y)$$

Prediction with Missing Values

- What if we don't observe $x^{2:D}$?
- Predict $P(y=\text{SPAM}|x^1)$

We can marginalize out the missing values!

$$P(y \mid x^1) = \sum_{x^{2:D}} P(y, x^{2:D} \mid x^1) = \sum_{x^{2:D}} \frac{P(x^{1:D}, y)}{P(x^1)}$$

How to efficiently sum over multiple missing values?

Conditional Independence to the Rescue!

$$P(y \mid x^1) = \sum_{x^{2:D}} P(y, x^{2:D} \mid x^1) = \sum_{x^{2:D}} \frac{P(x^{1:D}, y)}{P(x^1)}$$

From previous slide

$$P(x^{1:D}, y) = P(y) \prod_{d=1}^{D} P(x^d \mid y)$$

Definition of Naïve Bayes

$$\sum_{x^{2:D}} P(x^d, y) = P(y) \sum_{x^{2:D}} \prod_d P(x^d \mid y)$$

$$= P(y) P(x^1 \mid y) \prod_{d \in [2,D]} \sum_{x^d} P(x^d \mid y)$$

Swap Product & Sum due to independence!

$$= P(y) P(x^1 \mid y)$$

Marginalizes to 1!

Training

- Maximum Likelihood of Training Set: $S = \{(x_i, y_i)\}_{i=1}^{N}$

$$\arg\max_P P(S) = \arg\max_P \prod_i P(x_i, y_i)$$

$$= \arg\min_P \sum_i -\log P(x_i, y_i)$$

where the argmax/min is over all possible Naïve Bayes models.

Training

- Subject to Naïve Bayes assumption on structure of P(x,y), we only need to estimate P(y) and each each $P(x^d|y)$!
- This is just counting!

$$P(y = SPAM) = \frac{N_{y=SPAM}}{N}$$

Frequency of SPAM documents in training set

$$P(x^1 = 1 \mid y = SPAM) = \frac{N_{y=SPAM, x^1=1}}{N_{y=SPAM}}$$

Frequency of word x1 appearing in SPAM documents in training set

Training Derivation

- Define: $$P(x,y) = \frac{w_{x,y}}{\sum_{x',y'} w_{x',y'}}$$

Just a re-parameterization

$$\underset{P}{\operatorname{argmin}} \sum_i -\log P(x_i, y_i) = \underset{w}{\operatorname{argmin}} \sum_i \left[-\log w_{x_i,y_i} + \log \sum_{x',y'} w_{x',y'} \right]$$

training examples (x,y)

$$\partial_{w_{x,y}} = -\frac{N_{x,y}}{w_{x,y}} + \frac{N}{\sum_{x',y'} w_{x',y'}} \Rightarrow \frac{N_{x,y}}{N} = \frac{w_{x,y}}{\sum_{x',y'} w_{x',y'}} \Rightarrow P(x,y) = \frac{N_{x,y}}{N}$$

Frequency of (x,y) in training set!

Regularization

- Add "pseudo counts"
  - i.e. hallucinate some data

$$P(y = SPAM) = \frac{N_{y=SPAM} + \lambda P_{y=SPAM}}{N + \lambda}$$

Often just set pseudo counts to uniform distribution

$$P(x^1 = 1 \mid y = SPAM) = \frac{N_{y=SPAM \wedge x^1=1} + \lambda P_{y=SPAM \wedge x^1=1}}{N_{y=SPAM} + \lambda}$$

Sampling

- Can sample from distribution P(x,y)
  - First sample y:
    - Random uniform variable R
    - Set y=SPAM if R < P(y=SPAM) & y=NOT otherwise
  - Then sample each x^d:
    - Sample uniform variable R
    - Set x^d=1 if R < P(x^d=1|y) & x^d=0 otherwise

Sampling Example

- Sample P(y)
  - R = 0.5, so set y = SPAM
- Sample $P(x^1|y=SPAM)$
  - R = 0.1, so set $x^1 = 1$
- Sample $P(x^2|y=SPAM)$
  - R = 0.9, so set $x^2 = 0$

Can be done in either order

	x^1=Homework	x^2=Winner!
y=SPAM	$P(x^1=1\|y)=0.2$	$P(x^2=1\|y)=0.5$
y=NOT	$P(x^1=1\|y)=0.6$	$P(x^2=1\|y)=0.1$

	P(y)
y=SPAM	0.7
y=NOT	0.3

Sampling Example #2

- Sample P(y)
  - R = 0.9, so set y = NOT
- Sample $P(x^1|y=NOT)$
  - R = 0.5, so set $x^1 = 1$
- Sample $P(x^2|y=NOT)$
  - R = 0.05, so set $x^2 = 1$

	x^1=Homework	x^2=Winner!
y=SPAM	$P(x^1=1\|y)=0.2$	$P(x^2=1\|y)=0.5$
y=NOT	$P(x^1=1\|y)=0.6$	$P(x^2=1\|y)=0.1$

	P(y)
y=SPAM	0.7
y=NOT	0.3

Recap: Generative Models and Naïve Bayes

- Generative models (attempt to) model the whole data distribution
  - Can generate new data, tolerate missing values, etc.
  - Not as good at prediction as discriminative models

Recap: Generative Models and Naïve Bayes

- The Naïve Bayes model assumes all features are conditionally independent given the label
  - Greatly simplified model structure, but still generative
  - Compact representation
  - Easy to train
  - Easy to compute various probabilities
  - Not the most accurate for standard prediction

Next Lecture

- Hidden Markov Models in depth
  - Sequence Modeling
  - Requires Dynamic Programming
  - Implement aspects of HMMs in homework

- Recitation Tuesday:
  - Recap of Dynamic Programming (for HMMs)