Data and Models

Machine Learning and Pattern Recognition

Chris Williams

School of Informatics, University of Edinburgh

September 2015

(All of the slides in this course have been adapted from previous versions by Charles Sutton, Amos Storkey, David Barber.)

Outline

- Data
- Probabilistic Models of Data
- The Inverse Problem
- Simple example: learning about a Bernoulli variable
- Real example: Naive Bayes classifier

Readings: Murphy 3.3 up to and including 3.3.1, 3.5 up to and including 3.5.1.1 Barber 9.1.1, 9.1.3, 10.1-10.2

Data? All shapes and sizes.

- Data types:
  - Real valued, positive, vector (geometric), bounded, thresholded
  - Categorical data, hierarchical classes, multiple membership, etc
  - Ordinal data, binary data, partially ordered sets
  - Missing, with known error, with error bars (known measurement error)
  - Internal dependencies, conditional categories
  - Raw, preprocessed, normalised, transformed etc
  - Biased, corrupted, just plain wrong, in unusable formats
  - Possessed, promised, planned, non-existent

Attributes and Values

- Simple datasets can be thought of as attribute value pairs
- For example "state of weather" is an attribute and "raining" is a value
- "Height" is an attribute, and "4ft 6in" is a value
- In this course we will assume that the data have been transformed into a vector $\mathbf{x} \in \mathbb{R}^D$
- This transformation can be the most important part of a learning algorithm! Here's an example...

Categorical Data

- Each observation belongs to one of a number of categories. Orderless. E.g. type of fruit.
- 1-of-M encoding. Represent each category by a particular component of an attribute vector.

0	1	0	0
1	0	0	0
1	0	0	0
0	0	0	1
0	1	0	0
0	0	1	0

- Only one component can be 'on' at any one time. Attributes are not (cannot be) independent.

Practical Hint

Get to know your data!

Test your high level assumptions before you use them to build models...

Probabilistic Models of Data

Supervised Learning

$p(\mathbf{x}, y, \theta|\mathcal{M})$, where θ denotes the parameters of the model.
$\mathcal{D} = ((\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^N, y^N))$

Unsupervised Learning

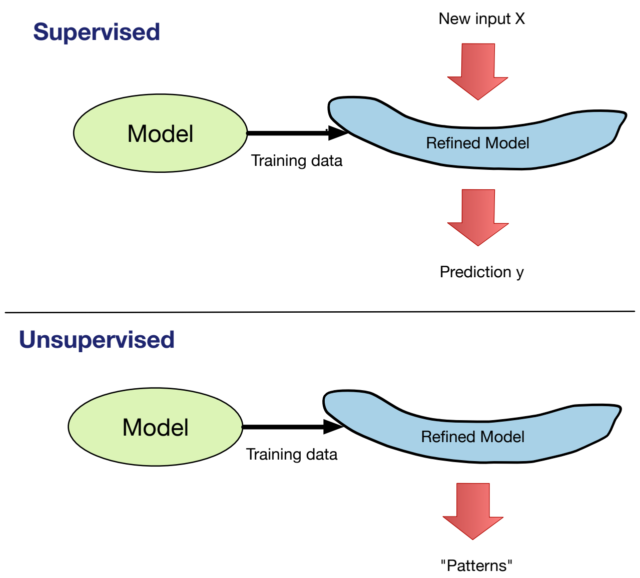
$p(\mathbf{x}, \theta|\mathcal{M})$, and data $\mathcal{D} = (\mathbf{x}^1, , \ldots, \mathbf{x}^N)$

Tasks

- **Prediction**: $p(y^*|\mathbf{x}^*, \theta, \mathcal{M})$, or $p(y^*|\mathbf{x}^*, \mathcal{D}, \mathcal{M})$
  unsupervised: $p(\mathbf{x}^*|\mathcal{D}, \mathcal{M})$
- **Learning**: $p(\theta|\mathcal{D}, \mathcal{M})$
- **Model Selection**: $p(\mathcal{D}|\mathcal{M})$

In the next two weeks, we'll see examples of using probabilistic models to do all of these things

Modelling



Generative and Discriminative Models

- Supervised setting
- *Discriminative* model: $p(y|\mathbf{x}, \mathcal{D})$
- *Generative* model:

$$p(y|\mathbf{x}, \mathcal{D}) \propto p(\mathbf{x}|y, \mathcal{D})p(y|\mathcal{D})$$

- With a generative model we can sample $\mathbf{x}$'s from the model to get artificial data
- Which approach is better?

The Inverse Problem

- We built a generative model, or a set of generative models on the basis of what we know (prior)
- Can generate artificial data
- BUT what if we want to *learn* a good distribution for data that we actually see? How is goodness measured?

Explaining Data

A particular distribution explains the data better if the data is more probable under that distribution: the **maximum likelihood** method

Likelihood

- $p(\mathcal{D}|\mathcal{M})$. The probability of the data $\mathcal{D}$ given a distribution (or model) $\mathcal{M}$. This is called the **likelihood**
- This is

$$L(\mathcal{M}) = p(\mathcal{D}|\mathcal{M}) = \prod_{n=1}^{N} p(x^n|\mathcal{M})$$

  i.e. the product of the probabilities of generating each data point individually
- This is a result of the independence assumption (indep $\rightarrow$ product of probabilities by definition)
- Key point: We consider this as a function of the *model*; the data is fixed
- Try different $\mathcal{M}$ (different distributions). Pick the $\mathcal{M}$ with the highest likelihood $\rightarrow$ Maximum Likelihood Method

Bernoulli Model

Example

Data: 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 1 1 0 1.

- Three hypotheses:
  - $\mathcal{M} = 1$ - From a fair coin. 1=H, 0=T
  - $\mathcal{M} = 2$ - From a die throw 1=1, 0 = 2,3,4,5,6
  - $\mathcal{M} = 3$ - From a double headed coin 1=H, 0=T

Bernoulli Model

Example

Data: 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 1 1 0 1.

- Three hypotheses:
  - $\mathcal{M} = 1$ - From a fair coin. 1=H, 0=T
  - $\mathcal{M} = 2$ - From a die throw 1=1, 0 = 2,3,4,5,6
  - $\mathcal{M} = 3$ - From a double headed coin 1=H, 0=T
- Likelihood of data. Let N_1=number of ones, N_0=number of zeros, with $N = N_0 + N_1$:

$$\prod_{n=1}^{N} p(x^n|\mathcal{M}) = p(1|\mathcal{M})^{N_1} p(0|\mathcal{M})^{N_0}$$

- $\mathcal{M} = 1$: Likelihood is $0.5^{20} = 9.5 \times 10^{-7}$
- $\mathcal{M} = 2$: Likelihood is $(1/6)^9 \ (5/6)^{11} = 1.3 \times 10^{-8}$
- $\mathcal{M} = 3$: Likelihood is $1^9 \ 0^{11} = 0$

Bernoulli model 2

Example

Data: 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 1 1 0 1.

- Continuous range of hypotheses: $\mathcal{M} = \pi$ - Generated from a Bernoulli distribution with parameter $p(x = 1|\pi) = \pi$.
- Likelihood:

$$\prod_{n=1}^{N} p(x^n|\pi) = \pi^{N_1}(1-\pi)^{N_0}$$

- Maximum likelihood hypothesis? Differentiate w.r.t. π to find maximum
- In fact usually easier to differentiate $\log p(\mathcal{D}|\mathcal{M})$: log is monotonic. So argmax $\log f(x) =$ argmax $f(x)$.

Bernoulli model 2

Example

Data: 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 1 1 0 1.

- Log likelihood:

$$L(\pi) = \log \prod_{n=1}^{N} p(x^n|\pi) = N_1 \log \pi + N_0 \log(1-\pi)$$

- Set $d/d\pi L(\pi) = N_1/\pi - N_0/(1-\pi)$ to zero to find maximum.
- So $N_1(1-\pi) - N_0\pi = 0$. This gives $\hat{\pi} = N_1/N$. Maximum likelihood result is unsurprising
- Warning: do we always believe all possible values of π are equally likely?

On the board

It's useful to plot this.

$$L(\pi) = \log \prod_{n=1}^{N} p(x^n|\pi) = N_1 \log \pi + N_0 \log(1-\pi)$$

Maximum Likelihood in General

- Model $\mathcal{M}$, data $\mathcal{D}$ and parameters θ
- Maximum likelihood estimator (MLE) obtained by

$$\hat{\theta} = \text{argmax}_{\theta} L(\theta; \mathcal{M}, \mathcal{D})$$

- MLE has several attractive statistical properties

Naive Bayes

Now let's look at a probabilistic generative model in a supervised setting

- Typical example: "Naive-Bayes Spam Filter" for classifying documents as spam (unwanted) or ham (wanted)
- (Not really a Bayesian method, in some sense—where that sense is the one we'll talk about next time).
- Basic (naive) assumption: conditional independence.
- Given the class (eg "Spam", "Not Spam"), whether one data item appears is independent of whether another appears.
- Invariably wrong! But useful anyway.

Conditional Independence, Parameters

- $x_1, x_2, \ldots, x_D$ are said to be conditionally independent given y iff

$$p(\mathbf{x}|y = c, \boldsymbol{\theta}) = \prod_{d=1}^{D} p(x_d|y = c, \theta_{dc})$$

  for $\mathbf{x} = (x_1, x_2, \ldots, x_D)^T$.
- $p(y = c) = \pi_c$

Naive Bayes

- The equation on the previous slide is in fact one part of the Naive Bayes Model. Extending for all the data $\{(\mathbf{x}^n, y^n)|n = 1, 2, \ldots, N\}$ we have:

$$p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{n} p(\mathbf{x}^n|y^n, \boldsymbol{\theta})p(y^n|\boldsymbol{\pi}) = \prod_{n} p(y^n|\boldsymbol{\pi}) \prod_{d=1}^{D} p(x_d^n|y^n, \boldsymbol{\theta}_d)$$

  for $\mathbf{x}^n = (x_1^n, \ldots, x_D^n)^T$.
- $\mathbf{x}^n$ is our attribute vector for data point n, and y^n the corresponding class label.
- We want to learn $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ from the data.
- We then want to find the best choice of y^* corresponding to a new datum $\mathbf{x}^*$ (inference).

Maximum Likelihood for Naive Bayes

- Simplest model: x_d is binary (presence or absence of word), y is binary (spam or ham).
- Already done this: $p(x_d|y)$ and $p(y)$ are both Bernoulli variables - see earlier. Just need to count to get maximum likelihood solution.
- $\hat{\pi}_{Spam}$ is (number of Spam documents)/(total number of documents)
- $\hat{\theta}_{d,Spam}$ is (number of spam documents that feature d turns up in)/(number of spam documents)

Spam

Sources: http://en.wikipedia.org/wiki/Spam_(Monty_Python),

http://commons.wikimedia.org/wiki/File:Spam_2.jpg

Whole Model

- We have built a *class conditional model* using the conditional probability of seeing each feature, given the document class (e.g. Spam/not Spam).
- Probability of Spam containing each feature. Probability of not Spam containing each feature. Estimated using maximum likelihood.
- Prior probability of Spam. Estimated using maximum likelihood.
- New document. Check the presence/absence of each feature. Build $\mathbf{x}^*$
- Calculate the Spam probability given the vector of word occurrence.
- How?

Inference in Naive Bayes

Use Bayes Theorem

$$p(Spam|\mathbf{x}^*, \boldsymbol{\theta}, \boldsymbol{\pi}) = \frac{\pi_{Spam} \prod_d p(x_d^*|Spam)}{p(\mathbf{x}^*|\boldsymbol{\theta}, \boldsymbol{\pi})}$$

where

$$p(\mathbf{x}^*|\boldsymbol{\theta}, \boldsymbol{\pi}) = \pi_{Ham} \prod_d p(x_d^*|Ham) + \pi_{Spam} \prod_d p(x_d^*|Spam)$$

by normalisation

Summary

- Given the data, and a model (a set of hypotheses - either discrete or continuous) we can find a maximum likelihood model/parameters for the data.
- Naive Bayes: Conditional independence
- Bag of words.
- Learning Parameters.
- Bayes Rule
- Next lecture: Bayesian methods.