Announcements

HW3 due tonight

Tutorial feedback back tonight

Tutorial due Apr 6 (Submission)

Tutorial peer evaluation: Apr 11 (Peer evaluation)

15-388/688 - Practical Data Science: Maximum likelihood estimation, naïve Bayes

Pat Virtue
Carnegie Mellon University
Spring 2022

Slide credits: CMU AI, Zico Kolter

Outline

Maximum likelihood estimation

Naive Bayes

Machine learning and maximum likelihood

Outline

Maximum likelihood estimation

Naive Bayes

Machine learning and maximum likelihood

Challenge

Assume that exam scores are drawn independently from the same Gaussian (Normal) distribution.

Given three exam scores 75, 80, 90, which pair of parameters is a better fit?

A) Mean 80, standard deviation 3

B) Mean 85, standard deviation 7

Use a calculator/computer.

Gaussian PDF: $p(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Estimating the parameters of distributions

We're moving now from probability to statistics

Estimating the parameters of distributions

We're moving now from probability to statistics

The basic question: given some data $x^{(1)}, \ldots, x^{(m)}$, how do I find a distribution that captures this data "well"?

In general (if we can pick from the space of all distributions), this is a hard question, but if we pick from a particular *parameterized family* of distributions $p(X; \theta)$, the question is (at least a little bit) easier

Question becomes: how do I find parameters θ of this distribution that fit the data?

Maximum likelihood estimation

Given a distribution $p(X;\theta)$, and a collection of observed (independent) data points $x^{(1)},\ldots,x^{(m)}$, the probability of observing this data is simply

$$p\left(x^{(1)},\ldots,x^{(m)};\theta\right) =$$

Basic idea of maximum likelihood estimation (MLE): find the parameters that maximize the probability of the observed data

$$\underset{\theta}{\text{maximize}} \prod_{i=1}^{m} p\left(x^{(i)};\theta\right) \equiv \underset{\theta}{\text{maximize}}$$

where $\ell(\theta)$ is called the **log likelihood** of the data

Seems “obvious”, but there are many other ways of fitting parameters

Parameter estimation for Bernoulli

Simple example: Bernoulli distribution

$$p(X = 1; \phi) = \phi, \qquad p(X = 0; \phi) = 1 - \phi$$

Given observed data $x^{(1)}, \ldots, x^{(m)}$, the "obvious" answer is:

$$\hat{\phi} = \frac{\#1\text{'s}}{\#\ \text{Total}} = \frac{\sum_{i=1}^{m} x^{(i)}}{m}$$

But why is this the case?

Maybe there are other estimates that are just as good, i.e.?

$$\phi = \frac{\sum_{i=1}^{m} x^{(i)} + 1}{m + 2}$$

Likelihood for Bernoulli

The likelihood for Bernoulli is given by

$$L(\phi) = \prod_{i=1}^{m} p\left(x^{(i)}; \phi\right)$$

Let's say we have a dataset of 3 heads and 2 tails:

	x
(1)	1
(2)	1
(3)	0
(4)	0
(5)	1

MLE for Bernoulli

Maximum likelihood solution for Bernoulli is given by

$$\underset{\phi}{\text{maximize}} \prod_{i=1}^{m} p\left(x^{(i)}; \phi\right) = \underset{\phi}{\text{maximize}}$$

Taking the negative log of the optimization objective (just to be consistent with our usual notation of optimization as minimization)

Derivative with respect to ϕ is given by

$$\frac{d}{d\phi}\ell(\phi) = \sum_{i=1}^{m}\left(\frac{x^{(i)}}{\phi} - \frac{1-x^{(i)}}{1-\phi}\right) = \frac{\sum_{i=1}^{m} x^{(i)}}{\phi} - \frac{\sum_{i=1}^{m}(1-x^{(i)})}{1-\phi}$$

MLE for Bernoulli, continued

Setting derivative to zero gives:

$$\frac{\sum_{i=1}^{m} x^{(i)}}{\phi} - \frac{\sum_{i=1}^{m}(1 - x^{(i)})}{1 - \phi} \equiv \frac{a}{\phi} - \frac{b}{1 - \phi} = 0$$
$$\Longrightarrow (1 - \phi)a = \phi b$$
$$\Longrightarrow \phi = \frac{a}{a + b} = \frac{\sum_{i=1}^{m} x^{(i)}}{m}$$

So, we have shown that the “natural” estimate of ϕ actually corresponds to the maximum likelihood estimate

MLE for Gaussian, briefly

For Gaussian distribution

$$p(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} \exp(-(1/2)(x-\mu)^2/\sigma^2)$$

Log likelihood given by:

$$\ell(\mu, \sigma^2) = -m\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2}\sum_{i=1}^{m}\frac{\left(x^{(i)}-\mu\right)^2}{\sigma^2}$$

Derivatives (see if you can derive these fully):

$$\frac{d}{d\mu}\ell(\mu, \sigma^2) = -\frac{1}{2}\sum_{i=1}^{m}\frac{x^{(i)}-\mu}{\sigma^2} = 0 \Rightarrow \mu = \frac{1}{m}\sum_{i=1}^{m}x^{(i)}$$

$$\frac{d}{d\sigma^2}\ell(\mu, \sigma^2) = -\frac{m}{2\sigma^2} + \frac{1}{2}\sum_{i=1}^{m}\frac{\left(x^{(i)}-\mu\right)^2}{(\sigma^2)^2} = 0 \Rightarrow \sigma^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)}-\mu\right)^2$$

Outline

Maximum likelihood estimation

Naive Bayes

Machine learning and maximum likelihood

SPAM Classification

Example

Training Data

Spam?	E-mail body
1	Money is free now
0	Pat teach 388
0	Pat free to teach
1	Sir money to teach
1	Pat free money now
0	Teach 388 now
0	Pat to teach 301

Vocabulary

388
free
is
money
now
Pat
Sir
teach
to
tomorrow

Test Data

Spam?	E-mail body
	Pat teach now

Poll 1

Assume:
Y is a binary random variable representing whether or not the email is spam, and X_i is a binary random variable representing whether or not the i-th word is in the email.

With a vocabulary of size 10, how may probability values are in the following probability table?

$$P(Y \mid X_1, \dots, X_{10})$$

$A.$ 10
$B.$ 11
$C.$ 110
$D.$ 22
$E.$ 2^{10}
$F.$ 2^{11}

	Vocabulary
1	388
2	free
3	is
4	money
5	now
6	Pat
7	Sir
8	teach
9	to
10	tomorrow

Naive Bayes modeling

Naive Bayes is a machine learning algorithm that rests relies heavily on probabilistic modeling

But, it is also interpretable according to the three ingredients of a machine learning algorithm (hypothesis function, loss, optimization), more on this later

Basic idea is that we model input and output as random variables $X = (X_1, X_2, \dots, X_n)$ (several Bernoulli, categorical, or Gaussian random variables), and Y (one Bernoulli or categorical random variable), goal is to find $p(Y|X)$

Naive Bayes assumptions

We're going to find $p(Y|X)$ via Bayes' rule

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} = \frac{p(X|Y)p(Y)}{\sum_y p(X|y)\, p(y)}$$

The denominator is just the sum over all values of Y of the distribution specified by the numeration, so we're just going to focus on the $p(X|Y)p(Y)$ term

Modeling full distribution $p(X|Y)$ for high-dimensional X is not practical, so we're going to make the **naive Bayes assumption**, that the elements X_i are conditionally independent given Y

$$p(X|Y) = \prod_{i=1}^{n} p(X_i|Y)$$

Poll 2

Assume:
Y is a binary random variable representing whether or not the email is spam, and
X_i is a binary random variable representing whether or not the i-th word is in the email.

True or False: $P(X_1 = 1 \mid Y = 0) = P(X_1 = 1 \mid Y = 1)$

	Vocabulary
1	388
2	free
3	is
4	money
5	now
6	Pat
7	Sir
8	teach
9	to
10	tomorrow

Modeling individual distributions

We're going to explicitly model the distribution of each $p(X_i|Y)$ as well as $p(Y)$

We do this by specifying a distribution for $p(Y)$ and a *separate* distribution and for each $p(X_i|Y = y)$

So assuming, for instance, that Y_i and X_i are binary (Bernoulli random variables), then we would represent the distributions

$$p(Y; \phi_{Y=1}), \qquad p(X_i|Y = 0; \phi_{Y=0,i}), \qquad p\left(X_i \middle| Y = 1; \phi_{Y=1,i}\right)$$

We then estimate the parameters of these distributions using MLE, i.e.

$$\phi_{Y=1} = \frac{\sum_{j=1}^{m} y^{(j)}}{m}, \qquad \phi_{y,i} = \frac{\sum_{j=1}^{m} x_i^{(j)} \cdot \mathbb{1}\{y^{(j)} = y\}}{\sum_{j=1}^{m} \mathbb{1}\{y^{(j)} = y\}}$$

Making predictions

Given some new data point x, we can now compute the probability of each class

$$p(Y = y \mid x) \propto p(Y = y) \prod_{i=1}^{n} p(x_i \mid Y = y) = \phi_y \prod_{i=1}^{n} \left(\phi_{y,i}\right)^{x_i} \left(1 - \phi_1^y\right)^{1-x_i}$$

After you have computed the right-hand side, just normalize (divide by the sum over all y) to get the desired probability

Alternatively, if you just want to know the most likely Y, just compute each righ-hand side and take the maximum

Example

Y	X_1	X_2
0	0	0
1	1	0
0	0	1
1	1	1
1	1	0
0	1	0
1	0	1
?	1	0

$p(Y = 1) = \phi_{Y=1} =$

$p(X_1 = 1 \mid Y = 0) = \phi_{Y=0,1} =$

$p(X_1 = 1 \mid Y = 1) = \phi_{Y=1,1} =$

$p(X_2 = 1 \mid Y = 0) = \phi_{Y=0,2} =$

$p(X_2 = 1 \mid Y = 1) = \phi_{Y=1,2} =$

$p(Y \mid X_1 = 1, X_2 = 0) =$

Potential issues

Problem #1: when computing probability, the product $\mathrm{p}(y) \prod_{i=1}^{n} p(x_i|y)$ quickly goes to zero to numerical precision

Solution: compute log of the probabilities instead

$$\log p(y) + \sum_{i=1}^{n} \log p(x_i|y)$$

Problem #2: If we have never seen either $X_i = 1$ or $X_i = 0$ for a given y, then the corresponding probabilities computed by MLE will be zero

Solution: Laplace smoothing, "hallucinate" one $X_i = 0/1$ for each class

$$\phi_{y,i} = \frac{\sum_{j=1}^{m} x_i^{(j)} \cdot \mathbb{1}\{y^{(j)} = y\} + 1}{\sum_{j=1}^{m} \mathbb{1}\{y^{(j)} = y\} + 2}$$

Other distributions

Categorical class

Let Y be the random variable for a class that takes on one of K possible categories $\{1, \ldots, K\}$ (rather than binary as we were doing before)

$$P(Y = y) = \phi_y = \frac{\sum_{j=1}^{m} \mathbb{1}\{y^{(j)} = y\}}{m}$$

Y	X_1	X_2
cat		
dog		
rat		
rat		
cat		
cat		

Y	X_1	X_2
1		
2		
3		
3		
1		
1		

Other distributions

Categorical feature conditioned on class

Assume the i-th feature takes on one of K possible categories $\{1, \ldots, K\}$ (rather than binary as we were doing before)

$$P(X_i = k \mid Y = y) = \phi_{y,i,k} = \frac{\sum_{j=1}^{m} \mathbb{1}\left\{x_i^{(j)} = k\right\} \cdot \mathbb{1}\{y^{(j)} = y\}}{\sum_{j=1}^{m} \mathbb{1}\{y^{(j)} = y\}}$$

Y	X_1	X_2
cat	blue	wood
dog	blue	metal
rat	green	metal
rat	red	paper
cat	red	wood
cat	blue	wood

Y	X_1	X_2
1	1	3
2	1	1
3	2	1
3	3	2
1	3	3
1	1	3

Other distributions

Though naive Bayes is often presented as "just" counting, the value of the maximum likelihood interpretation is that it's clear how to model $p(X_i|Y)$ for non-categorical random variables

Example: if x_i is real-valued, we can model $p(X_i|Y = y)$ as a Gaussian

$$p\left(x_i \middle| y; \mu^y, \sigma_y^2\right) = \mathcal{N}(x_i; \mu^y, \sigma_y^2)$$

with maximum likelihood estimates

$$\mu_y = \frac{\sum_{j=1}^m x_i^{(j)} \cdot \mathbb{1}\{y^{(j)} = y\}}{\sum_{j=1}^m \mathbb{1}\{y^{(j)} = y\}}, \sigma_y^2 = \frac{\sum_{j=1}^m (x_i^{(j)} - \mu^y)^2 \cdot \mathbb{1}\{y^{(j)} = y\}}{\sum_{j=1}^m \mathbb{1}\{y^{(j)} = y\}}$$

All probability computations are exactly the same as before (it doesn't matter that some of the terms are probability densities)

Other distributions

Gaussian features conditioned on class

$$\mu_y = \frac{\sum_{j=1}^{m} x_i^{(j)} \cdot \mathbb{1}\{y^{(j)} = y\}}{\sum_{j=1}^{m} \mathbb{1}\{y^{(j)} = y\}}, \sigma_y^2 = \frac{\sum_{j=1}^{m} (x_i^{(j)} - \mu^y)^\wedge 2 \cdot \mathbb{1}\{y^{(j)} = y\}}{\sum_{j=1}^{m} \mathbb{1}\{y^{(j)} = y\}}$$

	Score	Time
$Exam$	X_1	X_2
1	90	30
2	85	60
3	70	20
3	60	25
1	80	50
1	90	40

Outline

Maximum likelihood estimation

Naive Bayes

Machine learning and maximum likelihood

Machine learning via maximum likelihood

Many machine learning algorithms (specifically the loss function component) can be interpreted probabilistically, as maximum likelihood estimation

Recall logistic regression:

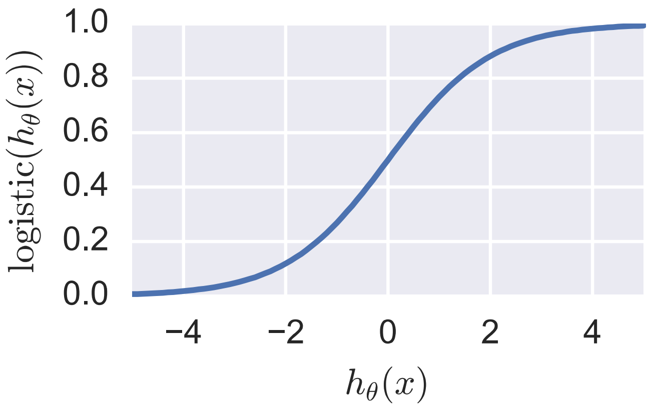
$$\operatorname*{minimize}_{\theta} \sum_{i=1}^{m} \ell_{\text{logistic}}(h_\theta(x^{(i)}), y^{(i)})$$

$$\ell_{\text{logistic}}(h_\theta(x), y) = \log(1 + \exp\left(-y \cdot h_\theta(x)\right))$$

Logistic probability model

Consider the model (where Y is binary taking on $\{-1, +1\}$ values)

$$p(y|x;\theta) = \operatorname{logistic}\big(y \cdot h_\theta(x)\big) = \frac{1}{1 + \exp(-y \cdot h_\theta(x))}$$



Under this model, the maximum likelihood estimate is

$$\underset{\theta}{\operatorname{maximize}} \sum_{i=1}^{m} \log p\big(y^{(i)} \big| x^{(i)}; \theta\big) \equiv \underset{\theta}{\operatorname{minimize}} \sum_{i=1}^{m} \ell_{\text{logistic}}\big(h_\theta(x^{(i)}), y^{(i)}\big)$$

Least squares

In linear regression, assume

$$y = \theta^T x + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2)$$
$$\Leftrightarrow p(y|x;\theta) = \mathcal{N}(\theta^T x, \sigma^2)$$

Then the maximum likelihood estimate is given by

$$\underset{\theta}{\text{maximize}} \sum_{i=1}^{m} \log p\left(y^{(i)} \middle| x^{(i)}; \theta\right) \equiv \underset{\theta}{\text{minimize}} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2$$

i.e., the least-squares loss function can be viewed as MLE under Gaussian errors

Other approaches possible too: absolute loss function can be viewed as MLE under Laplace errors