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

J. Zico Kolter
Carnegie Mellon University
Spring 2018

Outline

Maximum likelihood estimation

Naive Bayes

Machine learning and maximum likelihood

Outline

Maximum likelihood estimation

Naive Bayes

Machine learning and maximum likelihood

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

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

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

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

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:

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

But why is this the case?

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

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

MLE for Bernoulli

Maximum likelihood solution for Bernoulli given by

$$\operatorname*{maximize}_{\phi} \prod_{i=1}^{m} p(x^{(i)};\phi) = \operatorname*{maximize}_{\phi} \prod_{i=1}^{m} \phi^{x^{(i)}}(1-\phi)^{1-x^{(i)}}$$

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

$$\operatorname*{maximize}_{\phi} \ell(\phi) = \sum_{i=1}^{m}\left(x^{(i)}\log\phi + (1-x^{(i)})\log(1-\phi)\right)$$

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

Poll: Bernoulli maximum likelihood

Suppose we observe binary data $x^{(1)}, \dots, x^{(m)}$ with $x^{(i)} \in \{0{,}1\}$ with some $x^{(i)} = 0$ and some $x^{(j)} = 1$, and we compute the Bernoulli MLE

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

Which of following statements is necessarily true? (may be more than one)

1. For any $\phi' \neq \phi$, $p(x^{(i)}; \phi') \leq p(x^{(i)}; \phi\ )$ for all $i = 1, \dots, n$
2. For any $\phi' \neq \phi$, $\prod_{i=1}^{m} p(x^{(i)}; \phi') \leq \prod_{i=1}^{m} p(x^{(i)}; \phi\ )$
3. We always have $p(x^{(i)}; \phi') \geq p(x^{(i)}; \phi\ )$ for at least one i

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:

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

Derivatives (see if you can derive these fully):

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

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

Outline

Maximum likelihood estimation

Naive Bayes

Machine learning and maximum likelihood

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 $X = (X_1, X_2, \ldots, X_n)$ (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

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

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

$$p(Y;\phi_0), \qquad p(X_i|Y=0;\phi_i^0), \qquad p(X_i|Y=1;\phi_i^1)$$

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

$$\phi_0 = \frac{\sum_{j=1}^m y^{(j)}}{m}, \qquad \phi_i^y = \frac{\sum_{j=1}^m x_i^{(j)} \cdot 1\{y^{(j)} = y\}}{\sum_{j=1}^m 1\{y^{(j)} = y\}}$$

Making predictions

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

$$p(Y = y|x) \propto p(Y = y) \prod_{i=1}^{m} p(x_i|Y = y) = \phi_0 \prod_{i=1}^{m} (\phi_i^y)^{x_i} (1 - \phi_1^y)^{1-x_i}$$

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 right 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

$$p(Y=1) = \ \phi_0 =$$

$$p(X_1 = 1|Y=0) = \phi_1^0 =$$

$$p(X_1 = 1|Y=1) = \phi_1^1 =$$

$$p(X_2 = 1|Y=0) = \phi_2^0 =$$

$$p(X_2 = 1|Y=0) = \phi_2^1 =$$

$$p(Y|X_1 = 1, X_2 = 0) =$$

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

$$\phi_i^y = \frac{\sum_{j=1}^{m} x_i^{(j)} \cdot 1\{y^{(j)} = y\} + 1}{\sum_{j=1}^{m} 1\{y^{(j)} = y\} + 2}$$

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

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

with maximum likelihood estimates

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

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

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:

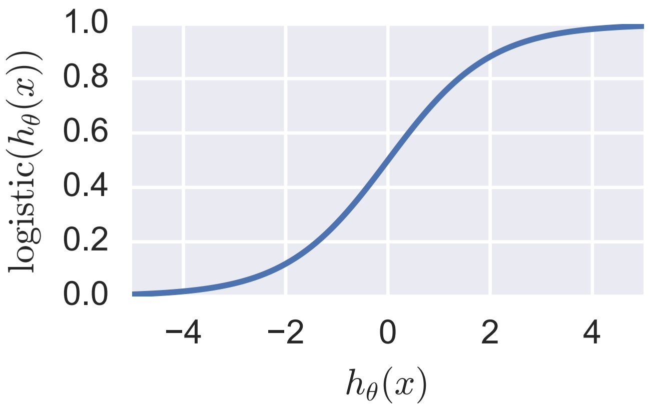
$$\underset{\theta}{\text{minimize}} \ \sum_{i=1}^{m} \ell_{\text{logistic}}(h_\theta(x^{(i)}), y^{(i)})$$

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

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

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

Least squares

In linear regression, assume

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

Then the maximum likelihood estimate is given by

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

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