CS 189 Introduction to Machine Learning
Spring 2018 DIS2

1 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) is a method of estimating the parameters of a statistical model given observations, by finding the parameters that maximize the likelihood of the observations. Concretely, given observations $y_1, y_2, \ldots, y_n$ distributed according to $p_\theta(y_1, y_2, \ldots, y_n)$ (here p_θ can be a probability mass function for discrete observations or a density for continuous observations), the likelihood function is defined as $L(\theta) = p_\theta(y_1, y_2, \ldots, y_n)$ and the MLE is

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} L(\theta).$$

We often make the assumption that the observations are *independent and identically distributed* or iid, in which case $p_\theta(y_1, y_2, \ldots, y_n) = f_\theta(y_1) \cdot f_\theta(y_2) \cdot \cdots \cdot f_\theta(y_n)$.

(a) Your friendly TA recommends maximizing the log-likelihood $\ell(\theta) = \log L(\theta)$ instead of $L(\theta)$. Why does this yield the same solution $\hat{\theta}_{\text{MLE}}$? Why is it easier to solve the optimization problem for $\ell(\theta)$ in the iid case? Write down both $L(\theta)$ and $\ell(\theta)$ for the Gaussian $f_\theta(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\mu)^2}{2\sigma^2}}$ with $\theta = (\mu, \sigma)$.

Solution: As the log is strictly monotonically increasing, maximizing $\ell(\theta) = L(\theta)$ and $L(\theta)$ will yield the same solution. Concretely, if θ^* is a unique maximum of $L(\theta)$, we have $L(\theta) < L(\theta^*)$ for all $\theta \neq \theta^*$ in the parameter space and therefore due to strict monotonicity of the log, $\ell(\theta) = \log L(\theta) < \log L(\theta^*) = \ell(\theta^*)$, which means θ^* is also a unique maximum of $\ell(\theta)$.

In the iid case, the log-likelihood decomposes into a sum

$$\ell(\theta) = \sum_{i=1}^{n} \log f_\theta(y_i)$$

and it is often easier to optimize over these sums rather than products:

Numerically: There are special algorithms like stochastic gradient descent available for sums that you will learn about later in lecture. Another reason is that forming the product of many probabilities will yield a very small number and it is easy to generate a floating point underflow this way. On the other hand, adding the logs of probabilities is a more stable operation because the partial sums stay in a reasonable range.

Analytically: Usually it is easier to compute the gradient of $\ell(\theta)$ than for $L(\theta)$. As an example, consider the case of a Gaussian distribution:

The likelihood function is

$$L(\theta) = \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \cdot e^{-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}}.$$

Taking logs yields

$$\ell(\theta) = \sum_{i=1}^{n} \log f_\theta(y_i) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2$$

which is much easier to minimize than $L(\theta)$.

(b) What is $\int_{y_1,y_2,\ldots,y_n} p_\theta(y_1, y_2, \ldots, y_n)\, dy_1 \cdots dy_n$? What is $\int_\theta p_\theta(y_1, y_2, \ldots, y_n) d\theta$?

Solution: The probability distribution is normalized, therefore by integrating over the observations we have

$$\int_{y_1,y_2,\ldots,y_n} p_\theta(y_1, y_2, \ldots, y_n)\, dy_1 \cdots dy_n = 1.$$

In terms of θ there is no such normalization. In fact, the integral can be divergent, consider the example of a Pareto density with fixed scale $x_m = 1$, $f_\alpha(y) = \alpha/y^{\alpha+1}$ for $y \geq 1$ in which case we have for a single observation $y_1 = 1$ that

$$\int_{\alpha>0} f_\alpha(1) d\alpha = \infty.$$

(c) The Poisson distribution is $f_\lambda(y) = \frac{\lambda^y e^{-\lambda}}{y!}$. Let $Y_1, Y_2, \ldots, Y_n$ be a set of independent and identically distributed random variables with Poisson distribution with parameter λ. Find the joint distribution of $Y_1, Y_2, \ldots, Y_n$. Find the maximum likelihood estimator of λ as a function of observations $y_1, y_2, \ldots, y_n$.

Solution:

The joint probability mass function is the product of the probability mass functions of all n independent variables y_i,

$p_\theta(y_1, y_2, \ldots, y_n) = \prod_{i=1}^{n} \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}$.

The log likelihood will thus be $\ell(\lambda) = \sum_{i=1}^{n}(y_i \log(\lambda) - \lambda - \log(y_i!))$

We find the maximum by finding the derivative and setting it to 0:

$\ell'(\lambda) = (\sum_{i=1}^{n} \frac{y_i}{\lambda}) - n = 0$. Hence, the estimate should be $\hat{\lambda} = \frac{\sum_{i=1}^{n} y_i}{n} = \bar{Y}$, which is the mean of the observations.

2 Intuition for Regularization

This problem is meant to be a short toy problem that explores the trade-off between the complexity of a model and its fit to the training data. If we make our models complicated enough, then we can fit pretty much anything. We avoid this using regularization. Before we do the math of regularization, let's think about why it's necessary.

One of our friendly TAs frequently goes to the pool to prepare for her upcoming triathlon. In the past month, she went to the pool 19 times. We want to look at the data and predict when she will go to the pool next month. Here is the data from this month:

Week 1: Sunday, Monday, Tuesday, Thursday, Friday
Week 2: Monday, Tuesday, Thursday, Saturday
Week 3: Sunday, Monday, Thursday, Friday, Saturday
Week 4: Monday, Tuesday, Thursday, Friday, Saturday

Consider the following four explanations for her swimming habits:

1. "She goes to the pool every day"
2. "She goes to the pool every day except Wednesdays"
3. "She goes to the pool every day except Wednesdays and even Sundays"
4. "She goes to the pool every day except Wednesdays and even Sundays and the Tuesday of third week of the month and the Saturday of the first week of the month and the Friday of the second week of the month."

(a) What is the right criterion for what makes a "good" rule? How can we validate it? Which rule is intuitively the best? **Solution:** A "good" rule is one that generalizes to unseen data. It can be validated by testing it on future data or hold out data.

Hopefully your intuition is that the second or third explanation are best since they both describe the data well and also find simple patterns in the data, therefore we might hope they will generalize.

(b) Out of the 28 days in the four weeks, how many does each rule get right? That is, on how many days does the rule predict whether or not she will be at the pool? **Solution:**

(a) 19

(b) 23

(c) 25

(d) 28

(c) Now, we have an intuitive notion that the later rules are more complex than the earlier rules. This is a rather difficult notion to formalize, but let's give it a shot by trying to measure the number of "cases" created by each rule. If we wanted to implement these rules in code, how many "case" statements (if/elif/elif/elif/else) would it take? Let's assume that we're only allowed to use AND statements, not OR statements (so that we can't combine cases). **Solution:** Here is an example of how we would implement Rule 3:

```
if today is 'Wednesday':
return False
elif today is 'Sunday' and week is even:
return False
else
return True
```

(a) 1 (we could think of this as 0, but then we should also ignore the else statements in the other problems)

(b) 2

(c) 3

(d) 6

(d) Which rule is the "best" rule if we only measure how well it does on the training data? **Solution:** Rule 4, because it predicts the correct behavior on all days in the training set, more than any other rule.

(e) Which rule is the "best" rule if we require that the "complexity" (as measured by the number of cases) is 2 or less? **Solution:** Rule 2 is the best, since rules 3 and 4 are banned by the constraint.

(f) Now, which rule is best if we don't set a strict requirement on the number of cases, but we penalize our correctness by 2 points for each case? **Solution:** Rules 2 and 3 tie for the best. The exact scores are:

(a) 17

(b) 19

(c) 19

(d) 16

(g) Does this match your intuition? How does this relate to regularization? **Solution:** Hopefully, it does. The "complexity" of the case statement acts as the regularization term. A motivation for introducing the regularization term is a principle called "Occam's Razor". It states that the simplest solution is often the preferred one and regularization is a way to formalize this intuition. In our case, the complexity is measured by the number of statements in the if condition and this is used to penalize more complex solutions. The hope is that simpler solutions can generalize better and for many regularizers in machine learning this can indeed be shown formally.

(h) On Week 5, the schedule was "Sunday, Monday, Tuesday, Thursday, Friday, Saturday" and on Week 6 it was "Tuesday, Thursday, Friday, Saturday". How does this change your conclusions? **Solution:** You should have increased confidence in Rule 3 now, because it explains the new data much better than Rule 2 and is only slightly more complex than Rule 3. Formally, you can go through (f) again with the new data and see that Rule 3 is preferred now.

(i) Is there any relation between this problem and ridge regression? **Solution:** Yes, the ℓ_2 regularization on the parameters in ridge regression is similar to the 2 points penalty for each extra case in this problem. In the ridge regression case, the complexity of the model is represented by the norm of the weights. Both of the regularization terms push the model away from selecting overly complex models.

3 Recalling Determinants

For a square matrix $\mathbf{A}$, the determinant $\det(\mathbf{A})$ is the oriented volume of the parallelepiped spanned by the columns or rows of $\mathbf{A}$. We then have (a) $\det\left(\prod_i \mathbf{A}_i\right) = \prod_i \det(\mathbf{A}_i)$, (b) $\det(\mathbf{A}) = \det\left(\mathbf{A}^\top\right)$, (c) for $\mathbf{T}$ an upper triangular matrix, $\det(\mathbf{T})$ is the product of the diagonal entries of $\mathbf{T}$ and (d) the determinant of a matrix is the product of its eigenvalues.

How can you compute the determinant of an arbitrary matrix using these properties? Think about writing the matrix as a product of elementary matrices! Find the determinant of $\mathbf{A} = \begin{bmatrix} 2 & 2 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}$.

Solution:

Using elementary matrices $\mathbf{E_1}, \mathbf{E_2}$ and $\mathbf{E_3}$, we get the reduced row echelon form of $\mathbf{A}$. In our case, $\mathbf{E_3 E_2 E_1 A} = \mathbf{I}$. Since we know the determinant of a triangular matrix, we can find $\det(\mathbf{A})$.

$\mathbf{E_1} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Multiply row 1 by 1/2.

$\mathbf{E_2} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Subtract row 2 from row 1.

$\mathbf{E_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}$ Subtract 3 times row 1 from row 3.

$1 = \det(\mathbf{I}) = \det(\mathbf{E_3 E_2 E_1 A}) = \det(\mathbf{E_3})\det(\mathbf{E_2})\det(\mathbf{E_1})\det(\mathbf{A}) = \frac{1}{2}\det(\mathbf{A}) \Rightarrow \det(\mathbf{A}) = 2$