6.867 Machine Learning

Problem Set 1

Due date: Monday September 27

Please address all questions and comments about this problem set to `6867-staff@csail.mit.edu`. You will need to use MATLAB for some of the problems but essentially all the code is provided. If you are not familiar with MATLAB, please consult

`http://www.ai.mit.edu/courses/6.867/matlab.html`

and the links therein.

Part I: background

Suppose we have a probability distribution or density $p(x;\theta)$, where x may be discrete or continuous depending on the problem we are interested in. θ specifies the parameters of this distribution such as the mean and the variance of a one dimensional Gaussian. Different settings of the parameters imply different distributions over x. The available data, when interpreted as samples $x_1, \ldots, x_n$ from one such distribution, should favor one setting of the parameters over another. We need a formal criterion for gauging how well any potential distribution $p(\cdot|\theta)$ "explains" or "fits" the data. Since $p(x|\theta)$ is the probability of reproducing any observation x, it seems natural to try to maximize this probability. This gives rise to the Maximum Likelihood estimation criterion for the parameters θ:

$$\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta} L(x_1, \ldots, x_n; \theta) = \arg\max_{\theta} \prod_{i=1}^{n} p(x_i|\theta) \tag{1}$$

where we have assumed that each data point x_i is drawn independently from the same distribution so that the likelihood of the data is $L(x_1, \ldots, x_n; \theta) = \prod_{i=1}^{n} p(x_i; \theta)$. Likelihood is viewed primarily as a function of the parameters, a function that depends on the data. The above expression can be quite complicated (depending on the family of distributions we are considering), and make maximization technically challenging. However, any monotonically increasing function of the likelihood will have the same maxima. One such function is log-likelihood $\log L(x_1, \ldots, x_n; \theta)$; taking the log turns the product into a sum, making derivatives significantly simpler. We will maximize the log-likelihood instead of likelihood.

Problem 1: Maximum Likelihood Estimation

Consider a sample of n real numbers $x_1, x_2, \ldots, x_n$ drawn independently from the same distribution that needs to be estimated. Assuming that the underlying distribution belongs to one of the following parametrized families, the goal is to estimate its parameters (each family should be treated separately):

$$\text{Uniform}: \quad p(x;a) = \frac{1}{a} \text{ for } x \in [0,a],\ 0 \text{ otherwise} \tag{2}$$

$$\text{Exponential}: \quad p(x;\eta) = \frac{1}{\eta}\exp(-x/\eta),\ \eta > 0 \tag{3}$$

$$\text{Gaussian}: \quad p(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{4}$$

1. (*10 points*) Derive the maximum likelihood estimators $\hat{a}_{\mathrm{ML}}$, $\hat{\eta}_{\mathrm{ML}}$, $\hat{\mu}_{\mathrm{ML}}$, $\hat{\sigma}^2_{\mathrm{ML}}$. The estimators should be obtained by maximizing the log-likelihood of the dataset under each of the families, and should be a function of $x_1, x_2, \ldots, x_n$ only.

To assess how well an estimator $\hat{\theta}$ recovers the underlying value of the parameter θ, we study its *bias* and *variance*. The bias is defined by the expectation of the deviation from the true value under the true distribution of the sample $(X_1, X_2, \ldots, X_n)$:

$$\mathrm{bias}(\hat{\theta}) = \mathrm{E}_{X_i \sim P(X|\theta)}\left[\hat{\theta}(X_1, X_2, \ldots, X_n) - \theta\right] \tag{5}$$

Biased (i.e. with a non-zero bias) estimators systematically under-estimate or over-estimate the parameter.

The variance of the estimator

$$\mathrm{var}(\hat{\theta}) = \mathrm{E}_{X_i \sim P(X|\theta)}\left[\left(\hat{\theta}(X_1, X_2, \ldots, X_n) - \mathrm{E}\left[\hat{\theta}(X_1, X_2, \ldots, X_n)\right]\right)^2\right] \tag{6}$$

measures the anticipated uncertainty in the estimated value due to the particular selection $(x_1, x_2, \ldots, x_n)$ of the sample. Note that the concepts of bias and variance of estimators are similar to the concepts of structural and approximation errors, respectively.

Estimators that minimize both bias and variance are preferred, but typically there is a trade-off between bias and variance.

2. (*10 points*) Show that $\hat{a}_{\mathrm{ML}}$ is biased (no need to compute the actual value of the bias), $\hat{\eta}_{\mathrm{ML}}$ and $\hat{\mu}_{\mathrm{ML}}$ are unbiased.

3. (*optional*) Show that $\hat{\sigma}^2_{\mathrm{ML}}$, equal to the sample variance $\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$, is biased. Show that the ML estimator of the variance becomes unbiased after multiplication with $n/(n-1)$. Let $\hat{\sigma}^2_{n-1}$ be this new estimator.

4. (*optional*) A standard way to balance the tradeoff between bias and variance is to choose estimators of lower *mean squared error*: $\mathrm{MSE}(\hat{\theta}) = \mathrm{E}_{X_i \sim P(X|\theta)}\left[(\hat{\theta} - \theta)^2\right]$. Show that $\mathrm{MSE}(\hat{\theta}) = \mathrm{bias}(\hat{\theta})^2 + \mathrm{var}(\hat{\theta})$ and that $\mathrm{MSE}(\hat{\sigma}^2_{\mathrm{ML}}) < \mathrm{MSE}(\hat{\sigma}^2_{n-1})$ even though $\hat{\sigma}^2_{\mathrm{ML}}$ is biased.

Problem 2: Maximum A-Posteriori Estimation

We want to determine the bias of an unfair coin for "heads" or "tails" from observing the outcome of a series of tosses. We model the coin by a single parameter θ that represents the probability of tossing heads.

Given n independent observed tosses $\mathcal{D} = \{x_1, \ldots, x_n\}$ out of which n_H are "heads", the likelihood function is:

$$p(\mathcal{D}|\theta) = \theta^{n_H}(1-\theta)^{n-n_H} \tag{7}$$

1. (*5 points*) Show that $\hat{\theta}_{\mathrm{ML}} = n_H/n$. Thus if we toss the coin only once and we see "tails" ($n = 1$ and $n_H = 0$), according to maximum likelihood flipping the coin should always result in "tails".

While the maximum likelihood estimator is accurate on large training samples, if data is very scarce the estimated value is not that meaningful (on small samples the variance of the estimator is very high and it overfits easily). In contrast, in **Maximum A-Posteriori (MAP) estimation** we compensate for the lack of information due to limited observations with an *a priori* preference on the parameters based on prior knowledge we might have. In the case of the coin toss for instance, even without seeing any tosses we can assume the coin should be able to show both "heads" and "tails" ($\theta \neq 0$).

We express the prior preference/knowledge about θ by a distribution $p(\theta)$ (the *prior*). Assuming that θ and the observed sample are characterized by an underlying joint probability $p(\theta, \mathcal{D})$, we can use the Bayes rule to express our adjusted belief about the parameters after observing the trials (the *posterior*):

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} \tag{8}$$

where $p(\mathcal{D}) = \int p(\mathcal{D}|\theta')p(\theta')d\theta'$ normalizes the posterior. Maximization of the posterior distribution gives rise to the *Maximum A-Posteriori* (MAP) estimate of the parameters:

$$\hat{\theta}_{\mathrm{MAP}} = \arg\max_{\theta} p(\theta|\mathcal{D}) = \arg\max_{\theta} p(\mathcal{D}|\theta)p(\theta) \tag{9}$$

As in maximum likelihood, to compute the MAP estimate it is often easier to maximize the logarithm $\log p(\theta) + \log p(\mathcal{D}|\theta)$.

For the coin toss we will consider separately each of the following priors:

$$\text{Discrete}: \quad p^1(\theta) = \begin{cases} 0.5 & \text{if } \theta = 0.5 \\ 0.5 & \text{if } \theta = 0.4 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$\text{Beta}: \quad p^2(\theta) = \frac{1}{Z}\theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{11}$$

Here α and β are *hyperparameters* that should be given, not estimated, and Z is a normalization constant needed to make $p^2(\theta)$ integrate to 1 whose actual value is not important.

2. (*10 points*) Prior $p^1(\theta)$ translates into a strong belief that the coin is either fair, or biased towards "tails" with a "heads" probability of 0.4. Express the MAP estimate $\hat{\theta^1}_{\text{MAP}}$ under this prior as a function of n_H/n.

3. (*10 points*) The Beta prior expresses the belief that θ is likely to be near $\alpha/(\alpha+\beta)$. The larger $\alpha+\beta$ is, the more peaked the prior, and the stronger the bias that θ is close to $\alpha/(\alpha+\beta)$. Derive $\hat{\theta^2}_{\text{MAP}}$ under the Beta prior and show that when n approaches infinity the MAP estimate approaches the ML estimate, thus the prior becomes irrelevant given a large number of observations.

4. (*optional*) Compare qualitatively $\hat{\theta^1}_{\text{MAP}}$ and $\hat{\theta}_{\text{ML}}$. Assuming that the coin has a true "heads" probability of 0.41, which of the two estimators is likely to learn it faster? If data is sufficient, which of the two estimators is better?

Part II: Polynomial Regression

Problem 3

In this problem, we explore the behavior of polynomial regression methods when only a small amount of training data is available. We use polynomial regression models of the form

$$y = w_0 + w_1 x + w_2 x^2 + \ldots + w_m x^m + \epsilon \tag{12}$$

$$= \mathbf{x}^T \mathbf{w} + \epsilon \tag{13}$$

where $\epsilon \sim N(0, \sigma^2)$ (zero mean Gaussian noise) and $\mathbf{x} = [1 \; x \; x^2 \; \ldots \; x^m]^T$. In a matrix form for all the training outputs, the model can be written as

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{e} \tag{14}$$

where $\mathbf{y} = [y_1, \ldots, y_n]^T$, $\mathbf{X} = [\mathbf{x}_1^T; \cdots; \mathbf{x}_n^T]$ depends on the polynomial order, and $\mathbf{e} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$. In other words, the outputs $\mathbf{y}$ are normally distributed with mean vector $\mathbf{X}\mathbf{w}$

and covariance matrix $\sigma^2\mathbf{I}$. The likelihood of the outputs, given the inputs, can therefore be expressed as

$$p(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2) = N(\mathbf{y}; \mathbf{X}\mathbf{w}, \sigma^2\mathbf{I}) \tag{15}$$

where $N(\mathbf{y}; \mu, \Sigma)$ is a multi-variate (here $n-$variate) Gaussian

$$N(\mathbf{y}; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} det(\Sigma)^{1/2}} \exp\{-\frac{1}{2}(\mathbf{y}-\mu)^T\Sigma^{-1}(\mathbf{y}-\mu)\} \tag{16}$$

We will begin by using a maximum likelihood estimation criterion for the parameters $\mathbf{w}$ that reduces to least squares fitting.

1. Consider a simple 1D regression problem. The data in `housing.data` provides information of how 13 different factors affect house price in the Boston area. (Each column of data represents a different factor, and is described in brief in the file `housing.names`.) To simplify matters (and make the problem easier to visualise), we consider predicting the house price (the 14th column) from the LSTAT feature (the 13th column).

   We split the data set into two parts (in `testLinear.m`), train on the first part and test on the second. We have provided you with the necessary MATLAB code for training and testing a polynomial regression model. Simply edit the script (`ps1_part2.m`) to generate the variations discussed below.

   (a) (*5 points*) Use `ps1_part2.m` to calculate and plot training and test errors for polynomial regression models as a function of the polynomial order (from 1 to 7). Use 250 training examples (set `numtrain=250`).

   (b) (*10 points*) Briefly explain the qualitative behavior of the errors. Which of the regression models are over-fitting to the data? Provide a brief justification.

   (c) (*10 points*) Rerun `ps1_part2.m` with only 50 training examples (set `numtrain=50`). Briefly explain key differences between the resulting plot and the one from part a). Which of the models are over-fitting this time?

   There are many ways of trying to avoid over-fitting. One way is to use a maximum a posteriori (MAP) estimation criterion rather than maximum likelihood. MAP criterion allows us to penalize parameter choices that we would not expect to lead to good generalization. For example, very large parameter values in linear regression make predictions very sensitive to slight variations in the inputs. We can express a preference against such large parameter values by assigning a prior distribution over the parameters such as simple Gaussian

$$p(\mathbf{w}; \alpha^2) = \mathcal{N}(\mathbf{0}, \alpha^2\mathbf{I}) \tag{17}$$

This prior decreases rapidly as the parameters deviate from zero. The single variance (hyper-parameter) α^2 controls the extent to which we penalize large parameter values.

This prior needs to be combined with the likelihood to get the MAP criterion. The MAP parameter estimate maximizes

$$\log\left(\, p(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2) p(\mathbf{w}; \alpha^2) \,\right) = \log p(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2) + \log p(\mathbf{w}; \alpha^2) \tag{18}$$

The resulting parameter estimates are biased towards zero due to the prior. We can find these estimates as before by setting the derivatives to zero.

2. (*15 points*) Show that

$$\hat{\mathbf{w}}_{MAP} = (\mathbf{X}^T\mathbf{X} + \frac{\sigma^2}{\alpha^2}\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{19}$$

3. (*5 points*). In the above solution, show that in the limit of infinitely large α, the MAP estimate is equal to the ML estimate, and explain why this happens

4. Let us see how the MAP estimate changes our solution in the housing-price estimation problem. The MATLAB code you used above actually contains a variable corresponding to the variance ratio `var_ratio` $= \frac{\sigma^2}{\alpha^2}$ for the MAP estimator. This has been set to a default value of zero to simulate the ML estimator discussed in class. In this part, you should vary this value from 1e-8 to 1e-4 in multiples of 10 (*i.e.* 1e-8, 1e-7, ..., 1e-4). A larger ratio corresponds to a stronger prior (smaller values of α^2 constrain the parameters $\mathbf{w}$ to lie closer to origin).

   (a) (*10 points*) Plot the training and test errors as a function of the polynomial order using the above 5 MAP estimators and 250 and 50 training points.

   (b) (*10 points*) Describe how the prior affects the estimation results.