6.891 Machine learning and neural networks

Problem set 1

Deadline: September 21, in class

Note: You will need to use MATLAB in this problem set. Information about Matlab can be found on the course web site[1]. Data files for the problems will be made available via the course Athena locker: `/mit/6.891`.

Reading: Lecture notes 1-3; DHS Chapters 1, 2.1–2.6, 5.1–5.8

Problem 1: regression

Here you'll be using a regression method to predict housing prices in suburbs of Boston. The **x** values or input vectors are 12 dimensional with the following interpretation

```
1. CRIM      per capita crime rate by town
2. ZN        proportion of residential land zoned for lots over
             25,000 sq.ft.
3. INDUS     proportion of non-retail business acres per town
4. CHAS      Charles River dummy variable (= 1 if tract bounds
             river; 0 otherwise)
5. NOX       nitric oxides concentration (parts per 10 million)
6. RM        average number of rooms per dwelling
7. AGE       proportion of owner-occupied units built prior to 1940
8. DIS       weighted distances to five Boston employment centres
9. RAD       index of accessibility to radial highways
10. TAX      full-value property-tax rate per $10,000
11. PTRATIO  pupil-teacher ratio by town
12. LSTAT    % lower status of the population
```

The y values or output scalars are the median value (in \$1000's) of owner-occupied homes in that area. Your goal is to build a model that can accurately predict y given an $\mathbf{x}$. You will find the data in four files, "hw1/boston_x.dat," "hw1/boston_y.dat", "hw1/boston_x_test.dat" and "hw1/boston_y_test.dat".

[1] http://www.ai.mit.edu/courses/6.891

a) Model the "boston_x.dat" and "boston_y.dat" data via linear regression using squared-error as the criterion to minimize. In other words $y = f(\mathbf{x}; \hat{\mathbf{w}}) = \hat{w}_0 + \sum_{i=1}^{12} \hat{w}_i x_i$, where $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{t=1}^{n} (\mathbf{y_t} - f(\mathbf{x_t}; \mathbf{w}))^2$; here n is the number of training examples.

   (note: MATLAB wants all vector indexes to be greater or equal to one; $w(0)$ won't work).

   After having created your linear regression model, use it to compute the mean squared error on the training instances and, separately, on the test instances that you can find in "boston_x_test.dat" and "boston_y_test.dat".

   Turn in the mean squared errors and your Matlab code.

b) Now that you've warmed up your Matlab skills, it's time to see if you can handle something a bit more complicated. For this part, do the same as in part a), but with an additive model using quadratic basis functions. In other words,

$$f(\mathbf{x}; \hat{\mathbf{w}}) = \hat{w}_0 + \sum_{i=1}^{12} \hat{w}_i x_i + \sum_{i \leq j} \hat{w}_{ij} x_i x_j \tag{1}$$

   Hint: simply treat this as a linear regression problem after numerically expanding the data matrices into larger feature matrices.

c) Compare the results from a) and b). Which regression model would you use? **(max 2 sentences)**.

d) Examine the linear coefficient you get in part b). Roughly, how many relevant basis functions do you think there are? Do you think that a linear regression model that would use only these "relevant" basis functions would achieve a lower mean squared error on the training/test set? **(max 2 sentences)**.

Problem 2: classification

Here you'll solve and analyze a classification problem involving binary hand-written digits similar to the ones you have already seen in lectures. The digits are provided to you in four data files: "hw1/digit_x.dat", "hw1/digit_y.dat", "hw1/digit_x_test.dat", and "hw1/digit_y_test.dat". You can view the digits in MATLAB, e.g., as follows

```
>> cd /mit/6.891/hw1
>> load digit_x.dat;
>> prettydigit(digit_x(1,:))
```

The digits that you see in the training and test sets are "3"s and "5"s. We have assigned $y = 1$ for all "3"s and $y = 0$ for all "5"s.

a) Estimate two multivariate Gaussian probability distributions, $P(\mathbf{x}|\mu_1, \Sigma)$ and $P(\mathbf{x}|\mu_0, \Sigma)$, for the two types of digits based on their respective training examples in "hw1/digit_x.dat". NOTE THAT THE COVARIANCES ARE FORCED TO BE EQUAL IN THE TWO DISTRIBUTIONS. Turn in just your MATLAB code, not the resulting means and the covariance matrix.

b) Show that when the covariance matrices are indeed equal, the posterior class probability $P(y|\mathbf{x}, \mu_1, \mu_0, \Sigma)$ computed from such a Gaussian mixture model, i.e.,

$$P(y = 1|\mathbf{x}, \mu_1, \mu_0, \Sigma) = \frac{P(\mathbf{x}|\mu_1, \Sigma)P(y = 1)}{P(\mathbf{x}|\mu_1, \Sigma)P(y = 1) + P(\mathbf{x}|\mu_0, \Sigma)P(y = 0)} \tag{2}$$

conforms to a logistic regression model:

$$P(y = 1|\mathbf{x}, \mathbf{w}) = g(\, w_0 + \sum_{i=1}^{d} w_i x_i \,) \tag{3}$$

Turn in a closed form expression for the linear coefficients $\mathbf{w}$.

From now on, we will assume that the prior frequencies are equal: $P(y = 1) = P(y = 0)$. This is indeed true in the training set "hw1/digit_x.dat".

c) We would like you to use a gradient ascent method to estimate the parameters of the logistic regression model. To this end, compute

$$\frac{\partial}{\partial w_i} \log P(y|\mathbf{x}, \mathbf{w}) \tag{4}$$

and turn in your brief derivation.

d) Perform on the order of 100 iterations of the gradient ascent update rule

$$w_i \leftarrow w_i + \epsilon \sum_{t=1}^{n} \frac{\partial}{\partial w_i} \log P(y_t|\mathbf{x}_t, \mathbf{w}), \quad i = 0, \ldots, 64 \tag{5}$$

Set the learning rate to $\epsilon = 1/\sqrt{n\,d}$, where n is the number of training examples, and set all the weights w_i initially to zero.

Turn in your MATLAB code (not the resulting weights $\mathbf{w}$).

e) Compute the number of classification errors on the test set ("hw1/digit_x_test.dat", "hw1/digit_y_test.dat") BOTH for the logistic regression model from part d) and for the Gaussian mixture (assuming equal prior frequencies) from part a). Briefly explain any differences. **(max 2 sentences)**

Hint: use the result of part b) to deal with the Gaussian mixture.