CS 335 Homework 3, Fall 2014

Dan Sheldon

Revision History

- Thursday Oct. 19: due date updated
- Friday Oct. 3: first posted

Instructions

All parts are due Friday 10/15 by noon

What to submit:

- `<yourlastname>_hw3.zip` - a single zip file of the directory containing all of your code
- `report.pdf` - your written work, unless you submit it by hard copy

Please submit scanned or typed reports as pdf files (as opposed to .docx, .jpg, etc)

Digital files should be submitted via moodle. Your answers to the other questions can be submitted as a pdf by moodle or as a hard copy in the dropbox outside Clapp 222B.

Problems

Let $g(z) = \dfrac{1}{1+e^{-z}}$ be the logistic function.

1. (5 points) Show that $\frac{d}{dz} g(z) = g(z)(1 - g(z))$.
2. (5 points) Show that $1 - g(z) = g(-z)$.
3. (5 points extra credit). Consider the log loss function for logistic regression simplified so there is only one training example:

$$J(\boldsymbol{\theta}) = -y \log h_{\boldsymbol{\theta}}(\mathbf{x}) - (1-y)\log(1 - h_{\boldsymbol{\theta}}(\mathbf{x})), \qquad h_{\boldsymbol{\theta}}(\mathbf{x}) = g(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$

Show that the partial derivative with respect to θ_j is:

$$\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) = (h_{\boldsymbol{\theta}}(\mathbf{x}) - y) x_j$$

.

4. **Logistic regression**. In this problem, you will implement logistic regression for book classification. Open, read, and run the file `books_classify.m`. It does the following:

   - Loads a data set for predicting whether a book is hardcover or paperback from two input features: the thickness of the book and the weight of the book
   - Normalizes the features
   - Has a placeholder for your implementation of logistic regression
   - Plots the data and the decision boundary of the learned model

   First, complete and test the functions in the following files:

   - `logistic.m`
   - `cost_function.m`
   - `gradient_descent.m`

   Follow the instructions in the comments in each file. After you complete the three functions, follow the instructions inside `books_classify.m` to do the following:

   - Learn $\boldsymbol{\theta}$ using gradient descent
   - Print the accuracy of the learned classifier on the training set
   - Plot the cost history
   - Tune the step size and number of iterations as necessary until the algorithm converges and the decision boundary looks sensible.

5. **SMS Spam Classification**. In this problem you will use your implementation of logistic regression to create a spam classifier for SMS messages. Each line of the data file `sms.txt` contains a label—either "spam" or "ham" (i.e. non-spam)—followed by a text message. Here are a few examples:[1]

   ```
   ham     Ok lar... Joking wif u oni...
   ham     Nah I don't think he goes to usf, he lives around here though
   spam    Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005.
           Text FA to 87121 to receive entry question(std txt rate)
           T&C's apply 08452810075over18's
   spam    WINNER!! As a valued network customer you have been
           selected to receivea £900 prize reward! To claim
           call 09061701461. Claim code KL341. Valid 12 hours only.
   ```

   To create features suitable for logistic regression, the following processing steps have already been applied:[2]

   - Words were converted to lowercase.
   - Punctuation and special characters were removed (but the $ and £ symbols are helpful for predicting spam and were preserved as a special token).
   - A dictionary was created containing the 2000 words that appeared most frequently in the entire set of messages.
   - Each message was encoded as a vector $\mathbf{x}^{(i)} \in \mathbb{R}^{2000}$. The entry $x_j^{(i)}$ is equal to the number of times the jth word in the dictionary appears in that message.
   - Some ham messages were discarded to have an equal number of spam and ham messages.
   - The data was split into a training set of 1000 messages and a test set of 400 messages.

[1] Line breaks were added for readability.

[2] If you are interested, you can read the script sms_prep.m to see how this was done.

Your job is to complete the code in `sms_classify.m`. There is already code to load the processed data set into the MATLAB workspace. See the comments in the file for explanations of the variables `X_train`, `y_train`, `X_test`, `y_test`, and `dict`. Follow the instructions in the file and complete code to do the following:

- Learn $\boldsymbol{\theta}$ by gradient descent
- Plot the cost history
- Make predictions and report the accuracy on the test set
- Test out the classifier on a few of your own text messages. Does it get them right?

6. (Extra credit) Use your own data—either SMS or email—to create a personalized spam classifier. You can either put it in the same format as `sms.txt` and use the script `sms_prep.m` to build the dictionary and features, or you can write your own scripts to import the data.