STAT 339: HOMEWORK 4 (APPROXIMATE INFERENCE AND CLUSTERING)

UPDATED: DUE ON BLACKBOARD BY CLASS TIME MONDAY 4/10)

Instructions. Turn in your writeup and code to Blackboard as an archive file (e.g., `.zip`, `.tar`, `.gz`) by the start of class on Monday 4/10. **Note: To make grading smoother, please include a main writeup file in your archive in pdf form with a file name like `cdawson.pdf` (sub your Obie ID). All plots and results should be included and described here, with references as appropriate to implementation files.**

As always, you may use any language you like for the progamming components of this assignment — the tasks are stated in a language-neutral way. You may also use any typesetting software to prepare your writeup, but the final document should be a PDF. LaTeXis encouraged; a reproducible research format in which code is embedded into the document (e.g., knitr, RMarkdown, Jupyter or IPython Notebook) is even more encouraged.

All data is available at `http://colindawson.net/data/<name>.csv`.

Date: Last Revised: April 3, 2017.

1. **(15 pts) Approximate Inference and Logistic Regression.** Implement a logistic regression solver using (i) maximum likelihood and (ii) a $\mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$ prior on the weights. In each case, use Newton-Raphson to find the MLE and the MAP estimates of $\mathbf{w}$, respectively. In the Bayesian case, approximate the full posterior using the Laplace approximation.

   (a) Test your solver on some of the binary classification datasets we have used before: `S1{Train/Test}.csv`, `S2{Train/Test}.csv` / `cancer_{train/test}.csv`.

   (b) For (at least) the 2D `S2` synthetic data, produce two plots, one for the MLE solver and one for the posterior predictive solver. In each plot, plot the data, color-coded by class, as well as a grid of points color-coded according to their predicted class probabilities using a gradient scheme that maps 0 to one extreme and 1 to another and interpolates between them (e.g., red vs. blue, with purple representing uncertainty), so we can see where the regions of high and low certainty are. In the posterior predictive case, you will need to sample weights from the approximated posterior, compute the class probabilities according to these weights, and average the probabilities over the sampled weights. You can use a library function (such as `numpy.random.multivariate_normal()`) to do the sampling.

   (c) Comment on the differences between the MLE and posterior-predictive methods.

   (d) For both the S2 and the cancer data, evaluate the two models on the test set by computing a weighted misclassification rate (i.e., the error for point n is the difference between the class label and the predicted probability of being in that class).

2. **(5 pts) Laplace Approximation for a Beta Distribution.** Find a formula for the Laplace approximation to a $\mathsf{Beta}(a, b)$ distribution. Plot the approximation against the true density for the following parameter values:

   (a) $a = 10, b = 10$

   (b) $a = 5, b = 15$

   (c) $a = 2, b = 18$

   (d) $a = 20, b = 180$

   and comment on the quality of the approximation in each case.

3. **(15 pts) Clustering with a Gaussian Mixture Model and the EM algorithm.** Implement the EM algorithm to find local maximum likelihood parameter values for a Gaussian Mixture model with a fixed number of clusters, K. The parameters are the D-dimensional mean vectors for each cluster $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, the $D \times D$ covariance matrices for each cluster, $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K$, and the mixture weights $\pi_1, \ldots, \boldsymbol{\pi}_K$. Your algorithm should also assign at each iteration the posterior "responsibility" vectors, q_n, $n = 1, \ldots N$, each of which is a K-dimensional array of weights that sum to 1. You may assume that the covariance matrices are diagonal, or if you want, tackle the more general case. Optional: implement K-means to initialize the cluster centers.

   (a) In the case where $D = 2$, your program should plot the data, color coding each point according to its maximum a posteriori cluster (or you could do something more sophisticated to depict the responsibilities instead of doing hard clustering).

   (b) Test your model with various K values and various initializations on the two petal dimensions of the iris data (Note that we are acting as though we don't know the species or even how many species there are), plotting the training set log likelihood by iteration.

   (c) Comment on how sensitive the algorithm is to the initial conditions, and what effect the choice of K has on the final training log likelihood.

4. **(5 pts) Cross-Validation in Clustering.** Implement 10-fold cross-validation to select the best value of K by maximizing the mean log likelihood on the validation set. Your code should plot the training and validation log likelihoods (normalize by sample size so that training and validation likelihood are on the same scale), for each K. Test your algorithm on the iris data: does it discover the correct number of clusters? Since we actually know the true species, how well do the clusters match the true species?

5. **(5 pts) Applying clustering to cancer microarray data.** This problem should not require any new implementation; just applying what you did in the previous two problems to a new dataset. The file `nci60_reduced.csv` comes from a dataset in which cell lines from 64 cancerous tumors were analyzed using a "microarray", which measures the degree to which particular genes are expressed in the sample. In the original data (obtained via the R package accompanying the ISL textbook), 6830 gene expression measurements were taken from every cell line. However, we do not want to cluster data with this many dimensions using

the simple techniques we have learned; so as in the digits data in HW3, I have preprocessed the data using Principle Components Analysis (PCA) to reduce the number of dimensions to 4.

(a) Using a GMM, estimating parameters using EM, and using 10-fold cross-validation, find a suitable number of clusters for this data, and plot the first two dimensions of the data, labeling by maximum likelihood cluster for your final choice of K.

(b) The diagnosed cancer types are listed in the file `nci_labels.csv`. Investigate the extent to which your clusters line up with the human-provided catgories.