STATS 306B Methods for Applied Statistics: Unsupervised Learning
Stanford University, Spring 2014

Problem Set 3

Due: Monday, May 19, 2014 (at the start of class)

Instructions:

- Please submit any code written for this assignment along with your derivations and plots.

Problem 1 (Kernelizing k-means)**.**

(a) Derive an optimization problem that is equivalent to the standard k-means optimization problem but that depends on the data only through the Gram matrix K with entries $K_{ij} = \langle x_i, x_j \rangle$.

(b) Derive the equivalent of Lloyd's algorithm for approximately optimizing the problem in part (a). (Your algorithm can only depend on the data through the kernel matrix K.)

Problem 2 (Compressed Probabilistic PCA and Collaborative Filtering)**.** Suppose that $y_1, \ldots, y_n \in \mathbb{R}^p$ are sampled i.i.d. from the following global-mean-zero probabilistic PCA model

- $z_i \overset{\text{iid}}{\sim} \mathcal{N}(0, I_{q \times q})$
- $y_i | z_i \overset{\text{ind}}{\sim} \mathcal{N}(\Lambda z_i, \sigma^2 I)$

for unknown parameters $\Lambda \in \mathbb{R}^{p \times q}$ and $\sigma^2 > 0$. However, you do not observe the y_i's directly. Rather, you observe a compression of each vector $x_i = P_i y_i \in \mathbb{R}^{d_i}$, where d_i is the known dimension of datapoint i, and P_i is an **arbitrary** known matrix in $\mathbb{R}^{d_i \times p}$ with full row rank. The complete log likelihood of $x_{1:n}$ and $z_{1:n}$ under this model has the form

$$\begin{aligned} &\log p(x_{1:n}, z_{1:n}; \theta) \\ &\quad = -\frac{1}{2} \sum_{i=1}^{n} \left[d_i \log \sigma^2 + \frac{1}{\sigma^2} (x_i - P_i \Lambda z_i)^\top (P_i P_i^\top)^{-1} (x_i - P_i \Lambda z_i) \right] + \text{constant} \end{aligned}$$

where θ represents all unknown parameters.

(a) Describe the E step of an EM algorithm based on this CLL.

(b) Compute the gradient of the ECLL with respect to the matrix Λ. (See Section 14.5 of the Factor Analysis chapter in your reading for instructions on computing gradients with respect to a matrix.)

(c) From this point on, assume that P_i is a projection matrix that selects a subset of coordinates $\mathcal{S}_i = \{j_1, \ldots, j_{d_i}\} \subset \{1, \ldots, p\}$ of y_i. That is, $P_i = \sum_{l=1}^{d_i} e_l e_{j_l}^\top = \sum_{j=1}^{p} \sum_{l=1}^{d_i} \mathbb{I}(j = j_l)\, e_l e_j^\top$, where e_l represents a standard basis vector (of appropriate dimension) with a 1 in entry l and

zeros elsewhere. Show that the gradient of the ECLL computed in the previous part can be rewritten as

$$\frac{1}{\sigma^2}\left[\sum_{j=1}^{p} e_j \sum_{i=1}^{n}\sum_{l=1}^{d_i} \mathbb{I}\left(j = j_l\right) x_{il} \mathbb{E}_{q_t}\left[z_i^\top\right]\right] - \frac{1}{\sigma^2}\left[\sum_{j=1}^{p} e_j \Lambda_{j\cdot} \sum_{i=1}^{n} \mathbb{I}\left(j \in \mathcal{S}_i\right) \mathbb{E}_{q_t}\left[z_i z_i^\top\right]\right]$$

where $\Lambda_{j\cdot} = e_j^\top \Lambda$ is the j-th row of Λ. This means that we can solve for each row of Λ separately to obtain an M-step update:

$$\Lambda_{j\cdot}^{(t+1)} = \sum_{i=1}^{n}\sum_{l=1}^{d_i} \mathbb{I}\left(j = j_l\right) x_{il} \mathbb{E}_{q_t}\left[z_i^\top\right] \left(\sum_{i=1}^{n} \mathbb{I}\left(j \in \mathcal{S}_i\right) \mathbb{E}_{q_t}\left[z_i z_i^\top\right]\right)^{-1}.$$

Note that in our coordinate projection setting, we have direct access to certain coordinates of y_i, those that appear in $\mathcal{S}_i$. Hence, we can state the above update equation more simply in terms of y_i:

$$\Lambda_{j\cdot}^{(t+1)} = \sum_{i=1}^{n} \mathbb{I}\left(j \in \mathcal{S}_i\right) y_{ij} \mathbb{E}_{q_t}\left[z_i^\top\right] \left(\sum_{i=1}^{n} \mathbb{I}\left(j \in \mathcal{S}_i\right) \mathbb{E}_{q_t}\left[z_i z_i^\top\right]\right)^{-1}.$$

You should use this update equation from now on.

(d) Find the M-step update for σ^2 in terms of $\Lambda^{(t+1)}$.

(e) The file `ratings-train.RData` (or, alternatively, `ratings-train.mat`) contains a subset of the Netflix Prize movie rating database stored as a matrix with 1000 rows and 100 columns. Every column of the matrix represents a different DVD title, every row represents a different customer, and every entry represents a preference rating in $\{1, \ldots, 5\}$ assigned to a particular title by a particular customer. Since not every customer rated every title, some entries are missing, and these are denoted by NAs. (The subset provided for this problem is unusually complete; only 1% of entries were observed in the full Netflix Prize dataset.) The file `ratings-test.RData` (or, alternatively, `ratings-test.mat`) contains a smaller set of additional ratings (encoded as the non-NA entries of a matrix) that should be withheld from the training set. Withhold these test entries from the training data by setting the corresponding training entries to NA.

Next, approximately mean center the training data by subtracting away the mean of observed entries in each column from each non-NA entry in that column. Subtract these column means (computed from the training data) from the observed entries of the corresponding columns of the test data matrix as well.

Finally, use the EM algorithm just developed to fit a compressed probabilistic PCA model with $q = 10$ factors to the incomplete data in this matrix by letting y_i represent the complete i-th row of this matrix (which was not fully observed), x_i represent the subvector of d_i observed entries in row i, and P_i represent the projection onto the observed d_i entries in row i.

(f) Let Ω represent the set of customer-title pairs (i, j) with ratings in the test file. Use the parameters learned in part (b) to infer the conditional means of these ratings under the factor analysis model given the observed data from part (b). Compare these conditional means $\hat{r}_{ij}$ to the true test ratings r_{ij} by computing (and reporting) the root mean squared error, $\sqrt{\frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}(\hat{r}_{ij} - r_{ij})^2}$.

Optional (for fun): Try swapping the roles of the test and training datasets above. Fit your EM algorithm (with a smaller number of factors like $q = 2$ or $q = 4$) using only the (appropriately mean-centered) test dataset and attempt to infer the unseen training dataset ratings. How accurate are your estimates?

Problem 3 (ICA and PCA)**.**

(a) The dataset `ica-data.csv` containing 1000 datapoints of dimension 3 was generated using the script `ica-data.R`. Write your own ICA procedure to carry out a FastICA analysis of this dataset (with 3 extracted components) following the treatment of Hyvärinen and Oja in Section 6.1 and Equation (44) of `http://www.sciencedirect.com/science/article/pii/S0893608000000265` with $G(u) = \log\cosh(u)$. Comment on the results.

(b) Perform a principal component analysis (with $k = 3$) of the same data, and compare the results.

Problem 4 (Canonical Correlation Analysis)**.** The file `hiv_phenotype.csv` contains information on several different HIV sequences (column names starting with "p") as well as the resistance of these viruses measured in vitro (column names starting with "fold"). The data is a subset of the database `http://hivdb.stanford.edu`. This response, called the phenotypic response, is the fold change replication capacity of these mutated viruses relative to "wildtype" HIV viruses to different protease inhibitors (ATV, DRV, FPV, IDV, LPV, NFV, SQV, TPV) a class of drugs used to treat HIV patients. An increase in fold change for a given virus indicates resistance to a particular drug, relative to wildtype.

(a) Carry out a standard canonical correlation analysis on the pair of datasets (the mutations and the log fold changes). Discard datapoints with missing values as needed. Describe how you selected the number of components to extract.

(b) Are the canonical directions for the mutations interpretable in the sense of having many values close to or equal to 0? What criterion do the canonical directions / vectors maximize? Can you suggest a modification that would yield canonical vectors that have values set to 0?

Problem 5 (Feedback)**.** (This "problem" is not graded.)

(a) How much time did you spend on this problem set?

(b) Which problems did you find valuable?