Homework 2
EM, Mixture Models, PCA, Dualitys

CMU 10-715: Machine Learning (Fall 2015)
http://www.cs.cmu.edu/~bapoczos/Classes/ML10715_2015Fall/
OUT: Oct 5, 2015
DUE: Oct 19, 2015, 10:20 AM

Guidelines

- The homework is due at 10:20 am on Monday October 19, 2015. Each student will given two late days that can be spent on any homeworks, but at most one late day per homework. Once you have used up your late days for the term, late homework submissions will receive no credit.
- Submit both a paper copy and an electronic copy through through the submission website: https://autolab.cs.cmu.edu/courses/10715-f15. You can sign in using your Andrew credentials. You should make sure to edit your account information and choose a nickname/handle. This handle will be used to display your results for any competition style questions on the class leaderboard.
- Some questions will be *autograded*. Please make sure to carefully follow the submission instructions for these questions.
- We recommend that you typeset your solutions using software such as LaTeX.If you write, ensure your handwriting is clear and legible. The TAs will not invest undue effort to decrypt bad handwriting.
- Programming guidelines:
  - **Octave:** You must write submitted code in Octave. Octave is a free scientific programming language, with syntax similar to that of MATLAB. Installation instructions can be found on the Octave website. (You can develop your code in MATLAB if you prefer, but you *must* test it in Octave before submitting, or it may fail in the autograder.)
  - **Autograding:** This problem is autograded using the CMU Autolab system. The code which you write will be executed remotely against a suite of tests, and the results used to automatically assign you a grade. To make sure your code executes correctly on our servers, you should avoid using libraries which are not present in the *basic* Octave install.
  - **Submission Instructions:** For each programming question you will be given a function signature. You will be asked to write a single Octave function which satisfies the signature. In the code handout linked above, we have provided you with a single folder containing stubs for each of the functions you need to complete. *Do not modify the structure of this directory or rename these files.* Complete each of these functions, then compress this directory *as a tar file* and submit to Autolab online. You may submit code as many times as you like.

    When you download the files, you should confirm that the autograder is functioning correctly by compressing and submitting the directory of stubs provided. This should result in a grade of zero for all questions.
  - **SUBMISSION CHECKLIST**
    - Submission executes on our machines in less than 20 minutes.
    - Submission is smaller than 2000K.
    - Submission is a `.tar` file.
    - Submission returns matrices of the *exact* dimension specified.

1 An EM algorithm for a Mixture of Bernoullis [Eric; 25 pts]

In this section, you will derive an expectation-maximization (EM) algorithm to cluster black and white images. The inputs $x^{(i)}$ can be thought of as vectors of binary values corresponding to black and white pixel values, and the goal is to cluster the images into groups. You will be using a mixture of Bernoullis model to tackle this problem.

For the sake of brevity, you do not need to substitute in previously derived expression in later problems. For example, beyond question 1.1.1 you may use $P(x^{(i)}|p^{(k)})$ in your answers.

1.1 Mixture of Bernoullis

1. (2pts) Consider a vector of binary random variables, $x \in \{0,1\}^D$. Assume each variable x_d is drawn from a Bernoulli(p_d) distribution, so $P(x_d = 1) = p_d$. Let $p \in (0,1)^D$ be the resulting vector of Bernoulli parameters. Write an expression for $P(x|p)$.

2. (2pts) Now suppose we have a mixture of K Bernoulli distributions: each vector $x^{(i)}$ is drawn from some vector of Bernoulli random variables with parameters $p^{(k)}$, we will call this Bernoulli($p^{(k)}$). Let $\{p^{(1)}, \ldots, p^{(K)}\} = \mathbf{p}$. Assume a distribution $\pi(k)$ over the selection of which set of Bernoulli parameters $p^{(k)}$ is chosen. Write an expression for $P(x^{(i)}|\mathbf{p}, \pi)$.

3. (2pts) Finally, suppose we have inputs $X = \{x^{(i)}\}_{i=1\ldots n}$. Using the above, write an expression for the log likelihood of the data X, $\log P(X|\pi, \mathbf{p})$.

1.2 Expectation step

1. (4pts) Now, we introduce the latent variables for the EM algorithm. Let $z^{(i)} \in \{0,1\}^K$ be an indicator vector, such that $z_k^{(i)} = 1$ if $x^{(i)}$ was drawn from a Bernoulli($p^{(k)}$), and 0 otherwise. Let $Z = \{z^{(i)}\}_{i=1\ldots n}$. What is $P(z^{(i)}|\pi)$? What is $P(x^{(i)}|z^{(i)}, \mathbf{p}, \pi)$?

2. (2pts) Using the above two quantities, derive the likelihood of the data and the latent variables, $P(Z, X|\pi, \mathbf{p})$.

3. (5pts) Let $\eta(z_k^{(i)}) = E[z_k^{(i)}|x^{(i)}, \pi, \mathbf{p}]$. Show that

$$\eta(z_k^{(i)}) = \frac{\pi_k \prod_{d=1}^D (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^{(k)})^{1 - x_d^{(i)}}}{\sum_j \pi_j \prod_{d=1}^D (p_d^{(j)})^{x_d^{(i)}} (1 - p_d^{(j)})^{1 - x_d^{(i)}}}$$

Let $\tilde{\mathbf{p}}, \tilde{\pi}$ be the new parameters that we'd like to maximize, so $\mathbf{p}, \pi$ are from the previous iteration. Use this to derive the following final expression for the E step in the expectation-maximization algorithm:

$$E[\log P(Z, X|\tilde{\mathbf{p}}, \tilde{\pi})|X, \mathbf{p}, \pi] = \sum_{i=1}^N \sum_{k=1}^K \eta(z_k^{(i)}) \left[\log \tilde{\pi}_k + \sum_{d=1}^D \left(x_d^{(i)} \log \tilde{p}_d^{(k)} + (1 - x_d^{(i)}) \log(1 - \tilde{p}_d^{(k)}) \right) \right]$$

1.3 Maximization step

1. (4pts) We need to maximize the above expression with respect to $\tilde{\pi}, \tilde{\mathbf{p}}$. First, show that the value of $\tilde{\mathbf{p}}$ that maximizes the E step is

$$\tilde{p}^{(k)} = \frac{\sum_{i=1}^N \eta(z_k^{(i)}) x^{(i)}}{N_k}$$

where $N_k = \sum_{i=1}^N \eta(z_k^{(i)})$.

2. (4pts) Show that the value of $\tilde{\pi}$ that maximizes the E step is

$$\tilde{\pi}_k = \frac{N_k}{\sum_{k'} N_{k'}}$$

The exponential families notation may be useful. Alternatively, you can use Lagrange multipliers.

2 Clustering Images of Numbers [Eric; 25 pts]

In this section you will use the above algorithm to cluster images of numbers. You will be using the MNIST dataset. Each input is a binary number corresponding to black and white pixels, and is a flattened version of the 28x28 pixel image.

We will use the following conventions:

- N is the number of datapoints, D is the dimension of each input, and K is the number of clusters.
- Xs is an $N \times D$ matrix of the input data, where row i is the pixel data for picture i.
- p is a $K \times D$ matrix of Bernoulli parameters, where row k is the vector of parameters for the kth mixture of Bernoullis.
- mix_p is a $K \times 1$ vector containing the distribution over the various mixtures.
- eta is a $N \times K$ matrix containing the results of the E step, so eta[i,k] $= \eta(z_k^{(i)})$.
- clusters is an $N \times 1$ vector containing the final cluster labels of the input data. Each label is a number from 1 to K.

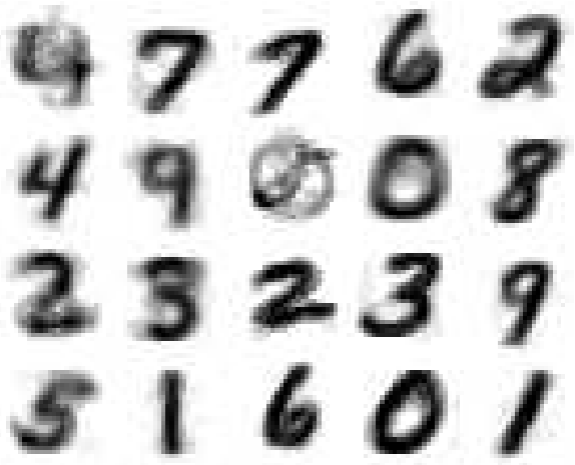
2.1 Programming (20pts)

1. Implement the E step of the algorithm within [eta] = Estep(Xs,p,mix_p), saving your calculated values within eta.
2. Implement the M step of the algorithm within [p,mix_p] = Mstep(Xs, model, alpha1, alpha2). p and mix_p returned by this function should contain the new values that maximize the E step. alpha1, alpha2 are Dirichlet smoothing parameters (explained below).
3. Implement [clusters] = MoBlabels(Xs,p,mix_p). This function will take in the estimated parameters and return the resulting labels that cluster the data.
4. Some hints and tips:
   - The autograder will separately grade the accuracy of your implementation separately for the E-step (4pts), M-step (4pts), and M-step with smoothing (2pts). Finally, it will run your implementation for several iterations on a subset of the MNIST dataset (8pts). A small subset of the MNIST dataset is provided for your convenience.
   - Use the log operator to make your calculations more numerically stable. In particular, pay attention to the calculation of $\eta(z_k^{(i)})$.
   - You will need to avoid zeros in π and $\mathbf{p}$ or else you will take $\log(0) = -\infty$. Use Dirichlet prior smoothing with the parameters α_1, α_2 when updating these variables:

$$\tilde{p}^{(k)} = \frac{\sum_{i=1}^{N} \eta(z_k^{(i)}) x^{(i)} + \alpha_1}{N_k + \alpha_1 D}$$

$$\tilde{\pi}_k = \frac{N_k + \alpha_2}{\sum_{k'} N_{k'} + \alpha_2 K}$$

   - Initialize your parameters $\mathbf{p}$ by randomly sampling from a Uniform$(0, 1)$ distribution and normalizing each $p^{(k)}$ to have unit length, and $\pi_k = 1/k$.
   - Running your implementation on the MNIST dataset with $K = 20$ clusters and $\alpha_1 = \alpha_2 = 10^{-8}$ for 20 iterations should give you some results similar to the following (our reference code runs in approximately 10 seconds):



2.2 Analysis

1. (5pts) For each cluster, reshape the pixels into a 28x28 matrix and print the resulting grayscale images. What do you see? Explain, and include one such image in your writeup. See https://www.gnu.org/software/octave/doc/interpreter/Representing-Images.html for help on printing the matrix, or use the provided helper function show_clusters(p, a, b) which will print the mixtures in p in an $a \times b$ grid.

2. (2pts) Using your implemented MoBlabels function, cluster the data. Using the true labels given in yTrain, how many unique digits does each cluster typically have? Are there any clusters that picked out exactly one digit?

3 Kernel PCA [Fish; 15 pts]

Principal component analysis (PCA) is used to emphasize variation and is often used to make data easy to explore and visualize. Suppose we have data $x_1, x_2, \cdots x_N \in \mathbb{R}^d$ that has zero mean. PCA aims at finding a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate, the second greatest variance on the second coordinate, and so on. The basis vectors of the new coordinate system are the eigenvectors of the co-variance matrix, i.e.,

$$\frac{1}{N}\sum_{i=1}^{N} x_i x_i^T = \sum_{i=1}^{d} \lambda_i v_i v_i^T, \tag{1}$$

where $v_1, v_2, \cdots, v_d$ are orthogonal ($< v_i, v_j >= 0$ for $i \neq j$). Then we can plot our data points on our new coordinate system. The position of each data points in the new coordinate system can be derived by projecting x on to the basis of the new coordinate system, i.e., $v_1, v_2, \cdots v_d$.

3.1 kernel PCA

Often we will want to make linear or non-linear transformation on the data so that they are projected to a higher dimensional space. Suppose there is a transformation function $\phi : \mathbb{R}^d \to \mathbb{R}^l$. We map all the data to a new space through this function, and we have $\phi(x_1), \phi(x_2), \cdots, \phi(x_N)$. We again want to do PCA on the transformed data in the new space. How can we do this?

1. (2pts) Write out the co-variance matrix, C, for $\phi(x_1), \phi(x_2), \cdots, \phi(x_N)$. (Define $\overline{\phi(x)} = \frac{1}{N}\sum_{j=1}^{N} \phi(x_j)$)

2. (2pts) Now we want to find the basis vectors for the orthogonalized new space by solving eigenvectors of C. Using the definition of an eigenvector, $\lambda v = Cv$, explain why v is a linear combination of $\left(\phi(x_1) - \overline{\phi(x)}\right), \left(\phi(x_2) - \overline{\phi(x)}\right), \cdots, \left(\phi(x_N) - \overline{\phi(x)}\right)$. Since v is a linear combination of these vectors, $v = \sum_{i=1}^{N} \alpha_i \left(\phi(x_i) - \overline{\phi(x)}\right)$.

3. (2pts) Before starting to derive α and find out v, we introduce kernel function here. A kernel function is in the form of $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. Notice that we do not need to know the function ϕ to calculate $k(x_i, x_j)$. We can simply define a function that is related to x_i, x_j and $k(x_i, x_j) = k(x_j, x_i)$. A classic example is Radial basis function (RBF) kernel, which is $k(x_i, x_j) = \exp(-||x_i - x_j||^2/2\sigma^2)$. In order to use these kernel function, we need to get rid of all the $\phi(x_i)$ and replace them with the kernel function. A kernel matrix is defined as $K \in \mathbb{R}^{N \times N}$ and $K_{ij} = k(x_i, x_j)$. Since we are dealing with non-centered data, we first use a modified kernel matrix $\tilde{K}_{ij} = \left\langle \left(\phi(x_i) - \overline{\phi(x)}\right), \left(\phi(x_j) - \overline{\phi(x)}\right) \right\rangle$. By using the results in question 3.1.1 and 3.1.2 and the definition of eigenvectors, show that

$$N\lambda\tilde{K}\alpha = \tilde{K}^2\alpha. \tag{2}$$

4. (2pts) Show that the solutions α in

$$N\lambda\alpha = \tilde{K}\alpha. \tag{3}$$

are also solutions for equation 2.

5. (3pts) Show that

$$\tilde{K} = (I - ee^T)K(I - ee^T). \tag{4}$$

Thus, by solving the eigenvectors of $\tilde{K}$, we get α.

6. (2pts)After obtaining α, we get the un-normalized basis vector v. To normalize it, what is the factor you need to multiply to v ?

7. (2pts) For a data point x, what is its position in the normalized new space? Explain why you can get the new coordinate without explicitly calculate $\phi(x)$.

4 Huber function and its application in solving dual problem [Fish; 35pts]

4.1 Huber function

Define a function

$$B(x, d) = \min_{\lambda \geq 0} \lambda + \frac{x^2}{\lambda + d}. \tag{5}$$

where $d > 0$.

1. (3pts) Show that the function

$$B(x, d) = \begin{cases} \frac{x^2}{d} & \text{if } |x| \leq d \\ 2|x| - d & \text{if } |x| > d \end{cases} \tag{6}$$

with the minimizer $\lambda^* = \max(0, |x| - d)$. B is often called a huber function.

2. (3pts) Is the function B convex in x? (Assume d is a fixed constant)

3. (3pts) Derive the gradient of function B at any given point (x, d). (Remember $d > 0$.)

4. (3pts) Function B is often used as a penalty term in classification or regression problems, which have the form

$$\min_x L(x) + B(x, d), \tag{7}$$

where L is the loss function, and d is a parameter with positive value. Describe how parameter d affects the penalty term B on the solution.

4.2 An application of using Huber loss to solve dual problem

Consider the optimization problem

$$\min_x \sum_{i=1}^{N} (\frac{1}{2} d_i x_i^2 + r_i x_i) \tag{8}$$

$$\text{s.t. } a^T x = 1, x_i \in [-1, 1] \text{ for } i = 1, 2, \cdots n. \tag{9}$$

1. (3pts) Show that the problem has strictly feasible solutions if and only if $||a||_1 > 1$.
2. (3pts) Re-express $x_i \in [-1, 1]$ as $x_i^2 \leq 1$, for $i = 1, 2, \cdots, n$. Write down the dual of this problem.
3. (6pts) Write down the KKT condition for this problem. Does it characterize the optimal solution?
4. (5pts) Show that we can further reduce the dual problem to a one-dimensional convex problem

$$\min_\mu \mu + \frac{1}{2} \sum_{i=1}^{N} B(r_i + \mu a_i, d_i), \tag{10}$$

where B is the Huber function defined in the previous section.

5. (3pts) Describe an algorithm to solve this dual problem. What is the time complexity for your algorithm?
6. (3pts) How can you recover an optimal primal solution x after solving the dual?