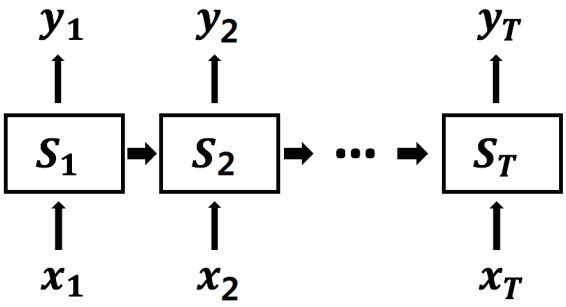
1 Back Propagation Through Time

Back propagation through time (BPTT) is a gradient-based technique for training certain types of recurrent neural networks (RNN). Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. BPTT begins by unfolding a recurrent neural network through time as shown in the figure below.



Here, $\boldsymbol{x}_t$ are the input sequence to the RNN and $\boldsymbol{s}_t$ are the hidden states of the RNN. Our goal is to predict the output sequence $\boldsymbol{y}_t$. For simplicity, we assume that dimension of input $\boldsymbol{x}_t$, output $\boldsymbol{y}_t$ and hidden state $\boldsymbol{s}_t$ are the same as M. Moreover, we ignore all the biases in our formulation. There are three connection weight matrices W_{IH}, W_{HH} and W_{HO} for the input-hidden, hidden-hidden and hidden-output connections. The behavior of the RNN can be described as the following dynamical system with non-linearity:

$$\boldsymbol{s}_t = \sigma(W_{IH}\boldsymbol{x}_t + W_{HH}\boldsymbol{s}_{t-1}), \boldsymbol{y}_t = W_{HO}\boldsymbol{s}_t,$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

BPTT algorithm is commonly used to train the RNN to obtain the weight matrices W_{IH}, W_{HH} and W_{HO} from training sequences $\{(\hat{\boldsymbol{x}}_{1,\ldots,T}, \hat{\boldsymbol{y}}_{1,\ldots,T})\}$ of length T. In this problem, we use the squared loss as the loss function. Let $\boldsymbol{y}_{1,\ldots,T}$ be the prediction of your RNN, we have loss function

$$L(\boldsymbol{y}_{1,\ldots,T}, \hat{\boldsymbol{y}}_{1,\ldots,T}) = \frac{1}{2}\sum_{i=1}^{T} ||\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i||_2^2.$$

(a) As a first step, you need to compute the gradient of loss function L with respect to $\boldsymbol{y}_t$, $\nabla_{\boldsymbol{y}_t} L$. For this problem, you can assume that there is only one training sequence $(\hat{\boldsymbol{x}}_{1,\ldots,T}, \hat{\boldsymbol{y}}_{1,\ldots,T})$.

(b) In this step, you need to compute the gradient of loss function L with respect to $\boldsymbol{s}_t$, $\nabla_{\boldsymbol{s}_t} L$. You should first compute $\nabla_{\boldsymbol{s}_T} L$ and then express $\nabla_{\boldsymbol{s}_t} L$ in terms of $\nabla_{\boldsymbol{s}_{t+1}} L$. It is where the name BPTT comes from.

(c) Use your answer in last two steps to derive the gradient of loss function L with respect to W_{IH}, W_{HH} and W_{HO}.

(d) When the length of sequence T is large, gradient descend method usually leads to bad performance due to vanishing gradients. As a cure for the problem, leaky hidden units can be

used to learn long range dependency. Namely, we use the following equation to update the hidden states:

$$\boldsymbol{s}_t = (1-\tau)\cdot \boldsymbol{s}_{t-1} + \tau \cdot \sigma(W_{IH}\boldsymbol{x}_t + W_{HH}\boldsymbol{s}_{t-1}),$$

where τ is a constant. You need to compute the gradient of loss function L with respect to W_{IH}, W_{HH} and W_{HO} when the leaky hidden units are used.

2 Kernel K-Means

In this problem, you will apply the kernel tricks to K-means algorithm to make it more powerful. Recall that the K-means algorithm optimizes the following objective: Given a set of data points $\{\mathbf{x}_n\}_{n=1}^N$, the method minimizes the following distortion measure (or objective or clustering cost):

$$D = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|_2^2$$

where $\boldsymbol{\mu}_k$ is the prototype of the k-th cluster. r_{nk} is a binary indicator variable. If $\mathbf{x}_n$ is assigned to the cluster k, r_{nk} is 1 otherwise r_{nk} is 0. For each cluster, $\boldsymbol{\mu}_k$ is the representative for all the data points assigned to that cluster.

Now assume that we apply a mapping $\phi(\mathbf{x})$ to map data points into feature space. Then, we define the objective function of kernel K-means as

$$\tilde{D} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\phi(\mathbf{x}_n) - \tilde{\boldsymbol{\mu}}_k\|_2^2,$$

where $\tilde{\boldsymbol{\mu}}_k$ is the center of the cluster k in the feature space.

(a) Show that the $\tilde{D}$ can be represented in terms of only kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x_i})^{\mathrm{T}}\phi(\mathbf{x_j})$.

(b) Describe and write down the equation of assigning a data point to its cluster. You answer should only consist of kernel value $K(\mathbf{x}_i, \mathbf{x}_j)$.

(c) Write down the pseudo-code of the complete kernel K-means algorithm including initialization of cluster centers.

3 EM algorithm

Zero-inflated Poisson regression is used to model count data that has an excess of zero counts. For example, the number of insurance claims within a population for a certain type of risk would be zero-inflated by those people who have not taken out insurance against the risk and thus are unable to claim. The observed data probability of observation i is:

$$p(x_i) = \begin{cases} \pi + (1-\pi)e^{-\lambda} & \text{if } x_i = 0 \\ (1-\pi)\frac{\lambda^{x_i}e^{-\lambda}}{x_i!} & \text{if } x_i > 0. \end{cases}$$

Your task in this problem is to design an EM to estimate the parameter π and λ from observed data $\{x_i\}_{i=1}^N$.

(a) Define a proper hidden variable z_i for the observations (Hint: you only need hidden variables for some observations) and use them to write down the complete likelihood function.

(b) Write down the update equations for both the E-Step and the M-step.

4 Programming

In this problem, you will implement three different clustering methods, K-means, Kernel K-means and Gaussian Mixture Model. You will evaluate the performance of your method on two synthetic datasets.

4.1 Data

You are provided with two datasets, `hw5_blob.mat` and `hw5_circle.mat`. Both datasets have two dimensions.

4.2 Implement k-means

As we studied in the class, k-means tries to minimize the following distortion measure (or objective function):

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \boldsymbol{\mu}_k||_2^2$$

where r_{nk} is an indicator variable:

$$r_{nk} = 1 \quad \text{if and only if } \mathbf{x}_n \text{ belongs to cluster } k$$

and $(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)$ are the cluster centers with the same dimension of data points.

(a) Implement k-means using random initialization for cluster centers. The algorithm should run until none of the cluster assignments are changed. Run the algorithm for different values of $K \in \{2, 3, 5\}$, and plot the clustering assignments by different colors and markers. (you need to report 6 plots, 3 for each dataset.)

(b) The k-means algorithm fails to separate the two circles in the `hw5_circle.mat` dataset. Please explain why this happens.

4.3 Implement kernel k-means

Implement kernel k-means you derived in Problem 2 and evaluate it on the `hw5_circle.mat` dataset. You should choose a kernel that can separate the two circles.

(a) Write down the choice of your kernel.

(b) Implement kernel k-means using random initialization for cluster centers (randomly pick data points as centers of the clusters). The algorithm should run until none of the cluster assignments are changed. Run the algorithm for $K = 2$, and plot the clustering assignments by different colors and markers. (you need to report 1 plot)

4.4 Implement Gaussian Mixture Model

In this problem, you need to implement the EM algorithm to fit a Gaussian Mixture model on the `hw5_blob.mat` dataset.

(a) Run 5 times of your EM algorithm with number of components $K = 3$, and plot the log likelihood of the data over iterations of EM for each of runs. (The x-axis is the number of iterations, and the y-axis is the log likelihood of the data given current model parameters. Please plot all five curves in the same figure)

(b) For the best run in terms of log likelihood, (1) Plot the most likely cluster assignment of each data point indicated by different colors and markers. (2) Report the mean and co-variance matrix of all the three Gaussian components.

5 Submission Instructions

Submission Instructions: You need to provide the followings:

- Provide your answers to problems 1-4 in PDF file, named as `CSCI567_hw6_fall15.pdf`. You need to submit the homework in both hard copy (at Locker #19 at PHE, with a box labeled as CSCI567-homework by 6pm of the deadline date) and electronic version as pdf file on Blackboard. If you choose handwriting instead of typing all the answers, you will get 40% points deducted.
- Submit ALL the code and report via Blackboard by 6 pm of the deadline date. The only acceptable language is MATLAB. For your program, you MUST include the main function called `CSCI567_hw5_fall15.m` in the root of your folder. After running this main file, your program should be able to generate all of the results needed for this programming assignment, either as plots or console outputs.You can have multiple files (i.e your sub-functions), however, the requirement is that once we unzip your folder and execute your main file, your program should execute correctly. Please double-check your program before submitting. You should only submit one `.zip` file. No other formats are allowed except `.zip` file. Also, please name it as `[lastname]_[firstname]_hw5_fall15.zip`.

Collaboration: You may collaborate. However, collaboration has to be limited to discussion only and you need to write your own solution and submit separately. You also need to list with whom you have discussed.