Machine Learning : E0270 2014
Assignment 3
Due Date : Apr 16th (By 12:00 PM)

1 Gaussian Processes

The GPML toolbox for MATLab provides a range of functionality for Gaussian Processes. Download the GPML toolbox from
`http://www.gaussianprocess.org/gpml/code/matlab/doc/`.

1. **Regression** You are given a regression dataset `synR1.mat`, $(\mathbf{X}, \mathbf{y}), \mathbf{X} \in \mathbb{R}^{N \times M}$ with N instances, each of dimension M, $\mathbf{y} \in \mathbb{R}^m$ m-dimensional vector, with values corresponding the instances in $\mathbf{X}$. Fit the data with Gaussian Process regression using the GPML toolbox using the helper scripts given in `gpml_regression_hwhelp.txt`. Experiment with the linear, polynomial_degree_2, polynomial_degree_3 and the squared exponential kernel(experiment with the width {0.001, 0.01, 0.1, 1, 10, 100} parameter using 5 fold crossvalidation using the folds given and take the best parameter). For each kernel type, report the squared error on the test data. Which kernel performs best ?

2. **classification** You are given the classification dataset `Spambase.mat`. Your task is to determine the class labels for the test set using Gaussian Processes for classification using the helper scripts in `gpml_classification_hwhelp.txt`. For the squared exponential kernel report the value of misclassification error with inference using Laplace approximation and Expectation propogation. [Note : You have seen the Laplace approximation during class. Expectation Propogation technique is an alternate method for evalulating the posterior]. Which performs better? **Please submit any scripts/commands you used for obtaining the results online by the due date as per the submission instructions**

2 Clustering : Gaussian Mixture Model

In this problem, you are expected to use MATLabs `gmdistribution.fit` and kmeans functions to cluster the points $(\mathbf{X}, \mathbf{y}), \mathbf{X} \in \mathbb{R}^{N \times 2}$ given in the `synCl1.mat` dataset.

1. Run the GMM algorithm (see `gmmhelp.txt` for more information) on the given data points. The dataset also contains two different initialization points. Run the `gmdistribution.fit` MATLab function with the options structure given (using the `Options` parameter) with each of the two

initialization structures given (using the `Start` parameter). Plot the per iteration likelihood for each of these runs (in the same figure).

2. Use the `cluster` function for each of these two runs and get cluster assignments $\{1 \leq Z_i \leq K\}_{i=1}^{N}$ for the points. Use the `scatter` function of MATLab to vizualize these points with a different color for each cluster.

3. Perform K-means clustering on this dataset with each of starting points provided. Plot the total squared error for each run with increasing iterations (in the same figure). With the obtained cluster assignments, use the `scatter` function of MATLab to vizualize these points with a different color for each cluster.

4. What are your observations from these experiments? What can you say about the underlying EM algorithm used for the maximum likelihood estimation of the GMM parameters? Does it reach the global optimum? What can you say about the k-means algorithm.

5. The K-means algorithm is closely connected to the GMM. In fact the GMM updates obtained during EM algorithm lead to the k-means algorithm with the assumption that all the Gaussians have the same covariance $\sigma^2 I$ with σ tending to 0. Show this.

   Please submit any scripts/commands you used for obtaining the results online by the due date as per the submission instructions

3 Expectation Maximization

1. **EM for Multinomial Mixtures** : Consider a mixture of K multinomials $p(X_i) = \sum_{k=1}^{K} p(Z_i)p(X_i|Z_i = k)$, for N observations $1 \leq i \leq N$ where $X_i|(Z_i = k) \sim Mult(X_i; \mu_k)$. We note that X_i is a categorical variable that can take a category between 1 and M. where $\sum_{j=1}^{M} X_{i,j} = 1, \forall i$ using the 1 in M notation and $p(X_i|Z_i = k) = \prod_{j=1}^{M} \mu_{k,j}^{X_{i,j}}$. The mixture distribution is also a multinomial distribution π such that $p(Z_i) = \prod_{k=1}^{K} \pi_k^{Z_{i,k}}$ using the one in K notation for Z_i. The parameters of this model are $\{\mu_{k,j} : 1 \leq k \leq K, 1 \leq j \leq M\}$ and $\{\pi_k : k = 1, \ldots, K\}$. Estimate the parameters of this model using the EM algorithm.

2. **MAP EM** : The MAP EM aims to maximize the posterior distribution by assuming a prior over the parameters θ. i.e it tries to maximize the posterior $p(\theta|X) \propto p(X|\theta)p(\theta)$ where X is the observed data and $p(\theta)$ is a prior for θ. It can be shown that, for the MAP EM, the E-Step remains the same as before, while in the M step, the quantity to be maximized is $Q(\theta, \theta^{old}) + ln(p(\theta))$. Note that $Q(\theta, \theta^{old})$ is the quantity maximized in the M step of our regular ML EM algorithm. (Refer to Bishop Section 9.3 for detailed notation for the ML EM algorithm).

   The EM algorithm for the mixture of Bernoulli is described in Bishop (Section 9.3.3 of Bishop). Extend this to MAP EM and write down the E-Step and M-step updates for MAP EM for Mixture of Bernoulli

where the parameter of each Bernoulli has a beta prior and the mixture distribution has a Dirichlet prior. For N datapoints, $1 \leq i \leq N$, $p(X_i|Z_i = k) = \mu_k^{X_i}(1-\mu_k)^{1-X_i}$ where each μ_k has a beta distribution prior with parameters a_k, b_k,i.e $p(\mu_k) = Beta(a_k, b_k)$. The mixture distribution is a multinomial π where $p(Z_i) = \prod_{k=1}^{K} \pi_k^{Z_{i,k}}$. The parameter π has a Dirichlet prior $p(\pi) = Dir(d_1, \ldots, d_k)$. [Refer to Bishop 2.1.1 and 2.2.1 for the form of the Beta and the Dirichlet distributions, alternately you can see Wikepedia].

4 Exponential Family

Several of the distributions encountered in class so far belong to a class of distributions called the exponential family. Distributions belonging to these family have sevaral common properties that prove to be useful in sevaral situations. Given a random variable X and a parameter η, a distribution belonging to exponential family takes the form

$$p(X|\theta) = h(X)e^{\eta^T u(X) - A(\eta)}$$

where η , called the natural parameter is a function of the original parameter θ and $u(X)$ is called the `sufficient statistic`.

[Note: A statistic $T(X)$ is a sufficient statistic for underlying parameter θ if $p(X = x|T(x) = t, \theta) = Pr(X = x|T(X) = t)$. This can equivalently be written as $p(\theta|T(X) = t, X) = p(\theta|T(X) = t)$. In other words, it is sufficient to store the value of T(X) instead of the actual data X to estimate parameter θ. However there is an easier way to check for sufficiency. Sufficiency factorization theorem: The statistic $T(x_1, \ldots, x_n)$ is sufficient for θ if the pdf (of pmf) can be factorized as follows : $p(x_1, \ldots, x_n) = \phi_1(x_1, \ldots, x_n)\phi_2(T(x_1, \ldots, x_n), \theta)$.]

1. Show that the Poisson, Binomial, exponential and Multivariate Normal distributions belong to the exponential family.

2. Find the maximum likelihood estimate η_{ML} for a distribution from the exponential family.

3. Suppose $X_1, X_2, \ldots X_N$ are drawn IID from an exponential family, what is the sufficient statistic $T(x_1, \ldots, x_n)$ for η.

5 Kernel K-Means

The Kernel K-means algorithm extends the k-means algorithm by clustering in the kernel space using a non-linear transformation ϕ mapping points in the input space to a higher dimensional space.

1. How will you modify the objective function of the K-means algorithm with N points and K clusters, $E_K(\mathbf{Z}, \mu) = \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{i,k}||\mathbf{x_i} - \mu_\mathbf{k}||^2$, for Kernel K-means so that the *Kernel trick* can be used. [Hint: Write the mean $\mu_\mathbf{k}$ in terms of $Z_{i,k}$ and $\phi(\mathbf{x_i})$, then expand]

2. What would the be Kernel K-Means algorithm given your new objective function? Would you still have the two alternating K-means steps of updating cluster assignments and means?

3. When do you think the kernel K-means algorithm will be useful over its non-Kernel counterpart? (Can you think of an example with points in R^2 where the Kernel extension to K-means is required for meaningful clustering)

4. Let $\mathbf{Z}$ be an $N \times K$ assignment matrix (for N data instances, K clusters) with values $Z_{i,k} = 1$ if $\mathbf{x_i}$ is assigned to cluster k. Let $N_k = \sum_{i=1}^{N} Z_{i,k}$ be the number of points assigned to cluster k and $\mathbf{L}$ be a $K \times K$ diagonal matrix with entries $L_{k,k} = 1/N_k$. Show that minimizing $E_K(\mathbf{Z}, \mu)$ in the kernel space is equivalent to finding $max_{\mathbf{Z}}\ trace(\mathbf{L^{1/2}Z^T KZL^{1/2}})$ where $\mathbf{K}$ is the Kernel Matrix.

   [Note : This formulation of the Kernel K-means objective function is important because it directly leads to clustering algorithm that attempts to find a global solution for this objective function]

6 PCA and Kernel PCA

1. Given data $\mathbf{x_i} \in R^D, 1 \leq i \leq N$, let C be the covariance matrix. Dimensionality reduction is achieved through PCA by projecting each point onto a lower K-dimensional subspace by computing the new representation $\mathbf{y_i} = U_K^T \mathbf{x_i}, \forall i$ where U_K is the matrix of top $K(< D)$ eigen vectors of C. Show that the projected data is not correlated in the new basis (Hint : compute covariance in new basis)

2. To extend PCA to the kernel PCA, the covariance matrix needs to be computed in the kernel space. Remember that even if the data is centered to have zero mean in the input space, this might not be the case in the kernel space. Compute the covariance matrix in the kernel space in terms of the kernel matrix K. [Remember you should not have to evaluate the transformation ϕ to compute the new covariance matrix in kernel space].

3. It can be shown that $\mathbf{u_k}$, the kth eigen vector can be written as a linear combination $\mathbf{u_k} = \sum_{i=1}^{N} \alpha_i \mathbf{x_i}$ of data instances for an appropriately chosen $\alpha_i, \forall i$. Given this information can you outline an algorithm for finding a low dimensional representation using kernal PCA?

4. When do you think kernel PCA is useful ?

Note : Datasets for the assignment are available at
http://drona.csa.iisc.ernet.in/ e0270/Jan-2014/Assignments/
Assignment3-Files/Files_data_codes.zip