Machine Learning 10-701 Apr 6, 2015
http://alex.smola.org/teaching/10-701-15/ due on Apr 13, 2015
Carnegie Mellon University

Homework 10

START HERE: Instructions

- The homework is due at 11:59pm on Apr 13th, 2015. Anything that is received after that time will be considered as late submission.
- Answers to everything but the coding question will be submitted electronically (e.g. as a PDF or handwritten and scanned).
- Please follow the instruction for code submission in problem correctly.
- The handout for question can be found on Autolab or here.
- Collaboration on solving the homework is allowed (after you have thought about the problems on your own). However, when you do collaborate, you should list your collaborators! You might also have gotten some inspiration from resources (books or online etc...). This might be OK only after you have tried to solve the problem, and couldn't. In such a case, you should cite your resources.
- If you do collaborate with someone or use a book or website, you are expected to write up your solution independently. That is, close the book and all of your notes before starting to write up your solution.

This is a programming assignment. You are asked to implement several algorithms in Octave. Octave is a free scientific programming language, with syntax (almost) identical to that of Matlab. Installation instructions can be found on the Octave website as linked above. If you've never used Octave or Matlab before, you may wish to check out this tutorial or this one.

For each question you will be given a single function signature, and asked to write a single Octave function which satisfies the signature. Please do *not* change the names of the functions and their variables. Do *not* modify the structure of the directories.

The problems are automatically graded 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. In order for your code to execute correctly on our servers you should avoid using libraries beyond the *basic* octave libraries.

The Autolab interface for this course can be found at `https://autolab.cs.cmu.edu/`. 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 the challenge question on the class leaderboard.

We have provided you with a single folder containing 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 as many times as you like (up to the due date).

Please *only* submit a tarball called "code.tar", which is compressed from "code" folder. Do not submit any other files such as data files.

If you have a question, please post it on Piazza `https://piazza.com/class/i4ivtbjbrt219e`. If you discover a bug or want to talk other technical issues, please send an email to Jin Sun `jins@andrew.cmu.edu`.

Notations

- **XTrain**: $\mathcal{R}^{nTrain \times f}$ is a matrix of training data, where each row is a training instance with f features.
- **XTest**: $\mathcal{R}^{nTest \times f}$ is a matrix of test data, where each row is a test instance with f features.
- **yTrain**: $\mathcal{R}^{nTrain \times 1}$ is a vector of labels (for classification) or real numbers (for regression).

Machine Learning 10-701 Apr 6, 2015
http://alex.smola.org/teaching/10-701-15/ due on Apr 13, 2015
Carnegie Mellon University

Homework 10

General Instructions

- Please refer to homework 9 for implementation. Please note that $q_{i,k}$, $Q(z_{i,k}|x_i)$ and n_{ik} (in hw9) are equivalent.
- Here is a tutorial on EM algorithm. If you are not familiar with EM, please walk through it at least once.

Suppose you want to figure out some parameters θ that maximizes the log-likelihood on data D with some observed variables x and some latent variables z. We can first write down the log-likelihood as follows:

$$\log \prod P(D;\theta) = \log \prod_i P(x_i;\theta) = \sum_i \log \sum_k P(x_i, z_{i,k};\theta)$$

Directly maximizing this likelihood is NP-complete. Instead, we try to maximize a concave lower bound derived with Jensen's Inequality:

$$\log \prod P(D;\theta) = \sum_i \log \sum_k P(x_i, z_{i,k};\theta) = \sum_i \log \sum_k Q(z_{i,k}|x_i) \frac{P(x_i, z_{i,k};\theta)}{Q(z_{i,k}|x_i)} \geq \sum_i \sum_k Q(z_{i,k}|x_i) \log \frac{P(x_i, z_{i,k};\theta)}{Q(z_{i,k}|x_i)}$$

$Q(z_{i,k}|x_i)$ is simply $P(z_{i,k}|x_i)$ for convenience in notation. Let's denote

$$F(Q,\theta) = \sum_i \sum_k Q(z_{i,k}|x_i) \log \frac{P(x_i, z_{i,k};\theta)}{Q(z_{i,k}|x_i)}$$

EM algorithm (in coordinate descent manner) works as follows:

- In E-step, fix θ, maximize F over Q.

$$\begin{aligned}
F(Q,\theta) &= \sum_i \sum_k Q(z_{i,k}|x_i) \log \frac{P(x_i, z_{i,k};\theta)}{Q(z_{i,k}|x_i)} = \sum_i \sum_k Q(z_{i,k}|x_i) \log \frac{P(z_{i,k}|x_i;\theta)P(x_i;\theta)}{Q(z_{i,k}|x_i)} \\
&= \sum_i \sum_k Q(z_{i,k}|x_i) \log \frac{P(z_{i,k}|x_i;\theta)}{Q(z_{i,k}|x_i)} + \sum_i \sum_k Q(z_{i,k}|x_i) \log P(x_i;\theta) \\
&= -\sum_i \sum_k Q(z_{i,k}|x_i) \log \frac{Q(z_{i,k}|x_i)}{P(z_{i,k}|x_i;\theta)} + \sum_i \log P(x_i;\theta) \sum_k Q(z_{i,k}|x_i) \\
&= -\sum_i D_{KL}(Q(z_i|x_i)||P(z_i|x_i;\theta)) + \log P(x_i;\theta)
\end{aligned}$$

D_{KL} denotes the Kullback Leibler divergence, a measure of the divergence of two distributions, which is defined as $D_{KL}(P||Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)}$. Since in E-step, θ is fixed, $F(Q,\theta)$ is maximized only by Q that maximizes the negative KL divergence. Because KL divergence is always non-negative. $D_{KL} = 0$ happens only when P and Q are the same.

Summary: In the E-step of t^{th} iteration, we derive $Q^{(t)} = \text{argmax}_Q F(Q, \theta^{(t-1)})$, namely $Q^{(t)}(z_{i,k}|x_i) = P(z_{i,k}|x_i;\theta^{(t-1)})$

- In M-step, fix Q, maximize F over θ.

$$F(Q,\theta) = \sum_i \sum_k Q(z_{i,k}|x_i) \log \frac{P(x_i, z_{i,k}; \theta)}{Q(z_{i,k}|x_i)} = \sum_i \sum_k Q(z_{i,k}|x_i) \log P(x_i, z_{i,k}; \theta) - Q(z_{i,k}|x_i) \log Q(z_{i,k}|x_i)$$
$$= \sum_i E_{z_i|x_i} \log P(x_i, z_i; \theta) + H(z_i|x_i)$$

$H(z|x)$ the entropy of $Q(z|x)$, which can be ignored since Q is fixed in M-step. θ should be derived with MLE on the first term.

Summary: In the M-step of t^{th} iteration, we derive $\theta^{(t)} = \operatorname{argmax}_\theta F(Q^{(t)}, \theta)$,
namely $\theta^{(t)} = \operatorname{argmax}_\theta \sum_i \sum_k Q^{(t)}(z_{i,k}|x_i) \log P(x_i, z_{i,k}; \theta)$.

1 Gaussian Mixture Model [40 pts]

In this problem you are asked to implement Gaussian Mixture model. Recall that mixture models are represented as:

$$x_1, x_2, \ldots, x_n \sim p(x) = \sum_k P(x|z=k)P(z=k)$$

The Gaussian Mixture Model assumes $x|z = k \sim \mathcal{N}(\mu_k, \Sigma_k)$. Denote $P(z = k)$ as π_k. The observed variable is x, the latent variable is z, and the parameters are μ, Σ, π

1.1 Expectation [20 pts]

Complete the function $E\_step(X, k, mu, sigma, pi)$ which returns $q \in \mathcal{R}^{n\times k}$, where n is the number of data points, k is the number of clusters. Recall that $q_{i,k} = P(z_{i,k}|x_i)$. $mu \in \mathcal{R}^{f\times k}$ are the means of k clusters. $sigma \in \mathcal{R}^{f\times f\times k}$ are the covariance matrices of k clusters. The covariance matrix of i^{th} cluster is simply $sigma(:, :, i)$. $pi \in \mathcal{R}^{k\times 1}$ are the priors for k clusters.

1.2 Maximization [20 pts]

Complete the function $M\_step(X, k, q, alpha, mu\_0, kappa\_0, sigma\_0, nu\_0)$ which returns $mu \in \mathcal{R}^{f\times k}$, $sigma \in \mathcal{R}^{f\times f\times k}$, $pi \in \mathcal{R}^{k\times 1}$. You can ignore $alpha, mu\_0, kappa\_0, sigma\_0, nu\_0$ in this problem but you will need them later.

2 Conjugate Smoothing [30 pts]

The original estimates for GMM may diverge with problems like infinite variance, zero probability, tiny clusters, etc. Conjugate smoothing applies priors to the variables. Recall the conjugate priors in exponential families (also in homework 5 and 9). In GMM, $z|\pi$ follows Multinomial distribution $Mult(\pi)$. We can apply a conjugate prior with π following Dirichlet distribution $Dir(\alpha)$. $X|y; \mu, \Sigma$ follows Gaussian distribution $\mathcal{N}(\mu, \Sigma)$. We can apply a conjugate prior with μ, Σ following Normal-Inverse-Wishart distribution $\mathcal{NIW}(\mu_0, \kappa_0, \Sigma_0, \nu_0)$.

Rework the function $M\_step()$ with conjugate smoothing.

Machine Learning 10-701 Apr 6, 2015
http://alex.smola.org/teaching/10-701-15/ due on Apr 13, 2015
Carnegie Mellon University

Homework 10

---

3 EM [30 pts]

3.1 Putting everything together [10 pts]

Complete the function $EM(X, k)$ which returns $mu, sigma, pi$. Since you need to tune parameters for the leaderboard problem, the autolab only checks the validity of your output for this question.

3.2 Leaderboard [20 pts]

Run your EM algorithm on a given data set with $k = 4$. The quality of your clustering algorithm will be evaluated with perplexity (lower is better), namely $\sqrt[n]{\prod_i^n \frac{1}{P(x_i;\theta)}}$. Students achieving baseline (0.15) will get 10 points. Students ranking top 10 on the leaderboard will get another 10 points.

Some hints:

- Pre-processing is crucial.
- Seeding your EM with K-means++ may help.
- You should design your stopping criteria. EM usually converges less than 30 iterations.
- Pick valid values for the priors. Check out Dirichlet and NIW in wikipedia.
- You can force Σ to be symmetric by doing $\Sigma = \frac{1}{2}(\Sigma + \Sigma^\top)$. This is helpful for some numerical issues in Matlab.