20124221: Introduction to Statistical Machine Learning
Winter 2012

Problem Set #3

Out: January 31, 2012

Due: Tuesday February 21, 2012, 23:59

Instructions

Please read these instructions carefully and follow them precisely. Feel free to ask the instructor if anything is unclear!

How and what to submit? Please submit your solutions electronically by e-mail to `greg@ttic.edu`, using the procedure below.

1. Create a directory ⟨`your_last_name`⟩`-ps3`) in which you will work on your solutions. All files mentioned below are assumed to reside in that directory (we will refer to it as the pset directory).

2. Generate the written parts of your solution in PDF. You can do this any way you want: typeset the solution in LaTeX(recommended), type it in Word or a similar program and convert/export to PDF, or even hand write the solution (legibly!) and scan it to PDF. Please name this document ⟨`your_last_name`⟩`-sol3.pdf`.

3. Create Matlab code (extension `.m`), data (`.mat`), image (`.png`, `.pdf` etc.) files needed in the pset directory. If file names are not specified explicitly in the assignment, please include a file `README` with brief description of the files.

4. When done, create an archive of the entire pset directory (please make sure it contains the directory node itself, not just the files) as `.zip` or `.tar.gz` file, and e-mail it to `greg@ttic.edu`.

Late submissions: there will be a penalty of 25 points for any solution submitted within 24 hours past the deadlne. No submissions will be accepted past then.

What is the required level of detail? When asked to derive something, please clearly state the assumptions, if any, and strive for balance: justify any non-obvious steps, but try to avoid superfluous explanations. When asked to plot something, please include the figure as well as the code used to plot it (and clearly explain in the `README` what the relevant files are). If multiple entities appear on a plot, make sure that they are clearly distinguishable (by color or style of lines and markers). When asked to provide a brief explanation or description, try to make your answers concise, but do not omit anything you believe is important.

When submitting code, please make sure it's reasonably documented, and describe succinctly in the `README` what is done in each `m`-file.

1 EM and generative models

In this part of the problem set we will derive and implement the EM algorithm for a discrete space, in which the observations are d-dimensional binary vectors (containing either 0 or 1). We will model distributions in this space as mixture of multivariate Bernoulli distributions. That is,

$$p\left(\mathbf{x} \,|\, \theta, \mathbf{p}\right) \;=\; \sum_{l=1}^{k} p_l p\left(\mathbf{x} \,|\, \theta_l\right), \tag{1}$$

where $\theta = [\theta_1, \ldots, \theta_l]$ are the parameters of the k components, and the mixing probabilities $\mathbf{p} = [p_1, \ldots, p_k]^T$ are subject to $\sum_l p_l = 1$.

The l-th component of the mixture is parametrized by a d-dimensional vector $\theta_l = [\theta_{l1}, \ldots, \theta_{ld}]^T$. The value of θ_{lj} is the probability of 1 in the j-th coordinate in a vector drawn from this distribution. Since the dimensions are assumed to be independent, given θ_l, the conditional distribution of $\mathbf{x}$ under this component is

$$p\left(\mathbf{x} \,|\, \theta_l\right) \;=\; \prod_{j=1}^{d} p\left(x_j \,|\, \theta_{lj}\right) \;= \prod_{j=1}^{d} \theta_{lj}^{x_j} (1 - \theta_{lj})^{1 - x_j}. \tag{2}$$

The hidden variables here are, just in the Gaussian mixture case, the identities of the component that generated each observation. We will denote the hidden variable associated with $\mathbf{x}_i$ by z_i. The EM with the model proceeds as follows. The E-step consists of computing the posterior of $p\left(z_i \,|\, \mathbf{x}_i; \theta, \mathbf{p}\right)$.

In the M-step, we need to update the estimates of θ and $\mathbf{p}$. These updates can be derived in a way similar to the derivation for the Gaussian mixture. They are[1]:

$$\theta_c^{(t+1)} = \frac{1}{N_c^{(t)}} \sum_{i=1}^{N} \mathbf{x}_i p\left(z_i = c \,|\, \mathbf{x}_i; \theta^{(t)}, \mathbf{p}^{(t)}\right), \tag{3}$$

$$p_c^{(t+1)} = \frac{N_c^{(t)}}{N}. \tag{4}$$

where N_c^t is the weighted "mass" of the c-th component,

$$N_c^{(t)} = \sum_{i=1}^{N} p\left(z_i = c \,|\, \mathbf{x}_i; \theta^{(t)}, \mathbf{p}^{(t)}\right),$$

and the notation $\theta_c^{(t)}, p_c^{(t)}$ stands for the values of the relevant parameters in the end of iteration t.

This model could be applicable, for instance, in modeling distributions of documents using binary features. Here, however, we will apply it to another domain–images of handwritten digits. We will use the data in `mnistData20x20`; it contains images of digits 0 through 9, that have been cropped to 20×20 pixels and pixel values normalized to be 0 or 1 (by thresholding the original pixels). The data are organized in `X`, `Xtest` and `Y`, `Ytest` in the same way as the MNIST data in PS2.

We have provided a skeleton for an implementation of EM with this model; you will need to complete this code. It consists of the following functions; you will find all of them useful. The relevant files are available on the course website:

`runem.m` the function that runs the EM iterations.

`estep.m`, `mstep.m` the E-step and the M-step implementation.

`conditional.m` computes $p\left(\mathbf{x}_i \,|\, \theta_l\right)$ for each i, l.

`logLikelihood.m` computes the conditional probabilities and the log-likelihood of the data under a given values of mixture parameters.

[1] See PRML Section 9.3.3 for the derivation.

Problem 1 **[30 points]**

Write the exact expression for the posterior probability $p(z_i = c \,|\, \mathbf{x}_i; \theta, \mathbf{p})$ in terms of $\mathbf{x}_i$ and the elements of θ and $\mathbf{p}$. Implement it in `estep.m`. Also, fill in the missing code for the component parameter updates in `mstep.m`.

Run your code on `X`, applying EM **separately** to each digit, with $k = 5$ components, using the call

```
[theta,p,logL]=runem(X(:,Y==y),5,1,100);
```

This will set stopping criterion to min. change of 1 in log-likelihood (and no more than 100 iterations). Once the EM has converged, we can visualize the parameters θ:

```
figure;colormap gray;
for k=1:5
  subplot(1,5,k); imagesc(reshape(theta(:,k),20,20)');axis image;
end
```

Turn in the plot obtained by the code above (for each digit class), and your interpretation of the components; what properties of the data do they capture?

End of problem 1

Advice: Running EM will take a while, even with an efficient implementation; typically, it should converge before 100 iterations. In practice, since the model found by EM depends on random initialization (first guess of θ), we would run EM a number of times, and take the results of the run that lead to the highest log-likelihood. You are welcome to do that, but it is OK to just run EM once and report the results of that single run.

We can use the mixture model learned for each digit class for classification. While above we arbitrarily decided to use five components in the mixture, a more principled way is to select the model based on an information criterion that balances the likelihood and model complexity. Here we will use the Bayesian Information Criterion (BIC). BIC for a model parametrized by θ on a data set X_N

$$BIC(\theta, X_N) \;=\; \log p(X_N; \theta) - \frac{\pi(\theta)}{2} \log N,$$

where $\pi(\theta)$ is the number of free parameters in the model. For instance, as discussed in class, mixture of two unconstrained Gaussians in d-dimensional space has $d(d+1)+2d+1$ parameters: $d(d+1)/2$ for each covariance matrix, d for each mean, and 1 for the priors (only one and not two since $p(1) = 1 - p(2)$). This criterion can be seen as a form of regularization; we will select the model with the highest value of BIC, instead of the model with the highest likelihood.

Problem 2 **[30 points]**
Write a Matlab function that performs BIC model selection with Bernoulli mixture EM. Using this function, apply the EM algorithm separately to the training data for each digit class from 0 to 9. Let $\theta_c = \{\theta_{c1}, \ldots, \theta_{ck}\}$ and $\mathbf{p}_c = \{p_{c1}, \ldots, p_{ck}\}$ be the estimated mixture parameters for class c, where k is the number of mixture components selected based on BIC. Turn in the plot of θs (similar to the plot in the previous problem) for each class-specific mixture.

Now write down the expression of the Bayes optimal classifier, under the assumption that the estimated mixture model is accurate (and assuming equal prior 1/10 for each digit class), in terms of a test input $\mathbf{x}$ and θ_c, $\mathbf{p}_c$. Implement this classifier in Matlab. In addition, construct a "baseline" Naive Bayes classifier, that models class-conditional distribution of each pixel as a Bernoulli distribution, independent of other pixels; estimate its parameters for this classifier using the closed-form ML solution for Bernoulli. Report the test error obtained with these two classifiers on `Xtest`, and briefly explain the results.

End of problem 2

2 Conditional mixture models

Here we will consider a two-level mixture model for a conditional distribution of output y for input $\mathbf{x}$:

$$p(y \mid \mathbf{x}) = \sum_{c=1}^{C} \pi_c \rho_c (y \mid \mathbf{x}; \theta_c) \tag{5}$$

where each ρ_c is itself a conditional mixture of L_c local conditional distributions,

$$\rho_c\left(y \,|\, \mathbf{x}\right) \;=\; \sum_{j=1}^{L_c} \lambda_{cj} \psi_{cj}\left(y \,|\, \mathbf{x};\, \theta_{cj}\right). \tag{6}$$

We assume that

$$\sum_c \pi_c \;\;=\; 1, \qquad \pi_c \;\;\geq\; 0 \;\forall c = 1, \ldots, C \tag{7}$$

$$\sum_j \lambda_{cj} \;\;=\; 1, \qquad \lambda_{cj} \;\;\geq\; 0 \;\forall c = 1, \ldots, C,\; j = 1, \ldots, L_c \tag{8}$$

It is not relevant for the purpose of this section whether the task underlying this model is regression or classification.

First, we will show that in general, this increased level of sophistication may not accomplish much.

Problem 3 **[10 points]**
Prove that the hierarchical model described by (5) and (6) is equivalent to a conventional (single-level) mixture model.

End of problem 3

Next we will consider an even more sophisticated model: the one in which each of the mixing coefficients at both levels is allowed to be an arbitray function of the input $\mathbf{x}$, that is, the model is

$$p\left(y \,|\, \mathbf{x}\right) \;=\; \sum_{c=1}^{C} \pi_c(\mathbf{x}) \rho_c\left(y \,|\, \mathbf{x};\, \theta_c\right) \tag{9}$$

and

$$\rho_c\left(y \,|\, \mathbf{x}\right) \;=\; \sum_{j=1}^{L_c} \lambda_{cj}(\mathbf{x}) \psi_{cj}\left(y \,|\, \mathbf{x};\, \theta_{cj}\right), \tag{10}$$

and the conditions in (7) are satisfied for every $\mathbf{x}$.

Problem 4 **[10 points]**
Show that the model described by (9) and (10) is still equivalent to a single-level model, of course with mixing coefficients that are allowed to be arbitrary functions of $\mathbf{x}$.

End of problem 4

Finally, consider a more specific model family: the one in which mixing coefficients at both levels are softmax functions, for instance,

$$\pi_c(\mathbf{x}) \;=\; \frac{e^{\mathbf{w}_c^T \mathbf{x}}}{\sum_l e^{\mathbf{w}_l^T \mathbf{x}}}$$

and similarly for λs.

Problem 5 **[20 points]**
Show that the two-level model with softmax mixing probabilities is *not* in general equivalent to a single-level model with softmax mixing probabilities.

End of problem 5

Advice: It is sufficient to construct a single counterexample. For instance, if you can show that a mixture of two components, one of which is itself a mixture of two components, can not be represented by a single level mixture with softmax mixing, you are done.