Coursera BETA Dong-Bang Tsai | Preferences All Courses About Contact Us Logout

Stanford University
Probabilistic Graphical Models

Daphne Koller, Kevin Murphy
Winter 2011-2012

Home

Review Quizzes

Theory Problems

Assignments

Assignment Questions

Video Lectures

Discussion Forums

Course Wiki

Lecture Slides

Course Schedule

Course Logistics

Course Information

Course Staff

Octave Installation

Feedback — Expectation Maximization

You achieved a score of 10.00 out of 10.00

Question 1

Bayesian Clustering using Normal Distributions. Suppose we are doing Bayesian clustering with K classes, and multivariate normal distributions as our class-conditional distributions. Let $\mathbf{X} \in \mathbf{R}^n$ represent a single data point, and $C \in \{1, 2, \ldots, K\}$ its unobserved class. Which of the following statement(s) is/are always true in the general case?

Your Answer		Score	Explanation
● $P(\mathbf{X} \mid C = c) \sim N(\mu_c, \Sigma_c) \quad \forall c \in \{1, 2, \ldots, K\}$, for some class-specific parameters μ_c and Σ_c that represent the distribution of data coming from the class c.	✓	1.00	This is the definition of having a multivariate normal distribution as the class-conditional distribution: given the class from which the data point came from, the distribution of the data point follows a multivariate normal distribution with mean and covariance parameters specific to its particular class.
Total		1.00 / 1.00	

Question 2

Hard Assignment EM. Continuing from the previous question, let us now fix each class-conditional distribution to have the identity matrix as its covariance matrix. If we use hard-assignment EM to estimate the class-dependent mean vectors, which of the following can we say about the resulting algorithm?

Your Answer		Score	Explanation
● It is equivalent to running standard k-means clustering with K clusters.	✓	1.00	You will always assign vertices to their closest (in Euclidean distance) cluster centroid, just as in k-means.
Total		1.00 / 1.00	

Question 3

***Hard Assignment EM.** Now suppose that we fix each class-conditional distribution to have the same diagonal matrix D as its covariance matrix, where D is **not** the identity matrix. If we use hard-assignment EM to estimate the class-dependent mean vectors, which of the following can we say about the resulting algorithm?

Your Answer		Score	Explanation
● It is an instance of k-means, but using a different distance metric rather than standard Euclidean distance.	✓	1.00	You will always assign vertices to their closest cluster centroid, just as in k-means. But here the definition of ``closest" is skewed by the covariance matrix so that it does not equally depend on each dimension and is thus not a Euclidean distance.
Total		1.00 / 1.00	

Question 4

EM Running Time. Assume that we are trying to estimate parameters for a Bayesian network structured as a binary (directed) tree (**not** a polytree) with n variables, where each node has at most one parent. We parameterize the network using table CPDs. Assume that each variable has d values. We have a data set with M instances, where some observations in each instances are missing. What is th tightest asymptotic bound you can give for the worst case running time of EM on this data set for K iterations?

Your Answer		Score	Explanation
● $O(KMnd^2)$	✓	1.00	At each iteration and for every instance, it is required to run exact inference over the given network. Using clique-tree calibration, the cost of inference is the number of cliques (n) times the size of the clique potential which is d^2 (due to the tree-structure of the network each clique can have only 2 variables in its scope).
Total		1.00 / 1.00	

Question 5

EM Running Time. Use the setting of question 4, but now we assume that the network is a polytree, in which some variables have several parents. What is the cost of running EM on this data set for K iterations?

Your Answer		Score	Explanation
● Can't tell using only the information given.	✓	1.00	We cannot tell because now the factors in the clique tree can be considerably larger than d^2 (but we do not how much larger they might be).
Total		1.00 / 1.00	

Question 6

***Optimizing EM.** Now, going back to the setting of the question 4 (each node has at most one parent), assume that we are in a situation where at most 2 variables in each data instance are unobserved (not necessarily the same 2 in each instance). Can we implement EM more efficiently? If so, which of the following reduced complexities can you achieve?

Your Answer		Score	Explanation
● $O(K(M+n)d^2)$	✓	1.00	In this case, the cost of the E-step is Md^2 since we can easily compute the probabilities of each possible completion of instances when only up to 2 variables are missing. We can use this to compute the expected sufficient statistics (where we will be summing over the M instances and up to d^2 possible completed instances). The cost of the M-step will be nd^2 (it is equal to the number of parameter values computed).
Total		1.00 / 1.00	

Question 7

***Optimizing EM.** Still in the setting of the question 4, now assume that we are in a situation where at most 2 variables in each data instance are unobserved, but it's the same 2 each instance. Can we implement EM more efficiently? If so, which of the following reduced complexities can you achieve?

Your Answer		Score	Explanation
●	✓	1.00	In this case, most of the graph is conditionally independent of the unobserved variables, so we can restrict our EM process to the sub-

$O(KMd^2)$			graph consisting of the unobserved variables and their Markov blankets, and fix parameters for the rest of the network once at the beginning, using standard MLE. Thus, the cost of updating a small subset of the parameters at each M-step will be no more than $O(d^2)$. In the E-step, for each instance, we will run inference over a small subset of variables at a cost of d^2 per instance. Accordingly, the cost of the E-step will be Md^2.
Total		1.00 / 1.00	

Question 8

EM Stopping Criterion. When learning the parameters $\theta \in \mathbf{R}^n$ of a graphical model using the EM algorithm, an important design decision is choosing when to stop training. Let $\ell_{\text{Train}}(\theta)$, $\ell_{\text{Valid}}(\theta)$, and $\ell_{\text{Test}}(\theta)$ be the log-likelihood of the parameters θ on the training set, a held-out validation set, and the test set, respectively. Let θ^t be the parameters at the t-th iteration of the EM algorithm. We can denote the change in the dataset log-likelihoods at each iteration with $\Delta\ell^t_{\text{Train}} = \ell_{\text{Train}}(\theta^t) - \ell_{\text{Train}}(\theta^{t-1})$ and the corresponding analogues for the validation set and the test set. Likewise, let $\Delta\theta^t = \theta^t - \theta^{t-1}$ be the vector of changes in the parameters at time step t.

Which of the following would be reasonable conditions for stopping training at iteration t? You may choose more than one option.

Your Answer		Score	Explanation
☑ $\Delta\ell^t_{\text{Valid}}$ becomes negative.	✓	0.20	Stopping when the log-likelihood starts to decrease on a held-out validation set is a good way to alleviate the problem of overfitting parameters to the training set.
☑ $\|\|\Delta\theta^t\|\|_2^2$ becomes small, i.e., it falls below a	✓	0.20	This is likely to return parameters that lie near to a local maximum of the log-likelihood function on the training set.

certain tolerance $\epsilon > 0$. Note: The ℓ_2 norm, also known as the Euclidean norm, is defined for any vector $x \in \mathbf{R}^n$ as $\|x\|_2^2 = \sum_{i=1}^n x_i^2$.			(In practice, this quantity is very rarely exactly 0 at the convergence point, due to issues with floating point inaccuracies and a potentially infinite number of steps required to converge to the exact value of the local maximum.)
□ $\Delta\ell_{\text{Test}}$ becomes negative.	✓	0.20	We never use the test set for parameter learning / selection. Instead, the test set should only be used for evaluation once we've selected our parameters based on the training and/or validation sets.
☑ $\|\Delta\theta^t\|_\infty$ becomes small, i.e., it falls below a certain tolerance $\epsilon > 0$. Note: The ℓ_∞ norm is defined for any vector $x \in \mathbf{R}^n$ as the largest component of x, $\|x\|_\infty = \max_{i=1}^n \lvert x_i \rvert$.	✓	0.20	This is likely to return parameters that lie near to a local maximum of the log-likelihood function on the training set. (In practice, this quantity is very rarely exactly 0 at the convergence point, due to issues with floating point inaccuracies and a potentially infinite number of steps required to converge to the exact value of the local maximum.)
□ $\Delta\ell_{\text{Test}}$ becomes small, i.e., it falls below a certain tolerance $\epsilon > 0$	✓	0.20	We never use the test set for parameter learning / selection. Instead, the test set should only be used for evaluation (after we've selected our parameters based on the training and/or validation sets).
Total		1.00 / 1.00	

Question 9

EM Parameter Selection. Once again, we are using EM to estimate parameters of a graphical model. We use n random starting points $\{\theta_i^0\}_{i=1,2,\ldots,n}$, and run EM to convergence from each of them to obtain a set of candidate parameters $\{\theta_i\}_{i=1,2,\ldots,n}$. We wish to select one of these candidate parameters for use. As in question 8, let $\ell_{\text{Train}}(\theta)$, $\ell_{\text{Valid}}(\theta)$, and $\ell_{\text{Test}}(\theta)$ be the log-likelihood of the parameters θ on the training set, a held-out validation set, and the test set, respectively.

Which of the following methods of selecting final parameters θ would be a reasonable choice? You may pick more than one option.

Your Answer		Score	Explanation
☑ Pick $\theta = \text{argmax}_{i=1,2,\ldots,n} \ell_{\text{Train}}(\theta_i)$.	✓	0.25	EM finds only local optima, so it is a good idea to find multiple local optima and pick the highest. This is to avoid ending up with poor local maxima of the log-likelihood function, i.e., those that have a relatively low log-likelihood with respect to other possible parameter values. Here, we are attempting to address the fact that we can't always optimize the function well, and not the problem of the optimum not generalizing well to the test data.
□ Pick $\theta = \text{argmax}_{i=1,2,\ldots,n} \ell_{\text{Test}}(\theta_i)$.	✓	0.25	We never use the test set for selecting any parameters; instead, we use it only for evaluating performance. This is because test set performance is meant as a measure of generalization ability, and we do not want to fit our parameters to the test set.
☑ Pick $\theta = \text{argmax}_{i=1,2,\ldots,n} \ell_{\text{Valid}}(\theta_i)$.	✓	0.25	Given enough data to create a validation set that is separate from both the training and test set, this could give better results than using the training set, as the problem of overfitting to the training set would be alleviated. It is not guaranteed that creating a held-out validation set (as opposed to adding the data to the training set) will always improve performance: if there is insufficient data, using all the available data for training could give better results.
□ Any one; the θ_i are all equivalent, since all of them are	✓	0.25	While all θ_i are indeed local maxima of the log-likelihood function with respect to the training

local maxima of the log-likelihood function.		data, each of these local maxima corresponds to a different value of the log-likelihood function. (It is important to remember that these are not **global** maxima.)
Total	1.00 / 1.00	

Question 10

EM and Convergence. When checking for the convergence of the EM algorithm, we can choose to measure changes in either the log-likelihood function or in the parameters. For a generic application, we typically prefer to check for convergence using the log-likelihood function. However, this is not always the case, especially when the values of the parameters are important in and of themselves. In which situations would we also be concerned about reaching convergence in terms of the parameters? Do not worry about the implementation details in the following models.

Your Answer		**Score**	**Explanation**
□ We are building a graphical model for medical diagnosis, where nodes can represent symptoms, diseases, predisposing factors, and so on. Our only aim is to maximize our chances of correctly predicting diseases that patients are suffering from.	✓	0.20	In this application, we are learning parameters to improve performance (i.e., medical diagnosis), and are not as interested in the parameters in and of themselves.
□ We have a graphical model in which each node is a superpixel, and we are using EM to learn the parameters that specify the relations between superpixels. Our end-goal is to build an image segmentation pipeline that is highly accurate.	✓	0.20	In this application, we are learning parameters to improve performance (i.e., image segmentation accuracy) and are not as interested in the

			parameters in and of themselves.
□ We have a graphical model in which each node represents an object part, and we are using EM to learn the parameters that specify the relations between object parts. Our end-goal is to build an image classification system that can accurately recognize the image as one of several known objects.	✓	0.20	In this application, we are learning parameters to improve performance (i.e., object recognition accuracy) and are not as interested in the parameters in and of themselves.
□ We are trying to transcribe human speech by building a Hidden Markov Model (HMM) and learning its parameters with the EM algorithm. The end-goal is correctly transcribing raw audio input into words.	✓	0.20	In this application, we are learning parameters to improve performance (i.e., transcription accuracy) and are not as interested in the parameters in and of themselves.
☑ We are trying to better understand high-energy physics by using a graphical model to analyze time-series data from particle accelerators. The hope is to elucidate the types of interactions between different particle types.	✓	0.20	In this application, we are interested in the actual value of the parameters, and we are using the data likelihood as a means of estimating these parameters accurately.
Total		1.00 / 1.00	