



Probabilistic Graphical Models

Daphne Koller, Kevin Murphy
Winter 2011-2012

[Home](#)
[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 — Parameter Learning in MNs

You achieved a score of 3.00 out of 3.00

Question 1

MRF Parameter Learning. Consider the process of gradient-ascent training for a log-linear model with k features, given a data set D with M instances. Assume for simplicity that the cost of computing a single feature over a single instance in our data set is constant, as is the cost of computing the expected value of each feature once we compute a marginal over the variables in its scope. Also assume that we can compute each required marginal in constant time after we have a calibrated clique tree.

Assume that we use clique tree calibration to compute the expected sufficient statistics in this model and that the cost of doing this is c . Also, assume that we need r iterations for the gradient process to converge.

What is the cost of this procedure? Recall that in big-O notation, same or lower-complexity terms are collapsed.

Your Answer	Score	Explanation
<input checked="" type="radio"/> $O(Mk + r(c + k))$	<input checked="" type="checkbox"/> 1.00	Before we start the gradient ascent process, we compute the empirical expectation for each of the k features by summing over their values for each of the M instances in our data set D ; the cost of this is Mk . Then, at each iteration, we use clique tree calibration at a cost c and extract the expected sufficient statistics from calibrated beliefs and update each of the k

parameters θ_i . Thus, the cost per iteration is $c + k$, and the total cost for r iterations is $r(c + k)$. Together with the initial computation of empirical expectations, we get a total cost of $O(Mk + r(c + k))$.

Total	1.00 / 1.00
-------	----------------

Question 2

CRF Parameter Learning. Consider the process of gradient-ascent training for a CRF log-linear model with k features, given a data set D with M instances. Assume for simplicity that the cost of computing a single feature over a single instance in our data set is constant, as is the cost of computing the expected value of each feature once we compute a marginal over the variables in its scope. Also assume that we can compute each required marginal in constant time after we have a calibrated clique tree.

Assume that we use clique tree calibration to compute the expected sufficient statistics in this model, and that the cost of running clique tree calibration is c . Assume that we need r iterations for the gradient process to converge.


What is the cost of this procedure? Recall that in big-O notation, same or lower-complexity terms are collapsed.

Your Answer	Score	Explanation
<input checked="" type="radio"/> $O(r(Mc + k))$	1.00	<p>When training the CRF, at each iteration we need to perform clique tree calibration and compute the expected value of each of the k features M times; thus, the computation at each iteration required $M(c + k)$ operations if done naively. If we aggregate the probabilities from these M clique trees into a single clique tree and then compute the feature value using the aggregated clique tree, we get $Mc + k$ operations per iteration; the procedure is correct due to linearity of expectations.</p>

Total 1.00 /
1.00

Question 3

Parameter Learning in MNs vs BNs. Compared to learning parameters in Bayesian networks, learning in Markov networks is generally...

Your Answer	Score	Explanation
<input checked="" type="radio"/> more difficult because we cannot use parallel optimization of subparts of our likelihood as we often can in BN learning.	 1.00	Correct. One trick that often makes Bayes Net learning more efficient is our ability to optimize each CPD independently after we have obtained our expected counts. Markov Net learning cannot be decoupled, as the partition function couples all parameters in Markov Nets.
Total	1.00 / 1.00	
