

10708 Graphical Models: Homework 4

Due November 12th, beginning of class

October 29, 2008

Instructions: There are six questions on this assignment. Each question has the name of one of the TAs beside it, to whom you should direct any inquiries regarding the question. Please submit your homework in two parts, one for each TA. Also, please put the TA's name on top of the homework.

The last problem involves coding. Do *not* attach your code to the writeup. Instead, copy your implementation to

`/afs/andrew.cmu.edu/course/10/708/your_andrew_id/HW4`

Refer to the web page for policies regarding collaboration, due dates, and extensions.

Note: Please remember to put your name and Andrew ID on the first page of your writeup.

1 Clique Tree Calibration Fixed-Point Dhruv [5 pts]

Prove that the clique beliefs $\pi(\mathbf{C}_i) = P(\mathbf{C}_i)$ and edge beliefs $\mu_{ij}(\mathbf{S}_{ij}) = P(\mathbf{S}_{ij})$ form a fixed point for the belief propagation algorithm for a clique tree, i.e., if we start BP with these beliefs, no messages will change them.

2 Markov Network Representations Amr [5 pts]

Figure 1 is a Markov Random Field where the potentials are defined on all cliques of three variables.

- (a) Convert the triangle graph on (A, B, C) with potential $\Psi(A, B, C)$ into a pairwise Markov Random Field by introducing a new variable X . Show the graph, as well as

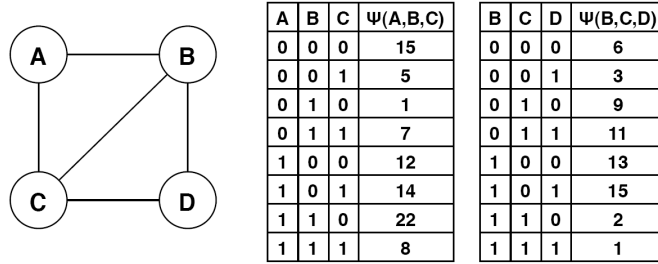


Figure 1: A chordal (triangulated) Markov network

the node and edge potentials in table form (*i.e.*, compute the values of the potentials in the pairwise MRF)

- (b) Convert the graph on (A, B, C, D) with potentials $\Psi(A, B, C)$ and $\Psi(B, C, D)$ into a pairwise Markov Random Field. Is the graph chordal? (*Note*: You do not have to compute the pairwise MRF potentials in your solution).

3 Hammersley-Clifford and Non-Positive Distributions

Amr [10 pts]

Complete the analysis of Example 4.3.5 (Koller & Friedman), showing that the distribution P defined in the example does not factorize over \mathcal{H} . (*Hint*: Use a proof by contradiction).

4 The Uniqueness of Minimal I-maps for MN Amr [10 pts]

Unlike directed models, where the minimal I-map for a given distribution is not unique, there exists a construction (Theorem 4.3.19 in Koller & Friedman) that results in the minimal I-map Markov Network (MN) for a positive distribution \mathcal{P} . The construction is as follows:

Theorem 4.3.19 Let \mathcal{P} be a positive distribution. For each node X , let $MB(X)$ be a *minimal* set of nodes U satisfying Eq. 1. We define a graph \mathcal{H} by introducing an edge $\{X, Y\}$ for all X and all $Y \in MB_{\mathcal{P}}(X)$. Then the Markov network \mathcal{H} is the unique minimal I-map for \mathcal{P} .

$$X \perp \mathcal{X} - \{X\} - U | U \tag{1}$$

Below we will prove the uniqueness and minimality of this construction. Consider some specific node X , and let \mathcal{U} be the set of all subsets U satisfying Eq. 1. Define \mathbf{U}^* to be the intersection of all $U \in \mathcal{U}$

- (a) Prove that $\mathbf{U}^* \in \mathcal{U}$. Conclude that $MB_P(X) = \mathbf{U}^*$.
- (b) Prove that if $P \models (X \perp Y | \mathcal{X} - \{X, Y\})$, then $Y \notin MB_P(X)$.
- (c) Prove that if $Y \notin MB_P(X)$, then $P \models (X \perp Y | \mathcal{X} - \{X, Y\})$.
- (d) Conclude that $MB_P(X)$ is precisely the set of neighbors of X in the graph \mathcal{H} defined above, showing that the construction above also produces a minimal I-map.

5 Variational Inference Amr [30 pts]

In this problem, you will investigate *mean field* approximate inference algorithms (Koller & Friedman 10.5). Consider the Markov network in Figure 2(a). Define edge potentials $\phi_{ij}(x_i, x_j)$ for all edges (x_i, x_j) in the graph. We can write

$$P(x_1, \dots, x_{12}) = \frac{1}{Z} \prod_{(i,j) \in E} \phi_{ij}(x_i, x_j)$$

- (a) Assume a fully factored mean field approximation Q (Figure 2(b)), parameterized by node potentials Q_i .
 - (i) Write down the update formula for $Q_1(x_1)$.
 - (ii) Write down the update formula for $Q_6(x_6)$.

In both cases, please expand out any expectations in the formulas (your answer should be in terms of Q_i and ϕ_{ij}).

- (b) Now we consider a structured mean field approximation Q (Figure 2(c)), parameterized by edge potentials $\psi_{ij}(x_i, x_j)$ for each edge (x_i, x_j) in Figure 2(c).

Using Theorem 10.5.15 (Koller & Friedman), the update equation for potential ψ_j is,

$$\psi_j(\mathbf{c}_j) \propto \exp \left[\sum_{\phi \in A_j} E_Q[\ln \phi | \mathbf{c}_j] - \sum_{\psi_k \in B_j} E_Q[\ln \psi_k | \mathbf{c}_j] \right],$$

where $A_j = \{\phi \in \mathcal{F} : \text{scope}(\phi) \text{ is not independent of } \mathbf{C}_j \text{ in } Q\}$ and $B_j = \{\psi_k : \mathbf{C}_k \text{ and } \mathbf{C}_j \text{ are not independent in } Q\}$.

- (i) Write down the update formula for $\psi_{34}(x_3, x_4)$

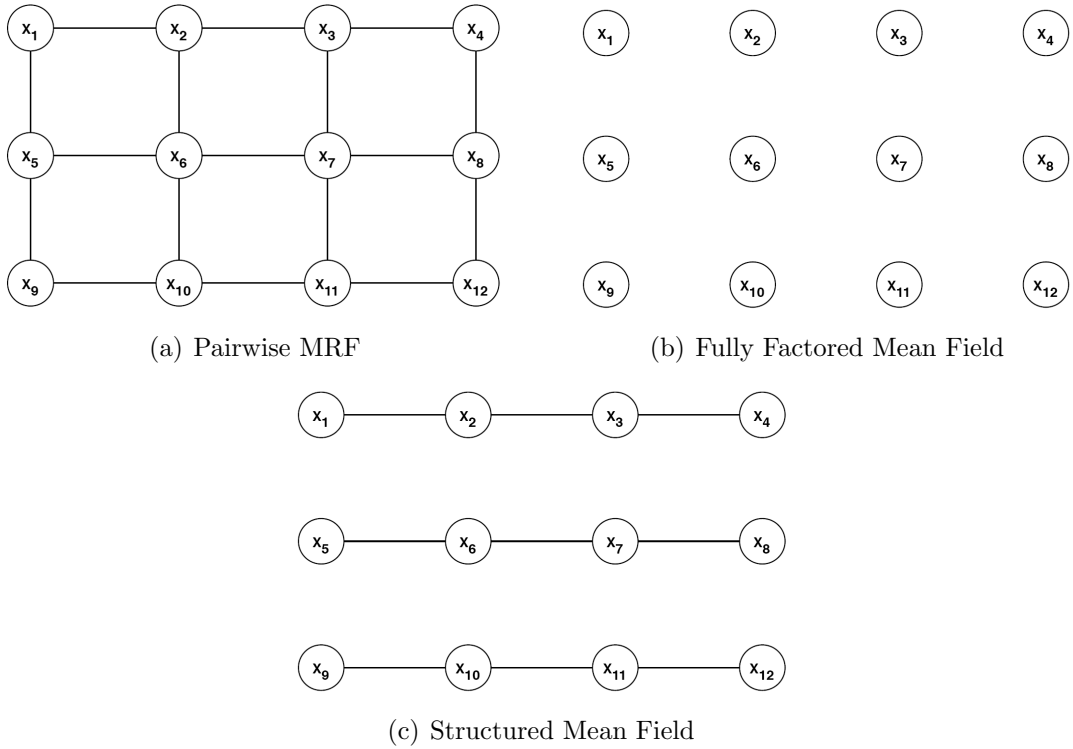


Figure 2: A pairwise Markov Random Field and the structure of two mean field approximations

- (ii) Write down the update formula for $\psi_{67}(x_6, x_7)$
 In both cases, write the required formula up to a proportionality constant. You can write it in terms of expected values, but do not include unnecessary terms.
- (iii) Write out the formula for $E_Q[\ln \phi_{12}(X_1, X_2)|x_3, x_4]$. Make sure to show how you would calculate the distribution that this expectation is over.
- (iv) Repeat for $E_Q[\ln \phi_{15}(X_1, X_5)|x_3, x_4]$.
- (v) Repeat for $E_Q[\ln \phi_{37}(X_3, X_7)|x_3, x_4]$.
- (c) For an $n \times n$ grid with k -ary variables:
 - (i) What is the computational complexity of a single potential update (like $Q_6(x_6)$) in the fully factored model?
 - (ii) What is the computational complexity of a single potential update (like $\psi_{67}(x_6, x_7)$) in the structured mean field model?

(Note: Do not include the cost of computing the normalization constant in your answer).

- (iii) Assuming No caching, what is the total cost in each case for a full iteration, i.e., that is computing the updates for all the potentials?
- (d) We would like to use caching to speed up computations in the structured mean field approach in Figure 2(c).
 - (i) What are the (conditional) marginal distributions under Q needed to calculate the update for $\psi_{34}(x_3, x_4)$.
 - (ii) Repeat for $\psi_{34}(x'_3, x'_4)$.
 - (iii) Repeat for $\psi_{12}(x_1, x_2)$.
 - (iv) Using the above intuition, sketch a scheme to schedule the updates over all $\psi_{XY}(X, Y)$ for all possible assignments to X and Y . You may use any exact inference algorithm as a subroutine.
 - (v) For an $n \times n$ grid with k -ary variables, what is the computational complexity for a single full iteration in your new scheme?

6 Semi-Supervised Image Segmentation with Loopy BP

Dhruv [40 pts]

Given an image of $l \times w$ pixels a K -ary segmentation is a clustering that assigns each pixel to one of K -classes, typically under the assumption that neighbouring pixels are more likely to belong to the same class.

In this question, you will implement an application of Loopy BP to the problem of interactive (or semi-supervised) image segmentation, where the goal is to produce a binary segmentation given user-provided scribbles or strokes on an image. Consider the image shown in figure 5(a). The user (let's call him Carlos) wants to cut himself out of this picture and paste it against a different background (he prefers mountains to beaches). Unfortunately, Carlos is also lazy, and is only willing to use a coarse paint-like interface and give us foreground/background labels for a few pixels. In figure 3(b), the green scribble corresponds to foreground and the blue scribble corresponds to background.

The most common graphical model approach for image segmentation represents the image as a grid-graph pairwise Markov random field (Figure 4) where each node corresponds to a pixel. Note that the value of a node is the cluster it belongs to.

Formally, the observed image is denoted $Y = \{Y_i\}$ and $X = \{X_i\}$, $X_i \in \{1 \dots K\}$ is the segmentation. The Markov random field has distribution

$$P(X, Y) = \frac{1}{Z} \prod_i \Phi(x_i, y_i) \prod_{(i,j) \in E} \Psi(x_i, x_j) \quad (2)$$



Figure 3: Interactive image segmentation

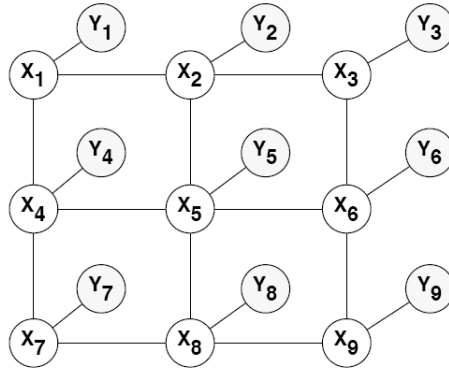


Figure 4: An example of a Markov Random Field for image segmentation

where Φ is the node potential¹, the effect that pixel y_i has on the label of x_i ; Ψ is the edge potential, how the label of x_i is influenced by the labels of its neighbours.

One common problem with this models is that even a (relatively) small image of size 300×300 pixels would contain 9×10^4 nodes in the MRF, and the size of the adjacency matrix for this graph would be 81×10^8 (although it would be sparse). A commonly used trick to get around this problem is to “over-segment” the image into small segments (called superpixels), and construct an MRF where the nodes correspond to superpixels instead of pixels. All the pixels in this superpixel image which have the same colour belong to the same superpixel, and will have the same foreground/background label because our MRF will assign labels on superpixels. Figure 5(b) shows a visualization of these superpixels where all the superpixels have been filled with random colours. Figure 5(c) shows what the MRF structure over superpixels would look like, where the nodes correspond to superpixels, and neighbouring (adjoining) superpixels have been connected by an edge. *Note:* The figure is a simplified visualization, and your resultant graph will have a lot more nodes and edges.

¹Also called the observation model or likelihood.

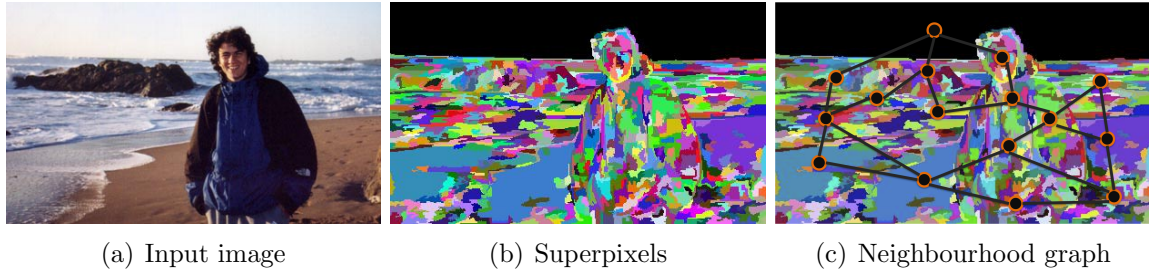


Figure 5: MRF on superpixels

6.1 Segmentation

This section describes what you need to implement. Please read the the readme file in the provided code for more details about the code provided. For this section we will redefine $X = \{X_i\}$ as the set of (binary) random variables representing class labels for superpixels and $Y = \{Y_i\}$ as the set of features extracted on superpixels ($i \in \{1, 2, \dots, S\}$, where S is the number of superpixels in the image) . The MRF distribution will stay the same (but the edge set E will change):

$$P(X, Y) = \frac{1}{Z} \prod_i \Phi(x_i, y_i) \prod_{(i,j) \in E} \Psi(x_i, x_j) \quad (3)$$

1. **[Pixel GMMs]** We have provided you with a scribble mask image (where a pixel has value 1 when it lies on the foreground scribble, 2 when it lies on the background scribble, and 0 otherwise). You will use the foreground and background pixels to learn Gaussian Mixture Models for each class. Set the number of components in each GMM to be 5. Your features for pixels will be Luv colour vectors. You may use external implementations for GMMs. A matlab implementation is available here:

<http://www.it.lut.fi/project/gmmbayes/>

Consider looking at the function `gmmb_create`.

Report the mean vectors (in the form of two 3×5 tables) for the foreground and background GMMs *in your writeup*.

2. **[Superpixel MRF structure]** We have provided you with a superpixel map, where each pixel holds the index of the superpixel it belongs to. Write a function called `convert_sp_labels_to_adj_mat.m` that takes in this matrix of superpixel labels and returns the adjacency matrix for the superpixel neighbourhood graph. If the image has been broken into S superpixels, this adjacency matrix will be of size $S \times S$, and will contain 1 whenever two superpixels are adjacent in the image (and 0 otherwise).

Hint: Consider scanning rows and columns of the superpixel map to find transitions.

Display the adjacency matrix using the function `imagesc` and **include** a snapshot *in your writeup*.

3. [**Superpixel MRF parameters**] Your features at superpixels (Y_i) will be the average colour (Luv vectors) of contained pixels. The node potential will be defined as the likelihood of these feature vectors under foreground and background GMMs:

$$\begin{aligned}\Phi(X_i = \text{fg}, y_i) &= P(y_i | \text{GMM}_{\text{fg}}) \\ \Phi(X_i = \text{bg}, y_i) &= P(y_i | \text{GMM}_{\text{bg}})\end{aligned}$$

Consider looking at the function `gmmb.pdf`.

The edge potential² will be defined as

$$\Psi(x_i, x_j) = \exp\{-\beta \times I(x_i \neq x_j)\},$$

where I is an indicator function.

4. [**Loopy BP**] Write a function `lbp.m` that takes as input the graph structure (adjacency matrix), the node potentials, and the edge potentials, and returns the MAP estimates of the states of the nodes via (sum-product) Loopy Belief Propagation.

Initialize $m_{ij}(x_i) = 1$ for all $i \neq j$. Stop running loopy belief propagation once the maximum absolute difference between an old message and a new message is less than 10^{-5} . Remember to compute the messages in log-space, for numerical stability. Also remember to normalize messages as listed on the lecture slides. You do not need to dampen messages to ensure convergence on this image.

5. [**Plotting Segmentations**] Vary β in the range $[0, 10]$ (take steps of size 2) and propagate superpixel labels back to pixel-level foreground/background segmentations (where all pixels in the image have been assigned to either foreground (1) or background (0)).

Include plots of these segmentations *in your writeup*. Comment on the behaviour as β increases. Why is $\beta = 0$ special? What would happen as $\beta \rightarrow \infty$?

6.2 Extra Credit [8 pts]

Show us your creative side. What can you do with a cut-out of Carlos? Paste Carlos against interesting backgrounds (smiling next to an alien ship or standing next to important presidents), and submit the images.

²Also called the Potts Model.