CS 683: Belief Propagation and VC Dimension

Shaomei Wu, June Andrews
shaomei@gmail.com, astuteAjax@gmail.com

April 25, 2008

1 MWM with Belief Propagation: A Summary

[1]

Before going into the new topic of VC dimension, let's summarize our discussion about the belief propagation algorithm with the maximal matching example.

Prove that the solution of belief propagation algorithm converges to the maximum weight matching (MWM) for given complete bipartite graph with n nodes at each side.

Assume that the MWM is unique, and the weight difference between MWM and the second maximal weight match is ϵ, where $\epsilon > 0$. Let w_{max} be the maximal weight for a single edge in the bipartite graph.

The outline of proof:

1. Construct a k levels tree-shape factor graph to represent the given MWM problem.
2. Run belief propagation algorithm over the factor graph and converge to an **optimal solution** Λ^{opt}.
3. Assume that Λ^{opt} is not identical as the **real MWM** Π in the bipartite graph. Without loss of generality, let's further assume that one of the incorrect edges in Λ^{opt} is at level 1 of tree. Then, there must be an **alternative path** P consisting of two segments: one from the level 1 Λ^{opt}

[1] This part of note is generated with great reference to the original paper by Mohsen Bayati, Devavrat Shah, and Mayank Sharma, *"Max-Product for Maximum Weight Matching: Convergence, Correctness, and LP Duality"*, in IEEE Transactions on Information Theory, VOL.54, NO.3, March 2008.

node all the way down to a leaf variable node, another from the level 1 Π **node** to another leaf variable node. By saying **alternative path**, we mean that the edges of P come from either Λ^{opt} or Π, alternatively.

4. P can be **decomposed** into at least $\frac{k}{n}$ **cycles** in the bipartite graph after adding two extra edges.

5. Since each cycle is a part of P, it is a chain of alternative edges from Λ^{opt} and Π.

   In each cycle, it suffices that the weight of edges from Π is greater than the weight of edges from Λ^{opt} by at least ϵ:

$$\sum_{C_{i,j}\in\Pi, C_{i,j}\in Cycle_l} Weight(C_{i,j}) - \sum_{C_{i,j}\in\Lambda^{opt}, C_{i,j}\in Cycle_l} Weight(C_{i,j}) > \epsilon. \tag{1}$$

6. By replacing edges from Λ^{opt} with edges from Π in each cycle, and removing the two extra edges, we get a new solution for the factor graph as Λ. Because the difference between Λ and Λ^{opt} comes from the substitution of Λ^{opt} edges with Π edges in all cycles (and also the two edges adding to P to form full cycles), assume the number of cycle is m, we then have:

$$Weight(\Lambda) - Weight(\Lambda^{opt}) > m\epsilon - 2w_{max} > \frac{k}{n}\epsilon - 2w_{max}. \tag{2}$$

   As Λ^{opt} is the optimum given by the belief propagation at level k, and the message passed in this factor graph has been designed such that the optimal solution will achieve maximal sum of weights of selected edges, with respect to all the constraints, there must be

$$Weight(\Lambda^{opt}) - Weight(\Lambda) > 0$$

   , together with the inequality (2), we have:

$$0 > Weight(\Lambda) - Weight(\Lambda^{opt}) > \frac{k}{n}\epsilon - 2w_{max}. \tag{3}$$

7. When k goes to infinity, $(\frac{k}{n}\epsilon - 2w_{max})$ will be greater than zero, hence contracdicts to the initial assumption that $\Lambda^{opt} \neq \Pi$.

Now we will show some more details of above proof with an example.

First assume that the original bipartite is given as in figure 1, where the edges in the maximal matching are solid and the other edges are dotted.

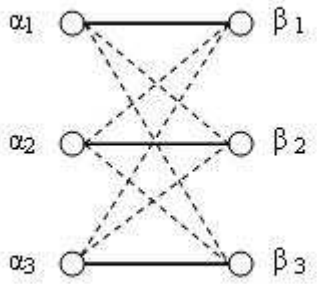

Figure 1: Bipartite graph

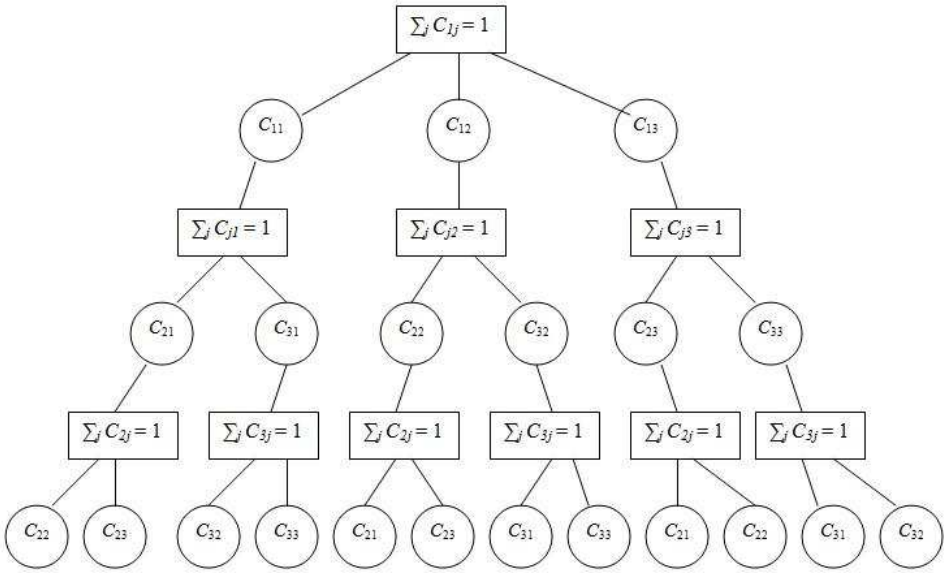

Figure 2: The factor graph (computational tree) for 2×3 bipartite graph

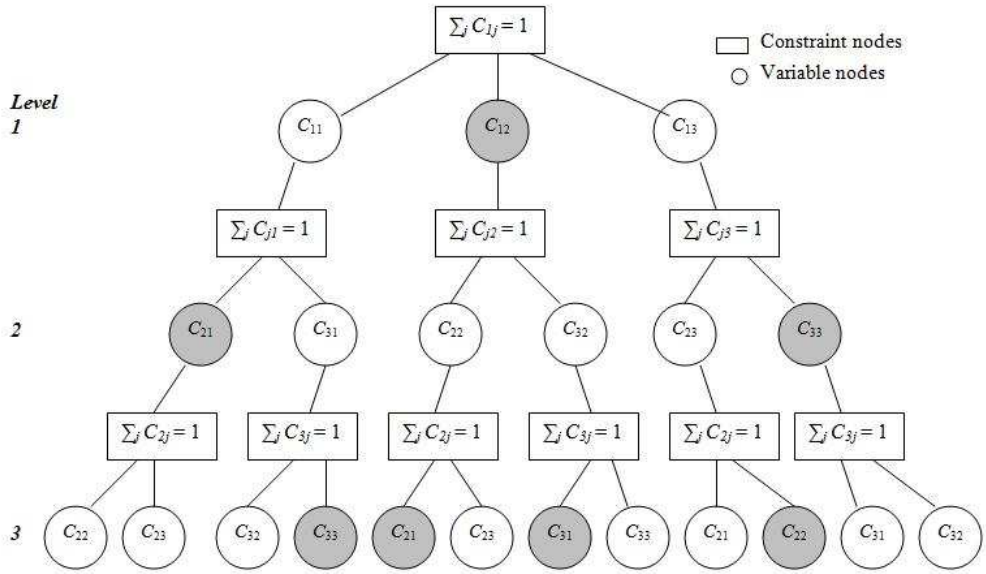

Figure 3: Optimum Λ^{opt} for the computational tree

Then we construct a 3-level factor graph as in figure 2. This graph can also be called a computational tree.

It's known that by running on a factor graph without loop, the message passing algorithm will converge to a solution, which is an assigment of variables nodes C_{ij}'s. Reflecting in the original bipartite graph, $C_{i,j} = 1$ means the edge (α_i, β_j) is selected. As presented in previous lectures, the messages passed between nodes are designed so that: for each constraint node, there is only one variable node selected among its adjacent variable nodes.

Assume that the optimal solution Λ^{opt} found by belief propagation algorithm is not the MWM in the original bipartite graph. As shown in figure 3, the shadowed nodes belong to Λ^{opt}.

As we know that the MWM $\Pi = \{C_{11}, C_{22}, C_{22}\}$, thus at the first level, node $C_{11} \in \Pi$, while $C_{12} \in \Lambda^{opt}$. Hence, we can find the alternative path $P = \{C_{22}, C_{21}, C_{11}, C_{12}, C_{22}, C_{21}\}$, as marked by the bold lines in figure 4. We can also see clearly from figure 4 that nodes from Π and nodes from Λ^{opt} appear alternatively along P.

Such alternative path P can be decomposed to a cycle C_1 and a subpath P_1 in the original bipartite graph, as shown in figure 5.

Now, for cycle C_1, it is clear that if we pick the edges $\{C_{11}, C_{22}\}$ and instead of $\{C_{12}, C_{21}\}$, we can gain weight for the matching by at least ϵ. Repeat this

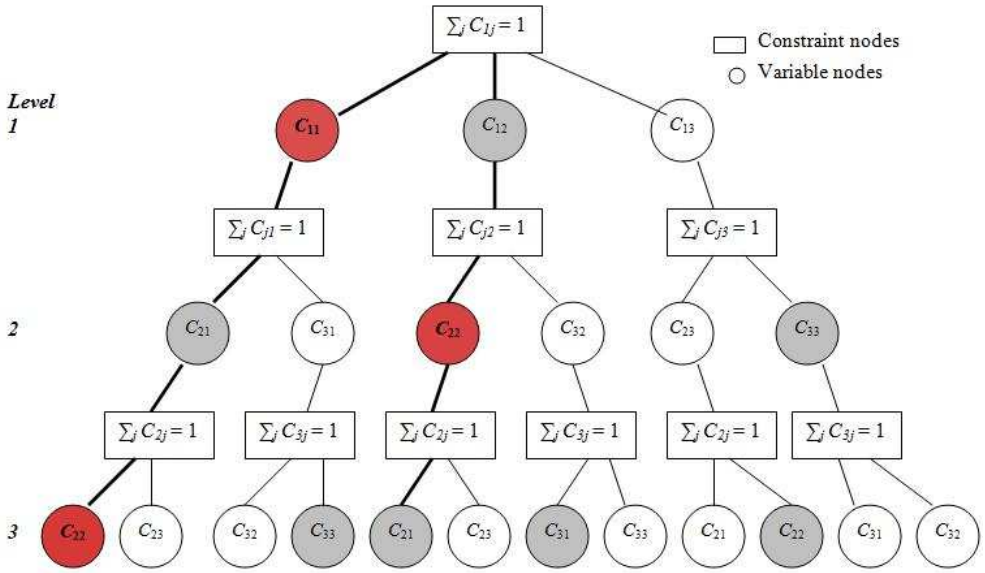

Figure 4: Alternative path in the computational tree

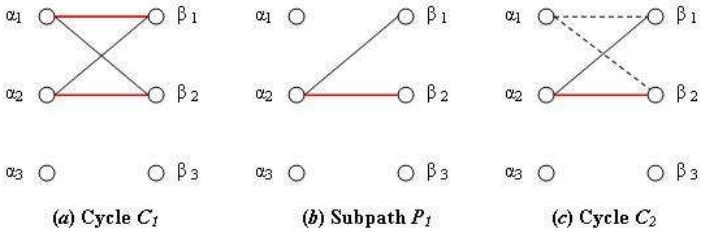

Figure 5: Projection of the alternative path P is decomposed to (a):cycle C_1 of length 4, and (b): path P_1 of length 2. Bying adding two extra edges C_{12}, C_{21}, we can complete P_1 to another cycle C_2.

process to cycle C_2. Eventually, by replacing edges from Λ^{opt} with edges from Π in all the cycles(including the cycle with two extra edges), we can get a new set of edges Λ'. In general, Λ' **may not be a solution** for the computational tree as Λ' may contain the **two extra edges**. Removing those two edges from Λ' if they exist, we will finally get a set of edges Λ which should also satisfy the constraints and thus be **a new solution** for the computational tree. In our particular example, the nodes in Λ are colored in red in figure 4.

Above reasoning gives us the following inequalities:

$$Weight(\Lambda') - Weight(\Lambda^{opt}) > m\epsilon,$$

$$Weight(\Lambda') - Weight(\Lambda) \leq 2w_{max}.$$

Combine them, we will get

$$Weight(\Lambda) - Weight(\Lambda^{opt}) > m\epsilon - 2w_{max} > \frac{k}{n}\epsilon - 2w_{max}.$$

2 VC Dimension

Given a set of data points in $\Re^2$ we want a way to compress the data, but retain enough information to answer all possible questions. For example a Censes Bureau may collect data $(age, income) \in \Re^2$ and want to be able to ask questions of the form how many people fall into the square of data $([age_1, age_2], [income_1, income_2])$. We could store every square needed to cover all possible subsets of data points and conquentially enough to answer all possible questions, but this is on the order of $O(2^n)$. However, if we store all equivalence classes of rectangles(ie the set of all rectangles defined by the data points, to become clear later), the amount of storage becomes $O(n^4)$. If we have the set of all rectangles defined by the data points, then we can answer all questions, as any answer must be itself a rectangle that reduces to a rectangle defined by the data points. Note: the change from $O(2^n)$ to $O(n^4)$ implies that some subsets can never be answers.

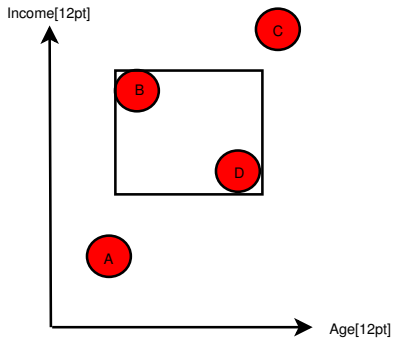

Figure 6: Census Bureau Data

Consider the example Census Bureau data. Note that any rectangle question whose answer is is the subset B, D is equivalent to the shown rectangle. Also note, that an answer of the subset A, C is impossible as any rectangle range would have to include both B and D. The term 'defined by the data' refers to the minimum rectangle needed to contain a given subset.

In general we note, without backup, that if the answer is a circular shape there are $O(n^3)$ equivalent circles defined by the data. If arbitray shapes can be answers then any subset is a possible answer and need $O(2^n)$ equivalent answers.

Given our understanding of precise compression, we want to now estimate a probability mass of how many data points are in a region. ie

$$|Probability\ of\ mass\ -\ estimate| < \epsilon\ with\ Probability\ (1-\delta) \tag{4}$$

where, δ depends on ϵ and VC dimension of the estimate.

To understand VC dimension we will need the concept of shattering, as VC dimension is the largest cardinality of a set that a given shape can shatter.

Definition 1 *If a shape shatters a set of data, S then, every subset $A \subset S$, of the data, can be enclosed in a corresponding shape and A^c is outside of the corresponding shape.*

Figure 7 presents an example of a rectangle *shattering* three points. Also, note that the shape can be expanded in many ways as long as the new shape is still considered to be in the class of the specified shape. In the figure, the class of shapes is rectangles with parallel and perpendicular edges to the axises. So we can use rectangles of different size and length-height ratios, but they are all oriented the same way.

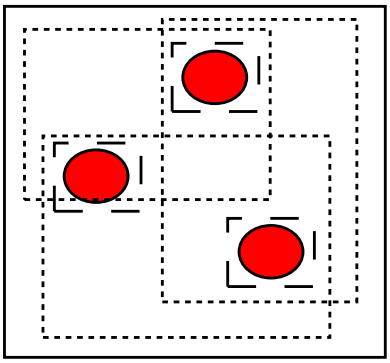

Figure 7: Shattering of 3 data points

Note: in general it is possible to find a set of shapes less than the VC dimension of a shape that the shape can not shatter, in most cases data points on a line can not be shattered.

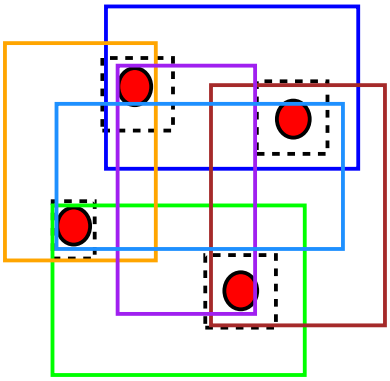

Figure 8: All subsets of size 2 encapsulated by oriented rectangles.

Using the shape of oriented rectangles used in figure 7, we now show that its VC dimension is 4.

In figure 8, we show the encapsulating of subsets of size 1 and 2 by oriented rectangles. It is not hard to find the recatngles needed for subsets of size 3 and 4. Hence VC dimension of the oriented rectangle is at least 4.

Now, to show that $\nexists$ a set of 5 points that oriented rectangles can shatter. To place the 5^{th} point, we must place it outside the convex hull of the rectangle defined by the first 4 points. (If it is placed within the convex hull of the subset of the first 4 points, then $\nexists$ a rectangle that contains the first 4 points but not the 5^{th}.) Likewise, if it is placed outside of the hull, then we can choose one of the first 4 points to not be in the subset, but inside the hull of the other points. Consquentially, the 5^{th} can only be placed on the boundary of the hull of the first 4 points. Now, since the first 4 points were shatterable, each side of a rectangle containing all of them, touches exactly one point. Hence the 5th point must share an edge with another point. Let the subset be such that it contains the 5^{th} point, but not the point that shares the edge with the 5^{th} point. This subset can not be ecapsulated by an oriented rectangle. Hence, there is no place a 5^{th} point can be placed, irregardless of the first 4 points. Hence oriented rectangles can not shatter 5 points, or any larger set. So, VC dimension of oriented rectangles is 4.

We leave estimation of Probability mass to next lecture.