Homework 9

Assigned: 21/11/2011.
Due: 02/12/2011.

Exercise 1. CONVEXITY OF BETHE VARIATIONAL RELAXATION
Recall the Bethe variational problem from the lecture:

$$\max_{\boldsymbol{\mu}\in\mathcal{M}_{\text{local}}} \boldsymbol{\theta}^T\boldsymbol{\mu} + \mathrm{H}_{\text{Bethe}}[\boldsymbol{\mu}], \quad \mathrm{H}_{\text{Bethe}}[\boldsymbol{\mu}] = \sum\nolimits_j \mathrm{H}[\mu_j(\boldsymbol{x}_{C_j})] - \sum\nolimits_i (n_i - 1)\mathrm{H}[\mu_i(x_i)],$$

$$\mathcal{M}_{\text{local}} = \left\{ \boldsymbol{\mu} \succeq \mathbf{0} \,|\, \sum\nolimits_{\boldsymbol{x}_{C_j\setminus i}} \mu_j(\boldsymbol{x}_{C_j}) = \mu_i(x_i),\ \sum\nolimits_{x_i} \mu_i(x_i) = 1 \right\}, \quad n_i = |\{j \,|\, i \in C_j\}|.$$

Also, recall that $\mu_j(\boldsymbol{x}_{C_j}) \mapsto \mathrm{H}[\mu_j(\boldsymbol{x}_{C_j})] - \mathrm{H}[\mu_i(x_i)]$ is concave, where $i \in C_j$ and $\mu_i(x_i) = \sum_{\boldsymbol{x}_{C_j\setminus i}} \mu_j(\boldsymbol{x}_{C_j})$ (you'll prove this below). In this exercise, you will show that the Bethe problem is a convex minimization (or concave maximization) problem for every connected factor graph with no more than one cycle.
Note: This exercise is about factor graphs, which are bipartite graphs, whose nodes come in two types (factor, variable), and whose edges are always between nodes of opposite type. Terms like "node", "edge", and "cycle" always refer to this graph, not a potential directed or undirected model graph it was constructed from.

1. **[4 points]** Let X, Y be discrete random variables, with joint distribution $P(X, Y)$. Prove that the conditional entropy as function of $P(X, Y)$,

$$P(X, Y) \mapsto \mathrm{H}[Y|X] = -\sum\nolimits_{X,Y} P(X, Y) \log P(Y|X)$$

   is concave. Note that this is not trivial: while $\mathrm{H}[X, Y]$ is known to be concave, $-\mathrm{H}[X]$ is convex, and $\mathrm{H}[Y|X]$ is the sum of these.
   Hint: Use the fact (you do not have to prove it) that the function $(\boldsymbol{q}, \boldsymbol{p}) \mapsto \mathrm{D}[\boldsymbol{q} \,\|\, \boldsymbol{p}]$, mapping a pair of distributions over the same variable(s) to their relative entropy, is convex. Pick the uniform distribution $U(Y)$ for Y (constant $1/|\mathcal{Y}|$ on the range $\mathcal{Y}$ for Y), and relate $\mathrm{H}[X|Y]$ to $\mathrm{D}[P(X, Y) \,\|\, P(X)U(Y)]$, where $P(X) = \sum_Y P(X, Y)$. Now, use the convexity of relative entropy. Please give a clean proof, using the definition of convexity.

2. **[5 points]** Recall the resource allocation game from the lecture. You assign variable nodes i to factor nodes j, where $i \in C_j$ (there is an edge between i and j), or, equivalently, entropy terms $\mathrm{H}[\mu_j(\boldsymbol{x}_{C_j})]$ to neg-entropy terms $-\mathrm{H}[\mu_i(x_i)]$, with the goal of matching up all terms in $\mathrm{H}_{\text{Bethe}}[\boldsymbol{\mu}]$. If this can be done without any neg-entropy terms left, the Bethe entropy is concave. Show that for any factor graph $\mathcal{G}$ that is a connected tree and any factor node j of $\mathcal{G}$, terms can be matched up, so that $\mathrm{H}[\mu_j(\boldsymbol{x}_{C_j})]$ is left at the end. This implies that $\mathrm{H}_{\text{Bethe}}[\boldsymbol{\mu}] - \mathrm{H}[\mu_j(\boldsymbol{x}_{C_j})]$ is concave.
   Hint: Analyze the following scheme, proving that it works. Designate factor node j as root. For every leaf node, move along its path to the root. Whenever stepping from factor to variable node, direct the edge along the direction of travel (draw an arrow). At the end, the matching is given by the directed edges $j' \to i'$. Please be precise in your arguments, making use of properties of trees (graphs without cycles). Why is the path from a leaf to the root unique? Why is every edge travelled along? Why can an

edge not be travelled along in both directions? Why does every factor node $j' \neq j$ have exactly one outgoing arrow? For every variable node i, why are there precisely $n_i - 1$ incoming arrows? Finally, state why this implies that $\mathrm{H}_{\mathrm{Bethe}}[\boldsymbol{\mu}] - \mathrm{H}[\mu_j(\boldsymbol{x}_{C_j})]$ is concave.

3. **[3 points]** Show that for a connected factor graph with a single cycle, $\mathrm{H}_{\mathrm{Bethe}}[\boldsymbol{\mu}]$ is concave, therefore the Bethe variational problem is a convex minimization problem (because the energy term is concave, and the constraint set $\mathcal{M}_{\mathrm{local}}$ is convex).
Hint: Cut an edge in the cycle, what do you get? Make use of the result you just showed (if you did not manage, use it anyway: it is true).

The convexity of the Bethe problem does not necessarily imply that LBP, applied to a graph with a single cycle, always converges. However, this can be proven. The solution obtained (pseudomarginals solving the Bethe problem) is not correct in general (recall that many of the counterexamples in the lecture are in fact graphs with a single cycle), but can be related analytically to the true solution. Unfortunately, all this does not imply much for graphs with more than one cycle, unless the resource allocation game works for them as well. This game can be played "fractionally", to derive a class of convex relaxations with associated reweighted LBP algorithms, a special case of which is Wainwright's TRW-LBP (your last exercise sheet).

Exercise 2. Cycle Inequalities
With the exception of the semidefinite variational inference relaxation discussed in the lecture, most others we have seen so far make use of the local consistency polytope $\mathcal{M}_{\mathrm{local}}$. Especially w.r.t. LP relaxations, where the only thing that matters is the outer bound on $\mathcal{M}$, we would like to have other options. It would be nice to have a method that starts with $\mathcal{M}_{\mathrm{local}}$, then adds additional inequalities (chops on the block of marble), until we have enough (our computational resources or our patience runs out). Methods like this are called *cutting plane* algorithms. In this exercise, you will learn about an interesting option, making use of cycle inequalities.
We will restrict ourselves to pairwise MRFs with binary variables $x_s \in \{0, 1\}$, with potentials on edges $(s, t) \in E$ and nodes $s \in V$, the graph being $\mathcal{G} = (V, E)$. Assume that $\mathcal{G}$ is connected, but not a tree. We'll use the usual binary indicator statistics $\boldsymbol{f}(\boldsymbol{x})$, so that the marginal components are $\mu_{st}(\tilde{x}, \tilde{x}') = \Pr\{x_s = \tilde{x}, x_t = \tilde{x}'\}$, $\mu_s(\tilde{x}) = \Pr\{x_s = \tilde{x}\}$. We are after a pretty generic method here, which evolves a sequence $\mathcal{M}_0 \supset \mathcal{M}_1 \supset \cdots \supset \mathcal{M}$, where $\mathcal{M}_0 = \mathcal{M}_{\mathrm{local}}$, the local marginalization polytope, and each subsequent set is smaller. We are not going to discuss applications of this technique in detail, but let me motivate the flavour to you. Suppose we want to approximate the MAP problem of solving the LP $\max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\theta}^T \boldsymbol{\mu}$. For some graphs $\mathcal{G}$, this problem is NP hard in general. In any relaxation, you evolve some pseudomarginal vector $\boldsymbol{\mu}$, living in a tractable superset of $\mathcal{M}$. In a cutting plane technique, this tractable set shrinks as you go. You iterate $\boldsymbol{\mu}_k \leftarrow \operatorname{argmax}_{\boldsymbol{\mu} \in \mathcal{M}_{k-1}} \boldsymbol{\theta}^T \boldsymbol{\mu}$, then find some new inequalities that hold for every $\boldsymbol{\mu} \in \mathcal{M}$, but are violated by $\boldsymbol{\mu}_k$. You add these inequalities to $\mathcal{M}_{k-1}$, which gives the smaller $\mathcal{M}_k$, and so on. This process may terminate with some $\mathcal{M}_k$ still not equal to $\mathcal{M}$, because generating new violated constraints may be too hard, but still it's better than just sticking with $\mathcal{M}_0$.

1. **[4 points]** Our graph $\mathcal{G} = (V, E)$ has cycles. Pick one of them, say $C \subset E$ (C contains the edges of the cycle). For any assignment, count the number of value flips along the cycle (theory people call $(s, t) \in C$ with $x_s \neq x_t$ a *cut*). This number has to be even, because you have to end at the value you started from. Show that for every cycle C and every subset of edges $F \subset C$, where $|F|$ is odd, this fact implies an inequality involving $\mu_{st}(x_s, x_t)$ along $(s, t) \in C$, for any $\boldsymbol{\mu} \in \mathcal{M}$. This is called a *cycle inequality*.
Hint: Use the indicator function $\sum_{(s,t) \in C \setminus F} \mathrm{I}_{\{x_s \neq x_t\}} + \sum_{(s,t) \in F} \mathrm{I}_{\{x_s = x_t\}}$. What's the smallest value for any assignment? Take expectations.

2. **[4 points]** Let us simplify notation. Define $\mu_{s=t} := \mu_{st}(0,0) + \mu_{st}(1,1)$, $\mu_{s\neq t} := \mu_{st}(0,1) + \mu_{st}(1,0)$. These are not coefficients of $\boldsymbol{\mu}$, but functions thereof. Your answer in the previous part depends on them. There are very many cycle inequalities for $\mathcal{G}$. Given some $\boldsymbol{\mu} \in \mathcal{M}_k$ at some point in our method, we have to find a violated inequality rapidly (trial and error is out of the question). Here is a construction how to do that (in order to understand it, draw it for some graph, say the complete graph with 4 nodes). We construct a new graph $\mathcal{G}'$ as follows. Start with two copies of $\mathcal{G}$, numbered 0 and 1. The nodes are called $s[0], s[1]$ for each $s \in V$. For each edge $(s,t) \in E$, there is $(s[j], t[j]) \in E'$, $j = 0, 1$, but also add the cross-edges $(s[0], t[1])$, $(s[1], t[0])$ to E'. This means that $|V'| = 2|V|$ and $|E'| = 4|E|$, which is reasonable. Now, put weights on the edges in E'. For each $(s,t) \in E$, weight $\mu_{s\neq t}$ to $(s[j], t[j])$, and weight $\mu_{s=t}$ to $(s[j], t[1-j])$, $j = 0, 1$.
   Now to the exercise. Suppose we have $\boldsymbol{\mu} \in \mathcal{M}_k$ and would like to find a violated cycle inequality, or otherwise output that $\boldsymbol{\mu}$ satisfies all possible cycle inequalities. Relate the *shortest path* (with the least sum of edge weights along the path) from $s[0]$ to $s[1]$ for any $s \in V$ to this problem. Shortest paths can be found efficiently by Dijktra's algorithm, which can be modified to search over all paths $s[0] \to s[1]$, $s \in V$, at the same time.
   Hint: What is the weight of any such path? How many times do you have to jump between the replicas 0, 1? Why not relate such cross-edge traversals with F? Be careful with your argument, mapping back a path $s[0] \to s[1]$ to a cycle in $\mathcal{G}$. Why can't it contain a subpath $t[j] \to t[1-j]$ for some $t \neq s$ and some $j \in \{0, 1\}$?

This renders a tractable method. Start with $\mathcal{M}_0$, find a MAP solution $\boldsymbol{\mu}_1$ there (say, by reweighted max-product). Find a violated cycle inequality, add it to the description of $\mathcal{M}_0$ to get $\mathcal{M}_1$, solve for $\boldsymbol{\mu}_2$ (this cannot be done by rewighted max-product anymore, but in their ICML 2008 paper, Kumar and Torr show how to extend this class of efficient dual LP solvers to respect cycle inequalities as well), iterate this process. Since $\boldsymbol{\mu}_k \in \mathcal{M}_{k-1} \setminus \mathcal{M}_k$, the sets shrink. The algorithm might terminate, or you may have enough and just stop. Mind you: even if you enforce all cycle inequalities, this does not get you $\mathcal{M}$, but still an outer bound. On the other hand, recall from the lecture that once you find some $\boldsymbol{\mu}_k$ which is *integral*, meaning that $\mu_s(0) \in \{0, 1\}$ for all $s \in V$, then you found a solution to the original LP. This is a lucky case, but it tends to happen on many problems. It might even happen with $\boldsymbol{\mu}_1$, but if $\boldsymbol{\mu}_1$ is fractional, you can run the cutting plane method above in order to improve your chances. When you stop with some $\boldsymbol{\mu}_k$ with is partly integral, there is in general no guarantee that this is part of a solution. You'll always have $\boldsymbol{\theta}^T \boldsymbol{\mu}_k$ as a lower bound to $\max_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\theta}^T \boldsymbol{\mu}$, although in general, you cannot say how close you are. Of course, $\boldsymbol{\mu}_k$ does not really constitute a solution to the original integer program, because it is not integral. In some cases, you can produce an integral vector from $\boldsymbol{\mu}_k$ by a rounding procedure, and possibly even obtain a relative approximation guarantee, but that depends on the specific problem.