CS 228T Problem Set 4

May 20, 2011

Instructions. The lengths listed for each problem are suggested *maximum* lengths for typed solutions, not minimum; solving the problems fully in less space is possible. Some questions may be related to published research papers, so do not refer to any outside sources to complete this assignment, in accordance with the honor code. If you work in groups, indicate in your solutions who you worked with.

1. *Markov network structure learning* (16 points, 2 pages).

   (a) Do exercise 20.16 from the book.

   (b) Do exercise 20.18 from the book.

2. *Cheeseman-Stutz lower bound* (14 points, 2 pages). Suppose we have a directed acyclic graph with parameters θ giving rise to an i.i.d. dataset $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with corresponding latent variables $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$, where each latent variable has cardinality k. Recall that the *incomplete data likelihood* is

   $$p(\mathbf{x} \,|\, \theta) = \sum_{\mathbf{z}} p(\mathbf{z} \,|\, \theta) p(\mathbf{x} \,|\, \mathbf{z}, \theta),$$

   and the *marginal likelihood*, or *evidence*, is

   $$p(\mathbf{x}) = \int_{\Theta} p(\theta) p(\mathbf{x} \,|\, \theta) \; d\theta = \int_{\Theta} p(\theta) \sum_{\mathbf{z}} p(\mathbf{z} \,|\, \theta) p(\mathbf{x} \,|\, \mathbf{z}, \theta) \; d\theta.$$

   The marginal likelihood is the key quantity used in Bayesian model selection tasks, such as structure learning, selecting the cardinality of latent variables, selecting the dimensionality of vectors of latent variables, and so on. Unfortunately, it is intractable to compute in most situations of interest, hence the need for approximations.

   Let $\hat{\theta}$ be the result of the M-step of EM, and let $\{p(\mathbf{z}_i \,|\, \mathbf{x}_i, \hat{\theta})\}_{i=1}^N$ be the set of posteriors over the hidden variables obtained in the subsequent E-step of EM. Let $\hat{\mathbf{z}} = \{\hat{\mathbf{z}}_i\}_{i=1}^N$ be a completion of the hidden variables such that $\hat{\mathbf{z}}_{ij} = p(\mathbf{z}_i = j \,|\, \mathbf{x}, \hat{\theta})$, for $i = 1, \ldots, N$. The *Cheeseman-Stutz approximation* to the marginal likelihood is given by

   $$\hat{p}_{\mathrm{CS}}(\mathbf{x}) := p(\mathbf{x}, \hat{\mathbf{z}}) \frac{p(\mathbf{x} \,|\, \hat{\theta})}{p(\hat{\mathbf{z}}, \mathbf{x} \,|\, \hat{\theta})}.$$

Show that the Cheeseman-Stutz approximation provides a lower bound to the marginal likelihood, *i.e.*, that

$$p(\mathbf{x}, \hat{\mathbf{z}}) \frac{p(\mathbf{x} \,|\, \hat{\theta})}{p(\hat{\mathbf{z}}, \mathbf{x} \,|\, \hat{\theta})} \leq p(\mathbf{x}).$$

Hints. Consider Jensen's inequality.

You can assume that there exists a $\hat{\mathbf{z}}_i$ such that

$$\log p(\hat{\mathbf{z}}_i, \mathbf{x} \,|\, \theta) = \sum_{\mathbf{z}_i} p(\mathbf{z}_i \,|\, \mathbf{x}, \hat{\theta}) \log p(\mathbf{x}_i, \mathbf{z}_i \,|\, \theta).$$

3. *Maximum entropy learning* (18 points, 2 pages).

   (a) Do exercise 20.9a from the book.

   (b) Do exercise 20.10 from the book, but use the following problem instead:

$$\begin{array}{lll} \text{maximize} & \sum_{i=1}^{n} H(\beta_i) & \\ \text{subject to} & \mathbb{E}_{\beta_i}[f_{ij}] = \mathbb{E}_{\mathcal{D}}[f_{ij}] & \forall i, j \\ & \sum_{c_i} \beta_i(c_i) = 1 & \forall i \\ & \beta_i(c_i) \geq 0 & \forall i, c_i, \end{array}$$

   with variables $\beta_i(c_i)$.

4. *Max-margin Markov networks* (17 points, 1 page). Let $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ be a labeled training set. Suppose we have a mapping from an input $\mathbf{x}$ to the corresponding Markov network graph $G(\mathbf{x}) = (\mathcal{V}, \mathcal{E})$, where the nodes $\mathcal{V}$ correspond to the variables in $\mathbf{y}$, and let $G_i = G(\mathbf{x}_i)$. Suppose the graph is a log-linear conditional random field that represents the conditional distribution $p(\mathbf{y} \,|\, \mathbf{x})$. In particular, suppose the CRF is defined via

$$\log p_{\mathbf{w}}(\mathbf{y} \,|\, \mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}) - \log Z_{\mathbf{w}}(\mathbf{x}).$$

   Moreover, assume that the graph is tree-structured. We now consider margin-based training approaches for this model.

   Maximum-margin estimation of this model can be formulated as solving the following optimization problem:

$$\begin{array}{lll} \text{minimize} & (1/2)\|\mathbf{w}\|_2^2 + C \sum_{i=1}^{N} \xi_i & \\ \text{subject to} & \mathbf{w}^T \delta\phi_i(\mathbf{y}) \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, & \forall i, \mathbf{y} \in \mathcal{Y} \\ & \xi_i \geq 0, & \forall i, \end{array}$$

   where $\delta\phi_i(\mathbf{y}) = \phi(\mathbf{x}_i, \mathbf{y}_i) - \phi(\mathbf{x}_i, \mathbf{y})$. (This is the primal structural SVM with margin rescaling.)

   The dual is given by

$$\begin{array}{lll} \text{maximize} & \sum_{i=1}^{N} \sum_{\mathbf{y}} \alpha_i(\mathbf{y}) \Delta(\mathbf{y}_i, \mathbf{y}) - (1/2) \| \sum_{i=1}^{N} \sum_{\mathbf{y}} \alpha_i(\mathbf{y}) \delta\phi_i(\mathbf{y}) \|_2^2 & \\ \text{subject to} & \sum_{\mathbf{y}} \alpha_i(\mathbf{y}) = C, & \forall i \\ & \alpha_i(\mathbf{y}) \geq 0, & \forall i, \mathbf{y}, \end{array}$$

with variables $\alpha_i(\mathbf{y})$. The primal formulation has exponentially many constraints in the number of labels, so the dual has exponentially many variables. For example, supposing that $\mathcal{Y} = \prod_{i=1}^{K} \mathcal{Y}_i$, where each $\mathcal{Y}_i = \{y_1, \ldots, y_d\}$, then K is the number of labels being simultaneously selected.

Assume that the loss function is decomposable, *i.e.*, that

$$\Delta(\mathbf{y}_i, \mathbf{y}) = \sum_{c \in C(G_i)} \Delta(\mathbf{y}_{ic}, \mathbf{y}_c),$$

where $C(G_i)$ denotes the cliques of G_i. (Hamming loss is a special case.) Assume that $\delta\phi_i$ decomposes similarly. Show how to use the structure of the model to reparameterize the dual as an equivalent problem with only polynomially many variables.

5. *Cutting plane methods for structural SVMs* (14 points, 1 page). Structural support vector machines extend standard support vector machines for structured prediction. The goal is to learn parameters $\mathbf{w}$ to predict a structured output $\mathbf{y} \in \mathcal{Y}$ from input features $\mathbf{x}$. Given a joint feature vector $\phi(\mathbf{x}, \mathbf{y})$ describing the relationship between inputs and outputs, we learn a linear prediction rule of the form

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$$

Let $\Delta(\mathbf{y}, \hat{\mathbf{y}})$ be the loss incurred by predicting $\hat{\mathbf{y}}$ for a given example with true output $\mathbf{y}$.

Let $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$ be the training set. The *n-slack* formulation of the structural SVM (with margin rescaling) is the problem

$$\begin{array}{lll} \text{minimize} & (1/2)\|\mathbf{w}\|_2^2 + (C/n)\sum_{i=1}^{n} \xi_i & \\ \text{subject to} & \mathbf{w}^T \delta\phi_i(\hat{\mathbf{y}}) \geq \Delta(\mathbf{y}_i, \hat{\mathbf{y}}) - \xi_i, & i = 1, \ldots, n, \ \hat{\mathbf{y}} \in \mathcal{Y} \\ & \xi_i \geq 0, & i = 1, \ldots, n, \end{array}$$

with variables $\mathbf{w}$ and ξ_i, where $\delta\phi_i(\mathbf{y}) = \phi(\mathbf{x}_i, \mathbf{y}_i) - \phi(\mathbf{x}_i, \mathbf{y})$. Here, we have one slack variable ξ_i for each training example. This problem can be reformulated using only a single slack variable ξ, yielding the 1-*slack* formulation

$$\begin{array}{ll} \text{minimize} & (1/2)\|\mathbf{w}\|_2^2 + C\xi \\ \text{subject to} & (1/n)\mathbf{w}^T \sum_{i=1}^{n} \delta\phi_i(\hat{\mathbf{y}}_i) \geq (1/n)\sum_{i=1}^{n} \Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i) - \xi, \quad \forall (\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_n) \in \mathcal{Y}^n \\ & \xi \geq 0, \end{array}$$

with variables $\mathbf{w}$ and ξ. Here, we only have a single slack variable ξ, but we have $|\mathcal{Y}|^n$ constraints, one for each combination of labels.

(a) Show that these two formulations are equivalent, in that any solution $\mathbf{w}^\star$ to the 1-slack formulation is also a solution to the n-slack formulation, and that $\xi^\star = (1/n)\sum_{i=1}^{n} \xi_i^\star$.

(b) The motivation for introducing the 1-slack formulation is that although it has $|\mathcal{Y}|^n$ constraints rather than $n|\mathcal{Y}|$, there is an efficient *cutting plane method* for solving it. Cutting plane methods form a class of optimization algorithms based on the use of *cutting planes*, which are hyperplanes that separate the current point from the optimal

points. Each iteration introduces a new cutting plane (corresponding to a particular constraint) that slices off more of the space that is infeasible. Cutting plane methods are often used in problems with with very large numbers of constraints, as we have here. It can be shown that using a cutting plane method to solve the 1-slack formulation here is much more efficient and scalable than the standard cutting plane method for the n-slack formulation discussed by Tsochantaridis et al.

Specifically, at each iteration, we introduce a *single* constraint by first solving

$$\text{maximize} \quad (1/n)\sum_{i=1}^n \Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i) - (1/n)\mathbf{w}^T \sum_{i=1}^n \delta\phi_i(\hat{\mathbf{y}}_i),$$

with variable $(\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_n) \in \mathcal{Y}^n$. The solution defines the most violated constraint. Show how to find the solution to this problem efficiently, *i.e.*, without simply searching over all $|\mathcal{Y}|^n$ possible joint assignments.

The overall algorithm involves solving the original problem above but only with the small subset of constraints introduced via this procedure (this subset of constraints is called the *working set*). It can be shown that the number of constraints that need to be introduced to converge to the solution is small and independent of the number of training examples.

6. *Latent structural SVMs* (11 points, 1 page). The standard structural SVM objective can equivalently be written without explicit use of slack variables, as in Yu and Joachims:

$$\text{minimize} \quad (1/2)\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \left(\max_{\hat{\mathbf{y}}}(\Delta(\mathbf{y}_i, \hat{\mathbf{y}}) + \mathbf{w}^T\phi(\mathbf{x}_i, \hat{\mathbf{y}})) - \mathbf{w}^T\phi(\mathbf{x}_i, \mathbf{y}_i)\right),$$

with variables $\mathbf{w}$. The objective of this problem is convex because it is the difference between a convex term and a linear term. Recall that the second term was shown to be a convex upper bound on the desired loss function $\Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i)$.

The latent structural SVM introduces latent variables $\mathbf{h}$, giving features $\phi(\mathbf{x}, \mathbf{y}, \mathbf{h})$ and the prediction rule

$$f_{\mathbf{w}}(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y},\mathbf{h}} \mathbf{w}^T\phi(\mathbf{x}, \mathbf{y}, \mathbf{h}).$$

The loss function Δ can also depend on the latent variables.

(a) The latent structural SVM objective is

$$(1/2)\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \left(\max_{\hat{\mathbf{y}},\hat{\mathbf{h}}} \left\{\mathbf{w}^T\phi(\mathbf{x}_i, \hat{\mathbf{y}}, \hat{\mathbf{h}}) + \Delta(\mathbf{y}_i, \hat{\mathbf{y}}, \hat{\mathbf{h}})\right\}\right) - C\sum_{i=1}^n \left(\max_{\mathbf{h}} \mathbf{w}^T\phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h})\right),$$

a difference of convex functions. Show that

$$\left(\max_{\hat{\mathbf{y}},\hat{\mathbf{h}}} \left\{\mathbf{w}^T\phi(\mathbf{x}_i, \hat{\mathbf{y}}, \hat{\mathbf{h}})) + \Delta(\mathbf{y}_i, \hat{\mathbf{y}}, \hat{\mathbf{h}})\right\}\right) - \left(\max_{\mathbf{h}} \mathbf{w}^T\phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h})\right)$$

is a valid upper bound on $\Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i(\mathbf{w}), \hat{\mathbf{h}}_i(\mathbf{w}))$.

(b) Algorithm 1 of Yu and Joachims describes a convex-concave procedure for optimizing the latent structural SVM objective, which is the difference of two convex functions. At each iteration, the algorithm requires finding a hyperplane v_t such that

$$-g(\mathbf{w}) \leq -g(\mathbf{w}_t) - v_t^T(\mathbf{w} - \mathbf{w}_t).$$

Using subdifferentials, characterize the set of vectors v_t that satisfy this inequality. Show that the particular choice

$$v_t = \sum_{i=1}^{n} \phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h}_i^*)$$

satisfies the desired inequality, where

$$\mathbf{h}_i^* = \operatorname*{argmax}_{\mathbf{h}} \mathbf{w}_t^T \phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{h})$$

for each i.

7. *First-order methods for ℓ_1 regularized loss minimization* (10 points, 2 pages). This question presents a state of the art algorithm for solving convex optimization problems involving ℓ_1 regularization. The use of ℓ_1 regularization as a heuristic to avoid exhaustive combinatorial search has become widespread because in a variety of problems (*e.g.*, structure learning, high-dimensional regression, signal processing), it is desirable to simultaneously fit the model and select a small subset of the variables of interest. This can help generalization performance and make the models more interpretable. For example, in some applications, there are a large number of candidate features, but only a small subset are expected to be predictive of a particular outcome. Moreover, it can be shown that the ℓ_1 norm is the tightest convex relaxation of the ℓ_0 pseudonorm (which requires combinatorial search to optimize), and that despite its being a heuristic, it (provably) happens to select the best subset of variables in a surprisingly wide range of situations.

Many modern algorithms for ℓ_1 regularized problems use *proximal operators* to effectively decouple the nonsmooth ℓ_1 term from the smooth unregularized objective. This allows for solving the problem without resorting to subgradient methods or other generic algorithms for nondifferentiable convex optimization, which are slow in general because they do not exploit the structure of the problem.

The *proximal operator* of a convex function h is

$$\mathbf{prox}_h(x) = \operatorname*{argmin}_z \left(h(z) + (1/2)\|z - x\|_2^2 \right).$$

The quadratic penalty term is called *proximal regularization*. When

$$h(x) = I_C(x) = \begin{cases} 0 & x \in C \\ +\infty & \text{otherwise,} \end{cases}$$

then $\mathbf{prox}_h$ is (Euclidean) projection onto the closed convex set C. In convex analysis, I_C is called the *indicator function* of the set C. If $h(x) = \lambda\|x\|_1$, then $\mathbf{prox}_h$ is given in closed

form by the *soft thresholding operator*

$$\mathbf{prox}_h(x)_i = S_\lambda(x)_i = \begin{cases} x_i - \lambda & x_i \geq \lambda \\ 0 & |x_i| \leq \lambda \\ x_i + \lambda & x_i \leq -\lambda. \end{cases}$$

This operator acts elementwise on x and can be computed very efficiently.

Note. All the functions in this problem are assumed to be extended-real-valued as needed for notational convenience; see §3.1.2 in Boyd and Vandenberghe for background.

(a) Consider the problem

$$\text{minimize} \quad f(x) = g(x) + h(x),$$

where g is convex and differentiable (with $\mathbf{dom}\, g = \mathbb{R}^n$) and h is convex and potentially nondifferentiable. Then the *proximal gradient algorithm* consists of the iteration

$$x^{k+1} := \mathbf{prox}_{\alpha^k h}\left(x^k - \alpha^k \nabla g(x^k)\right),$$

where $\alpha^k > 0$ is a step size and k is an iteration counter. Explain why this can be viewed as a generalization of the projected gradient algorithm. Briefly describe in English what this algorithm does $h(x) = \lambda\|x\|_1$.

(b) For the same problem, consider the algorithm

$$\begin{aligned} x^{k+1} &:= \mathbf{prox}_{\alpha^k h}\left(y^k - \alpha^k \nabla g(y^k)\right) \\ y^{k+1} &:= x^{k+1} + \frac{k-1}{k+2}\left(x^{k+1} - x^k\right). \end{aligned}$$

Here, y is an auxiliary variable. This algorithm is one of a class of methods known as *optimal first-order methods*, and is sometimes called FISTA (*fast iterative-shrinkage thresholding algorithm*) when $h(x) = \lambda\|x\|_1$. It can be shown that $f(x^k) - f^\star$, where $f^\star$ is the optimal value of the problem, decreases at least as fast as $O(1/k^2)$ if the step sizes are chosen appropriately. Moreover, it can be shown that this convergence rate is the *fastest possible* (in order) among the class of first-order methods, *i.e.*, methods that select x^{k+1} in

$$x^0 + \mathbf{span}\left\{\nabla f(x^0), \ldots, \nabla f(x^k)\right\}.$$

By comparison, gradient descent has a convergence rate of $O(1/k)$ (though in any case it cannot be used on ℓ_1 regularized problems).

Derive a FISTA-based method to fit an ℓ_1 regularized logistic regression model, *i.e.*, to carry out MAP estimation. Note that only the feature weights should be regularized, not the intercept term.

Note. Here, we use logistic regression since it can be viewed as the simplest example of a conditional random field.