CS 228T: A BIRD'S EYE VIEW

NEAL PARIKH

Abstract. This is a (rough draft of a) very high-level overview of some of the key ideas in graphical models, with a focus on topics covered in CS 228T. Many comments are left at an intuitive or casual level, technical conditions are omitted, and notation from the book is used without explanation. The bibliography contains many historical references and suggestions for further reading.

Discussions of graphical models are often divided into three major parts: *representation* (what they are), *inference* (what we do with them), and *learning* (how to pick which one to use). A fundamental characteristic of the subject is that these three facets are inextricably linked: efficient inference and learning algorithms rely on exploiting the model representation, representations are often chosen to ease inference and learning, and inference and learning themselves are intertwined. It is often worth keeping in mind the interactions among these three aspects when thinking about graphical models, rather than thinking about any one in isolation.

1. Representation

1. Modeling complex real world situations requires accounting for uncertainty, and probabilistic models are the natural way to do so.[1] This involves dealing with joint probability distributions over very large numbers of variables. These are not easy to work with either computationally (*e.g.*, computing singleton marginals requires huge multidimensional integrals) or statistically (estimating the exponential number of parameters in a full joint distribution would require massive amounts of data).
2. Graphical models provide a way around this by focusing on joint distributions that can be represented by graphs, ideally with a small number of edges. (Roughly speaking, the whole point is to erase most of the edges from the complete graph, which can represent any joint distribution over its nodes.) These allow for designing models over very large numbers of variables while managing computational cost and reducing the number of free parameters to be estimated.
3. The core idea is to assume that the *global* structure of the probability distribution is determined by composing *local* structures, each of which is much simpler to handle. Many of the benefits of graphical models come from the fact that the probabilistic structure and the graphical structure interact; for example, independence statements in probability can be translated into separation or reachability statements in graph theory.
4. There are two main classes of graphical models: directed models (Bayesian networks) and undirected models (Markov random fields). *Factor graphs* are also a convenient representation that can represent either directed or undirected models. Computationally, the major difference between directed and undirected models is that undirected models have a normalization constant

Date: June 1, 2011.

Prepared for CS 228T, Spring 2011, Stanford University.

[1]For some arguments for why one should use probability theory as opposed to something else, see [Pea88, chapter 1] or look up *Dutch book arguments*.

called the *partition function*.[2] This makes many tasks in undirected models harder, since the partition function[3] couples the local potentials and is often difficult to evaluate.

5. *Gaussians.* An important class of continuous graphical models is the class of Gaussian graphical models. It can be shown that multivariate Gaussians can be represented as Bayesian networks or as Markov random fields. There are several parametrizations of Gaussian distributions, including the usual form and the *information form* (in terms of the inverse covariance matrix Σ^{-1}), and the information form is often easier to work with in the context of graphical models. For example, the sparsity pattern of Σ^{-1} encodes the pairwise Markov properties for an MRF.

6. *Exponential models.* The most important class of graphical models consists of those that are members of the exponential family. In the undirected case, we usually work with the log-linear parametrization of MRFs directly (see below), which ensures they are exponential family models. In the directed case, these take the form of *conjugate-exponential models*. A prior distribution is *conjugate* to a likelihood function if the posterior is in the same family as the prior. All distributions in the exponential family have conjugate priors, so these models often involve picking an appropriate exponential family distribution for the data being modeled, then placing conjugate priors over the model parameters. For example, the Dirichlet is conjugate to the multinomial, so Dirichlet priors often appear in Bayesian models of discrete data. Conjugate-exponential models have many convenient properties. For example, the posterior can be computed by updating the prior parameters using sufficient statistics of the data, rather than by carrying out integrals explicitly. This can make inference in these models much simpler.

7. *Parametrization.*[4] In the general theory, we associate potential functions with the maximal cliques of the graph. By ranging over all possible potentials on the maximal cliques of a graph, we obtain all of the probability distributions that respect the Markov properties of the graph. In practical applications, large, fully-parametrized cliques are problematic both for computational reasons (inference is exponential in the clique sizes) and for statistical reasons (the estimation of large numbers of parameters requires large amounts of data). We usually prefer to work with reduced parametrizations that range over proper subsets of the set of all possible potential functions on maximal cliques. There are many ways to do this, *e.g.*, by building a potential function on a maximal clique from potentials on non-maximal cliques or by using features.

   (a) *Features.* Features let us assign parameters only to particular configurations of variables of interest (particular 'cells' in table factors), so the parametrization is significantly reduced. More generally, it is natural to consider 'features' as arbitrary functions on subsets of nodes. In the limiting case, if we use one binary feature for each cell in the table, we obtain a full parametrization of the potential function in which all cells in the table have an independently adjustable parameter.

   (b) *Exponential families.* There is a close relationship between exponential family models and graphical models. If we use an exponential representation for the contribution of each individual feature, then the product of potential functions leads to an exponential family representation for the joint distribution associated with the graphical model. (The features are the sufficient statistics.) Alternatively, we can also represent an arbitrary exponential family model as a graphical model by connecting nodes that appear together as arguments to the features. If we formulate the maximum entropy problem as finding the distribution with maximum entropy satisfying constraints on the expected values of a set of feature functions,

[2] Much of the terminology, particularly in the undirected case, comes from statistical physics and traces back to Gibbs [Gib76] and others.

[3] The partition function is denoted Z because of the German word *zustandssumme*, which means 'sum over states'.

[4] These comments are based on some unpublished notes by Michael Jordan.

this can be shown to be dual to maximum likelihood estimation in the exponential family with these features as sufficient statistics.

2. Inference

8. Given a graphical model (with fixed parameters), the term *inference* refers to computing marginal and conditional probabilities of interest from the full joint distribution. This involves summing or integrating over a number of variables, *e.g.*,

$$p(X_1) = \sum_{x_2} \sum_{x_3} \cdots \sum_{x_n} p(X_1, x_2, \ldots, x_n).$$

   The term *MAP inference* refers to computing modes over the whole graph or subsets of nodes.

9. *Exact inference.* Inference can be carried out exactly in any graphical model using the *clique tree algorithm*, also called the *junction tree algorithm*. This algorithm uses a special data structure called a clique tree to compute marginals over all cliques simultaneously; essentially, it is based on the idea of 'pushing sums in' as far as they will go, depending on the factorization of the full joint distribution. The time complexity of this algorithm is exponential in the tree-width of the graph, however, so this is only tractable for graphs with low tree-width (*e.g.*, trees have tree-width 1). This is another example where a probabilistic operation (marginalizing out variables) can be characterized using graph-theoretic properties of the model.

   (a) *Clique tree reparametrization.* A particularly important property of the clique tree algorithm is that it amounts to finding a reparametrization of the original joint distribution. In particular, the parametrization is in terms of *marginals* of cliques and sepsets. This is in contrast to the original factorization of the joint distribution, which may be in terms of conditional distributions (*e.g.*, in a Bayesian network). This parametrization plays a fundamental role in characterizing exact inference as an optimization problem and in variational inference and learning.

10. *Approximate inference.* When exact inference is intractable, we turn to *approximate inference* algorithms. There are two main categories: *sampling methods* and *variational methods*. Sampling algorithms rely on the use of Markov chain Monte Carlo methods for sampling from distributions that are difficult to sample from. Variational methods are based on optimization. Monte Carlo estimates are guaranteed to converge to the true value given enough samples from a fully mixed chain, but it can be difficult to diagnose mixing of Markov chains, and in very complex models even sampling methods can be slow. Variational methods are fast but produce approximate solutions, and it is difficult to quantify the quality of the approximations; on the other hand, some approximations are guaranteed to be an upper or lower bound on the quantity of interest.

11. *Markov chain Monte Carlo.* MCMC methods are based on constructing a Markov chain whose stationary distribution is the distribution we wish to sample from, and sampling from this chain. (Note that this means that the samples are not independent.) The most important MCMC algorithm is Metropolis-Hastings; many others are special cases. There are several important ideas that recur in the design of more advanced MCMC methods, such as *collapsing*, *auxiliary variables*, and *temperatures*. For further reading on MCMC methods, see [Nea93, Dia09, Dia11, LPW09, DSC98, BGJM10, KCGN98] and [Mac03, §29–§32].

   (a) *Metropolis-Hastings.* Simpler methods like importance sampling and rejection sampling rely on designing a proposal or sampling distribution $s(x)$ that is appropriately similar

to $\pi(x)$, the target distribution, yet easy to sample from. It can be difficult to find such distributions. Metropolis-Hastings instead uses a proposal $q(x \to x')$ that depends on the current state x, and it is not necessary for q to resemble π. The algorithm provides a constructive way to build a reversible chain that leaves π stationary; this is enforced by the Metropolis acceptance probability. The major tradeoff in designing proposals is achieving high acceptance rates while still making sufficiently large moves in the space. Many advanced methods are just different attempts to accomplish this in certain situations.

(b) *Gibbs sampling.* Gibbs sampling[5] is the MCMC equivalent of coordinate descent; it is a simple special case of Metropolis-Hastings that involves iteratively sampling from each variable conditioned on all other variables. In graphical models, each node often has a relatively small number of immediate dependencies, so this method can be easy to implement. It always accepts, but often mixes exceedingly slowly. The basic method can be improved by collapsing or *blocking*, in which groups of variables are iteratively resampled. Gibbs sampling illustrates the following general principle: Given multiple transition kernels that leave π stationary, it is possible to combine them in Metropolis-Hastings.

(c) *Collapsing.* In collapsing or *Rao-Blackwellization*, exact inference is carried out on some subset of the variable of interest, thus reducing the dimensionality of the space that the sampler needs to explore. In some cases, such as in conjugate-exponential models, this can simply involve (analytically) summing or integrating out some of the variables in the model; otherwise, it may involve running an exact inference algorithm. Collapsed Gibbs sampling is very widely used; *e.g.*, it is one standard way to perform inference in topic models and other conjugate-exponential Bayesian models.

(d) *Auxiliary variables.* Auxiliary variable methods pursue essentially the opposite idea as collapsed samplers: here, extra variables are introduced in order to make sampling easier, and in particular, to make it easier to take large steps in the state space. Two important auxiliary variable methods are Swendsen-Wang (for the Ising model) and slice sampling (for sampling from continuous distributions).

(e) *Temperatures.* Another important idea is that of temperature; this is used in methods like *annealed importance sampling*, a gold standard method in many situations. The basic idea is to interpolate between two extremes: a target distribution that is hard to sample from and an inaccurate distribution that is easy to sample from. This is sometimes called parallel tempering or replica exchange.[6]

12. *Variational inference.* Variational methods are based on finding a way to pose a given task as solving a particular optimization problem. Approximate solutions to the task can then be obtained by *relaxing* this optimization problem by simplifying the objective or the constraints.[7]

(a) *Exact inference as optimization.* Inference can be posed as the convex optimization problem

$$\begin{array}{ll} \text{minimize} & D(q \,\|\, p) \\ \text{subject to} & q \in \mathbb{M}(G), \end{array}$$

[5] Gibbs sampling is also known as *Glauber dynamics* or the *heat-bath algorithm* in physics.

[6] At a high level, the idea of interpolating between easy and hard problems appears in many places, such as in *homotopy methods* for optimization.

[7] The historical roots of variational methods lie in the calculus of variations, which accounts for the name. The particular class of variational methods used in graphical models come from statistical physics, and the ideas trace back to Feynman [Fey72] and others. The link to statistical physics was shown in [YFW01, YFW03].

where $\mathbb{M}(G)$ is the *marginal polytope*, a set containing all collections of marginal distributions over subsets of variables that can arise from a single valid joint distribution.

(i) Geometrically, the problem can be viewed as performing a particular nonlinear projection (called an *I-projection*) of p onto the marginal polytope. A major reason we minimize $D(q \,\|\, p)$ rather than $D(p \,\|\, q)$ (which would be *M-projection*) is because the former does not require carrying out inference in p.

(ii) Equivalently, the problem can be rewritten as maximizing an *energy functional* $F[\tilde{p}, q] = \mathbb{E}_q[\log \tilde{p}] + H(q)$ over the marginal polytope, where $p = \tilde{p}/Z$.

(iii) The constraints required to characterize the marginal polytope cannot be efficiently enumerated for graphs other than trees [DL09, KP82], so it is necessary to consider approximations to this problem, despite the fact that it is convex.

(iv) The classic belief propagation algorithm can be derived from a variational perspective as a particular fixed point method for solving this optimization problem (via the Lagrangian) when G is a tree.

(b) *Bethe approximation.* The Bethe approximation is a *tree-based* approximation to the exact inference problem, in the sense that the approximation is exact for trees.

(i) The entropy term $H(q)$ in the objective is replaced with a factored term $H_{\text{Bethe}}(q)$, yielding the factored energy functional, and the marginal polytope is replaced with the *local consistency polytope* $\mathbb{L}(G)$, a superset of the marginal polytope produced by only enumerating the local consistency constraints. When G is a tree, $\mathbb{M}(G) = \mathbb{L}(G)$ and $H_{\text{Bethe}}(q) = H(q)$, so the exact problem for trees involves maximizing the factored energy over the local polytope, which is tractable.

(ii) This problem is nonconvex because the objective is nonconcave in general. In particular, this means that any method used to solve it must settle for local optima.

(iii) Loopy belief propagation can be viewed as a fixed point method for finding local optima. Loopy BP can have trouble converging (even to local optima), but the Bethe approximation can also be solved using other algorithms, like the convex-concave procedure. However, loopy BP can also be very effective in certain classes of models, such as in the models used in modern coding theory.

(iv) It is possible to tighten this approximation in a number of ways. One way is to use approximations to the entropy term and the marginal polytope that are exact for graphs with tree-width higher than 1. A related idea is to enumerate 'higher-order' constraints that hold for any member of the marginal polytope but that can still be enumerated efficiently; the cycle inequalities are an example. However, these methods are not widely used in practice. See [WJ08, §4.2] and [DL09] for more details.

(c) *Mean field.* The mean field method approximates only the constraint set, and considers the subset of the marginal polytope consisting of fully factored distributions.

(i) More generally, we can consider subsets of the marginal polytope corresponding to simple graphical models in which exact inference is tractable. This is sometimes called *structured mean field*. Very roughly, the idea is that in some graphical models, there are still 'too many edges' to carry out exact inference efficiently, so we erase the complicating edges until the problem becomes simple enough; standard or 'naive' mean field erases all the edges.

(ii) Having said this, note that mean field is *not* equivalent to simply using a model without these extra edges in the first place. First, the true distribution p and the approximating distribution q interact when carrying out mean field inference, and only q is missing the complicating edges. Also, this only refers to carrying out inference, while we have still (somehow) fit the model parameters with the complicating edges.

(iii) *Solution quality and nonconvexity.* Mean field methods are fast and scale well, but it is difficult to make precise statements about the quality of the approximate solutions obtained (compared to MCMC methods), and only local optima can be obtained due to the nonconvexity of the problem.

(iv) *Lower bound property.* Perhaps the most important property of mean field approximations is that they provide a lower bound to the log partition function. This makes mean field the *only* natural choice in certain cases when an approximate inference algorithm is required as a *subroutine* in a larger procedure (*e.g.*, variational EM).

(d) *Other approximations.* There are other classes of variational methods based on other approximations to the exact inference problem. The two main classes of approximations considered here are nonconvex, but there are also convex approximations. For further information, see, for example, [WJ08, §4.3, §7, §9].

13. *MAP inference.* There are several major categories of MAP inference algorithms: the *max-product* or *Viterbi* algorithm; *move-making algorithms*; and methods based on duality.

(a) *Max-product.* The max-product belief propagation algorithm is the exact analogue of the standard sum-product algorithm, except that the summations are replaced with maximizations. As in standard inference, it is only used for models with low tree-width. There is also a loopy version for general cluster graphs.

(b) *Move-making.* The idea behind move-making is to only consider certain classes of 'moves' from the current labeling (much like in sampling methods), and then ensure that the algorithm is guaranteed to find local optima with respect to this restricted class of moves. Roughly, these methods are fast and work regardless of the tree-width of the graph, but require the MRF to be metric.

(i) *Binary variables.* MAP inference can be performed *exactly* in pairwise binary MRFs with submodular potentials, regardless of the structural complexity (*e.g.*, tree-width) of the underlying graph. The method is based on a constructing an auxiliary graph such that finding a minimal cut in this auxiliary graph corresponds to the MAP assignment in the original model.

(ii) *Nonbinary variables.* In this case, computing the exact MAP assignment is NP-Hard. The *α-expansion* and *(α, β)-swap* methods are based on taking greedy hill-climbing steps, where each step involves a globally optimal solution to a simplified problem.[8] When the original graph is a metric MRF, then the α-expansion and (α, β)-swap steps can be carried out optimally by running a min-cut algorithm on an appropriately constructed MRF. The α-expansion algorithm, for instance, allows each node to retain its label or switch to a fixed label α at each iteration. It is possible to show that the local optimum obtained by α-expansion is within a known factor of the global optimum; however, these bounds may be too loose in practice to be of much use.

[8] At a high level, this idea reappears in many places, including Newton's method, the EM algorithm, and the convex-concave procedure.

(c) *Dual decomposition.* Dual decomposition for MAP inference is an algorithm based on combining several ideas, each one of which is individually simple. At a high level, it involves solving the (convex) dual of the integer LP formulation of the MAP inference problem, which is guaranteed to provide a lower bound on the primal optimal value. We solve the dual by solving (combinatorial) MAP inference problems over tractable subsets of the original graph and then coordinating these to agree where they overlap.

(i) *Integer LP formulation.* In a discrete graphical model, the MAP inference task is evidently a combinatorial optimization problem, since it involves searching over the discrete (joint) label space. This problem can be expressed as an integer linear program, which is an LP where the variables are constrained to be in a discrete set (thus rendering the problem nonconvex).

(ii) *Tractable substructures.* First, we identify substructures of the model in which exact inference is tractable. This could include trees (*e.g.*, rows and columns of a grid), other graphs with low tree-width, or submodular components amenable to graph cuts.

(iii) *Consensus transformation.* We duplicate variables that are shared across multiple subproblems and then constrain them to agree. This evidently yields a problem that is equivalent to the original problem, but the effect of this is that in the dual problem, the subproblems will completely decouple and can be solved independently.

(iv) *Dual problem.* Recall that the dual of an optimization problem is convex, regardless of the convexity of the original problem, and the dual optimal value is a lower bound on the primal optimal value. Taking the dual of the consensus version of the MAP problem yields a convex problem that is separable across the tractable substructures. Note that we only relax the consistency constraints when forming the dual, so the integer constraints in the primal are preserved in the slave problems in the dual.

(v) *Dual subgradient method.* The dual function is concave but usually nondifferentiable, so it is necessary to use a subgradient method or equivalent to maximize it. The resulting algorithm can be viewed as a (continuous) master problem coordinating the behavior of a set of (combinatorial but tractable) slaves. Each master update will 'reprice' the cost of disagreement across slaves that do not agree where they overlap.

3. Learning

14. The term *learning* refers to the task of selecting which graphical model among a particular set of options is the best fit to a particular dataset. This could involve selecting a particular member of a parametric family of models (*i.e.*, *parameter learning*), or more generally, choosing from a set of different families (*i.e.*, *structure learning* or *model selection*). In structure learning, we want to learn the graph structure of the model or determine the type or dimensionality of latent variables (often in addition to learning the parameters). Roughly, model selection refers to determining something that changes the number or characteristics of the parameters in the model (*e.g.*, if there are more components in a mixture model, the model has more parameters). A major theme in graphical models is the deep link between inference and learning; the Bayesian approach goes so far as to unify the two.

15. The learning problem can be further classified in a number of ways: whether the model is directed or undirected, whether the model is generative or discriminative, whether the data is complete or incomplete, and how the learning task is defined (*e.g.*, maximum likelihood estimation, Bayesian learning, and so on). When selecting a training approach, it is important to think about several criteria, such as computation (*e.g.*, tractability, scalability, need for approximations); statistical

properties (*e.g.*, asymptotic consistency, overfitting, bias-variance tradeoff); and applicability (what task the model will ultimately be used for). We select a particular linear ordering of some of these topics, but really these methods should be organized across multiple axes, so do not read too much into the ordering in this section.

16. *Maximum likelihood estimation.* In maximum likelihood estimation, we define a parametric model $p(\mathbf{x} \mid \theta)$, parametrized by θ, and then maximize the (log) likelihood function

$$\ell(\theta) = \sum_{i=1}^{m} \log p(\mathbf{x}_i \mid \theta)$$

with variable θ, where $\{\mathbf{x}_i\}_{i=1}^m$ is the dataset. Maximum likelihood estimation can be viewed as computing the M-projection of the empirical distribution onto the model family.

    (a) *Duality.* Maximum likelihood is a convex dual of the maximum entropy problem. (This was touched on earlier, in the discussion of features and exponential families.)

    (b) *Regularization.* Standard maximum likelihood estimation will overfit the training data (unless a huge amount of data is available relative to the number of parameters being estimated). For example, the model will assign probability zero to any outcomes that happened not to appear in the training set. In practice, one would include some form of regularization on the parameters, or equivalently, place a prior on the parameters and carry out MAP estimation. Note that the term 'maximum likelihood' is often used even when regularization is involved.

17. *Directed models.* The fundamental characteristic of directed models is that the individual factors are normalized conditional distributions, and thus that the global factorization is fully normalized (at least when no evidence is present). The effect of this property is that many tasks decompose so each local distribution can be handled separately.

    (a) *Fully observed models.* In this case, the log likelihood decomposes as a sum of independent terms, one for each CPD in the network (assuming parameters are not shared across CPDs). This means each CPD's parameters can be estimated independently; in the discrete case, the estimate for each CPD parameter has a simple closed form solution ($\hat{\theta}_{x|u} = M[u, x]/M[x]$). Similar results hold for exponential family CPDs. (As usual, unregularized maximum likelihood will overfit, so usually one places a prior on the parameters.)

    (b) *Latent variable models.* The EM algorithm is the core method for performing maximum likelihood estimation in (usually directed) models with hidden variables. Introducing hidden variables makes the learning problem nonconvex, so we have to settle for local optima. The E-step involves computing the posterior of the hidden variables, and the M-step involves carrying out more or less standard maximum likelihood estimation using this 'soft completion' of the hidden variables in all the training instances. In this sense, it consists of alternating inference and learning. A useful property of EM is that we fit the parameters and obtain the posterior over the latent variables simultaneously.

        (i) *Free energy.* EM can be viewed from a variational perspective as maximizing (via coordinate ascent) a particular energy functional $F[\theta, q] = \mathbb{E}_q[\log \tilde{p}] + H(q)$. In particular, the E-step maximizes over q (the optimum is at $p(\mathbf{z} \mid \mathbf{x})$, so this reduces to inference) and the M-step corresponds to learning in a fully observed model. The major benefit of this perspective is that it provides a rigorous framework in which to extend or modify the standard EM algorithm in various ways.

(ii) *Variational EM.* In some models, the E-step is intractable to compute. This is because the normalization constant of the posterior $p(\mathbf{z} \mid \mathbf{x}) = p(\mathbf{z}, \mathbf{x})/p(\mathbf{x})$ is the *marginal likelihood* $p(\mathbf{x})$, which is frequently intractable to compute. A natural solution is to use variational inference for the E-step instead, and in particular, to use a mean field approximation $q(\mathbf{z})$ to $p(\mathbf{z} \mid \mathbf{x})$. Using a mean field approximation and then maximizing the same free energy F in the same fashion guarantees that we are still maximizing a lower bound on $p(\mathbf{x})$, simply by the lower bound property of mean field. Explicitly, in the free energy above, $\tilde{p}$ is $p(\mathbf{z}, \mathbf{x})$ and q is $q(\mathbf{z})$, and both of these factorize (p by the model definition and q by the mean field assumption).

18. *Undirected models.* In contrast with directed models, the fundamental characteristic of undirected models is that the local factors are arbitrary nonnegative functions, and so the partition function is needed to ensure the global distribution is normalized. All the local structures are then coupled through the partition function, and many of the decomposition properties of directed models fail to hold. This makes many tasks more difficult in undirected models.

    (a) *Maximum likelihood.* The lack of decomposability means that the parameter estimation problem does not have a closed form solution, but needs to be solved via an algorithm like gradient descent (or something better). Each step of the optimization algorithm requires running inference on the network, which can make this a difficult process. A significant amount of work has thus gone into finding alternate objectives that are easier to optimize, or into using approximate inference.

    (b) *Alternate objectives.* Because maximum likelihood estimation in undirected models is difficult (due to having to do inference in the inner loop), there are a number of alternate objectives for the learning problem that are easier to optimize (*e.g.*, because they somehow avoid dealing with the partition function). These include pseudolikelihood, contrastive divergence, and others. We focus on large margin methods here.

        (i) *Large margin.* In many situations, the goal of fitting a model is to use it to predict particular outcomes (*i.e.*, once we have the parameters we intend to carry out MAP inference to compute, say, some kind of optimal labeling of the data). The main idea behind large margin methods is to train the model in a way that is explicit about MAP inference (*i.e.*, structured prediction) being the main use case.

        (ii) *Structural support vector machines.* Structural support vector machines are analogues of standard support vector machines in which the goal is structured prediction rather than binary classification. This can also be viewed as a particular training method for conditional random fields $p(\mathbf{y} \mid \mathbf{x})$ (again, in the case where the goal is to use the CRF for prediction). It is also possible to include latent variables, though this makes learning more difficult.

        (iii) *Learning structured prediction models.* The setup of the problem means that we need to distinguish not between two outcomes, but exponentially many. The effect of this is that the optimization problem that needs to be solved is convex, but has exponentially many constraints. The models are thus trained using *cutting plane methods* that introduce constraints one at a time; these methods can be shown to converge before too many constraints are added. These cutting plane methods use MAP inference as a subroutine. If latent variables are included, the problem becomes a *difference of convex* optimization problem amenable to the convex-concave procedure.

19. *Bayesian learning.* The core idea of Bayesian learning is to reduce learning to the inference problem by treating parameters as random variables. Explicitly, we make a large joint model

$p(\mathbf{x}, \mathbf{z}, \theta)$ involving observed variables $\mathbf{x}$, latent variables $\mathbf{z}$, and the parameters θ, and then pose the learning problem as computing the posterior distribution $p(\theta \mid \mathbf{x})$ of the parameters given the data. This approach is much more resistant to overfitting. Another defining characteristic of Bayesian methodology is to maintain full distributions over unknown quantities rather than resorting to point estimators, and then integrating over these 'nuisance' parameters when making any predictions. For example, predicting future outcomes is done via the *predictive distribution* $p(x^{\text{new}} \mid \mathbf{x}) = \int p(x^{\text{new}} \mid \theta) p(\theta \mid \mathbf{x}) \, d\theta$. For more on the Bayesian approach to statistics, and some of the philosophical distinctions between the Bayesian and frequentist perspectives, see, *e.g.*, [Fre95, Ber02, GCSR04, Efr05].

(a) *Priors.* A major philosophical and practical issue in the use of Bayesian methods is where priors come from. The main approach considered in class was to use conjugate priors, whose major benefit is computational convenience. This is often the most pressing issue given the complexity of the models used in machine learning. However, there are many other approaches; for more on this, see the references above.

(b) *Variational Bayes.* Computing the posterior is often difficult (due to the marginal likelihood being the normalization constant), so we can again turn to variational inference algorithms to make this easier. In particular, using mean field inference to approximate the posterior guarantees a lower bound on the marginal likelihood, simply because of the lower bound property of mean field. When we use a mean field approximation that decouples the parameters and latent variables, this method is referred to as *variational Bayes*. Variational Bayesian learning algorithms often look similar to EM or variational EM algorithms. Variational EM and variational Bayesian algorithms are often used in complex latent variable models like topic models.

20. *Structure learning.* Structure learning can refer to a variety of tasks: learning the dimensionality of latent variables, learning the graph topology over a known set of variables, selecting from a set of candidate models, and so on. There are several approaches to structure learning (such as constraint-based learning and Bayesian model averaging), but we focus on score-based methods here for brevity. (Roughly, the Bayesian approach is mostly a natural extension of standard Bayesian models, except that the model structure is also thrown into the big joint model, and then we integrate over all the different models when making predictions.)

(a) *Score-based learning.* Score-based methods address learning as a model selection problem. We define a hypothesis space of potential models and a scoring function that measures how well the model fits the observed data. The problem is then to find the highest-scoring network structure (via search or some other procedure).

(b) *Marginal likelihood.* The *marginal likelihood* of the data is

$$p(\mathbf{x} \mid \mathcal{G}) = \int_{\Theta} p(\mathbf{x} \mid \theta_{\mathcal{G}}, \mathcal{G}) p(\theta_{\mathcal{G}} \mid \mathcal{G}) \, d\theta_{\mathcal{G}},$$

where $\mathcal{G}$ is the particular model or structure being scored. Note that this is fundamentally different from the likelihood score because we integrate over the parameters. If a prior over structures is also included, this gives the *Bayesian score*, but the prior tends to have little effect on the overall score, so the marginal likelihood is the key quantity. In fully observed Bayesian networks, the marginal likelihood can be computed efficiently, but this involves a difficult inference problem in partially observed Bayesian networks, and it is difficult to evaluate even with approximate inference in undirected models.

(c) *Cheeseman-Stutz approximation.* The Cheeseman-Stutz score is one approximation to the Bayesian score that is efficient to compute and also happens to provide a lower bound on

the marginal likelihood. This score is useful for Bayesian networks with hidden variables. We could also directly apply a variational Bayesian approach, which would also guarantee a lower bound as a consequence of the mean field lower bound; it can be shown that the variational bound is at least as tight as the Cheeseman-Stutz bound.

(d) ℓ_1 *regularization.*[9] In the undirected case, the marginal likelihood is difficult to deal with, even when using approximations. An important alternative is to use the ℓ_1 regularized likelihood score (*i.e.*, a MAP score); the use of regularization avoids the overfitting properties of the likelihood score, and the sparsity-inducing properties of ℓ_1 regularization implicitly perform model selection by effectively erasing a number of the edges. A useful property of the ℓ_1 regularized likelihood is that it is convex, which confers a number of advantages. However, we cannot simply solve this problem using an off-the-shelf optimization algorithm: Beginning with a fully connected model and then attempting to sparsify would result in an intractably large problem, so this problem is solved with a specialized algorithm.

21. *Bayesian nonparametrics.* The core idea in Bayesian nonparametrics is to move beyond (parametric) exponential family representations. The parametric distributions used as priors in classical Bayesian analysis are replaced with stochastic processes. Combining such a prior with a likelihood yields a posterior distribution that is also a stochastic process. Bayesian learning in this setting involves updating the prior stochastic process into the posterior process. There are several uses for models based on these ideas. (Though the use of Bayesian nonparametric models is a representational issue, many of the motivations for and challenges in using them involve inference and learning, so it is included in this section.) There are a number of uses of Bayesian nonparametric models and stochastic process priors we did not cover; for more on some of these, see [BGJ10] and the references in [Ble07].

(a) *Model selection.* Nonparametric models provide a different approach to model selection. In particular, there is no need to, say, fix the dimensionality of latent variables (*e.g.*, the number of clusters in a cluster model) in advance, and these values can be automatically learned from the data as part of the standard learning procedure. This said, the mathematical structure of the models does enforce certain implicit assumptions on the values of these quantities (*e.g.*, the number of clusters in a Dirichlet process mixture model is logarithmic in the number of examples).

(b) *Dirichlet processes.* The Dirichlet process is perhaps the most central stochastic process used in Bayesian nonparametric models. (Another important one is the Gaussian process.) The Dirichlet process is a measure over discrete distributions over a potentially unbounded number of outcomes. There are different perspectives on the Dirichlet process that lead to other closely related distributions, like the Chinese restaurant process or Pólya urn.

(c) *Exchangeability.* Exchangeability is a property that is particularly important in the context of Bayesian nonparametrics. A joint distribution is exchangeable if it is invariant under permuting the random variables; *i.e.*, the order of the data doesn't matter. (This is exactly like the 'bag of words' assumption in natural language processing.) There are two major implications of this property: first, the de Finetti theorem applies, and second, it significantly simplifies the inference algorithms for these models.

[9] More broadly, ℓ_1 regularization now plays a major role in high-dimensional statistics, machine learning, and signal processing; see, *e.g.*, [CP09, BDE09] for some more on this.

References

[BDE09] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[Bea03] M. J. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, 2003.

[Ber02] J. M. Bernardo. Bayesian statistics. *Encyclopedia of Life Support Systems*, 2002.

[BGJ10] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):7, 2010.

[BGJM10] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall / CRC Press, 2010.

[BHS+07] G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan, editors. *Predicting Structured Data*. The MIT Press, 2007.

[Ble07] D. M. Blei. Bayesian nonparametrics: course syllabus. Available at `www.cs.princeton.edu/courses/archive/fall07/cos597C/syllabus.html`, 2007.

[Bro86] L. D. Brown. *Fundamentals of Statistical Exponential Families*. Institute of Mathematical Statistics, 1986.

[CP09] E. J. Candès and Y. Plan. Near-ideal model selection by ℓ_1 minimization. *Annals of Statistics*, 37(5A):2145–2177, 2009.

[Dia09] P. Diaconis. The Markov chain Monte Carlo revolution. *Bulletin of the American Mathematical Society*, 46(2):179–205, 2009.

[Dia11] P. Diaconis. The mathematics of mixing things up. *Journal of Statistical Physics*, 2011.

[DL09] M. M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Springer-Verlag, 2009.

[Don00] D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Mathematical Challenges of the 21st Century*, pages 1–32, 2000.

[DSC98] P. Diaconis and L. Saloff-Coste. What do we know about the Metropolis algorithm? *Journal of Computer and System Sciences*, 57(1):20–36, 1998.

[Efr75] B. Efron. Defining the curvature of a statistical problem (with applications to second order efficiency). *Annals of Statistics*, 3(6):1189–1242, 1975.

[Efr78] B. Efron. The geometry of exponential families. *Annals of Statistics*, 6(2):362–376, 1978.

[Efr05] B. Efron. Modern science and the Bayesian-frequentist controversy. Technical report, Stanford University, 2005.

[Fey72] R. Feynman. *Lectures on Statistical Mechanics*. Addison-Wesley, 1972.

[Fre95] D. Freedman. Some issues in the foundation of statistics. *Foundations of Science*, 1(1):19–39, 1995.

[GCSR04] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. CRC Press, 2004.

[Gib76] J. W. Gibbs. On the equilibrium of heterogeneous substances. *Transactions of the Connecticut Academy of Arts and Sciences*, 1876.

[Jor98] M. I. Jordan, editor. *Learning in Graphical Models*. Kluwer Academic Publishers, 1998.

[KCGN98] R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal. Markov chain Monte Carlo in practice: A roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

[KF09] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[KP82] R. M. Karp and C. H. Papadimitriou. On linear characterizations of combinatorial optimization problems. In *Foundations of Computer Science*, pages 1–9, 1982.

[LPW09] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[Mac03] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[Nea93] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[SGJ10] D. Sontag, A. Globerson, and T. Jaakkola. Introduction to dual decomposition for inference. In *Optimization for Machine Learning*. MIT Press, 2010.

[WJ08] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[YFW01] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. *Advances in Neural Information Processing Systems*, pages 689–695, 2001.

[YFW03] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring Artificial Intelligence in the New Millennium*, 8:236–239, 2003.