Course: Model, Learning, and Inference: Lecture 4

Alan Yuille
Department of Statistics, UCLA
Los Angeles, CA 90095
yuille@stat.ucla.edu

Abstract

Steepest Descent. Discrete Iterative Optimization. Markov Chain Monte Carlo (MCMC).
NOTE: NOT FOR DISTRIBUTION!!

1 Introduction

These notes discuss inference algorithms that we can use for exponential models (without hidden variables). We cover discrete iterative updating and stochastic sampling (did not have time to cover belief propagation, simulated/deterministic annealing.Other algorithms max-flow/min-cut, dynamic programming will be covered in later lectures.

2 Discrete Iterative Optimization

To give context, we start by very briefly describing "continuous" methods – such as steepest/gradient descent, Newton-Raphson – for minimizing energy functions. There are standard textbooks on these methods. The basic idea of steepest descent is to define the update equation:

$$\frac{d\vec{x}}{dt} = -\frac{\partial f}{\partial \vec{x}} = -\vec{\nabla} f(\vec{x}), \tag{1}$$

where $f(\vec{x})$ is the function you want to minimize. This is guaranteed to continually decrease the value of the function because

$$\frac{df}{dt} = \frac{\partial f}{\partial \vec{x}}\frac{d\vec{x}}{dt} = -|\vec{\nabla} f(\vec{x})|^2. \tag{2}$$

COMMENT ABOUT LYAPONOV FUNCTIONS!! SUBSECTION ON LEVEL SETS!!

A problem with continuous methods is that they require discretizing t and replacing equation (1) by a discrete update rule:

$$\vec{x}_{t+1} = \vec{x}_t - \Delta \vec{\nabla} f(\vec{x}(t)), \tag{3}$$

where the choice of parameter Δ determines how well this approximates the continuous time version in equation (1) There is a tradeoff between how fast the algorithm is and how good is the approximation. Small Δ gives good approximation but slow speed – large Δ increases speed but may make the approximation so poor that the algorithm does not converge. Some possibilities are to determine Δ adaptively or to regularize the method. (WHAT IS A GOOD REFERENCE??).

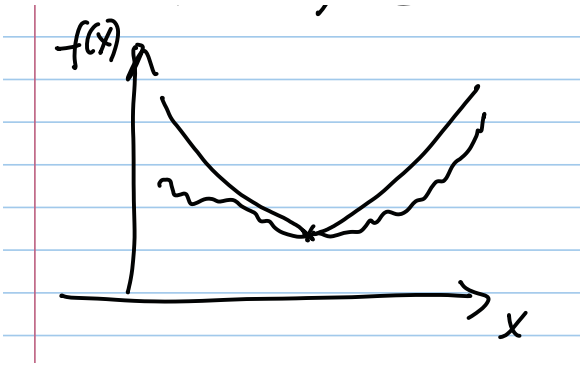

Figure 1: Put a variational bound on the function you want to minimize. Then minimize this bound. Then replace with a new bound and repeat.

Discrete iterative methods are guaranteed to converge (to local minima) and do not require choosing a step size. They are less well-known in general, but occur frequently in vision and machine learning, so we describe them in more detail here. Fortunately, and surprisingly, there is a general principle that underlies almost all discrete iterative methods (despite many of them being developed independently). This principle is called majorization or variational bounding (depending on whether you read the machine learning or the statistics literature). It includes, as a special cases, a method called CCCP developed by the author who proved that many existing algorithms – EM, GIS, Legendre transform optimization, discrete neural networks, Sinkhorn's method – can all be derived as examples of CCCP and hence of majorization/variational bounding.

3 Discrete Iterative Algorithms

Suppose we want to minimize $E(\vec{x})$. Majorization/variational bounding proceeds by obtaining a sequence of bounding functions $E_B(\vec{x}, \vec{x}_n)$ where $\vec{x}_n$ is the current state, see figure (1). The bounding functions must obey:

$$\begin{aligned} E_B(\vec{x}, \vec{x}_n) &\geq E(\vec{x}), \forall \vec{x}, \vec{x}_n \\ E_B(\vec{x}_n, \vec{x}_n) &= E(\vec{x}_n). \end{aligned} \tag{4}$$

Then the algorithm $\vec{x}_{n+1} = \arg\min_{\vec{x}} E_B(\vec{x}_n)$ is guaranteed to converge to a minimum of $E(\vec{x})$. The algorithm can make large moves from $\vec{x}_n$ to $\vec{x}_{n+1}$.

A special case of this approach is called CCCP (Yuille – it can be shown that most examples of variational bounding are equivalent to CCCP). Decompose the function $E(\vec{x})$ into a concave $E_c(\vec{x})$ and a convex part $E_v(\vec{x})$ so that: $E(\vec{x}) = E_c(\vec{x}) + E_v(\vec{x})$. Then define the update rule:

$$\vec{\nabla} E_v(\vec{x}_{n+1}) = -\vec{\nabla} E_c(\vec{x}_n). \tag{5}$$

By properties of convex and concave functions, this CCCP update rule is guaranteed to decrease the energy at each time-step.

It is easy to check (FIGURE) that the CCCP is a special case of majorization/variational bounding. Because it corresponds to the bound $E_B(\vec{x}, \vec{x}_n) = E_v(\vec{x}) + E_c(\vec{x}_n) + (\vec{x} - \vec{x}_n) \cdot \vec{\nabla} E_c(\vec{x}_n)$.

It can be shown that several well-known algorithms used in this course – EM, GIS, Sinkhorn – can be obtained as special cases of CCCP and hence of majorization/variational bounding (Yuille and Rangarajan – though the proofs requires changes of variables).

Here is a simple example $E(x) = x\log x + (1-x)\log(1-x) - (1/2)x^2$ for $0 \leq x \leq 1$. We can decompose $E(x)$ into a convex function $E_v(x) = x\log x + (1-x)\log(1-x)$ and a concave function $E_c(x) = -(1/2)x^2$. This decomposition gives a discrete update rule:

$$x_{n+1} = \frac{1}{1+\exp\{-x_n\}}. \tag{6}$$

3.1 Relations of Discrete Iterative to CCCP

A popular method for stabilizing gradient/steepest descent methods can be explained as an example of this discrete iterative framework.

Eyre studies gradient flow $d\vec{x}/dt = -\vec{\nabla}E(\vec{x})$ and proposes to decompose $E(\vec{x}) = E_1(\vec{x}) - E_2(\vec{x})$ where $E_1(.)$ and $E_2(.)$ are convex functions (i.e., $-E_2(\vec{x})$ is a concave function – so this is like a CCCP decomposition). His update rule is:

$$\vec{x}_{t+1} - \vec{x}_t = \Delta\{\vec{\nabla}E_2(\vec{x}_t) - \vec{\nabla}E_1(\vec{x}_{t+1})\}. \tag{7}$$

This is just CCCP with the decomposition $E_v(\vec{x}) = E_1(\vec{x}) + (1/2\Delta)\vec{x}\cdot\vec{x}$ and $E_c(\vec{x}) = -E_2(\vec{x}) - (1/2\Delta)\vec{x}\cdot\vec{x}$. Observe that Eyre's method looks like a small modification of the basic gradient/steepest descent $\vec{x}_{t+1} - \vec{x}_t = -\Delta\{\vec{\nabla}E(\vec{x}_t)$.

Legendre Transforms

The Legendre transform can be used to reformulate optimization problems in terms by introducing auxiliary variables. Let $E(\vec{x})$ be a convex function, define its Legendre transform to be $E^*(\vec{y}) = \min_{\vec{x}}\{E(\vec{x}) + \vec{x}\cdot\vec{y}\}$. It can be verified that $E^*(\vec{y})$ is concave. We recover $E(\vec{x})$ by the inverse transform: $E(\vec{x}) = \max_{\vec{y}}\{E^*(\vec{y}) - \vec{y}\cdot\vec{x}\}$.

Now suppose an energy function is expressed in CCCP form as $E(\vec{x}) = E_v(\vec{x}) + E_c(\vec{x})$ where $E_v(.)$ is convex and $E_c(.)$ concave. Minimizing $E(\vec{x})$ wrt $\vec{x}$ is equivalent to minimizing the function $E(\vec{x},\vec{y}) = E_v(\vec{x}) + \vec{x}\cdot\vec{y} + E_c^{-1*}(\vec{y})$ wrt $\vec{x},\vec{y}$, where E_c^{-1*} is the inverse Legendre transform of $E_c(.)$. Minimizing $E(\vec{x},\vec{y})$ by coordinate descent (i.e. wrt $\vec{x}$ and $\vec{y}$ alternatively) is equivalent to performing CCCP on $E(.)$.

4 Continous Formulations of the Weak Membrane Model

Mumford and Shah formulated image segmentation of a domain D as the minimization of a functional (called this because it depends on functions $J(\vec{x})$:

$$E[J,B] = C\int d\vec{x}(I(\vec{x}) - J(\vec{x}))^2 + A\int_{D/B}\vec{\nabla}J(\vec{x})\cdot\vec{\nabla}J(\vec{x})d\vec{x} + B\int_B ds, \tag{8}$$

where B is a boundary that separates D into subdomains $D = \bigcup D_i$, with $D_i \bigcap D_j = 0\ i \neq j$ and $B = \bigcup \partial D_i$.

This is directly analogous to the weak membrane models in previous lectures but there are some important differences: (i) the formulation is continuous in $\vec{x}$, (ii) the solution separates the images into disjoint sets $\{D_i\}$ (but the previous weak membrane model does not – see earlier lectures). More technically, there are mathematical issues about this formulation which for a long time meant that mathematicians did not know whether it had a well-defined minimum (or minima). Also the continuous formulation means that although we can formally define a Gibbs distribution using this energy functional – it is unclear whether this is a meaningful probability distribution.

It can be shown (Ambrosio-Tortorelli, Shen) that the Mumford-Shah model is the limit of a family of models of form:

$$E[J,z;\epsilon] = C\int d\vec{x}(I(\vec{x}) - J(\vec{x}))^2 + A\int d\vec{x}z(\vec{x})|\vec{\nabla}J(\vec{x})|^2 + B\int d\vec{x}\{\epsilon|\vec{\nabla}J(\vec{x})|^2 + \epsilon^{-1}\phi^2(z(\vec{x}))\}, \tag{9}$$

where $\epsilon > 0$ is a (small) parameter and $\phi(z)$ is a potential function. Two forms of $\phi(z)$ are $\phi_1(z) = (1-z)/2$ and $\phi_2(z) = 3z(1-z)$ (where $z \in [0,1]$). For $\phi_1(.)$ the edge set B is associated with the set of points such that $\phi_1(z) \approx 0$,

for $\phi_2(.)$ the edge set is associated with points such that $\phi_2(z) \approx 1/2$. As $\epsilon \mapsto 0$, the last two terms of the energy function converge to the edge set integral $\int_B ds$ (this is not-trivial).

This energy $E[J, z; \epsilon]$ can be minimized (local minima) by alternative minimization (coordinate descent) since $E[J, z; \epsilon]$ is a convex function of J (for fixed z) and a convex function of z (for each J).

An alternative energy functional is by Rudin, Osher and Fatemi:

$$E[J; I] = \int_D |\vec{\nabla} J| d\vec{x} + \frac{\lambda}{2} \int_D (J(\vec{x}) - I(\vec{x}))^2 d\vec{x} \tag{10}$$

Calculus of variations (NEED AN APPENDIX ON THIS) gives Euler-Lagrange equations:

$$\vec{\nabla} \cdot \{ \frac{\vec{\nabla} J}{|\vec{\nabla} J|} \} + \lambda (J - I), \tag{11}$$

which can be solved by steepest descent $(dJ)/(dt) = -\vec{\nabla} \cdot \{ \frac{\vec{\nabla} J}{|\vec{\nabla} J|} \} + \lambda (J - I)$. This is often solved by the lagged diffusion method which requires solving:

$$-\vec{\nabla} \cdot \{ |\vec{\nabla} J_n|^{-1} \vec{\nabla} J_{n+1} \} + \lambda (J_{n+1} - I) = 0. \tag{12}$$

The R-O-F model can be reformulated be introducing a new variable z and an energy function:

$$E[J, z] = \frac{1}{2} \int_D \{ z |\vec{\nabla} J|^2 + z^{-1} \} d\vec{x} + \frac{\lambda}{2} \int_D (J - I)^2 d\vec{x}. \tag{13}$$

Then the lagged-diffusion model is obtained as alternating minimization (coordinate descent) of $E[J, z]$ with respect to J and z.

This can also be shown to be an example of CCCP (Yuille).

(WHAT ABOUT LEVEL SETS?? OTHER PROPERTIES OF R-O-F?? CHECK MEYER's BOOK!!)

5 Stochastic Sampling: MCMC Introduction

MCMC gives a way to sample from any distribution $P(\vec{x})$. This enables us to estimate quantities such as $\vec{x}^* = \arg\max P(\vec{x})$ or $\sum_{\vec{x}} \vec{\phi}(\vec{x}) P(\vec{x})$. The advantage of MCMC is that it does not require knowing the normalization constant Z of the distribution $P(\vec{x}) = (1/Z) \exp\{-E(\vec{x})\}$. But MCMC is an art rather than a science.

A Markov chain is defined by transition kernel $K(\vec{x}|\vec{x}')$, such that $\sum_{\vec{x}} K(\vec{x}|\vec{x}') = 1$, $\forall \vec{x}'$ and $K(\vec{x}|\vec{x}') \geq 0$. We also require the constraint that for any $\vec{x}_0$ and $\vec{x}_N$ there exists a chain $\vec{x}_1, ..., \vec{x}_{N-1}$ such that $K(\vec{x}_i|\vec{x}_{i-1}) > 0$ for $i = 1, ..., N$ (i.e. so that the chain is *irreducible* – you can get to any state from any other state in a finite number of moves).

An MCMC for a distribution $P(\vec{x})$ is a special Markov chain where the transition kernel satisfies $\sum_{\vec{y}} K(\vec{x}|\vec{y}) P(\vec{y}) = P(\vec{x})$ – i.e. the target distribution $P(\vec{x})$ is a fixed point of the chain. In practice, most MCMC are designed to satisfy the more restrictive *detailed balance* condition (which implies the fixed point condition):

$$K(\vec{x}|\vec{y}) P(\vec{y}) = K(\vec{y}|\vec{x}) P(\vec{x}). \tag{14}$$

To run MCMC we give an initial condition $\vec{x}_0$ and repeatedly sample from $K(\vec{x}'|\vec{x})$ to get a sequence $\vec{x}_1, ..., \vec{x}_t, ...$ so that for sufficiently large t $\vec{x}_t$ is a sample from $P(\vec{x})$.

5.1 Metropolis-Hastings and Gibbs Sampler

Metropolis-Hastings is an ansatz for constructing a transition kernel that obeys the detailed balance condition. It is specified by:

$$K(\vec{y}|\vec{x}) = T(\vec{y}|\vec{x}) \min\{1, \frac{P(\vec{y})T(\vec{x}|\vec{y})}{P(\vec{x})T(\vec{y}|\vec{x})}\}, \text{ for } \vec{y} \cdot \vec{x}) \tag{15}$$

where $T(\vec{y}|\vec{x})$ is a conditional distribution and $K(\vec{y}|\vec{y})$ is defined to ensure that $\sum_{\vec{y}} K(\vec{y}|\vec{x}) = 1$ is satisfied for all $\vec{x}$. It can be checked that this satisfies detailed balance. The form of $T(\vec{y}|\vec{x})$ must be chosen to ensure that this is irreducible.

Metropolis-Hastings can be thought of as a two stage process. First, use the *proposal distribution* $T(\vec{y}|\vec{x})$ to generate a *proposal* $\vec{y}$. Accept the proposal with *acceptance probability* $\min\{1, \frac{P(\vec{y})T(\vec{x}|\vec{y})}{P(\vec{x})T(\vec{y}|\vec{x})}\}$. In practice, the convergence speed of Metropolis-Hastings algorithms depends on whether good proposals can be found (we will return to this issue later in the course).

A key property of Metropolis-Hastings algorithms (and other MCMC) is that they do not require knowing the normalization constant Z of the distribution $P(\vec{x}) = (1/Z)\exp\{-E(\vec{x})\}$ (observe that Z cancels in the acceptance probability).

What is the intuition for Metropolis-Hastings? First, we sample from $T(\vec{y}|\vec{x})$ to propose a move to $\vec{y}$. We accept this move with certainty if $E(\vec{y}) < E(\vec{x}) + \log \frac{T(\vec{x}|\vec{y})}{T(\vec{y}|\vec{x})}$ (i.e. the energy decreases after allowing for the proposal probability). But if $E(\vec{y}) > E(\vec{x}) + \log \frac{T(\vec{x}|\vec{y})}{T(\vec{y}|\vec{x})}$, then the move can still be accepted with probability. Hence, unlike steepest descent the state of an MCMC will not get stuck in a local minima because it can always increase the energy (with probability). However, MCMC will not converge to a fixed point but instead to a probability distribution.

Some history – the Metropolis sampler (with a very simple proposal distribution) was the first "modern" MCMC. It was developed to sample from statistical physics systems specified by a Gibbs distribution. This was done at Los Alamos directly after World War 2 - the story goes that they had a lot of computers (first generation) at Los Alamos because of designing the Atomic Bomb (the Manhattan project) and wanted to find a good use for them. The first author of the paper was Metropolis and two of the other authors were, or became, very famous. One was Edward Teller (Hungarian Physicist) sometimes called the father of the Hydrogen bomb and later famous for advocating the Star Wars missile defense system. A second was Murray Rosenbluth who for many years was considered the world expert on the development of Nuclear Power based on Fusion (as used in H-bombs). By coincidence, Metropolis apparently really like the weak membrane model (when I presented it at Los Alamos in the 1980's).

The Gibbs sampler is another MCMC (often very simple to implement). This is usually considered to be slower than Metropolis-Hastings (with good proposal distribution) but is easy to implement. It has transition kernels

$$K_r(\vec{x}|\vec{y}) = P(x_r|y_{N(r)})\delta_{\vec{x}/r,\vec{y}/r}, \;\; K(\vec{x}|\vec{y}) = \sum_r \rho(r) K_r(\vec{x}|\vec{y}), \tag{16}$$

where x_r denotes the states of a subset r of nodes, $\vec{x}/r$ is the state of all the nodes except r, $x_{N(r)}$ is the state of all nodes that are neighbors of r, and $P(x_r|y_{N(r)})$ is the distribution of x_r conditioned on its neighbors. $\rho(r)$ is a distribution. In words, we select a subset r of nodes with probability $\rho(\vec{y})$ and update their states by sampling from $P(x_r|y_{N(r)})$ keeping the other states fixed. It can be checked that the Gibbs transition kernel satisfies detailed balance.

Simple illustration of Gibbs sampling. Consider the Ising model defined on $\{x_i\}$ with $x_i \in \{-1, +1\}$.

$$P(x_1, ..., x_d) = \frac{1}{Z}\exp\{\mu \sum_{i=1}^{d-1} x_i x_{i+1}\}. \tag{17}$$

The graphical structure has nearest neighbors – i.e. site i is connected to sites $i+1$ and $i-1$ so $N(i) = \{i-1, i+1\}$ (except for $N(1) = \{2\}$ and $N(d) = \{d-1\}$). We let r correspond to nodes i. Then:

$$P(x_i|\vec{x}/i) = P(x_i|x_{N(i)}) = P(x_i|x_{i+1}, x_{i-1}). \tag{18}$$

To determine this, we write $P(x_i|\vec{x}/i) = P(\vec{x})/P(\vec{x}/i)$. We know $P(\vec{x})$ and $P(\vec{x}/i) = \sum_{x_i} P(\vec{x}) = F(\vec{x}/i)$ where $F(.)$ is some function which we can calculate – but this is not the most direct way. It is better to observe that $P(x_i|\vec{x}/i)$ is a function of x_i and $\vec{x}/i$ divided by a function of $\vec{x}/i$ and must be normalized (i.e., $\sum_{x_i} P(x_i|\vec{x}/i) = 1$). Hence $P(x_i|\vec{x}/i) = \exp\{\mu(x_{i-1}x_i + x_i x_{i+1})\}/f(x_{i-1}, x_{i+1})$ where, by normalization, we have $f(x_{i-1}, x_{i+1}) = \exp\{\mu(x_{i-1} + x_{i+1})\} + \exp\{-\mu(x_{i-1} + x_{i+1})\}$. Hence the conditional distributions are:

$$P(x_i|x_{N(i)}) = \frac{\exp\{\mu(x_{i-1}x_i + x_i x_{i+1})\}}{\exp\{\mu(x_{i-1} + x_{i+1})\} + \exp\{-\mu(x_{i-1} + x_{i+1})\}}. \tag{19}$$

The moral is that the conditional distribution $P(x_i|x_{N(i)})$ is usually straightforward to compute for MRF models (NEED AN APPENDIX ON MRF!!). Similarly, we get $P(x_1|x_{N(1)}) = \frac{\exp\{\mu(x_1 x_2)\}}{\exp\{\mu x_2\}+\exp\{-\mu x_2\}}$ and $P(x_d|x_{N(d)}) = \frac{\exp\{\mu(x_{d-1}x_d)\}}{\exp\{\mu x_{d-1}\}+\exp\{-\mu x_{d-1}\}}$.

We now define a Gibbs sampler by selecting a site $i\{1, ..., d\}$ from a uniform distribution $U(.)$ (s.t. $U(i) = 1/d,\ \forall i$). Then we sample from $P(x_i|x_{N(i)})$ to generate a new value for x_i (tossing a biased coin). Then we sample another site and continue.

Another use of Gibbs sampling is for *data augmentation* which is like an MCMC version of the EM algorithm. Recall that EM is applied in situations where you have a distribution $P(y, h|d)$ where y denotes variables that you want to estimate, h is hidden variables that you do not care about, and d is the input. EM estimates y and $q(h)$ (distribution on h) by coordinate descent (alternative minimization). By contrast, data augmentation repeatedly samples from $P(y|h, d)$ and $P(h|y, d)$ to generate samples y, h from $P(y, h|d)$ (like Gibbs sampling – see above). These samples give a non-parameter estimate $\{(y_\mu|, h_\mu)\}$ of $P(y, h|d)$ from which we can estimate a non-parametric distribution over h – simply $\{h_i\}$ – and can estimate $y^* = \arg\max P(y|d)$. Unlike EM, data augmentation is guaranteed to converge (but in practice it may be hard to known when the algorithm has converged).

DETAILED BALANCE EXAMPLE – FROM STAT 202C NOTES!!

5.2 Theory of MCMC for detailed balance

It is straightfoward to obtain converge results for MCMC (with detailed balance) but unfortunately they depend on properties of the transition kernel which are often hard or impossible to commute (some very clever people – Diaconis, Strook – have obtained bounds for convergence of MCMC but only with difficulty and – like most bounds – they are only of limited use).

To study MCMC with detailed balance the key observation is that the quantity $Q(x, y) = P(y)^{1/2}K(x|y)P(x)^{-1/2}$ is a symmetric matrix. This enables us to apply linear algebra. In particular, $Q(x, y)$ has d real eigenvectors $\{e^\mu(x)\}$ and eigenvalues $\{\lambda^\mu\}$ (where d is the dimension of the state x and the eigenvalues are ordered by their magnitude), and hence can be expressed as $Q(x, y) = \sum_{\mu=1}^n \lambda^\mu e^\mu(x)e^\nu(x)$. It can be shown that $\lambda^1 = 1$ (corresponding to the fixed point conditions $\sum_y K(x|y)P(y) = P(x)$) and that $|\lambda^i| < 1,\ i = 2, ..., d$. It follows that

$$K^M(y|x)P_0(x) = P(x) + \sum_{\mu=2}^{d} \alpha_\mu \{\lambda^\mu\}^M e^\mu(x)P(x)^{1/2}, \tag{20}$$

where $K^M(y|x)$ is matrix multiplication of the transition kernel with itself M times, $P_0(x)$ is the initial distribution and $\alpha_\mu = \sum_y P_0(y)e^\mu(y)P(y)^{-1/2},\ \ \mu = 2, ..., d$.

The main result is that the second term on the RHS of the equation decay exponentially fast (in M) with decay speed determined by the magnitude of the second biggest eigenvalue λ^2. This implies that samples from the MCMC converge to samples from $P(x)$ exponentially rapidly. The only problem is that computing λ^2 is often impossible.

CHECK MACKAY ON STOPPING CONDITIONS!! WHY USE MCMC?? ONLY WHEN YOU CANNOT SAMPLE DIRECTLY FROM $P(x)$!!