6.892 Machine learning seminar

http://www.ai.mit.edu/courses/6.892-ml

Prof. Tommi S. Jaakkola
tommi@ai.mit.edu

March 6, 1999

The EM-algorithm

The EM-algorithm is a standard tool in statistics for dealing with missing values in maximum likelihood parameter/structure estimation. Our presentation here deviates from the "standard" view of this algorithm. For the standard view see the classic paper by Dempster et al. (1977) and/or the refinement provided by Neal and Hinton (1998). We start by describing a simple algorithm for maximizing convex functions within non-convex sets. Carrying this algorithm over to probabilistic setting gives the EM-algorithm.

Constrained maximization of a convex function

Suppose we have a convex continuously differentiable function $f(x)$, $x \in \mathcal{X}$, and we would like to find the maximum of this function within some compact set $x \in \mathcal{X}_c \subseteq \mathcal{X}$. Neither $\mathcal{X}$ nor $\mathcal{X}_c$ need to be convex sets. More precisely, we want

$$x_c = \operatorname*{argmax}_{x \in \mathcal{X}_c} \{ \, f(x) \, \} \tag{1}$$

The set $\mathcal{X}_c$ may be fragmented and thus finding the maximum may require rather exhaustive search. To deal with this difficulty we develop instead a simple iterative algorithm that is no longer guaranteed to find the global maximum but will nevertheless always find a local maximum that depends on an initial guess x_0 provided by the user. Whenever the set $\mathcal{X}_c$ is convex, however, the iterative algorithm will find the global maximum.

Our assumptions guarantee that we can find a tangent plane for the function f at any point x_0:

$$L(x; x_0) = \nabla f(x_0)^T \, (x - x_0) + f(x_0) \tag{2}$$

Since f is convex we must have

1. $L(x; x_0) \leq f(x)$ for all $x \in \mathcal{X}$
2. $L(x_0; x_0) = f(x_0)$

Suppose now that we were able to find $x_1 \in \mathcal{X}_c$ such that $L(x_1; x_0) > L(x_0; x_0)$. Then the above properties imply that

$$0 < L(x_1; x_0) - L(x_0; x_0) = L(x_1; x_0) - f(x_0) \leq f(x_1) - f(x_0) \tag{3}$$

or, equivalently, $f(x_1) > f(x_0)$, which is what we want. By applying this argument iteratively, we get the following algorithm for maximizing f:

Algorithm 1

1. *Choose* $x_0 \in \mathcal{X}_c$.
2. *Compute* $L(x; x_{t-1})$.
3. *Find* $x_t \in \mathcal{X}_c$ *such that* $L(x_t; x_{t-1}) > L(x_{t-1}; x_{t-1})$.
4. *Iterate until* $L(x_t; x_{t-1}) - L(x_{t-1}; x_{t-1}) < \epsilon$

This algorithm generates a sequence $x_0, x_1, \ldots, x_t$ with the property that

$$f(x_0) < f(x_1) < \ldots < f(x_t) \tag{4}$$

and stops whenever

$$\operatorname*{argmax}_{x \in \mathcal{X}_c} L(x_t; x_{t-1}) = L(x_{t-1}; x_{t-1}) = f(x_{t-1}) \tag{5}$$

At this point we have reached a local maximum. Note that we may still be able to find $x_t \neq x_{t-1}$ such that $L(x_t; x_{t-1}) = f(x_{t-1})$; the local maximum is unique in terms of the value of the function, not in terms of x that achieves the value.

Finally note that to find x_t from x_{t-1} as required by the algorithm we do not need to evaluate $L(x; x_{t-1})$ but only the part that depends on x:

$$x_t = \operatorname*{argmax}_{x \in \mathcal{X}_c} \{\, L(x; x_{t-1}) \,\} = \operatorname*{argmax}_{x \in \mathcal{X}_c} \{\, f(x_{t-1})^T \, x \,\} \tag{6}$$

The stopping criterion also relies on computing only $f(x_{t-1})^T \, x$:

$$L(x_t; x_{t-1}) - L(x_{t-1}; x_{t-1}) = f(x_{t-1})^T \, x_t - f(x_{t-1})^T \, x_{t-1} \tag{7}$$

The EM-algorithm

Suppose we have a parametric model

$$P(x|\theta) = P(x_1, \ldots, x_n|\theta) \tag{8}$$

over $x \in A$ where $\theta \in \Theta$ denotes the parameters. Let x_v be the set of visible or observed variables. We would like to maximize

$$\log P(x_v|\theta) = \log \sum_{x \setminus x_v} P(x \setminus x_v, x_v|\theta) \tag{9}$$

with respect to the parameters $\theta \in \Theta$. Typically there are multiple observations and possibly over different sets of variables. Our simple case here is readily extended to the more general setting.

The EM-algorithm proceeds to maximize the log-likelihood by starting with an initial guess θ_0 and successively refining the parameters by finding the maximum of

$$Q(\theta|\theta_0) = E_{\theta_0}\{\, \log P(x \setminus x_v, x_v|\theta) \mid x_v\} = \sum_{x \setminus x_v} P(x \setminus x_v|x_v, \theta_0)\, \log P(x|\theta) \tag{10}$$

with respect to $\theta \in \Theta$. The advantage of this formulation is that due to the factorization of the joint distribution, $\log P(x|\theta)$ often reduces to a simple sum that considerably simplifies the maximization step. The computation of the Q-function, i.e., carrying out the expectation over the missing variables, is called the *E-step* and the maximization with respect to θ is known as the *M-step*. We can write the algorithm as follows:

Algorithm 2

1. *Choose* $\theta_0 \in \Theta$.
2. *E-step: compute* $Q(\theta|\theta_{t-1})$.
3. *M-step: find* $\theta_t \in \Theta$ *such that* $Q(\theta_t|\theta_{t-1}) > Q(\theta_{t-1}|\theta_{t-1})$
4. *Iterate until* $Q(\theta_t|\theta_{t-1}) - Q(\theta_{t-1}|\theta_{t-1}) < \epsilon$

This is similar to our previous algorithm. We can establish the connection formally as follows. Let Φ be the space of all real vectors (functions) $\phi(x)$ whose components are indexed by the configurations $x \in A$. Also let Φ_c be the set of all vectors (functions) $\phi(x) = \log P(x|\theta)$ for $\theta \in \Theta$. Thus maximizing the log-likelihood is equivalent to maximizing

$$\log \sum_{x \setminus x_v} P(x|\theta) = \log \sum_{x \setminus x_v} e^{\log P(x|\theta)} = \log \sum_{x \setminus x_v} e^{\phi(x)} = f(\phi) \tag{11}$$

with respect to $\phi \in \Phi_c$. It can be shown that $f(\phi)$ is a convex function of ϕ. This is a well known property in the physics literature and can be shown easily by computing the Hessian and showing that it must be positive (semi-)definite. The gradient of f with respect to a particular component of ϕ, say $\phi(\tilde{x})$ is given by

$$\begin{aligned}
\frac{\partial f(\phi)}{\partial \phi(\tilde{x})} &= \frac{1}{\sum_{x \backslash x_v} e^{\phi(x)}} \sum_{x \backslash x_v} \delta(x = \tilde{x}) e^{\phi(x)} = \frac{1}{\sum_{x \backslash x_v} e^{\phi(x)}} e^{\phi(\tilde{x})} \delta(x_v = \tilde{x}_v) && (12) \\
&= \frac{1}{P(x_v|\theta)} P(\tilde{x}|\theta)\, \delta(x_v = \tilde{x}_v) && (13)
\end{aligned}$$

so long as $\phi \in \Phi_c$. Thus

$$\begin{aligned}
\nabla f(\phi_0)^T \phi &= \sum_{\tilde{x}} \left[\frac{1}{P(x_v|\theta_0)} P(\tilde{x}|\theta_0)\, \delta(x_v = \tilde{x}_v) \right] \log P(\tilde{x}|\theta) && (14) \\
&= \sum_{x \backslash x_v} \frac{P(x|\theta_0)}{P(x_v|\theta_0)} \log P(x|\theta) && (15) \\
&= \sum_{x \backslash x_v} P(x \setminus x_v | x_v, \theta_0) \log P(x|\theta) = Q(\theta|\theta_0) && (16)
\end{aligned}$$

which completes the connection. The analysis of the EM-algorithm follows the analysis of the previous algorithm.

References

A. Dempster, N. Laird, and D. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B* **39**:1-38.

R. Neal and G. Hinton (1998). A new view of the EM algorithm that justifies incremental and other variants. Learning in graphical models. M. I. Jordan (ed.). MIT Press.