Maximum Entropy Principle

Lecture notes for Cmput466/551 5/Apr/05

S Wang

Motivating example [1]: Suppose we wish to model an expert translator's decisions concerning the proper French rendering of the English word *in*. Our model of the expert's decisions assigns to each French word or phrase f an estimate, $p(f)$, of the probability that the expert would choose f as a translation of *in*.

We might discover that the expert translator always chooses among the following five French phrases: {*dans, en, a, au cours de, pendant*}. With this information in hand, we can impose our first constraint on our model p:

$$p(dans) + p(en) + p(a) + p(au\ cours\ de) + p(pendant) = 1$$

Knowing only that the expert chose exclusively from among these five French phrases, the most intuitively appealing model is

$$\begin{aligned} p(dans) &= 1/5 \\ p(en) &= 1/5 \\ p(a) &= 1/5 \\ p(au\ cours\ de) &= 1/5 \\ p(pendant) &= 1/5 \end{aligned}$$

Suppose we notice that the expert chose either *dans* or *en* 30% of the time. We could apply this knowledge to update our model of the translation process by requiring that satisfy two constraints:

$$\begin{aligned} p(dans) + p(en) &= 1/3 \\ p(dans) + p(en) + p(a) + p(au\ cours\ de) + p(pendant) &= 1 \end{aligned}$$

In the absence of any other knowledge, a reasonable choice for p is again the most uniform–that is, the distribution which allocates its probability as evenly as possible, subject to the constraints:

$$\begin{aligned} p(dans) &= 3/20 \\ p(en) &= 3/20 \\ p(a) &= 7/30 \\ p(au\ cours\ de) &= 7/30 \\ p(pendant) &= 7/30 \end{aligned}$$

Say we inspect the data once more, and this time notice another interesting fact: in half the cases, the expert chose either *dans* or *a*. We can incorporate this information into our model as a third constraint:

$$\begin{aligned} p(dans) + p(en) &= 1/3 \\ p(dans) + p(en) + p(a) + p(au\ cours\ de) + p(pendant) &= 1 \\ p(dans) + p(a) &= 1/2 \end{aligned}$$

We can once again look for the most uniform p satisfying these constraints, but now the choice is not as obvious. As we have added complexity, we have encountered two problems. First, what exactly is meant by "uniform," and how can one measure the uniformity of a model? Second, having determined a suitable answer to these questions, how does one find the most uniform model subject to a set of constraints like those we have described?

The maximum entropy method answers both these questions. Intuitively, the principle is simple: model all that is known and assume nothing about that which is unknown. In other words, given a collection of facts, choose a model which is consistent with all the facts, but otherwise as uniform as possible. This is precisely the approach we took in selecting our model at each step in the above example.

Maximum entropy (maxent) principle:

$$\max_{p(x)} H(p) = -\int_x p(x)\log p(x)dx \tag{1}$$

$$\text{s.t.} \int_x p(x)f_i(x) = \sum_x \tilde{p}(x)f_i(x),\ i = 1,\cdots,N \tag{2}$$

$$\int_x p(x) = 1 \tag{3}$$

$f_i(x)$: feature function
$\tilde{p}(x) = \frac{1}{M}$ number of times that x occurs in the train data $\tilde{\mathcal{X}} = (x_1,\cdots,x_M)$, empirical distribution of x

Fact Maximum entropy = Minimum KL distance between $p \in C$ and uniform distribution $\mathcal{U}$:

$$\begin{aligned}\min_p D(p||\mathcal{U}) &= \min_p \int_x p(x)\log\left(\frac{p(x)}{\mathcal{U}}\right)dx \\ &= \min_p \left(\int_x p(x)\log p(x)dx - \int_x p(x)\log\frac{1}{|\mathcal{U}|}dx\right) \\ &= \max_p \left(-\int_x p(x)\log p(x)dx + \log|\mathcal{U}|\right)\end{aligned}$$

Denote the set of $p(x)$ satisfying the constraints (2) as C.

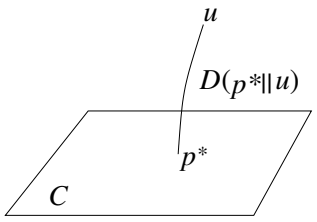

Figure 1: Maximum entropy over C= Minimum KL distance between $p \in C$ and uniform distribution.

Solving maxent: Maxent is a convex optimization problem with concave objective function over a set of linear constraints. This is called **primal** problem in optimization. We form the Lagrangian function using Lagrangian multipliers λ,

$$L(p,\lambda) = -\int_x p(x)\log p(x)dx + \lambda_0\left[\int_x p(x)dx - 1\right] + \sum_{i=1}^N \lambda_i\left[\int_x p(x)f_i(x)dx - \sum_x \tilde{p}(x)f_i(x)\right] \tag{4}$$

Differentiating with respect to $p(x)$, the xth component of p to obtain

$$\frac{\partial L(p,\lambda)}{\partial p(x)} = -\log p(x) - 1 + \lambda_0 + \sum_{i=1}^N \lambda_i f_i(x) \tag{5}$$

Setting this to 0, we obtain the form of the maximizing density

$$p(x) = e^{\lambda_0 - 1 + \sum_{i=1}^N \lambda_i f_i(x)} \tag{6}$$

where $\lambda_0, \lambda_1, \cdots, \lambda_N$ are chosen so that p satisfies the constraints.

In machine learning, people prefer to use another form

$$p_\lambda(x) = \frac{1}{Z_\lambda} e^{\sum_{i=1}^N \lambda_i f_i(x)} \tag{7}$$

where $Z_\lambda = \int_x e^{\sum_{i=1}^N \lambda_i f_i(x)} dx$ is called normalization constant or partition function to ensure (3). $\lambda_0 = -\log Z_\lambda + 1$ in this case.

Plug (7) into Lagrangian function (4), we obtain the unconstrained **dual** optimization problem

$$\min_\lambda \Psi(\lambda) = \log Z_\lambda - \sum_x \tilde{p}(x) \left(\sum_{i=1}^N \lambda_i f_i(x) \right) \tag{8}$$

Minimizing $\Psi(\lambda)$ is equivalent to maximizing

$$\mathcal{L}(\lambda) = \sum_x \tilde{p}(x) \log p_\lambda(x) = -\log Z_\lambda + \sum_x \tilde{p}(x) \left(\sum_{i=1}^N \lambda_i f_i(x) \right) \tag{9}$$

(MLE over exponential family $p_\lambda(x)$, Markov random field)

$$\min_\lambda \Psi(\lambda) = -\max_\lambda \mathcal{L}(\lambda) \tag{10}$$

$\Longrightarrow$ Maximum entropy subject to (2) $\equiv$ maximum likelihood estimation over exponential family $p_\lambda(x), \lambda \in \Re^N$.

Fact Maximum likelihood estimation over exponential family $p_\lambda(x)$ = Minimum KL distance between empirical distribution and exponential family, $D(\tilde{p}(x)||p_\lambda(x))$.

Theorem 1 *Let $p^*(x) = p_\lambda(x) = e^{\lambda_0 + \sum_{i=1}^N \lambda_i f(x)}$, where $\lambda_0, \cdots, \lambda_N$ are chosen so that p^* satisfies (2) and (3). Then p^* maximizes $H(p)$ over all probability densities p satisfying constraints (2) and (3).*

Proof:

$$\begin{aligned}
H(p^*) - H(p) &= -\int_x p^*(x) \log p^*(x) dx + \int_x p(x) \log p(x) dx \\
&= -\int_x p^*(x) \left(\lambda_0 + \sum_{i=1}^N \lambda_i f(x) \right) dx + \int_x p(x) \log p(x) dx \text{ (by definition of } p^*) \\
&= -\left(\lambda_0 + \sum_{i=1}^N \lambda_i \sum_x \tilde{p}(x) f_i(x) \right) + \int_x p(x) \log p(x) dx \ (p^* \text{ satisfies constraints}) \\
&= -\int_x p(x) \left(\lambda_0 + \sum_{i=1}^N \lambda_i f(x) \right) dx + \int_x p(x) \log p(x) dx \ (p \text{ satisfies constraints}) \\
&= -\int_x p(x) \log p^*(x) dx + \int_x p(x) \log p(x) dx \text{ (by definition of } p^*) \\
&= D(p||p^*) \geq 0
\end{aligned}$$

∎

Example 1. Let the constraints be $EX = 0$, $EX^2 = \sigma^2$. Then the form of the maxent distribution is

$$p(x) = e^{\lambda_0 + \lambda_1 x + \lambda_2 x^2}$$

We first recognize that this distribution has the same form as a normal distribution. It has mean 0 and variance σ^2. Hence maxent $\Rightarrow N(0, \sigma^2)$.

Example 2. (Dice, no constraints) $X \in S = \{1, 2, 3, 4, 5, 6\}$. Maxent is the uniform distribution, $p(x) = 1/6$ for $x \in S$.

Example 3. $x \in [0, \infty)$ and $EX = \mu$. Then the form of the maxent distribution is

$$p(x) = e^{\lambda_0 + \lambda_1 x}, \quad x \geq 0$$

It's easy to recognize that this distribution has the same form as an exponential distribution.

Theorem 2 *(Pythagorean Theorem) Let* $\Theta = \left\{\lambda \in \Re^m : \int g(x) \exp\left(\sum_{i=1}^N \lambda_i f_i(x)\right)\, dx < \infty\right\}$ *and* $\mathcal{E} = \left\{p_\lambda : p_\lambda(x) = \frac{1}{Z_\lambda} \exp\left(\sum_{i=1}^N \lambda_i f_i(x)\right),\ \lambda \in \Theta\right\}$. *If there exists* $p^* \in C \cap E$, *then* p^* *satisfies*

$$D(p||p_\lambda) = D(p||p^*) + D(p^*||p_\lambda) \quad \forall p \in C, \forall p_\lambda \in \mathcal{E}, \lambda \in \Theta \tag{11}$$

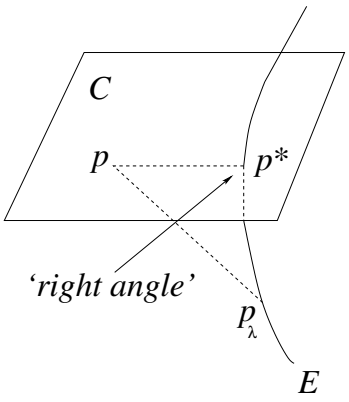

Figure 2: Pythagorean theorem.

Computing the optimal parameters of maxent solution

For all but the most simple problems, the λ^* that maximize $\mathcal{L}(\lambda)$ cannot be found analytically. Instead, we must resort to numerical methods. Denote $\underline{f}(x) = (f_1(x), \cdots, f_N(x))^T$, $\mathcal{L}(\lambda)$ has the following properties:

$$\frac{\partial \mathcal{L}(\lambda)}{\partial \lambda} = -\frac{\partial \log Z_\lambda}{\partial \lambda} + \sum_x \tilde{p}(x) f(x) = -E_{p_\lambda(x)}(\underline{f}(x)) + E_{\tilde{p}(x)}(\underline{f}(x)) \tag{12}$$

$$\frac{\partial^2 \mathcal{L}(\lambda)}{\partial \lambda \partial \lambda^T} = -\frac{\partial^2 \log Z_\lambda}{\partial \lambda \partial \lambda^T} = -Var_{p_\lambda(x)}(\underline{f}(x)) \text{ (negative definite)} \tag{13}$$

The log-likelihood function is concave in each λ. A variety of numerical methods can be used to calculate λ^*, for example, gradient ascent and conjugate gradient.

Here we describe a nice simple algorithm called the *Iterative Scaling* algorithm, that can be used to find λ^*. We assume $f_i(x) \geq 0$ and $\sum_{i=1}^N f_i(x) = 1$. (If $f_i(x)$ is sometimes negative, we can replace $f_i(x)$ by $f_i(x) + c$ for some constant c. Because of normalization, this does not change the corresponding exponential distribution. If $\sum_{i=1}^N f_i(x) < 1$, we can add a new feature $f_0(x) = 1 - \sum_{i=1}^N f_i(x)$. Since a linear combination over all the features including f_0 is the same as one over just the original features, this again does not change the problem or the exponential distributions that can be represented.)

We are looking for a sequence of vectors $\lambda^1, \lambda^2, ...$ and we want $\mathcal{L}$ to be going up for each subsequent vector. So in a single time step say from t to $t+1$ the difference can be defined as below, and we want to lower bound this. Denote $\delta\lambda_i^t = \lambda_i^{t+1} + \lambda_i^t$,

$$\begin{aligned}
\Delta\mathcal{L} &= \mathcal{L}(\lambda^{t+1}) - \mathcal{L}(\lambda^t) \\
&= -\log Z_{\lambda^{t+1}} + \sum_x \tilde{p}(x) \left(\sum_{i=1}^N \lambda_i^{t+1} f_i(x)\right) + \log Z_{\lambda^t} - \sum_x \tilde{p}(x) \left(\sum_{i=1}^N \lambda_i^t f_i(x)\right) \\
&= -\log \frac{Z_{\lambda^{t+1}}}{Z_{\lambda^t}} + \sum_{i=1}^N \delta\lambda_i^t \left(\sum_x \tilde{p}(x) f_i(x)\right)
\end{aligned} \tag{14}$$

Now,

$$\frac{Z_{\lambda^{t+1}}}{Z_{\lambda^t}} = \frac{\int_x \exp\left(\sum_{i=1}^N \lambda_i^{t+1} f_i(x)\right) dx}{Z_{\lambda^t}}$$

$$
\begin{aligned}
&= \frac{\int_x \exp\left(\sum_{i=1}^N \lambda_i^t f_i(x)\right) \exp\left(\sum_{i=1}^N \delta\lambda_i^t f_i(x)\right) dx}{Z_{\lambda^t}} \text{ (plug in the value of} \lambda_i^{t+1}) \\
&= \int_x p_{\lambda^t}(x) \exp\left(\sum_{i=1}^N \delta\lambda_i^t f_i(x)\right) dx \\
&\leq \int_x p_{\lambda^t}(x) \sum_{i=1}^N f_i(x) e^{\delta\lambda_i^t} dx \text{ (Jensen's inequality exp is convex, } \underline{f} \text{ is a distribution)} \\
&= \sum_{i=1}^N e^{\delta\lambda_i^t} \int_x p_{\lambda^t}(x) f_i(x) dx \qquad (15)
\end{aligned}
$$

Plug (15) into (14), we have

$$\Delta\mathcal{L} \geq \sum_{i=1}^N \delta\lambda_i^t \left(\sum_x \tilde{p}(x) f_i(x)\right) - \log\left(\sum_{i=1}^N e^{\delta\lambda_i^t} \int_x p_{\lambda^t}(x) f_i(x) dx\right) \qquad (16)$$

We have derived a lower bound of the change in the loss function. Now, to optimize, we can take the derivative to choose the $\delta\lambda_i^t$ that is the largest.

$$\frac{\partial}{\delta\lambda_i^t} = \left(\sum_x \tilde{p}(x) f_i(x)\right) - \frac{e^{\delta\lambda_i^t} \int_x p_{\lambda^t}(x) f_i(x) dx}{\left(\sum_{i=1}^N e^{\delta\lambda_i^t} \int_x p_{\lambda^t}(x) f_i(x) dx\right)} = 0 \qquad (17)$$

The thing to notice is that if we have any solution for this, say $\delta\lambda'^t_i$ we can add a constant c and get another solution, $\delta\lambda_i^t = \delta\lambda'^t_i + c$. The reason is that the constants get cancelled. We will choose this constant in such a way that the denominator of the second term equals 1.

By using that trick we can set

$$\delta\lambda_i^t = \log\left(\frac{\sum_x \tilde{p}(x) f_i(x)}{\int_x p_{\lambda^t}(x) f_i(x) dx}\right) \qquad (18)$$

The iterative scaling algorithm works iteratively by calculating successive λ^t's in each round, for $t = 1, 2, \cdots$

$$\lambda_i^{t+1} = \lambda_i^t + \log\left(\frac{\sum_x \tilde{p}(x) f_i(x)}{\int_x p_{\lambda^t}(x) f_i(x) dx}\right), \quad i = 1, \cdots, N \qquad (19)$$

which monotomically increases the log-likelihood.

Theorem 3 *Denote p^* as the optimal maxent distribution, then as $t \to \infty$, $p_{\lambda^t} \to p*$.*

Regularized maximum entropy (RME) principle

In situations of limited traing data, we can't get an accurate estimate of empirical feature expectations. We use regularized maximum entropy principle instead,

$$\max_{p,a} \quad H(p) - U(a) \qquad (20)$$

$$\text{s.t} \int_x p(x) f_i(x) = \sum_x \tilde{p}(x) f_i(x) + a_i, \quad i = 1, ..., N \qquad (21)$$

Here $a = (a_1, ..., a_N)$, a_i is the error for each constraint, and $U : \Re^N \to \Re$ is a convex function which has its minimum at 0. The function U penalizes errors in satisfying the constraints, and can be used to penalize deviations in the more reliably observed constraints to a greater degree than deviations in less reliably observed constraints.

The key observation to finding optimal solutions is to note that they are intimately related to finding *maximum a posteriori* (MAP) solutions: Given a penalty function U over errors a, an associated *prior* U^* on λ can be obtained by setting U^* to the convex (Fenchel) conjugate of U [2]. Vice versa, given the convex conjugate cost function U^*, the corresponding penalty function U can be derived by using the property of *Fenchel biconjugation*; that is, the conjugate of the conjugate of a convex function is the original convex function, $U = U^{**}$.

To illustrate, consider a quadratic penalty $U(a) = \sum_{i=1}^N \frac{1}{2}\sigma_i^2 a_i^2$. Here the convex conjugate $U^*(\lambda) = \sum_{i=1}^N \frac{\lambda_i^2}{2\sigma_i^2}$ can be determined by setting $a_i = \frac{\lambda_i}{\sigma_i^2}$; which specifies a Gaussian prior on λ. A different example can be obtained by considering the Laplacian prior on λ, $U^*(\lambda) = ||\lambda||_1 = \sum_{i=1}^N |\lambda_i|$, which leads to the penalty function

$$U(a) = \begin{cases} 0 & ||a||_\infty = \max_{i=1}^N |a_i| \le 1 \\ \infty & \text{otherwise} \end{cases}$$

that forces hard inequality constraints.

Note that in each case, given a prior U^*, the standard MAP estimate maximizes the penalized log-likelihood $R(\lambda) = \sum_x \tilde{p}(x) \log p_\lambda(x) - U^*(\lambda)$.

$\Longrightarrow$ Maximum regularized entropy subject to (21) $\equiv$ Maximum penalized log-likelihood over exponential family $p_\lambda(x)$ with penalty U^*.

Maximum conditional entropy and conditional random fields

Denote Y as observed data, Z missing data and $X = (Y, Z)$ complete data. Given training data $\tilde{D} = [(y_1, z_1), \cdots, (y_M, z_M)]$.

Primal problem: maximum conditional entropy

$$\max_{p(z|y)} \left(-\sum_y \tilde{p}(y) \int_z p(z|y) \log p(z|y) dz \right) \tag{22}$$

$$\text{s.t.} \sum_y \tilde{p}(y) \int_z p(z|y) f_i(y,z) = \sum_{y,z} \tilde{p}(y,z) f_i(x), \ i = 1, \cdots, N \tag{23}$$

$$\int_z p(z|y) = 1, \quad \forall y \tag{24}$$

Dual problem: maximum likelihood over conditional random fields

$$\sum_{y,z} \tilde{p}(y,z) \log p_\lambda(z|y) \tag{25}$$

where $p_\lambda(z|y) = \frac{1}{Z_\lambda(y)} \exp\left(\sum_{i=1}^N f_i(y,z)\right)$ and $Z_\lambda(y) = \int_z \left(\sum_{i=1}^N f_i(y,z)\right) dz$ is a normalization constant to ensure (24) for each y.

$\Longrightarrow$ Maximum conditional entropy $\equiv$ maximum conditional likelihood over conditional random fields $p_\lambda(z|y)$.

References

[1] A. Berger, S. Della Pietra, and V. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39-71, 1996.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*, Cambridge University Press, 2004

[3] T. Cover and J. Thomas. *Elements of Information Theory*, John Wiley, 1991.

[4] S. Della Pietra, V. Della Pietra and John Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, April, 1997.