In this lecture we study maximum entropy models and how to use them to model natural language classification problems. Maximum Entropy models are probabilistic models. As in the general Bayesian approach a model is selected from a class of hypotheses based on the data observed. Maximum entropy modeling offers a clean and philosophically appealing way to select the "best" model without making any (independence or other) assumptions on the data. We will also discuss some computational issues and a learning theory view of this approach.
The notes are based on notes of Adwait Ratnaparkhy

1 Introduction

The *Principle of Maximum Entropy* [Jay68, Goo63] states that when one searches for a probability distribution p that satisfies some constrains (evidence), the correct one to choose is the one that maximizes the uncertainty (or: entropy) subject to these constrains.

> ...in making inferences on the basis of partial information we must use that probability distribution which has maximum entropy subject to whatever is known. This is the only unbiased assignment we can make; to use any other would amount to arbitrary assumption of information which by hypothesis we not have.

Consider a typical NLP classification problem:

The *from* needs to be completed. (The form needs to be completed).

In this case, as in most classification problems, instances (sentences, say) are elements in the instance space X', and class labels ("from", "form" in this case), are taken from a discrete set C. We are typically interested in modeling the joint probability distribution over the space $X = X' \times C$. So, according to Jaynes, that probability distribution we should look for is a probability distribution over X which is *consistent with the evidence we have* and which maximizes

$$H(p) = -\sum_{x \in X} p(x) \log p(x),$$

where here $x =< x', c > \in X' \times C$

2 Representing Evidence

The representation of evidence will determine the form of the probability distributions we consider. We will encode facts (actually, statistics) about the observed data as *features*.

Features are *conditions* over the instances $x \in X$. Formally, we can define them as characteristic functions

$$\chi : X \to [0, 1].$$

That is, each feature distinguishes a certain subset of the instance space $(X' \times C)$ – all those elements that satisfy χ. We will denote the set of all features by $\mathcal{X}$.

Given a set of features we can encode the constrains. In the Maximum Entropy Paradigms this is encoded by requiring that the expected value of each feature under the target probability distribution is the same as the expected value of the feature under the empirical distribution. That is,

$$\forall \chi \in \mathcal{X}, E_p \chi = E_{\tilde{p}} \chi.$$

Here, $\tilde{p}$ is the observer probability distribution in the training sample S and

$$E_{\tilde{p}} \chi = \sum_{x \in X} \tilde{p}(x)\chi(x) = \sum_{x \in S} \tilde{p}(x)\chi(x),$$

where the latter equality assumes no smoothing. Similarly,

$$E_p \chi = \sum_{x \in X} p(x)\chi(x).$$

Notice that χ are binary functions and can also be thought as *events* in X. Therefore

$$E_p \chi \equiv p(\chi)$$

and, similarly

$$E_{\tilde{p}} \chi \equiv \tilde{p}(\chi),$$

where the last term simply refers to the maximum likelihood estimate of the event χ using the sample S.

That is, **we are looking for a distribution p that has the same marginals as the empirical one $\tilde{p}$.**

Notice that this is the same class of distributions that we considered when we worked with the naive Bayes algorithm.

Only that there we chose a specific member of this class of distributions, the product distribution.

Here, we are using the principle of Maximum entropy and search for a distribution p^* such that

$$p^* = argmax_{p \in \mathcal{P}} H(p),$$

where the search is in a class $\mathcal{P}$ of distributions defined by

$$\mathcal{P} = \{p | E_p \chi = E_{\tilde{p}} \chi, \forall \chi \in \mathcal{X}\}.$$

Later on we will show that the sought after distribution must have a form equivalent to:

$$p^*(x) = k \Pi_{\chi_i \in \mathcal{X}} \alpha_i^{\chi_i(x)}, 0 < \alpha_i < \infty$$

where k is a normalization factor and the α_is are the model parameters. We can also call α_i the *weight* of the feature χ_i, especially when we look at the logarithmic representation:

$$\log p^*(x) = k' + \sum_{\chi_i \in \mathcal{X}} \log \alpha_i \chi_i(x).$$

(But notice that the constant k' depends on x.)

3 Maximum Entropy: Flow of Story

- The notion of entropy and KL divergence
  - Important property: positivity.
- Two classes of distributions:
  1. $\mathcal{P}$: All probability distributions that satisfy the constraints
  2. $\mathcal{Q}$: All probability distributions that can be written in a certain way (exponential form; log-linear)
- An important property of $\mathcal{P}$ and $\mathcal{Q}$: The Pythagorean Theorem.
- Fundamental Theorem: The "best" distribution in $\mathcal{P}$ (that satisfies the constraints has an exponential form (is in $\mathcal{Q}$).
- Maximum Likelihood perspective: In the class of exponential form distributions, searching for maximum likelihood and for maximum entropy is the same thing.
- Algorithmic issues: classical view and modern view

4 The Notion of Entropy

For a given random variable X, how much information is conveyed in the message that $X = x$?

In order to quantify this statement we can first agree that the amount of information in the message that $X = x$ should depend on how likely it was the X would equal x.

In addition, it seems reasonable to assume that the more unlikely it was that X would equal x, the more informative would the message be.

For instance, if X represents the sum of two fair dice, then there seems to be more information in the message that $X = 12$ than there would be in the message that $X = 7$ since the former events happens with probability 1/36 and the latter 1/6.

Let's denote by $I(p)$ the amount of information contained in the message that an event whose probability is p has occurred. Clearly $I(p)$ should be non negative, decreasing function of p.

To determine its form, let X and Y be independent random variables, and suppose that

$$P\{X = x\} = p \qquad P\{Y = y\} = q.$$

How much information is contained in the message that X equals x and Y equals y?

Note that since knowledge of the fact that X equals x does not affect the probability that Y will equal y (since X, Y are independent), it seems reasonable that the additional amount of information contained in the statement the $Y = y$ should equal $I(q)$. Therefore, the amount of information in the message that $X = x$ and $Y = y$ is $I(p) + I(q)$.

On the other hand:

$$P\{X = x, Y = y\} = P\{X = x\}P\{Y = y\} = pq$$

which implies that the the amount of information in the message that $X = x$ and $Y = y$ is $I(pq)$.

Therefore, the function I should satisfy the identity

$$I(pq) = I(p) + I(q)$$

If we define

$$I(2^{-p}) = G(p)$$

we get from the above that

$$G(p + q) = I(2^{-(p+q)}) = I(2^{-p}2^{-q}) = I(2^{-p}) + I(2^{-q}) = G(p) + G(q)$$

However, it can be shown that the only monotone functions G that satisfy the this functional relationship are those of the form

$$G(p) = cp$$

for some constant c. Therefore, we must have that

$$I(2^{-p}) = cp,$$

or, letting $z = 2^{-p}$,

$$I(z) = -clog_2(z)$$

for some positive constant c. It is traditional to assume that $c = 1$ and say that the information is measured in units of *bits*.

Consider now a random variable X, which must take on one of the values $x_1, \ldots x_n$ with respective probabilities $p_1, \ldots p_n$.

As $-log(p_i)$ represents the information conveyed by the message that X is equal to x_i, it follows that the expected amount of information that will be conveyed when the value of X is transmitted is given by

$$H(X) = -\sum_{i=1}^{n} p_i log_2(p_i)$$

This quantity is know in information theory as the **entropy** of the random variable X.

Definition 4.1 (Entropy) *Let p be a probability distribution over a discrete domain X. the entropy of p is*

$$H(p) = -\sum_{x \in X} p(x) \log p(x).$$

Notice that we can think of the entropy as

$$H(p) = -E_p \log p(x).$$

Since $0 \leq p(x) \leq 1$, $1/p(x) > 1$, $\log 1/p(x) > 0$ and therefore $0 < H(p) < \infty$.

Intuitively, the entropy of the random variable measures the uncertainty of the random variable (or: how much we know about its value when we know the distribution p). The value of $H(p)$ is thus maximal and equal to $\log |X|$ when p is the uniform distribution.

Example 4.1 *Consider a simple language L over a vocabulary of size 8. Assume that the distribution over L is uniform: $p(l) = 1/8, \forall l \in L$.*

Then,

$$H(L) = -\sum_{l=1}^{8} p(l) log p(l) = -log\frac{1}{8} = 3$$

Indeed, if you want to transmit a character in this language, the most efficient way is to encode each of the 8 characters in 3 bits. There isn't a more clever way to transmit these messages. An optimal code sends a message of probability p in $-logp$ bits.

On the other hand, if the distribution over L is:

$$\{1/2, 1/81/8, 1/8, 1/32, 1/32, 1/32, 1/32\}$$

Then:

$$H(L) = -\sum_{l=1}^{8} p(l) log p(l) = [1/2 \cdot 1 + 3(1/8 \cdot 3) + 4(1/32 \cdot 5) = 1/2 + 9/8 + 20/32 = 2.25.$$

Definition 4.2 (Joint and Conditional Entropy) *Let p be a joint probability distribution over a pair of discrete random variables X, Y. The average amount of information needed to specify both their values is the joint entropy:*

$$H(p) = H(X, Y) = -\sum_{y \in Y} \sum_{x \in X} p(x, y) \log p(x, y).$$

The conditional entropy of Y given X, for $p(x, y)$ expresses the average additional information one needs to supply about Y given that X is known:

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X = x) = \sum_{x \in X} p(x)[-\sum_{y \in Y} p(y|x) log p(y|x)] = -\sum_{y \in Y} \sum_{x \in X} p(x, y) \log p(y|x).$$

Definition 4.3 (Chain Rule) *The chain rule for entropy is given by:*

$$H(X, Y) = H(X) + H(Y|X).$$

More generally:

$$H(X_1, X_2, \ldots X_n) = H(X_1) + H(X_2|X_1) + \ldots + H(X_1, X_2, \ldots X_{n-1}).$$

5 Max Entropy: Examples

Consider the problem of selecting a probability distribution over two variables, $X = X' \times C$, where $X' = \{x, y\}$ and $C = \{0, 1\}$.

We will assume that the only feature we have is

$$\{c = 0\},$$

which, written in a functional form is:

$$\chi(x', c) = 1 \quad \text{when} \quad c = 0 \quad \text{and} \quad \chi(x', c) = 0 \quad \text{otherwise} \quad .$$

After observing the sample S we determine that the constrain is

$$E_{\tilde{p}}\chi = E\tilde{p}\{c = 0\} = p(x, 0) + p(y, 0) = 0.6.$$

Notice that since p is a probability distribution we can think about it as if we have two features, $\{c = 0\}$ and $\{c = 1\}$.

Now, from all the probability distributions with this marginal, we need to select the one that has the maximal entropy. Intuitively, this is the one that has the least additional constrains or, the most uncertainty.

We need to fill this table:

X'\C	0	1
x		
y		
total	0.6	1.0

One way to do it is:

X'\C	0	1	
x	0.3	0.2	
y	0.3	0.2	
total	0.6		1.0

In this case, the entropy is

$$\begin{aligned} H(p) &= -\sum_{x \in X} p(x) \log p(x) \\ &= 0.3 \log 0.3 + 0.3 \log 0.3 + 0.2 \log 0.2 + 0.2 \log 0.2 \\ &= 1.366 / \ln 2 = 1.97 \end{aligned}$$

(I used $\log_2$)

Another way is:

X'\C	0	1	
x	0.5	0.3	
y	0.1	0.1	
total	0.6		1.0

Here we get:

$$\begin{aligned} H(p) &= -\sum_{x \in X} p(x) \log p(x) && (1) \\ &= 0.5 \log 0.5 + 0.3 \log 0.3 + 0.1 \log 0.1 + 0.1 \log 0.1 && (2) \\ &= 1.16 / \ln 2 = 1.68 && (3) \end{aligned}$$

Notice that the maximum entropy you can get for a distribution over a domain of size 4 is $-log1/4 = log4 = 2$.

A *feature* is simply a specification of a **list of cells in the joint probability table**. Therefore a constrain is just a requirement on the sums of the probability mass in these cells.

In general, though, there is no *closed form solution* to finding the distribution that satisfies the constrains and has the maximum entropy and there will be a need to resort to an iterative algorithm.

5.1 An NLP example

Consider the context sensitive spelling correction examples that you have looked at before. The notion of a *feature* is the same as here. Notice the difference from FEX, where you have interacted with the program at the level of *types* of features.

In this case, the distribution we consider is over the space of instances of the form:

$$\{(x, c) = (((w_1, t_1), (w_2, t_2), \ldots, (w_{i-1}, t_{i-1}), , (w_{i+1}, t_{i+1}), \ldots (w_k, t_k)), c)\},$$

where $c \in$ {accept,except} is the target word and the sentence is represented as a set of pairs, word and pos tag, and i is the index of the target word in the sentence.

We can then define a features like:

- $\chi(x, c) = 1$ iff `the word before target is "the" and target is "accept"`
- $\chi(x, c) = 1$ iff `the word after the target is a proposition`

In order to generate the constrains, for each of these features we will count the number of times we see this feature active in the training sample, and divide by the size of the training sample. (Exactly was what we will do in the case of naive Bayes!)

Once we have written down all the constrains, the joint probability distribution we seek is uniquely determined (as will be shown shortly) and we only need to compute it.

Once it is given, making a decision is easy. Let p^* be the joint probability distribution selected, then:

$$c = argmax_{c \in C} p^*((x, c)).$$

6 Maximum Entropy: Flow of Story

- The notion of entropy and KL divergence
  - Important property: positivity.
- Two classes of distributions:
  1. $\mathcal{P}$: All probability distributions that satisfy the constraints
  2. $\mathcal{Q}$: All probability distributions that can be written in a certain way (exponential form; log-linear)
- An important property of $\mathcal{P}$ and $\mathcal{Q}$: The Pythagorean Theorem.
- Fundamental Theorem: The "best" distribution in $\mathcal{P}$ (that satisfies the constraints has an exponential form (is in $\mathcal{Q}$).
- Maximum Likelihood perspective: In the class of exponential form distributions, searching for maximum likelihood and for maximum entropy is the same thing.
- Algorithmic issues: classical view and modern view

7 Information Theory Preliminaries

Definition 7.1 (Notation)

- *X': space of possible instances (examples)*
- *C: space of possible classes*
- *Joint Space: $X = X' \times C$*
- *S: Training sample*
- *A feature function: $\chi : X \to \{0, 1\}$*
- *Class of all features: $\mathcal{X}$.*
- *A probability distribution over X: $p : X \to [0, 1]$*
- *Observed distribution in S: $\tilde{p} : X \to [0, 1]$*
- $E_p\chi = \sum_{x \in X} p(x)\chi(x)$
- $E_{\tilde{p}}\chi = \sum_{x \in X} \tilde{p}(x)\chi(x)$

- *Class of constrained distributions:*

$$\mathcal{P} = \{p | E_p\chi = E_{\tilde{p}}\chi, \forall \chi \in \mathcal{X}\}$$

- *Class of Max Entropy Distributions:*

$$\mathcal{Q} = \{p | p(x) = k\Pi_{\chi_i \in \mathcal{X}} \alpha_i^{\chi_i(x)}, 0 < \alpha_i < \infty\}$$

Definition 7.2 (Entropy) *For a probability distribution p over a discrete domain X the entropy of p is*

$$H(p) = -\sum_{x \in X} p(x) \log p(x).$$

Notice that we can think of the entropy as

$$H(p) = -E_p \log p(x).$$

Since $0 \leq p(x) \leq 1$, $1/p(x) > 1$, $\log 1/p(x) > 0$ and therefore $0 < H(p) < \infty$.

Intuitively, the entropy of the random variable measures the uncertainty of the random variable (or: how much we know about its value when we know p). The value of $H(p)$ is thus maximal and equal to $\log |X|$ when p is the uniform distribution.

Definition 7.3 (Relative Entropy; Kullback-Liebler Distance) *Let p, q be two probability distributions over a discrete domain X. The relative entropy between p and q is*

$$D(p, q) = D(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

Notice that $D(p||q)$ is not symmetric. It could be viewed as

$$D(p||q) = E_p \log \frac{p(x)}{q(x)},$$

the expectation according to p of $\log p/q$. It is therefore unbounded and not defined if p gives positive support to instances q does not.

Lemma 7.1 (KL-Divergence) *For any probability distributions p, q, $D(p||q) \geq 0$ and equality holds if and only if $p = q$.*

Lemma 7.2 (Pythagorean Property) *Let $\mathcal{P}, \mathcal{Q}$ be as defined above. Let $p \in \mathcal{P}, q \in \mathcal{Q}$ and $p^* \in \mathcal{P} \cap \mathcal{Q}$. Then,*

$$D(p||p^*) + D(p^*||q) = D(p||q).$$

Proof: If $t \in \mathcal{Q}$,

$$\log t(x) = k + \sum_{\chi_i \in \mathcal{X}} \log \alpha_i \chi_i(x).$$

Therefore, for any $r, s \in \mathcal{P}, t \in \mathcal{Q}$, since

$$\sum_x r(x)\chi_i(x) = \sum_x s(x)\chi_i(x)$$

(due to the fact that these are $E_r\chi_i$ and $E_s\chi_i$ resp., and the definition of $\mathcal{P}$), we have

$$\begin{aligned}
\sum_{x\in X} r(x)\log t(x) &= \sum_x r(x)[k + \sum_{\mathcal{X}} \log \alpha_i \chi_i(x)] \\
&= k[\sum_x r(x)] + [\sum_{\mathcal{X}} \log \alpha_i \sum_x r(x)\chi_i(x)] \\
&= k[1] + [\sum_{\mathcal{X}} \log \alpha_i \sum_x r(x)\chi_i(x)] \\
&= k[\sum_x s(x)] + [\sum_{\mathcal{X}} \log \alpha_i \sum_x s(x)\chi_i(x)] \\
&= \sum_x s(x)[k + \sum_{\mathcal{X}} \log \alpha_i \chi_i(x)] \\
&= \sum_{x\in X} s(x)\log t(x)
\end{aligned}$$

Now, since $p^* \in \mathcal{P} \cap \mathcal{Q}$

$$\begin{aligned}
D(p||p^*) + D(p^*||q) &= \sum_x p(x)\log p(x) - \sum_x p(x)\log p^*(x) + \sum_x p^*(x)\log p^*(x) - \sum_x p^*(x)\log q(x) \\
&= \sum_x p(x)\log p(x) - \sum_x p(x)\log p^*(x) + \sum_x p(x)\log p^*(x) - \sum_x p(x)\log q(x) \\
&= \sum_x p(x)\log p(x) - \sum_x p(x)\log q(x) \\
&= D(p||q)
\end{aligned}$$

where we have used the previous equality twice- once for $p, p^* \in \mathcal{P}$ w.r.t $q \in \mathcal{Q}$ and once for $p, p^* \in \mathcal{P}$ w.r.t $p^* \in \mathcal{Q}$. ■

8 Maximum Entropy

Now we can get the Maximum Entropy property:

Theorem 8.1 (Maximum Entropy Property) *Let $\mathcal{P}, \mathcal{Q}$ be as defined above. Let $p \in \mathcal{P}$ and $p^* \in \mathcal{P} \cap \mathcal{Q}$. Then,*

$$p^* = argmax_{p \in \mathcal{P}} H(p).$$

Furthermore, p^* is unique.

In words: p^*, the "best" distribution (in the sense of maximizing the entropy) has an exponential form; and it is unique.

Proof: Assume that $p \in \mathcal{P}$ and $p^* \in \mathcal{P} \cap \mathcal{Q}$. Let u be the uniform distribution over X, that is $u(x) = \frac{1}{|X|}$. Notice the $u \in \mathcal{Q}$ (by taking $\forall i, \alpha_i \equiv 1$). Therefore, by lemma 7.2

$$D(p||u) = D(p||p^*) + D(p^*||u)$$

and by lemma 7.1

$$D(p||u) \geq D(p^*||u)$$

which means

$$-H(p) - \log \frac{1}{|X|} \geq -H(p^*) - \log \frac{1}{|X|}$$

and therefore

$$-H(p) \geq -H(p^*).$$

This shows that all distributions $p^* \in \mathcal{P} \cap \mathcal{Q}$ have entropy that is not smaller than any $p \in \mathcal{P}$. The fact that actually the equality never holds and the distribution we seek is the one with the maximal entropy is due to lemma 7.1.

Otherwise, $H(p) = H(p^*)$ which implies that $D(p||u) = D(p^*||u)$, which in turns means that $D(p||p^*) = 0$, and $p = p^*$.

■

9 Relations to Maximum Likelihood

We have shown that the maximum entropy model subject to the marginal constrains is of the form

$$p^*(x) = k\Pi_{\chi_i \in \mathcal{X}} \alpha_i^{\chi_i(x)}, 0 < \alpha_i < \infty.$$

Now, consider the sample S and the empirical distribution $\tilde{p}$ determined based on it. If we would like to adopt the maximum likelihood approach, our goal would be to maximize the likelihood of the data. Let p be any probability distribution; then we would like to maximize

$$L(p) = \Pi_{x \in S} p(x).$$

Or, equivalently,

$$LL(p) = \sum_{x \in S} \log p(x) = \sum_{x \in X} \tilde{p}(x) \log p(x).$$

Assume that we are not looking for the global ML distribution (that satisfies the constraints).; instead, let's look for the most likely distribution in the space of those that have an exponential form. The most likely, turns out to be also the max entropy.

Theorem 9.1 *Let $\mathcal{P}, \mathcal{Q}, LL(p)$ be as defined above. If $p^* \in \mathcal{P} \cap \mathcal{Q}$ then,*

$$p^* = argmax_{q \in \mathcal{P}} LL(q).$$

Furthermore, p^* is unique.

Interpretation: *If you assume an exponential form, searching for the most likely distribution will also give you the maximum entropy distribution.*

Proof: Assume that $\tilde{p}$ is the observed distribution of x in the sample S for $x \in X$. Clearly, $\tilde{p} \in \mathcal{P}$. Let $q \in \mathcal{Q}$ and $p^* \in \mathcal{P} \cap \mathcal{Q}$. Therefore, by lemma 7.2

$$D(\tilde{p}||q) = D(\tilde{p}||p^*) + D(p^*||q)$$

and by lemma 7.1

$$D(\tilde{p}||q) \geq D(\tilde{p}||p^*)$$

which means

$$-H(\tilde{p}) - \sum_{x \in X} \tilde{p}(x) \log q(x) \geq -H(\tilde{p}) - \sum_{x \in X} \tilde{p}(x) \log p^*(x)$$

and therefore

$$H(\tilde{p}) + LL(q) \leq H(\tilde{p}) + LL(p^*).$$
$$LL(q) \leq LL(p^*).$$

This shows that the likelihood of the data according to all distributions $p^* \in \mathcal{P} \cap \mathcal{Q}$ is not smaller than any other distribution in $\mathcal{Q}$. The fact that actually the equality never holds is due to lemma 7.1.

Otherwise, $LL(q) = LL(p^*)$ which implies that $D(\tilde{p}||q) = D(p^*||q)$, which in turns means that $D(p^*||q) = 0$, and $p^* = q$. ■

Notice that this *does not mean* that the ML model is the same as the ME model. It just mean that the ML model of *this* form is the ME model. It could be, however, that the real ML model has a different form.

10 Algorithmic Issues

We have proved the **Duality theorem:**

Consider the class of constrained distributions:

$$\mathcal{P} = \{p| \ \ \forall \chi \in \mathcal{X} : E_p \chi = \sum_{y,x_i} p(y|x_i)\chi(x_i, y) \equiv \sum_{y,x_i} \chi(x,y) = E_{\tilde{p}}\chi\}$$

and the Class of Max Entropy Distributions:

$$\mathcal{Q} = \{p|p(y|x) = \frac{\exp\{\sum_{\chi_i \in \mathcal{X}} w_i \chi_i(x,y)\}}{\sum_y \exp\{\sum_{\chi_i \in \mathcal{X}} w_i \chi_i(x,y)\}}\}$$

Here we can think of the distribution p as a vector of size $|X| \times |Y|$.

We proved that there is a unique distribution $p^* \in \mathcal{P} \cap \mathcal{Q}$ so that

$$p^* = argmax_{p \in \mathcal{P}} LL(p),$$

and

$$p^* = argmax_{p \in \mathcal{P}} H(p).$$

The remaining question now is then how to find the coefficients of the Maximum Entropy distribution, given the data sample S. These coefficients can be viewed as the Lagrange multipliers of the χs in some optimization problem, although we will not pursue this view here.

The algorithmic direction is based on the last Theorem. As shown above we are looking for the ML solution under the assumption that the distribution is an element in the class $\mathcal{Q}$.

There is *no closed form solution* to this problem. (On the other hand, there is a closed form solution to the ML problem under the product distribution assumption - the naive Bayes solution).

We wrote the likelihood as:

$$LL(p) = \sum_{x \in S} \log p(x) = \sum_{x \in X} \tilde{p}(x) \log p(x),$$

so conceptually, we only need to maximize this expression. That is, to find the coefficients α_i of the exponential family distribution p, with

$$\log p(x) = k + \sum_{\chi_i \in \mathcal{X}} \log \alpha_i \chi_i(x),$$

that maximize the likelihood.

In order to see how this will eventually yield an algorithmic approach, let's recall what we said for the naive Bayes model. There, we had a model that made decisions as follows:

Predict $c = 1$ if and only if

$$\frac{P(c=1)\Pi_i P(x_i|c=1)}{P(c=0)\Pi_i P(x_i|c=0)} > 1$$

.

Denoting: $p_i = P(x_i = 1|c = 1), q_i = P(x_i = 1|c = 0)$, we have:

Predict $c = 1$ if and only if

$$\frac{P(c=1)\Pi_i p_i^{x_i}(1-p_i)^{1-x_i}}{P(c=0)\Pi_i q_i^{x_i}(1-q_i)^{1-x_i}} > 1.$$

For the case of two classes we got that the optimal Bayes behavior is given by a linear separator:

$$\log\frac{P(c=1)}{P(c=0)} + \sum_i \log\frac{1-p_i}{1-q_i} + \sum_i (\log\frac{p_i}{1-p_i} - \log\frac{q_i}{1-q_i})x_i > 0,$$

That is

$$\sum_i w_i x_i + b \equiv w \cdot x + b > 0.$$

We then went on to compute the *posterior probabilities* are given by the logistic function of a linear function of the features:

$$P(c=1|x) = \frac{1}{1+\exp\{-w\cdot x+b\}}.$$

As a side note, in the more general case of k class labels, we get that the *posterior probabilities* are given by the slightly more complicated term, the *softmax function* of a linear combination of the features:

$$P(c^i=1|x) = \frac{\exp\{w^i\cdot x+b^i\}}{\sum_i \exp\{w^i\cdot x+b^i\}}.$$

Exactly the same thing holds for probability distributions in the exponential family. We can always write the posterior as a logistic function (or a softmax function). The optimization problem that we will solve will therefor become a logistic regression problem.

The algorithmic approaches become now different methods that attempt to optimize the vector of coefficient in order to maximize the log likelihood. The **conditional exponential model** (also called: log linear, maximum entropy) is:

$$P(y^i|\chi) = P(y^i|\chi, w) = \frac{\exp\{w \cdot \chi\}}{\sum_i \exp\{w \cdot \chi\}} = \frac{\exp\{z^i\}}{\sum_i \exp\{z^i\}},$$

where $z^i = w \cdot \chi(x, y_i)$, with w denoting the weight vector and χ denoting the feature vector representation of the example.

Note that now we think of w as a longer vector, of size $|X| \times |Y|$ and hide the dependence on the value of y in the feature functions.

The goal is to choose the parameters w such that the conditional likelihood of the data given this model is maximized.

Note again that this is the same problem that we have been discussing over and over in this part of the class.

There are many other ways to choose the vector w.

- Here, and in the NB way, we are doing it by going the ML way.
  - For NB - the weights that provide the maximize the likelihood can be computed in a closed form, given some independence assumptions on the
  - Here, we will have to use search techniques to find w
- Perceptron, Winnow, SVM, are driven by an explicit loss function (error).

We can write the conditional log likelihood of the data as:

$$L(Y|S, w) = \sum_{(y,\chi)} \log P(y|\chi, w) = \sum_{x \in S} z^i - \sum_{x \in S} \log \{\sum_i \exp\{z^i\}\}, \quad (4)$$

where the first terms is the empirical counts, and the second is the normalization. This is the reason these models are called **log-linear models**.

Now we need to compute the gradient of the log-likelihood with respect the parameters.

$$\begin{aligned}\frac{dL}{dw} &= \sum_{x\in S}\chi(x,y) - \sum_{x\in S}\frac{\sum_y \chi(x,y)\exp\{z^i\}}{\sum_y \exp\{z^i\}} = \\ &= \sum_{x\in S}\chi(x,y) - \sum_{x\in S}\sum_y \chi(x,y)\frac{\exp\{z^i\}}{\sum_y \exp\{z^i\}} = \\ &= \sum_{x\in S}\chi(x,y) - \sum_{x\in S}\sum_y \chi(x,y)p(y|x,w)\end{aligned}$$

Notice that the first term really represent the *empirical counts* while the second one represents the *expected counts* with respect to the true distribution p. So, we can write the gradient of the log-likelihhod with respect to w as

$$G(w) = \frac{dL}{dw} = E_{\tilde{p}}(\chi) - E_p(\chi). \tag{5}$$

The significance of this is two fold:

- This the first time we see in an explicit way a probabilistic methods that actually tries to fit the data. But, we don't yet have a generalization rational for it.
- Algorithmically, all methods are using expression 5 for the first derivative.

The likelihood function in Eq.4 is concave over the parameter space, it has a global maximum, where the gradient is zero. However, simply setting $\frac{dL}{dw} = 0$ and solving for w does not yield a closed form solution, so we need to proceed iteratively.

All parameter estimation algorithms the following general form:

- At each step, adjust an estimate of the parameters $w(k)$ to a new estimate $w(k+1)$ based on the divergence between the estimated probability distribution $w(k)$ and the empirical distribution $\tilde{p}$.
- Continue until successive improvements fail to yield a sufficiently large decrease in the divergence.

The methods for computing the updates at each search step differs substantially. As we shall see, this difference can have a dramatic impact on the number of updates required to reach convergence.

10.1 Iterative Scaling

Until recently, the most popular method for iteratively refining the model parameters is Generalized Iterative Scaling (GIS), due to Darroch and Ratcliff (1972).

GIS scales the probability distribution $p(k)$ by a factor proportional to the ratio of $E_p(\chi)$ to $E_{p(k)}(\chi)$ with the restriction that The GIS procedure requires the constrain that

$$\forall x \in X, \sum_{\chi \in \mathcal{X}} \chi(x) = C$$

for some constant C. (That is, the number of active features in each example is the same.) If this is not the case, a correction feature can be added.

Basically, that means that the update rule of GIS is:

$$w_j^{(n+1)} = w_j^{(n)} [\frac{E_{\tilde{p}} \chi_j}{E_{p^{(n)}} \chi}]^{1/C}$$

(Additively, to be consistent with other methods, you add the log of this term).

The step size, and thus the rate of convergence, depends on the constant C: the larger the value of C, the smaller the step size. In case not all rows of the training data sum to a constant, the addition of a correction feature effectively slows convergence to match the most difficult case.

There are several improved GIS methods that attempt to get around these difficulties.

10.2 First Order Methods

The most obvious way of making explicit use of the gradient is by the method of steepest ascent. The gradient of a function is a vector which points in the direction in which the functions value increases most rapidly. Since our goal is to maximize the log-likelihood function, a natural strategy is to shift our current estimate of the update rule:

$$\delta(k) = \alpha(k) G(k)),$$

where the step size $\alpha(k)$ is chosen to maximize $L(p(k) + \delta(k))$.

Finding the optimal step size is itself an optimization problem, though only in one dimension and, in practice, only an approximate solution is required to guarantee global convergence.

Since the log-likelihood function is concave, the method of steepest ascent is guaranteed to find the global maximum.

However, while the steps taken on each iteration are in a very narrow sense locally optimal, the global convergence rate of steepest ascent is very poor. Different methods of *line search* can be used to accelerate that.

Conjugate gradient methods are still first order methods that attempt to accelerate the line search by choosing a search direction which is a linear combination of the steepest ascent direction and the previous search direction. The step size is selected by an approximate line search, as in the steepest ascent method. While theoretically equivalent, they use slightly different update rules and thus show different, often better, numeric properties.

10.3 Second Order Methods

Another way of looking at the problem with steepest ascent is that while it takes into account the *gradient* of the log-likelihood function, it fails to take into account its *curvature*, or the gradient of the gradient.

The usefulness of the curvature is made clear if we consider a second-order Taylor series approximation of

$$L(w+\delta) : L(w+\delta) \approx L(w) + \delta G(w) + \frac{1}{2}\delta^2 H(w)\delta$$

) where H is the *Hessian matrix* of the log-likelihood function, the matrix of its second partial derivatives with respect to w.

Setting this to 0 and solving with respect to δ gives Newton's method:

$$\delta(k) = H^{-1}(w(k))G(w(k))$$

.

Newtons method converges very quickly but it requires the computation of the inverse of the Hessian matrix on each iteration. The evaluation of the Hessian matrix is computationally impractical, and Newtons method is not competitive with iterative scaling or first order methods.

Variable metric or quasi-Newton methods avoid explicit evaluation of the Hessian by building up an approximation of it using successive evaluations of the gradient. Basically, H^{-1} is replaced by a local approximation of it.

Variable metric methods also show excellent convergence properties and can be much more efficient than using true Newton updates, but for large scale problems with hundreds of thousands of parameters, even storing the approximate Hessian is prohibitively expensive.

For such cases, there are *limited memory variable metric methods*, which implicitly approximate the Hessian matrix using a lot less space.

For NLP applications these seem to be, as of today, the best ME methods around.

(Although, very few people use it).

The *Generalized Iterative Scaling* [DR72], or *GIS* is a procedure that finds the parameters $\{\alpha_1, \ldots \alpha_k\}$ of the unique distribution $p^* \in \mathcal{P} \cap \mathcal{Q}$.

The GIS procedure requires the constrain that

$$\forall x \in X, \sum_{\chi \in \mathcal{X}} \chi(x) = C$$

for some constant C. (That is, the number of active features in each example is the same.)

If this is not the case, choose C to be

$$C = max_{x \in X} \sum_{\chi \in \mathcal{X}} \chi(x)$$

and add a "correction" feature $\chi_l, l = k + 1$, such that

$$\forall x \in X, \chi_l(x) = C - \sum_{i=1}^{k} \chi_i(x).$$

(Note that unlike all other features, χ_l ranges in $[0, C]$, where C can be greater than 1. Furthermore, that GIS assumes that at least one feature is active in all examples, that is,

$$\forall x \in X, \exists \chi \in \mathcal{X}, \chi(x) = 1.$$

Theorem 10.1 *The following procedure will converge to $p^* \in \mathcal{P} \cap \mathcal{Q}$.*

1. $\alpha_j^{(0)} = 1$
2. $\alpha_j^{(n+1)} = \alpha_j^{(n)} [\frac{E_{\tilde{p}} \chi_j}{E_{p^{(n)}} \chi}]^{1/C}$

where

$$E_{p^{(n)}} \chi = \sum_{x \in X} p^{(n)}(x) \chi(x),$$

is the expectation of χ_j according to the distribution $p^{(n)}$, defined by

$$p^{(n)}(x) = \pi \Pi_{j=1}^{l} (\alpha_j^{(n)})^{\chi(x)}.$$

It can be shown [DR72] that this algorithm convergences to the Maximum Entropy distribution (not *local* maximum as happens for other algorithms). Also, the likelihood of the distributions considered in this process are non-decreasing, that is

$$D(\tilde{p} || p^{(n)} 1) \leq D(\tilde{p} || p^{(n)}).$$

The *Improved Iterative Scaling* algorithm [BPP96] finds p^* without the use of the correction feature.

10.4 Details of the Computation

Each iteration of the GIS requires the quantities $E_{\tilde{p}}\chi$ and $E_p\chi$. The first is straight forward given the training sample $S, |S| = N$:

$$E_{\tilde{p}}\chi = \frac{1}{N}\sum_{i=1}^{N}\chi(x_i).$$

(Notice that N is the number of examples, that is, event tokens rather than types, in the data). However, the computation of the model's features expectation,

$$E_{p^{(n)}}\chi = \sum_{x\in X} p^{(n)}(x)\chi(x)$$

in a model with k, potentially overlapping, features require touching 2^k events. That is, the distribution $p^{(n)}$ can be viewed as defined over 2^k regions, defined by the subset of the features active in it. For each of these regions we can compute $p^{(n)}$ and then compute the distribution.

Instead, we can approximate over the data we have, assuming that $p(x', c) = p(c|x')\tilde{p}(x')$.

$$\begin{aligned} E_{p^{(n)}}\chi &= \sum_{x\in X} p^{(n)}(x)\chi(x) && (6)\\ &= \sum_{(x',c)\in X} p^{(n)}(c|x')p(x')\chi(x',c) && (7)\\ &\approx \sum_{(x',c)\in X} p^{(n)}(c|x')\tilde{p}(x')\chi(x',c) && (8)\\ &= \sum_{i=1}^{N}\tilde{p}(x')\sum_{c\in C} p^{(n)}(c|x')\chi(x',c) && (9) \end{aligned}$$

which sums only over contexts that we actually see in S.

In practice, one should terminate this procedure after some fixed number of iteration, of when the improvement in the log-likelihood is negligible. The running time is dominated by the estimation of the expectation at each iteration, which is $O(NCA)$, where N is the training set size, C is the number of classes and A is the average number of features active in each example. (This is due to the need to compute the constant π, in the computation $p(c|x') = p(c, x')/p(x')$.

11 Related Reading

Several works on the theory of Maximum Entropy and it's use in NLP are listed below. Also listed are several applications papers that cover a variety of applications.

References

[BPP96] Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[DR72] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

[Goo63] I.J. Good. Maximum entropy for hypothesis formulation, especially for multidimensional contingency tables. *Annals of Mathematical Statistics*, 34:911–934, 1963.

[Jay68] E. T. Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, SSC-4(3):227–241, September 1968.