EÖTVÖS LORÁND UNIVERSITY

Biszak Előd

Maximum Entropy Models

Supervisor: Lukács András

Faculty of Science
MSc in Mathematics

May 2013

Contents

List of Figures

List of Algorithms

1 About Entropy

In this chapter we will introduce the concept of entropy and show some of it's basic characteristics. For a X discrete random variable that takes values from a finite set $\mathcal{X}$, $\triangle$ will denote the set of all possible distributions of X, that is

$$\triangle_X = \{\, p : \mathcal{X} \to [0,1] \mid \sum_{x \in \mathcal{X}} p(x) = 1 \,\}.$$

$supp(p)$ denotes the support of $p \in \triangle_X$, that is

$$supp(p) = \{\, x \in \mathcal{X} \mid p(x) > 0 \,\}.$$

For two discrete random variables X, Y with possible values from $\mathcal{X}$ and $\mathcal{Y}$ respectively we will denote the set of all possible conditional distributions $p(y|x)$ by $\triangle_{Y|X}$.

$$\triangle_{Y|X} = \{\, p : \mathcal{X} \times \mathcal{Y} \to [0,1] \mid \sum_{y \in \mathcal{Y}} p(y|x) = 1, \forall x \in \mathcal{X} \,\}$$

For a conditional distribution $p \in \triangle_{Y|X}$ p_x denotes the distribution of Y given $x \in \mathcal{X}$. For technical reasons we will define $0 \cdot \log 0 = 0$, which is consistent with the limit: $\lim_{x \to 0} x \log x = 0$.

1.1 Entropy

Definition 1.1. *Given a random variable X that takes values from a finite set $\mathcal{X}$ we define the entropy of a distribution $p \in \triangle$ as follows*

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

where log *denotes the natural logarithm.*

The entropy is mainly defined with logarithm of base 2, but for technical reasons we will use the natural logarithm. Note that they differ only by linear factor.

1.1.1 Basic characteristics

Now we show some basic characteristics of entropy that will be useful in the following chapters.

Lemma 1.1. *$H(p) \geq 0$ for any $p \in \triangle$.*

Proof. Since $0 \leq p(x) \leq 1$, $-\log p(x) \geq 0$ for all $x \in \mathcal{X}$. □

Lemma 1.2. *$H(p) = 0$ if and only if $|supp(p)| = 1$.*

Proof. Since $-x \log x > 0$ for any $x \in (0, 1)$ the possible values of p are 0 and 1. □

Lemma 1.3. *For any $p \in \triangle$*

$$H(p) \leq \log |\mathcal{X}|$$

with equality if and only if p is the uniform distribution, i.e., $p(x) = 1/|\mathcal{X}|$ for all $x \in \mathcal{X}$.

Proof. Being $\sum_{x \in \mathcal{X}} p(x) = 1$ and $\log(x)$ convex, according to the Jensen's inequality:

$$H(p) = \sum_{x \in supp(p)} p(x) \log \left(\frac{1}{p(x)} \right) \leq \log \left(\sum_{x \in supp(p)} p(x) \cdot \frac{1}{p(x)} \right) = \log |supp(p)| \leq \log |\mathcal{X}|,$$

and equality holds if and only if $|supp(p)| = |\mathcal{X}|$ and $p(x_1) = p(x_2)$ for all $x_1, x_2 \in \mathcal{X}$, that is if $p(x) = \frac{1}{|\mathcal{X}|}$ for every $x \in \mathcal{X}$. □

Lemma 1.4. *$H(p_1, p_2, \ldots, p_n) = H(p_2, p_1, \ldots, p_n)$, where $H(p_1, \ldots, p_n) = -\sum_{i=1}^{n} p_i \log p_i$. That is H is symmetric.*

Lemma 1.5. *$H(p) = -\sum_{i=1}^{n} p_i \log p_i$ is continuously differentiable and strictly concave on $\mathbb{R}_+^n$.*

Proof. $H(p)$ is clearly continuously differentiable on $\mathbb{R}_+^n$. We focus on the concavity. Suppose that $p, q \in \mathbb{R}_+^n$ and λ such that $p + \lambda q \in \mathbb{R}_+^n$. Then

$$H(p + \lambda q) = -\sum_{i=1}^{n} (p_i + \lambda q_i) \log (p_i + \lambda q_i).$$

The derivatives with respect to λ are

$$\frac{dH}{d\lambda} = -\sum_{i=1}^{n} q_i \log\left(p_i + \lambda q_i\right) + q_i, \qquad \frac{d^2H}{d\lambda^2} = -\sum_{i=1}^{n} \frac{q_i^2}{p_i + \lambda q_i}.$$

The latter is strictly negative. □

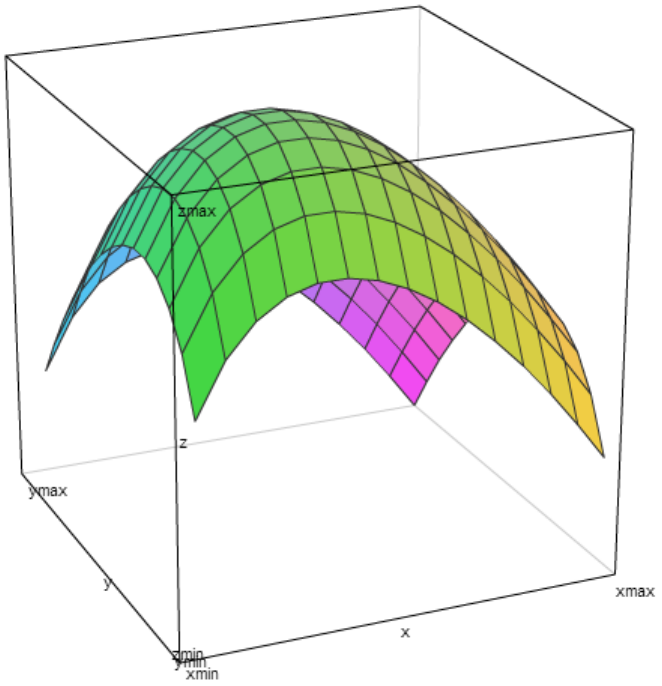

FIGURE 1.1: The entropy function in two dimensions. $H(x, y) = -(x \log x + y \log y)$, $(x, y, z) \in [0, 1] \times [0, 1] \times [0, 1]$

1.2 Conditional entropy

Definition 1.2. *Given two discrete random variable Y and X, that take values from $\mathcal{X}$ and $\mathcal{Y}$ respectively. Let q be the distribution of X. We define the conditional entropy of*

a conditional distribution $p \in \triangle_{Y|X}$, as the expectation of $H(p_x)$. Namely

$$\begin{aligned} H(p) &= \sum_{x\in\chi} q(x)H(p_x) \\ &= \sum_{x\in\mathcal{X}} q(x)\left(-\sum_{y\in\mathcal{Y}} p(y|x)\log p(y|x)\right) \\ &= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} q(x)p(y|x)\log p(y|x) \\ &= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log p(y|x) \end{aligned}$$

where $p(x,y) = q(x)p(y|x)$.

1.2.1 Basic characteristics

Lemma 1.6. *$H(p) \geq 0$ for all $p \in \triangle_{Y|X}$.*

Proof. Since $H(p_x) \geq 0$ for all $x \in \mathcal{X}$, $\sum_{x\in\mathcal{X}} q(x)H(p_x) \geq 0$. □

Lemma 1.7. *$H(p) = 0$ for some $p \in \triangle Y|X$ if and only if the value of Y is completely determined by the value of X.*

Proof. It is equivalent to being $H(p_x) = 0$ for all $x \in \mathcal{X}$, that is $|supp(p_x)| = 1$. □

Lemma 1.8. *For any $p \in \triangle_{Y|X}$*

$$H(p) \leq \log|\mathcal{Y}|$$

with exact equality if and only if p_x is the uniform distribution for all $x \in \mathcal{X}$.

Proof.

$$H(p) = \sum_{x\in\mathcal{X}} q(x)H(p_x) \leq \sum_{x\in\mathcal{X}} p(x)\log|Y| = \log|Y|,$$

according to Lemma 1.3, and equality holds if and only if $H(p_x) = \log|Y|$ for every $x \in \mathcal{X}$. □

Lemma 1.9. *Holding the distribution of X, q fixed the conditional entropy is concave and continuously differentiable in $\mathbb{R}^n_+$, where $n = |\mathcal{X}||\mathcal{Y}|$.*

Proof. As we saw in the previous section the function $f(x_1, \ldots, x_n) = -\sum_{i=1}^{n} x_i \log x_i$ is strictly concave and continuously differentiable on $\mathbb{R}^n_+$. For fixed q

$$H(p) = -\sum_{x \in \mathcal{X}} q(x) \sum_{y \in \mathcal{Y}} p(y|x) \log p(y|x)$$

the conditional entropy is a convex combination of such functions. □

2 | Maximum Entropy Principle

The characteristics of entropy seen in the previous chapter gives us a reasonable choice for measure the uniformity of a distribution by it's entropy. That is the higher the entropy the higher the uniformity. The maximum entropy principle consits of selecting the most uniform distribution of a set of possible distributions, that is the one with maximum entropy. In this chapter we will discuss how can we use the maximum entropy principle to model a random process given a large number of samples. Suppose that we are given a random process that produces an output x that can take values from a finite set $\mathcal{X}$. We would like to model this process given a set of observed values $O = \{ x_i \mid i = 1, \ldots, N \}$. In practical tasks that use maximum entropy typically a particular $x \in \mathcal{X}$ will either not occur at all in the sample or only occur a few times at most. We will use the term model for a distribution p on $\mathcal{X}$. We will denote the set of all possible distributions on $\mathcal{X}$ by $\triangle$ meaning

$$\triangle = \{ p : \mathcal{X} \to \mathbb{R}^+ \mid \sum_{x \in \mathcal{X}} p(x) = 1 \}.$$

We will denote the empirical distribution of the training sample by $\tilde{p}$, namely

$$\tilde{p}(x) = \frac{\sum_{i=1}^{N} \chi_x (x_i)}{N} \quad \text{for} \quad x \in \mathcal{X},$$

where $\chi_{\bar{x}}$ is the indicator function of $\bar{x}$ defined as

$$\chi_{\bar{x}}(x) = \begin{cases} 1 & \text{if } x = \bar{x} \\ 0 & \text{otherwise.} \end{cases}$$

We will introduce the concept of *feature* functions. A feature function is a non-negative valued function on $\mathcal{X}$. We denote the expected value of a feature function f respect to the distribution p by $E_p [f]$, namely

$$E_p [f] = \sum_{x \in \mathcal{X}} f(x) p(x).$$

We will have a set of feature functions $\{ f_i \mid i = 1, \ldots, n \}$ and we will force our model to accord with the corresponding statistics what we get from the training sample. That is we would like a distribution p which is in the subset C of $\triangle$ defined by

$$C = \{ p \in P \mid E_p [f_i] = E_{\tilde{p}} [f_i] \text{ for } i = 1, \ldots, n \}.$$

We will use the term *constraints* for these equations. The principle of maximum entropy dictates that we select the distribution p^* of C that maximizes the entropy. Using the notation and formula of entropy defined in the previous section the optimization problem we are left to solve is

$$p^* = \underset{p \in C}{argmax} \left(- \sum_{x \in \mathcal{X}} p(x) \log p(x) \right) = \underset{p \in C}{argmax} H(p).$$

As we seen in the previous chapter the entropy is bounded from below by zero and from above by $\log |\mathcal{X}|$ with the uniform distribution on $\mathcal{X}$. So $H(p)$ is a continuous, strictly convex, bounded function in $\triangle$, furthermore C is bounded, closed, convex and non-empty subset of $\mathbb{R}^{|\mathcal{X}|}$ since $\tilde{p} \in C$. That is $H(p)$ reaches it's maximum in a unique $q^* \in C$.

2.1 Parametric form, Lagrange multipliers

We have the constraint optimization (*primal*) problem introduced in the previous section

$$p^* = \underset{p \in C}{argmax} H(p).$$

Using the Lagrange multiplicator method we obtain the following parametric form

$$\Lambda(p, \lambda) = H(p) + \sum_{i=1}^{n} \lambda_i (E_p [f_i] - E_{\tilde{p}} [f_i]) + \mu (\sum_{x \in \mathcal{X}} p(x) - 1)$$

We call the λ_i and the μ parameters Lagrange multipliers and the Λ the Lagrangian function. We denote by λ the vector containing μ and λ_i for $i = 1, \ldots, n$. Holding λ fixed we can compute the unconstrained maximum of the Lagrangian over all $p \in P$. We will denote this optimal distribution p_λ, namely

$$p_\lambda = \underset{p \in P}{argmax} \Lambda(p, \lambda).$$

Substituting p_λ to Λ we get a function that only depends on λ.

$$\Psi(\lambda) = \Lambda(p_\lambda, \lambda)$$

We call Ψ the *dual* function. From the theory of contstraint optimization we get that maximizing the dual function in λ yields a solution for the original primal problem. The method of Lagrange multipliers yields a necessary condition for constrained optimization problems, namely that if the $H(p)$ function has a constrained maximum in a then there exists an λ for which (a, λ) is a stationary point of the Lagrangian meaning that all the partial derivatives are zero. Furthermore according to the Kuhn-Tucker theorem, since $H(p)$ and the constraints are all continuously differentiable and $H(p)$ is concave and the constraints are linear such a stationary point identifies this optimal solution **uniquely**. So given the Lagrangian in the form

$$\begin{aligned}\Lambda(p,\lambda) &= H(p) + \sum_{i=1}^{n} \lambda_i (E_p[f_i] - E_{\tilde{p}}[f_i]) + \mu(\sum_{x\in\mathcal{X}} p(x) - 1) \\ &= -\sum_{x\in\mathcal{X}} p(x)\log p(x) + \sum_{i=1}^{n} \lambda_i \cdot \sum_{x\in\mathcal{X}} (p(x)f(x) - \tilde{p}(x)f(x)) + \mu\left(\sum_{x\in\mathcal{X}} p(x) - 1\right)\end{aligned}$$

Derivating the Lagrangian by $p(x)$

$$\frac{\partial\Lambda}{\partial p(x)} = -\log p(x) - 1 + \sum_{i=1}^{n} \lambda_i f_i(x) + \mu$$

And for a stationary point

$$-\log p(x) - 1 + \sum_{i=1}^{n} \lambda_i f_i(x) + \mu = 0 \qquad \forall x \in \mathcal{X},$$

solving the equaition we get

$$p(x) = \frac{1}{e^{1-\mu}} \cdot e^{\sum_{i=1}^{n} \lambda_i f_i(x)}.$$

Taking in mind that p is a distribution over $\mathcal{X}$, we can compute the exact value of the μ parameter. This parameter is responsable for the distribution to sum to 1 over all $x \in \mathcal{X}$.

$$\sum_{x\in\mathcal{X}} \frac{1}{e^{1-\mu}} \cdot e^{\sum_{i=1}^{n} \lambda_i f_i(x)} = 1$$

$$\mu = 1 - \log \sum_{x\in\mathcal{X}} e^{\sum_{i=1}^{n} \lambda_i f_i(x)}$$

So the p_λ distribution which maximizes Λ for some λ takes the form

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

where

$$Z_\lambda = \sum_{x \in \mathcal{X}} e^{\sum_{i=1}^{n} \lambda_i f_i(x)}.$$

Substituting this parametric form of the optimal distribution to the dual function Ψ we get

$$\Psi(\lambda) = -\sum_{x \in \mathcal{X}} p_\lambda(x) \log p_\lambda(x) + \sum_{i=1}^{n} \lambda_i \left(\sum_{x \in \mathcal{X}} p_\lambda(x) f_i(x) - \sum_{x \in \mathcal{X}} \tilde{p}(x) f_i(x) \right) + \mu \left(1 - \sum_{x \in \mathcal{X}} p_\lambda(x) \right)$$

that is

$$\Psi(\lambda) = -\log Z_\lambda + \sum_{i=1}^{n} \lambda_i E_{\tilde{p}} [f_i].$$

It yields to an unconstrained optimization problem which identifies uniquely our optimal solution. That is the most important practical consequence of the method of Lagrange multipliers.

2.2 Relation to Maximum Likelihood

We define the *log-likelihood* of the model p with respect to the empirical distribution $\tilde{p}$ by

$$L_{\tilde{p}}(p) = \log \prod_{x \in \mathcal{X}} p(x)^{\tilde{p}(x)} = \sum_{x \in \mathcal{X}} \tilde{p}(x) \log p(x).$$

In our case for a given $\lambda \in \mathbb{R}^n$

$$\begin{aligned} L_{\tilde{p}} p_\lambda &= \sum_{x \in \mathcal{X}} \tilde{p}(x) \log p_\lambda(x) \\ &= \sum_{x \in \mathcal{X}} \tilde{p}(x) \log \left(\frac{1}{Z_\lambda} e^{\sum_{i=1}^{n} \lambda_i f_i(x)} \right) \\ &= \sum_{x \in \mathcal{X}} \tilde{p}(x) \left(-\log Z_\lambda + \sum_{i=1}^{n} \lambda_i f_i(x) \right) \\ &= -\log Z_\lambda + \sum_{i=1}^{n} \lambda_i E_{\tilde{p}} [f_i] \\ &= \Psi(\lambda) \end{aligned}$$

Theorem 2.1. *Maximizing the log-likelihood of the training sample in the parametric family p_λ is equivalent to maximizing the entropy over C.*

2.3 Conditional distributions

There is another variation of the maximum entropy approach which is widely used in applications. Suppose that we are given two discrete random variables X and Y with support $\mathcal{X}$, $\mathcal{Y}$ respectively. We would like to model the conditional distribution $Y|X$ given a huge number of observed values $O = \{ (x_i, y_i) \mid i = 1, \ldots, N \}$. We define feature functions similarly to the ones in the previous section as non-negative valued functions on $\mathcal{X} \times \mathcal{Y}$. Since in practical tasks it is often the case that modeling the distribution of X and the joint distribution (X, Y) is untractable we define the *constraint* equations as follows.

$$\sum_{(x,y)\in O} \tilde{p}(x,y) f(x,y) = \sum_{(x,y)\in O} \tilde{p}(x) p(y|x) f(x,y),$$

where $\tilde{p}$ denotes the empirical distribution on the training data. We will force our model to satisfy these constraints. According to the maximum entropy principle we want to select the model which is most uniform that is the one with maximum **conditional entropy**. As we saw in the previous chapter it is a reasonable choice for measuring uniformity of conditional distributions.

$$H(p) = -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p(x,y) \log p(y|x) \approx -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \tilde{p}(x) p(y|x) \log p(y|x)$$

If we do the straightforward calculations as we did in the previous section we get the following parametric form of the optimal distribution

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} e^{\sum_{i=1}^n \lambda_i f_i(x,y)}$$

where

$$Z_\lambda(x) = \sum_{y\in\mathcal{Y}} e^{\sum_{i=1}^n \lambda_i f_i(x,y)}.$$

The corresponding dual function of λ will be

$$\Psi(\lambda) = -\sum_{x\in\mathcal{X}} \tilde{p}(x) \log Z_\lambda(x) + \sum_{i=1}^n E_{\tilde{p}}\left[f_i\right].$$

Note that the very same proposition holds for conditional ditributions as in the previous section.

$$L_{\tilde{p}}(p) = \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \tilde{p}(x,y) \log p(y|x)$$

Which for the exponential distributions p_λ

$$\begin{aligned}
L_{\tilde{p}}(p) &= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \tilde{p}(x,y)\log\left(\frac{1}{Z_\lambda(x)}e^{\sum_{i=1}^n \lambda_i f_i(x,y)}\right) \\
&= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \tilde{p}(x,y)\left(-\log Z_\lambda(x) + \sum_{i=1}^n \lambda_i f_i(x,y)\right) \\
&= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} \tilde{p}(x,y)\log Z_\lambda(x) + \sum_{i=1}^n \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \tilde{p}(x,y)\lambda_i f_i(x,y) \\
&= -\sum_{x\in\mathcal{X}} \tilde{p}(x)\log Z_\lambda(x)\sum_{y\in\mathcal{Y}} \tilde{p}(y|x) + \sum_{i=1}^n E_{\tilde{p}}\left[f_i\right] \\
&= -\sum_{x\in\mathcal{X}} \tilde{p}(x)\log Z_\lambda(x) + \sum_{i=1}^n E_{\tilde{p}}\left[f_i\right] \\
&= \Psi(\lambda)
\end{aligned}$$

3 Reformulation, the Kullback-Leibler divergence

In the previous chapter we saw the existence and unicity of the maximum entropy model as a consequence of the Kuhn-Tucker theorem. In this chapter we will introduce a self-contained proof of a generalization of the model [1]. Before the generalization some concepts need to be learnt about.

3.1 Kullback-Leibler divergence

Let $\mathcal{X}$ be a finite set. We denote the set of all possible probability distributions on $\mathcal{X}$ by $\triangle$ as in the previous chapter, that is

$$\triangle = \{\, p : \mathcal{X} \to [0,1] \mid \sum_{x \in \mathcal{X}} p(x) = 1 \,\}$$

Definition 3.1. *Suppose we have two probabilty distribution $p, q \in \triangle$ such that whenever $q(x) = 0$ for some $x \in \mathcal{X}$ $p(x) = 0$ holds (absolute continuity denoted by $p << q$). We define the Kullback-Leibler divergence of such distributions as follows*

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

In a sense that if $0 \cdot \log 0$ or $0 \cdot \log \frac{0}{0}$ appears in the formula it is represented by zero. In the case of $q(x) = 0$ and $p(x) \neq 0$ for some x we define the KL-divergence ∞.

Lemma 3.1. *The following lemmas demonstrate some of the basic properties of the Kullback-Leibler divergence.*

1. $D(p||q) \geq 0$, $\forall p, q \in \triangle$
2. $D(p||q) = 0$ *if and only if* $p = q$

3. *$D(p||q)$ is continuously differentiable in q*

4. *$D(p||q)$ is strictly convex both in p and q*

Proof. We concentrate on the first two statements.

$$\begin{aligned} -D(p||q) &= -\sum_{x\in supp(p)} \log\frac{p(x)}{q(x)} = \sum_{x\in supp(p)} p(x)\log\frac{q(x)}{p(x)} \\ &\leq \log\sum_{x\in supp(p)} p(x)\frac{q(x)}{p(x)} = \log\sum_{x\in supp(p)} q(x) \\ &\leq \log\sum_{x\in\mathcal{X}} = \log 1 = 0 \end{aligned}$$

Since $\log t$ is a strictly concave function of t, we have equality if and only if $\frac{q(x)}{p(x)}$ is constant everywhere, i.e. $p(x) = q(x)$ for all $x \in \mathcal{X}$. □

Lemma 3.2. *Suppose that $u \in \triangle$ is the uniform and $p \in \triangle$ is an arbitrary probability distribution on $\mathcal{X}$. Then*

$$D(p||u) = -H(p) + \log|\mathcal{X}|$$

Proof. It is consequence of simple calculation, that is

$$\sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{u(x)} = \sum_{x\in\mathcal{X}} p(x)\log p(x) + \log|\mathcal{X}|\sum_{x\in\mathcal{X}} p(x) = -H(p) + \log|\mathcal{X}|$$

□

As a consequence of Lemma 3.2 we can see that maximizing the entropy on a subset $\bar{\triangle}$ of $\triangle$ is equivalent to minimizing the Kullback-Leibler divergence respect to the uniform probability distribution of $\mathcal{X}$.

Lemma 3.3. *Suppose $q \in \triangle$ is some arbitrary probability distribution and $\tilde{p}$ is the empirical distribution of some training data $x_1, x_2, \ldots x_N$. Then the log-likelihood of q respect to the training data:*

$$L_{\tilde{p}}(q) = \sum_{x\in\mathcal{X}} \tilde{p}(x)\log q(x) = -D(\tilde{p}||q) - H(\tilde{p})$$

Proof. It is a consequence of simple calculation, namely

$$\begin{aligned} -D(\tilde{p}||q) &= -\sum_{x\in\mathcal{X}} \tilde{p}(x) \log \frac{\tilde{p}(x)}{q(x)} = -\sum_{x\in\mathcal{X}} \tilde{p}(x) \left(\log \tilde{p}(x) - \log q(x)\right) \\ &= \underbrace{\sum_{x\in\mathcal{X}} \tilde{p}(x) \log q(x)}_{L_{\tilde{p}}(q)} \underbrace{- \sum_{x\in\mathcal{X}} \tilde{p}(x) \log \tilde{p}(x)}_{H(\tilde{p})} \end{aligned}$$

□

That is maximizing the log-likelihood in some subset $\bar{\triangle}$ of $\triangle$ for some training data is equivalent to minimizing the Kullback-Leibler divergence to the empirical distribution of this training data.

3.2 Generalized Gibbs distribution

Definition 3.2. *For a function $h : \mathcal{X} \to \mathbb{R}$ and a distribution $p \in \triangle$ we define the generalized Gibbs distribution [1].*

$$(h \circ p)(x) = \frac{1}{Z_p(h)} e^{h(x)} q(x)$$

The $Z_p(h)$ parameter is just a normalizing factor that ensures for p_h to sum to 1 over $x \in \mathcal{X}$. It can be written as:

$$Z_p(h) = \sum_{x\in\mathcal{X}} e^{h(x)} p(x) = E_p[e^h]$$

We will use only some special functions for the generalized Gibbs distribution. Let $f = (f_1, f_2, \dots f_n)$ be a set of feature functions on $\mathcal{X}$ as defined in the previous chapter. For any $\lambda \in \mathbb{R}^n$ we define

$$(\lambda \circ p)(x) = \left((\lambda \cdot f) \circ p\right)(x) = \frac{1}{Z_p(\lambda \cdot f)} e^{\sum_{i=1}^n \lambda_i f_i(x)}.$$

Lemma 3.4. *The following lemmas show some basic characteristics of the generalized Gibbs distribution*

1. *The function $(\lambda, p) \to \lambda \circ p$ is smooth in $(\lambda, p) \in \mathbb{R}^n \times \triangle$*
2. *The derivative of $D(q||\lambda \circ p)$ with respect to λ is*

$$\frac{d}{dt}|_{t=0} D(p||(t\lambda) \circ q) = \lambda \cdot (E_p[f] - E_q[f])$$

Note that with a special choice of $p \in \triangle$, namely the uniform distribution we get the following formula

$$(\lambda \circ p) = \frac{1}{Z_p(\lambda \cdot f)} e^{\sum_{i=1}^{n} \lambda_i f_i}$$

which is the same formula that appears in Section 2.1.

3.3 Reformulation

Now we can get to the generalization of the maximum entropy principle. We define two sets of distributions on $\mathcal{X}$. Let $f = (f_1, f_2, \dots f_n)$ be a set of features functions, $q_0 \in \triangle$ an arbitrary distribution, $\tilde{p}$ a reference distribution.

$$P(f, \tilde{p}) = \{\, p \in \triangle : E_p[f] = E_{\tilde{p}}[f] \,\}$$

$$Q(f, q_0) = \{\, (\lambda \cdot f) \circ q_0 : \lambda \in \mathbb{R}^n \,\}$$

We let $\bar{Q}(f, q_0)$ denote the closure of $Q(f, q_0)$ in $\triangle$ with respect to the topology it inherits as a subset of $\mathbb{R}^{|\mathcal{X}|-1}$.There are some basic characteristics that we will use in the following: $q_0 \in Q(f, q_0)$, $\tilde{p} \in P(f, \tilde{p})$. There are two other distributions with high significance.

$$\underset{q \in \bar{Q}(f, q_0)}{argmin}\ D(\tilde{p}||q)$$

$$\underset{q \in P(f, \tilde{p})}{argmin}\ D(q||q_0)$$

As a consequence of Lemma 3.3 we see that the first choice of distribution is equivalent to maximizing the log-likelihood on $\bar{Q}(f, q_0)$. Similarly Lemma 3.2 yields that with a special choice of q_0, namely the uniform distribution the second minimalization is equivalent to maximizing the entropy on $P(f, \tilde{p})$.

3.4 Existence and unicity

In this section we will show a self-contained proof on the existence and unicity of the maximum entropy distribution.

Theorem 3.5. *Suppose that we are given two probability distributions on $\mathcal{X}$ $\tilde{p}, q_0 \in \triangle$ such that $\tilde{p} << q_0$. Then there exist a unique $q^* \in \triangle$ satisfying:*

1. $q^* \in P \cap \bar{Q}$
2. $D(p||q) = D(p||q^*) + D(q^*||q)$, $\forall p \in P$, $\forall q \in \bar{Q}$

3. $q^* = \underset{q \in \bar{Q}}{argmin}\, D(\tilde{p}||q)$

4. $q^* = \underset{p \in \tilde{p}}{argmin}\, D(p||q_0)$.

Moreover, any of the four properties defines q^* uniquely.

Lemma 3.6. *Given $\tilde{p}, q_0 \in \triangle$, $\tilde{p} << q_0$ then $P \cap \bar{Q} \neq \emptyset$.*

Proof. Let q^* be $argmin_{q \in \bar{Q}} D(p||q)$. To see that it exists uniquely, note that $\tilde{p} << q_0$ so $D(p||q)$ is not identically ∞ on $\bar{Q}$. Also, the KL-divergence is strictly convex with respect to q as we saw in Lemma 3.1. Since $\bar{Q}$ is closed it determines q^* uniquely. We have to prove that $q^* \in P$. Since for any $\lambda \in \mathbb{R}^n$ $\lambda \circ q^*$ is in $\bar{Q}$ and the definition of p^* $\lambda \to D(\tilde{p}, \lambda \circ q^*)$ reaches it's minimum in $\lambda = 0$. Hence the derivative with respect to λ disappears in 0. So using the formula of the derivative from Lemma 3.4 follows $E_{q^*}[f] = E_{\tilde{p}}[f]$, meaning that $q^* \in P$. □

Lemma 3.7. *Given $q^* \in P \cap \bar{Q}$, $p \in P$ and $q \in \bar{Q}$ then*

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

Proof. Let $p \in P(f, \tilde{p})$ and $q \in \bar{Q}(f, q_0)$ such that $q = (\lambda \cdot f) \circ q_0$ for some $\lambda \in \mathbb{R}$. Some straightforward calculation yields

$$\begin{aligned}
&D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = \\
&\sum_{x \in \mathcal{X}} p(x) \log p(x) - \sum_{x \in \mathcal{X}} p(x) \log \left(\frac{1}{Z_{q_0}(\lambda \cdot f)} e^{\lambda \cdot f} q_0(x) \right) = \\
&\sum_{x \in \mathcal{X}} p(x) \log p(x) + \log Z_{q_0}(\lambda \cdot f) \sum_{x \in \mathcal{X}} p(x) - \sum_{x \in \mathcal{X}} p(x) \log q_0(x) - \underbrace{\sum_{x \in \mathcal{X}} p(x)(\lambda \cdot f)}_{\lambda \cdot E_p[f]}
\end{aligned}$$

Let p_1, p_2 be two arbitrary distributions from $P(f, \tilde{p})$ and q_1, q_2 from $\bar{Q}(f, q_0)$ for some $\lambda_1, \lambda_2 \in \mathbb{R}^n$ respectively. It follows that

$$\begin{aligned}
& (D(p_1||q_1) - D(p_1||q_2)) - (D(p_2||q_1) - D(p_2||q_2)) = \\
= \;& \lambda_2 \cdot E_{p_1}[f] - \log Z_{q_0}(\lambda_2 \cdot f) - \lambda_1 \cdot E_{p_1}[f] + \log Z_{q_0}(\lambda_1 \cdot f) - \\
& (\lambda_2 \cdot E_{p_2}[f] - \log Z_{q_0}(\lambda_2 \cdot f) - \lambda_1 \cdot E_{p_2}[f] + \log Z_{q_0}(\lambda_1 \cdot f)) \\
= \;& (\lambda_2 - \lambda_1) \cdot (E_{p_1}[f] - E_{p_2}[f]) \\
= \;& (\lambda_2 - \lambda_1) \cdot (E_{\tilde{p}}[f] - E_{\tilde{p}}[f]) = 0
\end{aligned}$$

The lemma follows by taking $p_1 = q_1 = q^*$. □

Proof. Now we get to the proof of Theorem 3.4. Let q^* be any distribution from $P(f,\tilde{p})\cap \bar{Q}(f,q_0)$. Such q^* exists by Lemma 3.6. It satisfies Property 2 by Lemma 3.7. Let q be an arbitrary distribution in $\bar{Q}(f,q_0)$. It follows that

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

by Property 2 and Lemma 3.1. Similarly let p be an arbitrary distribution in $P(f,\tilde{p})$ then

$$D(p,q_0) = D(p||q^*) + D(q^*||q_0) \geq D(q^*||q_0).$$

It remains to prove that every of these properties defines q^* uniquely. Suppose that $q^* \in P \cap \bar{Q}$ and q' satisfies

1. Property 1. Then by Lemma 3.7 $0 = D(q'||q') = D(q'||q^*) + D(q^*,q')$, hence $D(q'||q^*) = 0$ by Lemma 3.1 follows that $q' = q^*$.
2. Property 2. Then the same argument holds
3. Property 3. Then $D(\tilde{p}||q^*) \leq D(\tilde{p}||q') = D(\tilde{p}||q^*) + D(q^*,q')$, hence $D(q^*,q') \leq 0$ meaning that $q^* = q'$.
4. Property 4. Then $D(q^*||q_0) \leq D(q'||q_0) = D(q'||q^*)+D(q^*,q_0)$, hence $D(q',q^*) \leq 0$ meaning that $q^* = q'$.

□

4 Computation

In the previous chapter we saw a self-contained proof of the existence and unicity of the maximum entropy model. Now we get to it's computation. As we saw in Chapter 2 we can compute the maximum entropy model solving an unconstrained maximization problem, namely maximizing the dual function

$$\Psi(\lambda) = -\log Z_\lambda + \sum_{i=1}^{n} \lambda_i E_{\tilde{p}}\left[f_i\right].$$

For all but the most simple cases the λ that maximizes Ψ cannot be found analytically. Instead we must resort to numerical methods. From the perspective of numerical optimization the function Ψ is well behaved, since it is smooth and convex. Therefore a variaty of numerical methods can be used to calculate the optimal λ^*. Such methods include coordinate-wise ascent, in which λ^* is computing by iteratively maximizing $\Psi(\lambda)$ one coordinate at a time. Other general purpose methods include conjugate gradient methods and quasi-Newton methods (Section 4.2 on page 25). There is an optimization method specifically tailored to the maximum entropy problem called the Improved Iterative Scaling algorithm. In the following sections we present the algorithm as it was introduced by Della Pietra et al. [1] and we also give a proof of the algorithm's monotonicity and convergence.

4.1 Improved Iterative Scaling

In this section we present the Improved Iterative Scaling algorithm introduced by Della Pietra et al. [1]. The algorithm generalizes the Darroch-Ratcliff procedure [2], which requires, in addition to the non-negativity, that the features functions satisfy $\sum_{i=1}^{n} f_i(x) = 1$ for all $x \in \mathcal{X}$.

Algorithm 4.1.1 Improved Iterative Scaling

Require: A reference distribution $\tilde{p}$, an initial model such that $\tilde{p} << q_0$ and non-negative features $f_1, f_2, \dots f_n$

Ensure: The distribution $q^* = \underset{q \in Q(\bar{f}, q_0)}{argmin}\ D(\tilde{p}||q)$

$q^{(0)} := q_0$

$k := 0$

while $q^{(k)}$ has not converged **do**

For each $i = 1, \dots, n$ let $\gamma_i^{(k)}$ be the unique solution of

$$q^{(k)}[f_i e^{\gamma_i^{(k)} f_\#}] = \tilde{p}[f_i]$$

where

$$f_\#(x) = \sum_{i=1}^{n} f_i(x)$$

$q^{(k+1)} := \gamma^{(k)} \circ q^{(k)}$

$k := k + 1$

end while

$q^* := q^{(k)}$

The key step of the algorithm is solving the non-linear equation

$$q^{(k)}[f_i e^{\gamma_i^{(k)} f_\#}] = \tilde{p}[f_i].$$

A simple and effective way of doing it is by Newton's method.

4.1.1 Monotonicity and convergence

In this section we will show a self-contained proof of the monotonicity and convergence of the Improved Iterative Scaling algorithm as it was proved by Della Pietra et al. [1]. We have L, the log-likelihood function of a distribution respect to the empirical distribution $\tilde{p}$ as we saw in the previous chapter

$$L(q) = -D(\tilde{p}||q) - H(\tilde{p}).$$

Before we get to proof of the monotonicity and convergence some concepts have to be introduced.

Definition 4.1. *A function $A : \mathbb{R}^n \times \triangle \to \mathbb{R}$ is an auxiliary function for L if*

1. *For all $q \in \triangle$ and $\gamma \in \mathbb{R}^n$*

$$L(\gamma \circ q) \geq L(q) + A(\gamma, q)$$

2. *$A(\gamma, q)$ is continuous in $q \in \triangle$ and C^1 in $\gamma \in \mathbb{R}^n$ with*

$$A(0, q) = 0 \text{ and } \frac{d}{dt}|_{t=0} A(t\gamma, q) = \frac{d}{dt}|_{t=0} L(t\gamma \circ q).$$

What we use auxiliary functions for is very simple. We can construct an iterative algorithm that increases the value of L in each step so that $q^{(0)} = q_0$ for some $q_0 \in \mathbb{R}^n$ and $q^{(k+1)} = \gamma^{(k)} \circ q^{(k)}$ where $\gamma^{(k)}$ is where the function $A(\gamma, q^{(k)})$ reaches it's supremum. From the first property we can see that $L(q^{(k)})$ increases monotically. For the complete proof of the convergence we need to introduce a more general concept.

Definition 4.2. *Let $\bar{\mathbb{R}}$ denote the $\mathbb{R} \cup -\infty$ the partially extended real numbers with the usual topology. The operations of addition and exponentiation extend continuously to $\bar{\mathbb{R}}$. Let $\mathcal{S}$ be the open subset of $\bar{\mathbb{R}}^n \times \triangle$ defined by*

$$\mathcal{S} = \{ (\gamma, q) \in \bar{\mathbb{R}}^n \times \triangle \mid q(\omega) e^{\gamma \cdot f(\omega)} > 0 \text{ for some } \omega \in \mathcal{X} \}.$$

Observe that $\mathbb{R}^n \times \triangle$ is a dense subset of $\mathcal{S}$. The map $(\gamma, q) \to \gamma \circ p$, which we defined for only γ where every coordinate was finite extends uniquely to a continuous map on $\mathcal{S}$ to $\triangle$. Note that by the definition of $\mathcal{S}$ the normalization constant in the definition of $\gamma \circ q$ is well defined even if γ is not finite, namely

$$Z_q(\gamma \cdot f) = E_q \left[e^{\gamma \cdot f} \right] > 0.$$

We will denote by $\mathcal{S}_m$ for $m \in \triangle$ a projection of $\mathcal{S}$ to $\bar{\mathbb{R}}^n$, that is

$$\mathcal{S}_m = \{ \gamma \in \bar{\mathbb{R}}^n \mid (\gamma, m) \in \mathcal{S} \}$$

Definition 4.3. *We call a function $A : \mathcal{S} \to \bar{\mathbb{R}}$ an extended auxiliary function for L if when restricted to $\mathbb{R}^n \times \triangle$ it is an auxiliary function and if, in addition, it satisfies Property 1 of the Definition 4.1 for any $(\gamma, q) \in \mathcal{S}$ even if γ is not finite.*

Note that if an auxiliary function extends to $\mathcal{S}$ continuously then the extended function is an extended auxiliary function since Property 1 holds by the continuity.

Lemma 4.1. *If m is a cluster point of $q^{(k)}$, then $A(\gamma, m) \leq 0$ for all $\gamma \in \mathcal{S}_m$.*

Proof. Since m is a cluster point, there exists a sub sequence $q^{(k_l)}$ of $q^{(k)}$ such that $q^{(k_l)}$ converges to m. Then for any $\gamma \in \mathcal{S}_m$

$$A(\gamma, q^{(k_l)}) \leq A(\gamma^{(k_l)}, q^{(k_l)}) \leq L(q^{(k_l+1)}) - L(q^{(k_l)}) \leq L(q^{(k_{l+1})}) - L(q^{(k_l)})$$

The first inequality follows from the definition of $\gamma^{(k_l)}$, that is it is where $A(\gamma, q^{(k_l)})$ reaches it's supremum. For the second and third inequalities note that the iterative algorithm defined by A is still increases $L(q^{(k)})$ in every step for extended auxiliary functions as well. Taking limits the lemma follows from the continuity of A and L. □

Lemma 4.2. *If m is a cluster point of $q^{(k)}$, then $\frac{d}{dt}|_{t=0} L((t\gamma) \circ m) = 0$ for any $\gamma \in \mathcal{S}_m$.*

Proof. By Lemma 4.1 we get that $A(\gamma, m) \leq 0$ for every $\gamma \in \mathcal{S}_m$. From Property 2 of an auxiliary function, namely $A(0, m) = 0$ follows that $\gamma = 0$ is a maximum of $A(\gamma, m)$, that is

$$0 = \frac{d}{dt}|_{t=0} A(t\gamma, m) = \frac{d}{dt}|_{t=0} L((t\gamma) \circ m),$$

where the second equality is also the consequence of the second property of auxiliary functions. □

Lemma 4.3. *Suppose that $q^{(k)}$ has only one cluster point m. Then $q^{(k)}$ converges to m.*

Proof. Suppose the contrary. Then exists an open subset of $\triangle$, B and a subsequence $q^{(k_l)}$ of $q^{(k)}$ such that $q^{(k_l)} \notin B \; \forall l$. Since $\triangle$ is compact $q^{(k_l)}$ has a cluster point m' which is not in B contradicting the assumption being m a unique cluster point. □

Theorem 4.4. *Suppose $q_0 \in \triangle$, $q^{(k)}$ is a sequence in $\triangle$ such that*

$$q^{(0)} = q_0 \text{ and } q^{(k+1)} = \gamma^{(k)} \circ q^{(k)} \text{ for } k \geq 0$$

where $\gamma^{(k)}$ satisfies

$$\gamma^{(k)} \in \mathcal{S}_{q^{(k)}} \text{ and } A(\gamma^{(k)}, q^{(k)}) \geq A(\gamma, q^{(k)}) \text{ for any } \gamma \in \mathcal{S}_{q^{(k)}}.$$

Then $L(q^{(k)})$ increases monotically to $\max_{q \in \bar{Q}} L(q)$ and $q^{(k)}$ converges monotically to $q^* = \operatorname{argmax}_{q \in \bar{Q}} L(q)$.

Proof. Suppose that m is a cluster point in $q^{(k)}$. From Lemma 4.2 follows that

$$\frac{d}{dt}|_{t=0} L((t\gamma) \circ m) = 0.$$

As a consequence of Lemma 3.4 $m \in P \cap \bar{Q}$, but as discussed in the previous chapter $q^* \in P \cap \bar{Q}$ is unique, hence q^* is the only cluster point in $q^{(k)}$ and from Lemma 4.3 $q^{(k)}$ converges to q^*. □

Now we get to the proof of monotonicity and convergence of the Improved Iterative Scaling algorithm. It is based on applying Theorem 4.4 to a particular choice of auxiliary function. For $q \in \triangle$ and $\gamma \in \mathbb{R}^n$, define

$$A(\gamma, q) = 1 + \gamma \cdot E_{\tilde{p}}[f] - \sum_{x \in \mathcal{X}} q(x) \sum_{i=1}^{n} f(i|x) e^{\gamma_i f_\#(x)},$$

where $f(i|x) = \frac{f_i(x)}{f_\#(x)}$. Note that A defined on $\mathbb{R}^n \times \triangle$ extends to a continuous function on $\bar{\mathbb{R}}^n \times \triangle$. One can see easily that maximizing A is equivalent to solving the equation defined in algorithm 4.1.1.

Lemma 4.5. *The function $A : \bar{\mathbb{R}}^n \times \triangle \to \mathbb{R}$ defined earlier is an extended auxiliary function for L.*

Proof. Note that the the logarithm function is concave and the exponential function is convex, hence

$$e^{\sum_i t_i a_i} \leq \sum_i t_i e^{a_i} \text{ for } t_i \geq 0, \sum_i t_i = 1,$$

$$\log x \leq x - 1 \text{ for } x > 0.$$

Since A extends to a continuous function on $\bar{\mathbb{R}}^n \times \triangle$, it suffices to prove that it satisfies Property 1 and 2 of auxiliary functions.

- Property 1:

$$\begin{aligned}
L(\gamma \circ q) - L(q) &= \sum_{x \in \mathcal{X}} \tilde{p}(x) \log \left(\frac{1}{Z_\lambda(\lambda \cdot f)} e^{\lambda \cdot f} q(x) \right) - \sum_{x \in \mathcal{X}} \tilde{p}(x) \log q(x) \\
&= \sum_{x \in \mathcal{X}} \tilde{p}(x) \left(-\log Z_\lambda(\lambda \cdot f) + \lambda \cdot f + \log q(x) - \log q(x) \right) \\
&= \gamma \cdot E_{\tilde{p}}[f] - \log \sum_{x \in \mathcal{X}} q(x) e^{\gamma \cdot f(x)} \\
&\geq \gamma \cdot E_{\tilde{p}}[f] + 1 - \sum_{x \in \mathcal{X}} q(x) e^{\gamma \cdot f(x)} \\
&\geq \gamma \cdot E_{\tilde{p}}[f] + 1 - \sum_{x \in \mathcal{X}} q(x) \sum_{i=1}^{n} f(i|x) e^{\gamma_i \cdot f_\#(x)} \\
&= A(\gamma, q).
\end{aligned}$$

- Property 2: this is consequence of straightforward calculations:

$$A(0, q) = 1 - \sum_{x \in \mathcal{X}} q(x) \sum_{i=1}^{n} f(i|x) = 1 - \sum_{x \in \mathcal{X}} q(x) = 1 - 1 = 0$$

$$
\begin{aligned}
\frac{d}{dt}|_{t=0} A((t\gamma), q) &= t(\gamma \cdot E_{\tilde{p}}[f]) - \sum_{x \in \mathcal{X}} q(x) \sum_{i=1}^{n} f(i|x)\gamma_i f_{\#}(x) e^{t\gamma_i f_{\#}(x)} \\
&= \sum_{x \in \mathcal{X}} q(x) \sum_{i=1}^{n} \gamma_i f_i(x) = \sum_{x \in \mathcal{X}} q(x)(\gamma \cdot f)(x) = E_q[\gamma \cdot f] \\
L_{\tilde{p}}((t\gamma) \circ q) &= \sum_{x \in \mathcal{X}} \tilde{p}(x) \log \left(\frac{1}{Z_q(t\gamma)} e^{t(\gamma \cdot f)(x)} q(x) \right) \\
&= \log Z_q(t\gamma) + \sum_{x \in \mathcal{X}} \tilde{p}(x) t(\gamma \cdot f)(x) + \tilde{p}(x) \log q(x) \\
&= \log Z_q(t\gamma) + t(\gamma \cdot E_{\tilde{p}}[f]) + \tilde{p}(x) \log q(x) \\
\frac{d}{dt}|_{t=0} L_{\tilde{p}}((t\gamma) \circ q) &= \frac{1}{e^{t(\gamma \cdot f)(x)}} \sum_{x \in \mathcal{X}} (\gamma \cdot f)(x) e^{t(\gamma \cdot f)(x)} q(x) \\
&= \sum_{x \in \mathcal{X}} q(x)(\gamma \cdot f)(x) = E_q[\gamma \cdot f]
\end{aligned}
$$

□

4.1.2 Conditional case

In Chapter 2 we introduced the conditional maximum entropy model, using Lagrange multipliers we got the form of the optimal model

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} e^{\sum_{i=1}^{n} f_i(x,y)}$$

where

$$Z_\lambda(x) = \sum_{y \in \mathcal{Y}} e^{\sum_{i=1}^{n} f_i(x,y)}.$$

We can obtain the optimal model by maximizing $\Psi(\lambda)$ dual function, that is

$$\Psi(\lambda) = -\sum_{x \in \mathcal{X}} \tilde{p}(x) \log Z_\lambda(x) + \sum_{i=1}^{n} E_{\tilde{p}}[f_i].$$

In this section we show how these optimal parameters can be calculated. The algorithm and the proof are both highly related to the ones in the previous section. We define the generalized Gibbs distribution for conditional distributions as follows.

$$(\gamma \circ q)(y|x) = \frac{1}{Z_{\gamma \cdot f}(x)} e^{\sum_{i=1}^{n} \lambda_i f_i(x,y)} q(y|x).$$

where

$$Z_{\gamma \cdot f}(x) = \sum_{y \in \mathcal{Y}} e^{\sum_{i=1}^{n} \lambda_i f_i(x,y)} q(y|x).$$

Note that the notations of the following algorithms are the ones used with conditional probablilities.

Algorithm 4.1.2 Improved Iterative Scaling (conditional case)

Require: A reference distribution $\tilde{p}$ and non-negative features $f_1, f_2, \dots f_n$
Ensure: The optimal parameter values γ_i^*, optimal model p_γ
Let be $q^{(0)}$ the uniform conditional distribution, that is $q^{(0)}(y|x) = \frac{1}{|\mathcal{Y}|}$ for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$.
$k := 0$
while $q^{(k)}$ has not converged **do**
For each $i = 1, \dots, n$ let $\gamma_i^{(k)}$ be the unique solution of

$$q^{(k)}[f_i e^{\gamma_i^{(k)} f_\#}] = \tilde{p}[f_i]$$

that is

$$\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \tilde{p}(x) q^{(k)}(y|x) f_i(x,y) e^{\gamma_i^{(k)} f_\#(x,y)} = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \tilde{p}(x,y) f_i(x,y)$$

where

$$f_\#(x,y) = \sum_{i=1}^{n} f_i(x,y)$$

$q^{(k+1)} := \gamma^{(k)} \circ q^{(k)}$
$k := k + 1$
end while
$q^* := q^{(k)}$

According to the results of the previous section for the proof of monotonicity and convergence of the IIS for conditional models we are left to prove that the function

$$A(\gamma, q) = 1 + \gamma \cdot E_{\tilde{p}}[f] - \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \tilde{p}(x) p(y|x) f(i|x) e^{\gamma_i f_\#(x)}$$

is an auxiliary function for the log-likelihood, namely

$$L_{\tilde{p}}(p) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \tilde{p}(x,y) \log \left(\frac{1}{Z_\lambda(x)} e^{\sum_{i=1}^{n} \lambda_i f_i(x,y)} \right).$$

Proof.

- Property 1: A very similar argument holds for conditional model to the one in the previous section.

$$\begin{aligned}
L(\gamma \circ q) - L(q) &= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \tilde{p}(x,y) \log\left(-\log Z_{\gamma\cdot f}(x) + \gamma\cdot f + \log q(y|x) - \log q(y|x)\right) \\
&= \gamma\cdot E_{\tilde{p}}[f] - \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \tilde{p}(x,y) \log\left(\sum_{y\in\mathcal{Y}} q(y|x) e^{\sum_{i=1}^n \gamma_i f_i(x,y)}\right) \\
&\geq \gamma\cdot E_{\tilde{p}}[f] - \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \tilde{p}(x,y) \left(\left(\sum_{y\in\mathcal{Y}} q(y|x) e^{\sum_{i=1}^n \gamma_i f_i(x,y)}\right) - 1\right) \\
&= \gamma\cdot E_{\tilde{p}}[f] + 1 - \sum_{x\in\mathcal{X}} \left(\sum_{y\in\mathcal{Y}} q(y|x) e^{\sum_{i=1}^n \gamma_i f_i(x,y)}\right) \sum_{y\in\mathcal{Y}} \tilde{p}(x,y) \\
&= \gamma\cdot E_{\tilde{p}}[f] + 1 - \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \tilde{p}(x) q(y|x) e^{\sum_{i=1}^n \gamma_i f_i(x,y)} \\
&\geq \gamma\cdot E_{\tilde{p}}[f] + 1 - \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \tilde{p}(x) p(y|x) f(i|x) e^{\gamma_i f_{\#}(x)} \\
&= A(\gamma, q).
\end{aligned}$$

- Property 2: A very similar straightforward calculation to the one in the normal case holds.

□

4.2 General numerical approaches

4.2.1 Conjugate Gradient Methods

Conjugate gradient (CG) methods comprise a class of unconstrained optimization algorithms which are characterized by low memory requirements and strong local and global convergence properties. We have the following unconstrained optimization problem

$$\min\{f(x) | x \in \mathbb{R}^n\}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function bounded from below. $\nabla_x f$ will denote the gradient of f. We present the general algorithm for a conjugate gradient method [5].

Algorithm 4.2.1 Conjugate Gradient Method

Require: A continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$
Ensure: The $x^* \in \mathbb{R}^n$ for which $f(x^*)$ is a local minimum
Let $x_0 \in \mathbb{R}^n$ the initial guess
$\Delta x_0 := -\nabla_x f(x_0)$
$\alpha_0 := \underset{\alpha}{argmin}\, f(x_0 + \alpha \Delta x_0)$. α_0 is obtained by a line search.
$x_1 := x_0 + \alpha_0 \Delta x_0$
$s_0 := \Delta x_0$
$k := 1$
while Some tolerance criterion is reached **do**
Calculate the steepest direction: $\Delta x_k := -\nabla_x f(x_k)$
Compute β_k according to one of the formulas below
Update the conjugate direction: $s_k := \Delta x_k + \beta_k s_{k_1}$
Perform a line search: $\alpha_k := \underset{\alpha}{argmin}\, f(x_k + \alpha s_k)$
Update the position: $x_{k+1} := x_k + \alpha_k s_k$, $k := k + 1$
end while
$x^* := x_k$

The most used formumlas for computing β_k are

- Fletcher-Reeves:
$$\beta_k := \frac{\Delta x_k^T \Delta x_k}{\Delta x_{k-1}^T \Delta x_{k-1}}$$
- Polak-Ribière:
$$\beta_k := \frac{\Delta x_k^T (\Delta x_k - \Delta x_{k-1})}{\Delta x_{k-1}^T \Delta x_{k-1}}$$
- Hestenes-Steifel:
$$\beta_k := -\frac{\Delta x_k^T (\Delta x_k - \Delta x_{k-1})}{s_k (\Delta x_{k-1}^T \Delta x_{k-1})}.$$

The algorithm stops when it finds the minimum, determined when no progress is made after a direction reset (i.e. in the steepest descent direction), or when some tolerance criterion is reached.

4.2.2 Quasi-Newton Methods

We have the same problem presented in the previous chapter. quasi-Newton methods [6] are based on Newton's method to find the stationary point of a functions, where the gradient is 0. In higher dimensions, Newton's method uses the gradient and the Hessian matrix of second derivatives of the function to be minimized. The size of the Hessian is quadratic in the number of parameters. Some practical applications often use tens of thousands or even millions of parameters, so even storing the full Hessian is not practical. In quasi-Newton methods the Hessian matrix does not need to be computed.

The Hessian is updated by analyzing successive gradient vectors instead. Quasi-Newton methods are a generalization of the secant method to find the root of the first derivative for multidimensional problems.
Now we present the general algorithm of quasi-Newton methods. The Taylor series of $f(x)$ around an iterate is:

$$f(x_k + \Delta x) \approx f(x_k) + \nabla_x f(x_k)^T \Delta x + \frac{1}{2} \Delta x^T B \Delta x,$$

where B is the approximation of the Hessian matrix. The gradient of this approximation with respect to Δx is

$$\nabla_x f(x_k + \Delta x) \approx \nabla_x f(x_k) + B\Delta x$$

and setting this gradient to zero provides a Newton step:

$$\Delta x = -B^{-1} \nabla_x f(x_k).$$

The Hessian approximation B is chosen to satisfy

$$\nabla_x f(x_k + \Delta x) = \nabla_x f(x_k) + B\Delta x.$$

In more than one dimension B is under determined. The various quasi-Newton methods differ in their choice of the solution to the secant equation (in one dimension, all the variants are equivalent). Given the current approximation B_k of the Hessian matrix one step of the procedure is

- $\Delta x_k := -\alpha_k B_k^{-1} \nabla_x f(x_k)$ with α chosen by line search
- $x_{k+1} := x_k + \Delta x_k$
- $y_k := \nabla_x f(x_{k+1}) - \nabla_x f(x_k)$. It is used to update the approximation of B_{k+1} and it's inverse H_{k+1}.

The most popular formulas for B_{k+1} and H_{k+1} are:

- Davidon–Fletcher–Powell (DFP) method

$$B_{k+1} := \left(I - \frac{y_k \Delta x_k^T}{y_k^T \Delta x_k}\right) B_k \left(I - \frac{\Delta x_k y_k^T}{y_k^T \Delta x_k}\right) + \frac{y_k y_k^T}{y_k^T \Delta x_k}$$

$$H_{k+1} := H_k + \frac{\Delta x_k \Delta x_k^T}{y_k^T \Delta x_k} - \frac{H_k y_k y_k^T H_k^T}{y_k^T H_k y_k}$$

- Broyden–Fletcher–Goldfarb–Shanno (BFGS) method

$$B_{k+1} := B_k + \frac{y_k y_k^T}{y_k^T \Delta x_k} - \frac{B_k \Delta x_k \left(B_k \Delta x_k\right)^T}{\Delta x_k^T B_k \Delta x_k}$$

$$H_{k+1} := \left(I - \frac{y_k \Delta x_k^T}{y_k^T \Delta x_k}\right)^T H_k \left(I - \frac{y_k \Delta x_k^T}{y_k^T \Delta x_k}\right) + \frac{\Delta x_k \Delta x_k^T}{y_k^T \Delta x_k}$$

- Broyden method

$$B_{k+1} := B_k + \frac{y_k - B_k \Delta x_k}{\Delta x_k^T \Delta x_k} \Delta x_k^T.$$

$$H_{k+1} := H_k + \frac{\left(\Delta x_k - H_k y_k\right) \Delta x_k^T H_k}{\Delta x_k^T H_k y_k}$$

5 Related Models

In this chapter we will present certain models that are based on the maximum entropy principle and their application in a specific relational learning task. Our example will be a very fundamental problem related to natural language processing and information extraction, the named-entity recognition. Named-entity recognition consists of locating and classifying atomic elements in text into predefined categories (*labels*) such as name, organization, place, date. Suppose we are given a large number of training samples $x_1, x_2, \ldots x_n$ and some labels $y_1, y_2, \ldots y_n$ generated by a human expert. Our task will be to set up a stochastic model that accurately represents the random process that generated the $(x_1, y_1), \ldots (x_2, y_2)$ pairs that for a given $\bar{x}_1, \ldots, \bar{x}_m$ sequence of observed values will generate the a label sequence $\bar{y}_1, \ldots, \bar{y}_m$ that is most likely according to the process. Named-entity recognition is a very interesting example for two reasons. Hence the nature of natural languages the observed sequence $x_1, \ldots, x_n$ – that will be tipically words or sequence of words – is too complex to model the big the training data may be. The other reason is that there can be complex long distance dependencies so in the observed sequence as in the label sequence. One could make some independence assumption as we will in the following sections, but it can lead to reduced performance.

5.1 The Hidden Markov Model

Hidden Markov models introduced by Rabiner [7] are a powerful probabilistic tool for modeling sequential data and have been applied with success to many text-related task, such as part-of-speech tagging, text segmentation and information extraction. In this section we will introduce the concept of Hidden Markov Models and discuss some of the basic algorithms in more detail.

An example of the hidden markov model is the urn problem. In a room which cannot be seen by an observer there is a genie. The room contains N different urns each of which contains a known mix of balls. The genie chooses randomly an urn in the room and draws a single ball from it. Then he puts the ball onto a conveyor belt, where the observer can

observe the sequence of the balls, without knowing the sequence of urns from which they were drawn from. The genie has a strategy to choose urns so that choosing the n-th urn only depends on the previous urn chosen.

5.1.1 The mathematical model

Suppose that we have a finite set of hidden states $S = \{ S_i \mid i = 1, \ldots, N \}$, and a finite set of possible observation values $V = \{ O_i \mid i = 1, \ldots, M \}$. Q_t is a radnom variable in time t which takes values from S, V_t is the observed value that takes values from O. The observed value V_t only depends on the current hidden state, and the current state only depends on the previous state and not on the ones before the previous one. This is called the *Markov property*. We will denote by a_{ij} the *transition probabilities* from one state S_i to S_j, namely

$$a_{ij} = P(Q_t = S_j | Q_{t-1} = S_i) \qquad \text{for } t = 1...T$$

One can see the transition probabilities do not depend on the observation time. This is called *stationary* property. We will denote the *emission probabilities* by b_{ij} for $i = 1, \ldots, N$ and $j = 1, \ldots, M$ the probability of a possible observed value depending on the current state.

$$b_{ij} = P(O_t = V_i | Q_t = S_j)$$

We will denote the distribution of the initial state by π, namely

$$\pi_i = P(Q_1 = S_i)$$

We can characterized the model with three parameters $\lambda = (A, B, \pi)$ where A is an $N \times N$ matrix constructed from the transition probabilities, B is an $N \times M$ matrix constructed from the emission probabilities and π is vector of size N which is the distribution of the initial state. There are three basic task according to the model:

- *model training*: Learning the parameters of $\lambda = (Q, B, \pi)$. A possible approach is the *Baum-Welch* algorithm.
- *decoding*: finding the most likely sequence of hidden states given a sequence of observations. An efficent algorithm for this task is the *Viterbi* algorithm.
- *inference*: Calculate the probability of a sequence of observed values. A possible solution is the *Forward-Backward* algorithm.

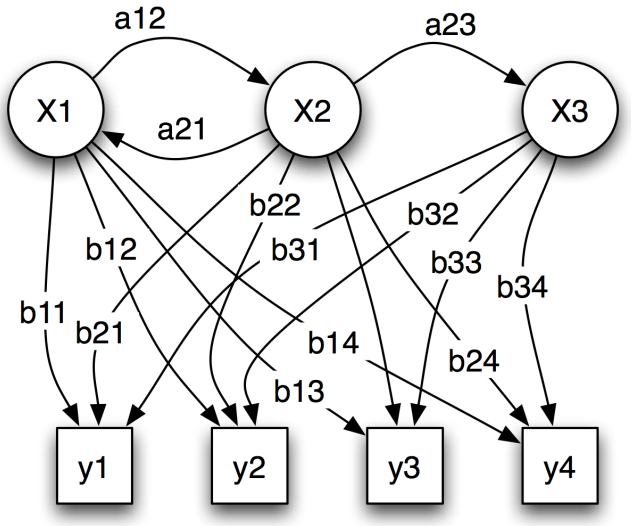

FIGURE 5.1: Hidden Markov Model example

5.1.2 The Forward-Backward algorithm

We have hidden markov model defined by it's parameters $\lambda = (A, B, \pi)$, and a sequence of possible observed values $\bar{O}_t$ for $t = 1, \ldots, N$. Our task is to determine the probability of this observed sequence. If we know the sequence of hidden states $(\bar{Q})$ we can easily calculate the probability of the output $\bar{O}$.

$$P(\bar{O}|\bar{Q}, \lambda) = \prod_{t=1}^{T} P(\bar{O}_t|\bar{Q}_t, \lambda) = \prod_{t=1}^{T} b_{\bar{Q}_t}(\bar{O}_t)$$

The probability that the model sequence of hidden states is $\bar{Q}_1, \ldots \bar{Q}_T$:

$$P(\bar{Q}|\lambda) = \pi(\bar{Q}_1) a_{\bar{Q}_1\bar{Q}_2} a_{\bar{Q}_2\bar{Q}_3} \ldots a_{\bar{Q}_{T-1}\bar{Q}_T}$$

The probability of the sequence of observed values $\bar{O}_1 \ldots \bar{O}_T$:

$$P(\bar{O}|\lambda) = \sum_{Q} P(\bar{O}|Q, \lambda) P(Q|\lambda) = \sum_{Q} \pi(Q_1) a_{Q_1}(O_1) \prod_{t=2}^{T} a_{Q_{t-1}Q_t} b_{Q_t}(O_t)$$

We can see that there is an exponential number of member in the summa. So we cannot calculate it directly. An efficent solution is the **Forward** algorithm. The main idea is to calculate the probability of a the sequence $\bar{O}_1 \ldots \bar{O}_t$ for every t adding that we arrive to

hidden state S_i for time t. We will denote these probabilities by $\alpha_t(i)$

$$\alpha_t(i) = P(O_1 \dots O_t, Q_t = S_i|\lambda)$$

We can calculate recursively these probabilities for every t and every i. Summing them for every i at time T gives us the probability we are looking for, since it is the probability of the entire sequence $\bar{O}_1 \dots \bar{O}_T$ for every final hidden state S_i. The recursion:

$$\alpha_1(i) = \pi(S_i)b_i(\bar{O}_1) \qquad i = 1, \dots, N$$

$$\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i)a_{ij}b_j(O_{t+1}) \qquad t = 1, \dots, T-1, j = 1, \dots, N$$

So summing them up in time T

$$P(\bar{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

The running time of the algorithm is $(O)(TN^2)$ since the calculation for an $\alpha_t(i)$ is N steps and we have to repeat the calculation NT times. The **Backward** algorithm is another possibly solution to the problem. It is pretty similar to the *Forward* algorithm. We define the backward variables $\beta_t(i)$ for $t = 1, \dots, T$, $i = 1, \dots, N$

$$\beta_t(i) = P(\bar{O}_{t+1}, \dots, \bar{O}_T, Q_t = S_i|\lambda)$$

The recursion:

$$\beta_T(i) = 1 \qquad i = 1, \dots, N$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}b_j(\bar{O}_{T+1}\beta_{t+1}(j))$$

5.1.3 The Viterbi algorithm

Suppose we have a sequence of observed values $\bar{O}_t$ for every $t = 1, \dots, N$. The *Viterbi* algorithm [7] is an efficient solution for finding the most likely sequence of hidden states which produces $\bar{O}$. The algorithm is very similar to the *Forward* algorithm, the difference is that we maximize in every step instead of summing. We calculate recursively the most likely sequence of hidden states $\bar{Q}_1, \dots, \bar{Q}_t$ for every $t = 1, \dots, T$ for the sequence of observed values $\bar{O}_1, \dots, \bar{O}_t$.

$$\delta_t(i) = \max_{Q_1, \dots, Q_t} \left(P(Q_t = S_i, \bar{O}_1, \dots, \bar{O}_t)\right)$$

We define the variable $\psi_t(i)$ for storing the state sequence where the maximum is reached. The recursion:

$$\delta_1(i) = \pi_i b_i(\bar{O}_1) \qquad \psi_t(i) = 0 \qquad i = 1, \ldots, N$$

$$\delta_t(j) = \max_{i=1,\ldots,N} \delta_{t-1}(i) a_{ij} b_j(\bar{O}_t) \qquad t = 2 \ldots T \quad j = 1, \ldots, N$$

$$\psi_t(j) = \operatorname*{argmax}_{i=1,\ldots,N} \delta_{t-1}(i) a_{ij} b_j(\bar{O}_t) \qquad t = 2 \ldots T \quad j = 1, \ldots, N$$

Finally to calculate the probability of the most likely hidden state sequence

$$P = \max_{i=1,\ldots,N} (\delta_T(i))$$

We can get the member of the most likely sequence for time T:

$$\bar{Q}_t = \operatorname*{argmax}_{i=1,\ldots,N} \delta_T(i)$$

We can calculate recursively the actual sequence of hidden states where the maximum is reached using the ψ variables:

$$\bar{Q}_t = \psi_{t+1}(\bar{Q}_{t+1}) \qquad t = T-1, \ldots, 1$$

Similarly to the *Forward* algorithm the running time is $\mathcal{O}(TN^2)$

5.1.4 The Baum-Welch algorithm

Suppose we are given a random process which we want to model by the hidden markov model. We get to observe the process for a while so we have a $\bar{O}_t$ for $t = 1, \ldots, T$ the observation values. The *Baum-Welch* algorithm helps us to find the right parametrization $\lambda = (A, B, \pi)$ for the hidden markov model which produces the sequence of observed values $\bar{O}$. For the algorithm an initial parametrization $\lambda_0 = (A_0, B_0, \pi_0)$ is required. The algorithm will improve this parametrization in every iteration. We will define two auxiliary variables ξ and γ. $\xi_t(i, j)$ will be the probability that system is in the S_i state at time t and in the S_j state at time $t + 1$ given the current model. Namely

$$\xi_t(i, j) = P(Q_t = S_t, Q_{t+1} = S_j | \bar{O}, \lambda)$$

We can calculate $\xi_t(i, j)$ using the α and β variables used in the *Forward-Backward* algorithm

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(\bar{O}_{t+1}) \beta_{t+1}(j)}{P(\bar{O}|\lambda)}$$

$\gamma_t(i)$ is the probability that the system is in the S_i hidden state given the current model λ and the sequence of observed values $\bar{O}$.

$$\gamma_t(i) = P(Q_t = S_t|\bar{O}, \lambda)$$

The following equation holds for γ and ξ:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$$

If we summerize the $\gamma_t(i)$-t for every t we get the expected number of times the system was in state S_i

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{Expected number of times the system is in state } S_i$$

Similarly if we summerize $\xi_t(i,j)$ for every t we get the expected number of transition from S_i to S_j

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{Expected number of transition from } S_i \text{ to } S_j$$

We can estimate accordingly a better model $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$.

$$\begin{aligned} \bar{\pi}_i &= \gamma_1(i) \\ \bar{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \\ \bar{b}_j(k) &= \frac{\sum_{t=1,\ldots,T, \text{ if } \bar{O}_T = V_k} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned}$$

Baum et al. [8] has shown that if $\bar{\lambda} \neq \lambda$ then $P(\bar{O}|\bar{\lambda}) > P(\bar{O}|\lambda)$, so iterating the procedure we get better and better model. It is important that the parametrization given by the Baum-Welch algorithm is highly dependent on the initial parametrization.

5.2 The Maximum Entropy Markov Model

There are two problems with the tradition approach of Hidden Markov models in applications such as named-entity recogniton. In these tasks it may beneficial for the observations to be whole lines of text, or even entire documents, so all possible observations is not reasonably enumerable. The other problem is that it inappropriately uses a joint model in order to solve a conditional problem. We will introduce the concept

of Maximum Entropy Markov models [9] in this chapter which adress both of these concerns.

5.2.1 The mathematical model

Similarly to Hidden Markov models suppose that we have a finite set of hidden states $S = \{ S_i \mid i = 1, \ldots, N \}$ (also called as *labels*), and the set of possible observation values $V = \{ O_i \mid i = 1, \ldots, M \}$. Now instead of the independency assumption that HMM makes we will only assume that the probability of the current state only depends on the previous state and the current observation value. So the transition and emission functions will be replaced by one single function the probability of $s \in S$ given the previous label s' and the current observation o. Since in our model it is independent from the time parameter t we will use the following notation:

$$P(s|s', o) = P(Q_t = s|Q_{t-1} = s', O_t = o)$$

It reflects the fact that we do not really care about the probability of the given observation sequence, but the probability of the label sequence they induce. The use of state-observation transition functions rather than separate transition and observation functions in HMMs allows us to model transitions in terms of multiple non-independent features of observations. To model this conditional probability we will use the maximum entropy principle introduced in Chapter 2. First, we will split this conditional distributions into N different conditional distribution $P_{s'}(s|o)$ and we will treat them separately. Suppose that we are given n features $f_1, f_2, \ldots f_n$ that depend only on the observed value o and the current state. As described in Chapter 2 the model that is consistent with these features and maximizes the entropy takes the form

$$P_{s'}(s|o) = \frac{1}{Z(o, s')} e^{\sum_{i=1}^{n} \lambda_i f_i(s,o)}$$

where λ_i are parameters to learn and $Z(o, s')$ is a normalization factor that makes the distribution sum to one across all next state s.

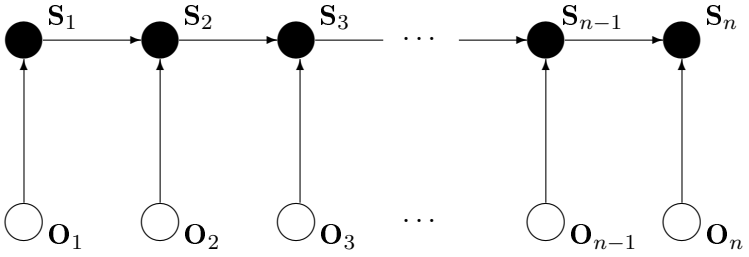

FIGURE 5.2: Maximum Entropy Markov Model example

5.2.2 The modified Viterbi algorithm

Suppose we are given a set of observation values $o_1, o_2, \ldots o_m$. The modified *Viterbi* algorithm finds the label sequence that is most likely, it is closely related to the *Viterbi* algorithm introduced in Section 5.1.3, we redefine the variables $\delta_t(s)$ for every $t = 1, \ldots, T$ and $s \in S$ as the most likely state s at time t given the observation sequence up to time t. This probability initially can be easily expressed with the defined probabilities:

$$\delta_1(s) = P(s|o_1).$$

The recursive *Viterbi* step is then

$$\delta_{t+1}(s) = \max_{s' \in S} \left(\delta_t(s') P_{s'}(s|o_{t+1}) \right).$$

The algorithm is then pretty much the same as described in the previous section. We can redefine the forward and backward variables similarly. In the new model $\alpha_t(s)$ for every $t = 1, \ldots, T$ and $s \in S$ as the probability of being in state s in time t given the observation values up to t and also $\beta_t(s)$ as the probability of starting from s at time t given the observation sequence after time t.

5.2.3 Parameter Learning

Suppose that we are given a sequence of observation values $o_1, o_2, \ldots o_m$ with the corresponding label sequence $s_1, s_2, \ldots s_n$. The Improved Iterative Scaling algorithm introduced in Chapter 3 finds iteratively the λ_i values that form the maximum entropy solution for each transition function. We present the algorthm in the following pseudocode.

Algorithm 5.2.1 Parameter learning in Maximum Entropy Markov Models

Require: sequence of observation values $o_1, o_2, \ldots o_m$, the corresponding label sequence $s_1, s_2, \ldots s_n$

Ensure: A maximum-entropy-based Markov model that takes unlabeled sequence of observations and predicts their correspoding labels

for $s' \in S$ **do**

Deposit state-observation pairs (s,o) into their corresponing previous state s' as traingin data for $P_{s'}(s|o)$

Find the maximum entropy solution by running IIS

end for

5.3 Markov Random Fields

In this section we present some basic concepts of Markov Random Fields, although detailed discussion is out of the scope of this document. See [10] for a detailed discussion.

Definition 5.1. *Let $G = (V, E)$ be an undirected graph, X a random variable such that X is indexes by the V and X_v takes values from $\mathcal{X}$ $\forall v \in V$. X is Markov Random Field if it satisfies the following properties:*

1. *$p(x) > 0$ for every $x \in \mathcal{X}$*
2. *for every $v \in V$*

$$p(X_v = x_v | X_w = x_w, w \neq v) = p(X_v = x_v | X_w = x_w, wv \in E).$$

Let $\mathcal{C}$ denote the set of all cliques of a graph. *Clique*s are maximal subgraphs such that every two vertices are connected.

Definition 5.2. *Let G, X be as before. We call p, the distribution of X, a Gibbs-distribution if p factorizes respect to the cliques of G, that is it can be written as*

$$p(x) = \prod_{c \in \mathcal{C}(G)} \Psi_c(x),$$

where $\Psi_c(x) > 0$ for every $x \in \mathcal{X}^{|V|}$ and Ψ_c only depends on $x_c = \{ x_v \mid v \in c \}$, that is on the coordinates corresponding to the clique c. We call Ψ_c local functions for $c \in \mathcal{C}(G)$.

Theorem 5.1. *(Hammersly-Clifford) Let $G = (V, E)$, X, p be as before. X is Markov Random Field if and only if p is a Gibbs-distribution.*

5.4 Conditional Random Fields

In this section we introduce the concept of Conditional Random Fields. Conditional Random Fields (CRFs) are a sequence modeling framework that has all the advantages of Maximum Entropy Markov Models. The critical difference between CRFs and MEMMs is that MEMM uses a per-state exponential models for the conditional probabilities of next states given the current state, while CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence. CRFs include state-of-the-art models for named entity recognition by the ability to model long distance dependencies between labels. In the followings we present the mathematical definition of CRFs [11] as well as basic algorithms for parameter estimation and inference [11],[12].

5.4.1 The mathematical model

Let $\mathcal{Y}$ be a finite set of possible labels. X, Y are discrete random variables such that $Y = (Y_1, Y_2, \ldots, Y_T)$ where Y_i takes values from $\mathcal{Y}$ for every $i = 1, \ldots, T$. We want to model the random variable $Y|X$.

Definition 5.3. *Let $G = (V, E)$ be an undirected graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X, the random variables Y_v obey the Markov property with respect to G, that is*

$$p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, vw \in E).$$

In other words Y given X is a Markov Random Field with respect to the graph G, that is according to the Hammersley-Clifford theorem from the previous chapter the probability $p(\mathbf{y}|\mathbf{x})$ takes the form

$$p(\mathbf{y}|\mathbf{x}) = \prod_{c \in \mathcal{C}(G)} \Psi_c(\mathbf{x}, y_c)$$

for all $y \in \mathcal{Y}^n$, $x \in \mathcal{X}$. Troughout the whole section we will assume that all the local functions has the form

$$\Psi_c(\mathbf{x}, \mathbf{y}) = \exp\left\{ \sum_{k=1}^{n(c)} \lambda_{ck} f_{ck}(\mathbf{x}, y_c) \right\},$$

for some $f = \{ f_c i \mid c \in \mathcal{C}(G), i = 1, \ldots, n(c) \}$ none-negative feature functions. The joint distribution then takes the form

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \prod_{c \in \mathcal{C}(G)} \exp\left\{ \sum_{k=1}^{n(c)} \lambda_{ck} f_{ck}(\mathbf{x}, y_c) \right\},$$

where $Z(x)$ is the normalizing factor that ensures that for a given $x \in \mathcal{X}$ the distribution sums to 1 over all $y \in \mathcal{Y}^n$. In addition, practical models rely extensively on parameter tying. To denote this, we partition the factors of G into $\mathcal{C} = \{ C_1, C_2, \ldots, C_P \}$ where each C_p is a *clique template* whose parameters are tied. We also require that the clicques corresponding to the same clique template have the same size. Then the CRF can be written as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \prod_{C_p \in \mathcal{C}} \prod_{c \in C_p} \Psi_c(\mathbf{x}, y_c, \theta_p),$$

where each factor is parameterized as

$$\Psi_c(x_c, y_c, \theta_p) = \exp\left\{ \sum_{i=1}^{n(p)} \lambda_{pi} f_{pi}(\mathbf{x}, y_c) \right\}$$

and the normalization function is

$$Z(x) = \sum_{y \in \mathcal{Y}^n} \prod_{C_p \in \mathcal{C}} \prod_{c \in C_p} \Psi_c(\mathbf{x}, y_c, \theta_p).$$

5.4.2 Inference

In some simple cases such as in Linear Chain Conditional Random Fields, some modifications of the Viterbi algorithm or other dynamic programming methods can be used for inference, but in the general computing exactly the most likely label sequence given an observation sequence is untractable. In this section we show how to solve this problem using *Gibbs sampling* [13], a simple Monte Carlo method used to perform approximate inference in probabilistic models. Monte Carlo methods are a simple and effective class of methods for approximate inference based on sampling. Suppose that we are given an observation sequence $\mathbf{O}$ that was produced by the random variable $\mathcal{X}$, $\mathcal{Y}$ the finite set of possible labels and a trained CRF (F) to model the conditional ditribution $p_F(Y|\mathcal{X})$, where $Y = (Y_1, \ldots, Y_n)$ such that every Y_i is a random variable that takes values from $\mathcal{Y}$. We want to find the most likely label sequence $y_1, \ldots, y_n$ $y_i \in \mathcal{Y}$ given $\mathbf{O}$. Gibbs sampling provides a clever solution. It defines a Markov chain in the space of possible label assignments such that the stationary distribution of the Markov chain is the joint distribution over the labels. The chain is defined in very simple terms: from each label sequence we can only transition to a label sequence obtained by changing the label at any one position i, and the distribution over these possible transitions is just

$$P(\mathbf{y}^{(t)}|\mathbf{y}^{(t-1)}) = P_F(y_i^{(t)}|\mathbf{y}_{-i}^{(t-1)}, \mathbf{O}) \tag{5.1}$$

where $\mathbf{y}_{-i}$ is all labels except the one in position i. This probability can be efficently computed with our model F since

$$P_F(y_i^{(t)}|\mathbf{y}_{-i}^{(t-1)}, \mathbf{O}) = \frac{P_F(y^{(t)}|\mathbf{O})}{\sum_{y \in \mathcal{Y}} P_F(\mathbf{y}_{i=y}^{(t-1)}|\mathbf{O})}$$

where $\mathbf{y}_{i=y}^{(t)}$ denotes the sequence that follows by changing the label at position i to y. Now we need an efficent way to find the vertice with maximum probability. We cannot just transition greedily to higher probability sequences at each step, because the space is extremely non-convex. Instead of that we borrow a technique of non-convex optimization called simulated annealing. This technique consists of modify the probability in 5.1 in

every time t, given a sequence $c = \{ c_1, c_2, \ldots, c_T \}$ such that $0 < c_i <= 1$

$$P'(\mathbf{s}^{(t)}|\mathbf{s}^{(t-1)}) = \frac{P_F(s_i^{(t)}|\mathbf{s}_{-i}^{(t-1)}, \mathbf{O})^{1/c_t}}{\sum_{j=1}^{n} P_F(s_j^{(t)}|\mathbf{s}_{-j}^{(t-1)}, \mathbf{O})^{1/ct}}.$$

This annealing technique has been shown to be an effective technique for stochastic optimization [13].

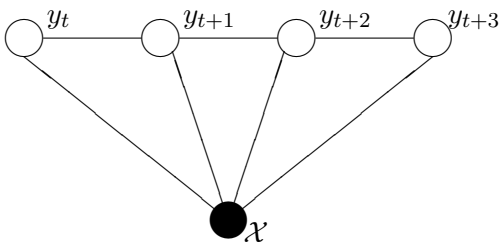

FIGURE 5.3: A sepcial case of CRF, the corresponding graph is a chain.

Bibliography

[1] Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, April 1995. URL http://arxiv.org/pdf/cmp-lg/9506014.pdf.

[2] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972. URL http://ftp.cs.nyu.edu/~roweis/csc412-2006/extras/gis.pdf.

[3] Rong Jin, Rong Yan, Jian Zhang, and Alex G. Hauptmann. A faster iterative scaling algorithm for conditional exponential model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, April 2003. URL http://repository.cmu.edu/cgi/viewcontent.cgi?article=1994&context=compsci.

[4] Joshua Goodman. Sequential conditional generalized iterative scaling. *Proc. ACL*, pages 9–16, July 2002. URL http://acl.ldc.upenn.edu/P/P02/P02-1002.pdf.

[5] William W. Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1):35–58, 2006. URL http://www.caam.rice.edu/~zhang/caam454/pdf/cgsurvey.pdf.

[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Section 10.9. quasi-newton or variable metric methods in multidimensions. *Cambridge University Press, Numerical Recipes: The Art of Scientific Computing (3rd ed.)*, 2007.

[7] Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[8] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970. URL http://www.jstor.org/discover/10.2307/2239727?uid=3738216&uid=2&uid=4&sid=21101939396641.

[9] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. *Proc. ICML*, pages 591–598, 2000. URL http://www.ai.mit.edu/courses/6.891-nlp/READINGS/maxent.pdf.

[10] J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. 1971. URL http://www.statslab.cam.ac.uk/~grg/books/hammfest/hamm-cliff.pdf.

[11] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*, pages 282–289, 2001. URL http://www.cis.upenn.edu/~pereira/papers/crf.pdf.

[12] Charles Sutton and A. McCallum. An introduction to conditional random fields for relational learning. *MIT Press*, 2006. URL http://people.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf.

[13] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. *ACL*, pages 363–370, 2005. URL http://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf.

[14] L. Adam Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1): 39–71, March 1996. URL http://acl.ldc.upenn.edu/J/J96/J96-1002.pdf.