Lesson 6

Stochastic Stringsets

6.1 Stochastic Stringsets and Parametric Language Models

So far, we have mostly studied the problem of learning grammars which classify strings as belonging to some formal language or not. That is we have studied the learning of functions of the form $f : \Sigma^* \to \{0, 1\}$. Many are also interested in learning probability distributions over $Sigma^*$, that is functions of the form $f : \Sigma^* \to [0, 1]$ subject to the constraint that $\sum_{w \in \Sigma^*} f(w) = 1$. Functions which classify strings yield formal languages (i.e. stringsets), and functions which assign probabilities to strings yield stochastic formal languages (stochastic stringsets). This section is about learning stochastic stringsets, for which there is a vast literature we only scratch the surface of here.

Nonetheless, there are a few major lessons I want to get across.

1. Both long-term (e.g. PAC) and short-term learning criteria (e.g MLE) exist which can be used to understand the general behavior of learning algorithms.
2. The problem of learning functions which assign probabilities to strings typically comes down to assigning values to parameters. In other words, the concept class and possible hypotheses are defined parametrically. Learning the correct function then means finding the true values of the parameters. (There are also non-parametric models and methods. These actually do not eliminate parameters altogether, but they do have a bit more flexibility about what the parameters are. The parameters are not fixed.)
3. The problem of learning functions which assign probabilities to strings typically assumes that the data has been generated i.i.d. according to the true values of the parameters.
4. When the parametric models are deterministic, theoretical solutions exist. Still, additional heuristics are usually needed in real-world applications (Jurafsky and Martin, 2008).
5. When the parametric models are non-deterministic, there are guarantees to find "local optima" but not the true parameter values, unless various additional assumptions are made on the hypothesis space.
6. In general, independence-like assumptions (that factor the concept class in a natural way) are extremely useful for both deterministic and non-deterministic models (Ghahramani and Jordan, 1997; Hsu *et al.*, 2012; Shibata and Heinz, 2019).
7. How do stochastic stringsets relate to linguistics? Many equate likelihood with acceptability but this is a conceptual error (Heinz and Idsardi, 2017).
8. That said, there is a lot of really interesting work on learning stochastic stringsets and functions of the form $f : \Sigma^* \to \mathbb{R}$ more generally.

6.1.1 Parametric Models

We begin with the simplest parametric model I know of: the unfair coin. A fair coin has equal probability of landing heads or tails when flipped. An unfair coin has probabilty θ of landing heads and probability $1 - \theta$ of landing tails. So a fair coin is the special case of an unfair coin when $\theta = 0.5$. This probability θ is the sole parameter in our model of unfair coins.[1]

Another very simple parametric model is often called a unigram model. In this model, there are parameters for each symbol in the alphabet in addition to a parameter signaling the end of the sequence. It assumes that the probability of the next event in the sequence are completely

[1] This unfair coin model does not yield a probability distribution over $\{H, T\}^*$ because there is no end to the sequence of flips.

independent. For example if $\Sigma = \{a, b, c\}$ then there are four parameters: θ_a, θ_b, θ_c, and $\theta_{\ltimes}$. These must sum to 1. The probability of a string $\sigma_1\sigma_2 \ldots \sigma_n$ is simply the product $\theta_{\sigma_1}\theta_{\sigma_2} \ldots \theta_{\sigma_n}\theta_{\sigma_{\ltimes}}$.

For both simple parametric models above, the concept class is defined simply by considering all possible values the parameters can take on. We will denote this set of possible parameter values with Θ. Since the parameters are real numbers in $[0, 1]$, these hypothesis spaces are infinite in size (but obviously very well structured).

6.1.2 What does learning mean?

With these two simple models in mind, we can begin to formulate learning problems. In both cases we want algorithms which provide "good" estimations of the parameter values from data. Parametric models usually have more than one parameter, but they only ever have finitely many parameters. Notationally, a single symbol $\theta \in \Theta$ is used to represent all parameter values.

In the case of the unfair coin, what kind of procedure gives us a good estimate of θ from observed coin flips of some unfair coin? In the case of the unigram model, what kinds of procedure gives us a good estimate of θ from an observed sample of strings? Both these questions amount to coming up with values for the parameters in the model that are "close" to the true values. In both examples, the assumption is that the function we want to find is the function generating the data.

6.1.3 Learning Criteria

Here are some common learning criteria in this setting.

- A learning algorithm A **is a consistent estimator for a parametric model** M_Θ iff for all $\theta \in \Theta$, given data randomly presented i.i.d. according to θ, the algorithm's estimates of the parameters $A(D) = \hat{\theta}$ approaches the true value of θ. Formally, $forall\epsilon > 0, \exists n$ such that $\forall m > n$ we have $[\hat{\theta}_m - \theta < \epsilon]$.
- A learning algorithm A **is a maximum likelihood estimator for a parametric model** M_Θ iff given any finite set of data D randomly presented i.i.d. according to θ, the algorithm's estimates of the parameters $A(D) = \hat{\theta}$ maximizes the probability of D with respect to all other possible values of the parameters in M. Formally, for all $theta' \in \Theta$ $\Pr_M(D \mid \hat{\theta}) \geq \Pr_M(D \mid \theta')$. (Note $\Pr_M(D \mid \hat{\theta})$ is often called the **likelihood function**.)

It is well-known that estimators that maximize the likelihood are also consistent estimators.

Other learning criteria of course have been studied too. We start here because they are simple, straightforward, and encompass both long-term and short-term criteria for understanding the general behavior of learning algorithms.

6.1.4 Finding the MLE

There are generally two strategies for finding MLE: analytic solutions and hill-climbing.

The simple parametric models above, the unfiar coin and the unigram model, have analytic solutions. In both cases, one can prove that the relative frequencies of the observed symbols yields the MLE. Specifically, for the unfair coin, given n flips of the coin,a the estimate $\hat{\theta} = \frac{c(H)}{n}$ is the MLE where $c(H)$ is the number of times the coin landed heads. For the unigram model, for each symbol $\sigma \in \Sigma$, the estimate $\hat{\theta_\sigma} = \frac{c(\sigma)}{\sum_{\sigma \in \Sigma} c(\sigma)}$ is the MLE. For a proof of the latter, see the appendix.

In the absence of analytic solutions, the hill-climbing strategy is used. This strategy first sets the parameter values arbitrarily, and then repeats the following process. It finds the direction of steepest ascent—called the gradient—and adjusts the paremeters a small amount in that direction. This is repeated until the new parameter values are not that different from the previous parameter values. In that case, we are at, or arguably very close, to the top of the hill. Hill-climbing is not guaranteed to find the MLE, though is is guaranteed to find local optima. Hill-climbing is a general strategy to find local optima of any kind (not just for finding the MLE).

6.1.5 Probabilistic Determinsitic Finite-State Acceptors

Every deterministic finite-state acceptor (DFA) is a parametric model defining a family of stochastic stringsets. The parameters are probabilities on the transitions at each state (including ending) subject to the constraint that these sum to 1. The analytic solution for unigram model generalizes to every state in the DFA (see appendix). This yields the general strategy for finding the MLE classes defined with DFA. Pass the dataset D through the DFA counting each time a transition is passed. Once all the counts have been obtained convert the counts into probabilities via normalization.

Note there are many interesting classes of DFA, most of which have *not* been studied in this way.

6.1.6 N-gram Models

N-gram models are stochastic strictly k-local stringsets where $n = k$. The states correspond to the most recent $n-1$ symbols. Each transition is thus the probability that a symbol σ is generated given the previous $n - 1$ symbols. So a trigram's models parameters are of the form $Pr(c \mid ab)$ where $a, b, c \in \Sigma$. This corresponds to the transition from state ab with input c. In the figure they correspond to the class D (Definite).

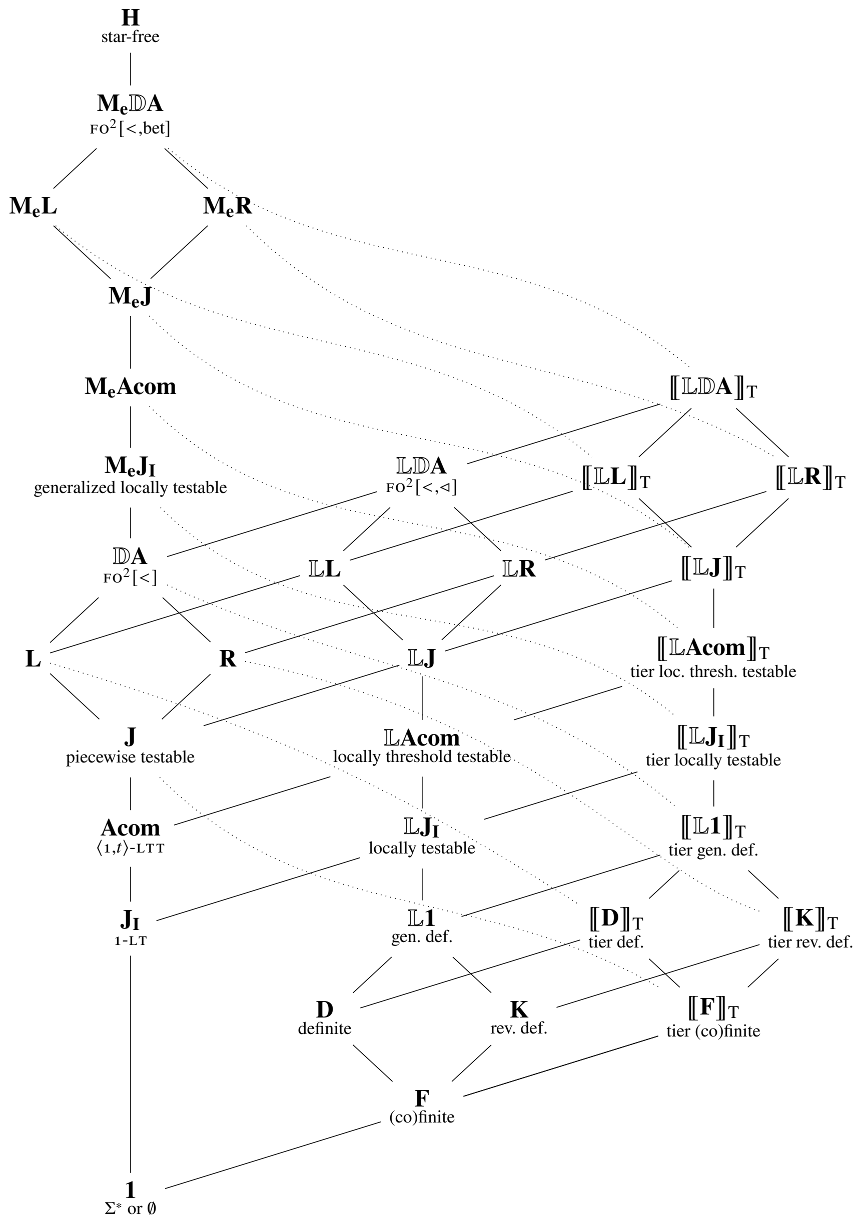
6.1.7 Smoothing and Interpolation

The MLE may not make good predictions in practice, especially in the context of sparse data. Smoothing and interpolation are two common techniques (Jurafsky and Martin, 2008).

I find interpolation interesting because at its heart the idea is to combine models. Suppose you have three models M_1, M_2, and M_3 (say three DFA). Given a data set D you can find the MLE estimates for each their parameters. Interpolation would be a meta model with three hyperparameters h_1, h_2, h_3 which sum to 1 in order to assign a probability to a string as follows.

$$Pr(w) = h_1 M_1(w) + h_2 M_2(w) + h_3 M_3(w)$$

In practice, values for the hyperparameters are found using the validation/development data set.



Some subregular classes for sets and functions over sequences, circa 2022

6.1.8 Probabilistic Non-determinstic Finite-State Acceptors

PNFA are equivalent in expressivity to Hidden Markov Models. The main method used here is expectation-maximization ().

6.1.9 Probabilistic Context Free Grammars

6.1.10 Entropy

6.1.11 Maximum Entropy Models

6.1.12 Perplexity

6.1.13 Likelihood and Acceptability

Appendix

Probability distributions over words

A *stochastic language* is a probability distribution over all logically possible words. This distribution is given by a vector of parameters Θ and some formula for how probabilities of words are determined given Θ. The range of values Θ can take while preserving a well-formed probability distribution over words yields a *family* of distributions. A sample of data D from the stochastic language is a (multi)set of words drawn from this distribution (i.i.d).

The likelihood function is the probability of the data D given a vector of parameters Θ (so the likelihood function is always discussed in the context of some family of stochastic languages).

$$L_\Theta(D) = \prod_{w \in D} Pr_\Theta(w) \tag{6.1}$$

The maximum likelihood estimate of the data D with respect to the vector of parameters Θ are those parameter values which maximize the likelihood function.

$$MLE_\Theta(D) = \hat{\Theta} = \underset{\Theta}{\text{argmax}}\ L_\Theta(D) \tag{6.2}$$

Unigram models

There are $|\Sigma| + 1$ parameters. For all $\sigma \in \Sigma$, we write θ_σ for the $Pr(\sigma)$, and $\theta_\sharp$ for the probability of the word ending. We refer to these parameters collectively as Θ.

Let $\Sigma_\sharp = \Sigma \cup \{\sharp\}$. By definition

$$\sum_{\sigma \in \Sigma_\sharp} \theta_\sigma = 1 \tag{6.3}$$

The probability of a word according to a unigram model is

$$Pr_\Theta(w = \sigma_1 \ldots \sigma_m) = \left(\prod_{1\leq i\leq m} \theta_{\sigma_i}\right) \theta_\sharp \tag{6.4}$$

$$= \left(\prod_{\sigma\in\Sigma} {\theta_\sigma}^{c(\sigma)}\right) \theta_\sharp \tag{6.5}$$

It follows for some (multi)set D that the likelihood function is

$$Pr_\Theta(D) = \left(\prod_{\sigma\in\Sigma} {\theta_\sigma}^{c(\sigma)}\right) {\theta_\sharp}^{|D|} \tag{6.6}$$

For convenience we let $c(\sharp)$ denote $|D|$, and so we can rewrite Equation 6.6 as

$$L_\Theta(D) = Pr_\Theta(D)$$

$$= \left(\prod_{\sigma\in\Sigma} {\theta_\sigma}^{c(\sigma)}\right) {\theta_\sharp}^{c(\sharp)} \tag{6.7}$$

$$= \prod_{\sigma\in\Sigma_\sharp} {\theta_\sigma}^{c(\sigma)} \tag{6.8}$$

We want to find the values of θ_σ that maximize the likelihood function. For reasons that will become clear momentarily, we first rewrite the likelihood form in a slightly more complex form, by singling out some other $\tau \in \Sigma_\sharp$. Recall by definition that

$$\theta_\tau = 1 - \sum_{\sigma\in\Sigma_\sharp/\{\tau\}} \theta_\sigma \tag{6.9}$$

In other words, once the $|\Sigma|$ other parameters are fixed, θ_τ is too. This is because the parameters are not independent (recall Equation 6.3).

Thus:

$$\prod_{\sigma\in\Sigma_\sharp} {\theta_\sigma}^{c(\sigma)} = \left(\prod_{\sigma\in\Sigma_\sharp/\{\tau\}} {\theta_\sigma}^{c(\sigma)}\right) {\theta_\tau}^{c(\tau)} \tag{6.10}$$

$$= \left(\prod_{\sigma\in\Sigma_\sharp/\{\tau\}} {\theta_\sigma}^{c(\sigma)}\right) \left(1 - \sum_{\sigma\in\Sigma_\sharp/\{\tau\}} \theta_\sigma\right)^{c(\tau)} \tag{6.11}$$

We take the partial derivative of Equation 6.11, for all $\bar{\sigma} \in \Sigma_\sharp/\{\tau\}$, with respect to each $\theta_{\bar{\sigma}}$ using the product and chain rules below.

product rule: $\frac{d}{dx}(f(x)\cdot g(x)) = \frac{d}{dx}f(x)\cdot g(x) + \frac{d}{dx}g(x)\cdot f(x)$

chain rule: $\frac{d}{dx}f(g(x)) = \frac{d}{dx}f(g(x))\cdot\frac{d}{dx}g(x)$

The partial derivatives yield a system of equations. Each equation has the form shown in Equation 6.12.

$$\begin{aligned}\frac{\partial L}{\partial\theta_{\bar\sigma}} = {} & c(\bar\sigma){\theta_{\bar\sigma}}^{c(\bar\sigma)-1}\left(\prod_{\sigma\in\Sigma_\sharp/\{\tau,\bar\sigma\}}{\theta_\sigma}^{c(\sigma)}\right)\left(1-\sum_{\sigma\in\Sigma_\sharp/\{\tau\}}\theta_\sigma\right)^{c(\tau)} \\ & - c(\tau)\left(\prod_{\sigma\in\Sigma_\sharp/\{\tau\}}{\theta_\sigma}^{c(\sigma)}\right)\left(1-\sum_{\sigma\in\Sigma_\sharp/\{\tau\}}\theta_\sigma\right)^{c(\tau)-1}\end{aligned} \tag{6.12}$$

We substitute θ_τ back in and simplify

$$\frac{\partial L}{\partial\theta_{\bar\sigma}} = c(\bar\sigma){\theta_{\bar\sigma}}^{c(\bar\sigma)-1}\left(\prod_{\sigma\in\Sigma_\sharp/\{\tau,\bar\sigma\}}{\theta_\sigma}^{c(\sigma)}\right){\theta_\tau}^{c(\tau)} - c(\tau)\left(\prod_{\sigma\in\Sigma_\sharp/\{\tau\}}{\theta_\sigma}^{c(\sigma)}\right){\theta_\tau}^{c(\tau)-1} \tag{6.13}$$

$$= c(\bar\sigma){\theta_{\bar\sigma}}^{c(\bar\sigma)-1}\left(\prod_{\sigma\in\Sigma_\sharp/\{\bar\sigma\}}{\theta_\sigma}^{c(\sigma)}\right) - c(\tau)\left(\prod_{\sigma\in\Sigma_\sharp/\{\tau\}}{\theta_\sigma}^{c(\sigma)}\right){\theta_\tau}^{c(\tau)-1} \tag{6.14}$$

$$={\theta_{\bar\sigma}}^{c(\bar\sigma)-1}{\theta_\tau}^{c(\tau)-1}\left(\prod_{\sigma\in\Sigma_\sharp/\{\bar\sigma,\tau\}}{\theta_\sigma}^{c(\sigma)}\right)(c(\bar\sigma)\theta_\tau - c(\tau)\theta_{\bar\sigma}) \tag{6.15}$$

For each $\bar\sigma \in \Sigma_\sharp/\{\tau\}$, we obtain an equation and we want solve this system of equations where, for each $\bar\sigma \in \Sigma_\sharp/\{\tau\}$, it is the case that $\frac{\partial L}{\partial\theta_{\bar\sigma}} = 0$.

In order to make $\frac{\partial L}{\partial\theta_\tau} = 0$, either the left term or the right term in Equation 6.15 must equal 0. The left term can equal zero by making any of the parameters θ_σ ($\sigma \neq \tau$) equal to zero. However, assuming the nonzero counts, the likelihood function (Equation 6.8) will also be zero.[2] This is probably not going to be a maximum. Therefore we consider what is necessary to make the right term equal to zero. In other words for each $\bar\sigma \in \Sigma_\sharp/\{\tau\}$, we want to solve this system of equations.

$$c(\bar\sigma)\theta_\tau - c(\tau)\theta_{\bar\sigma} = 0 \tag{6.16}$$

proof 1

We can solve for both $\theta_{\bar\sigma}$ and θ_τ.

$$\theta_{\bar\sigma} = \frac{c(\bar\sigma)\theta_\tau}{c(\tau)} \tag{6.17}$$

And similarly, solving for θ_τ yields

$$\theta_\tau = \frac{c(\tau)\theta_{\bar\sigma}}{c(\bar\sigma)} \tag{6.18}$$

[2] Also assuming that $0^0 = 1$.

Note the equations above hold for any $\bar{\sigma} \in \Sigma_\sharp/\{\tau\}$. Thus for any $\sigma, \bar{\sigma} \in \Sigma_\sharp/\{\tau\}$, $\theta_{\bar{\sigma}}$ and θ_σ can be related by substituting Equation 6.18 for θ_τ in Equation 6.17 as follows.

$$\theta_\sigma = \frac{c(\sigma)}{c(\tau)} \frac{c(\tau)\theta_{\bar{\sigma}}}{c(\bar{\sigma})} \tag{6.19}$$

which simplifies to

$$\theta_\sigma = \frac{c(\sigma)\theta_{\bar{\sigma}}}{c(\bar{\sigma})} \tag{6.20}$$

We use this result in Equation 6.20 back in Equation 6.16 since, for all $\sigma \in \Sigma/\{\tau, \bar{\sigma}\}$, every θ_σ can be written in terms of $\theta_{\bar{\sigma}}$, $c(\sigma)$, and $c(\bar{\sigma})$.

This substitution happens in Equation 6.23. We can then solve for $\theta_{\bar{\sigma}}$ in terms of the counts $c(\sigma)$ for all $\sigma \in \Sigma_\sharp/\{\tau\}$. Here is Equation 6.16 again.

$$c(\bar{\sigma})\theta_\tau - c(\tau)\theta_{\bar{\sigma}} = 0 \tag{6.16}$$

$$c(\bar{\sigma}) \left(1 - \sum_{\sigma \in \Sigma_\sharp/\{\tau\}} \theta_\sigma \right) - c(\tau)\theta_{\bar{\sigma}} = 0 \tag{6.21}$$

$$c(\bar{\sigma}) \left(1 - \theta_{\bar{\sigma}} - \sum_{\sigma \in \Sigma_\sharp/\{\bar{\sigma},\tau\}} \theta_\sigma \right) - c(\tau)\theta_{\bar{\sigma}} = 0 \tag{6.22}$$

$$c(\bar{\sigma}) \left(1 - \theta_{\bar{\sigma}} - \sum_{\sigma \in \Sigma_\sharp/\{\bar{\sigma},\tau\}} \frac{c(\sigma)\theta_{\bar{\sigma}}}{c(\bar{\sigma})} \right) - c(\tau)\theta_{\bar{\sigma}} = 0 \tag{6.23}$$

$$c(\bar{\sigma}) - c(\bar{\sigma})\theta_{\bar{\sigma}} - \sum_{\sigma \in \Sigma_\sharp/\{\bar{\sigma},\tau\}} c(\sigma)\theta_{\bar{\sigma}} - c(\tau)\theta_{\bar{\sigma}} = 0 \tag{6.24}$$

$$c(\bar{\sigma}) - \sum_{\sigma \in \Sigma_\sharp} c(\sigma)\theta_{\bar{\sigma}} = 0 \tag{6.25}$$

$$c(\bar{\sigma}) - \theta_{\bar{\sigma}} \sum_{\sigma \in \Sigma_\sharp} c(\sigma) = 0 \tag{6.26}$$

$$\frac{c(\bar{\sigma})}{\sum_{\sigma \in \Sigma_\sharp} c(\sigma)} = \theta_{\bar{\sigma}} \tag{6.27}$$

And there we have it! In fact the MLE estimate of $\theta_{\bar{\sigma}}$ is the relative frequency of $\bar{\sigma}$'s occurences with respect to all other $\sigma \in \Sigma_\sharp$. Since τ was arbitrarily selected, the value of θ_τ is determined in exactly the same way (and of course is fixed as this by the results of the other θ_σ).

Proof 2

Recall the system of equations for all $\bar{\sigma} \in \Sigma_\sharp/\{\tau\}$:

$$c(\bar{\sigma})\theta_\tau - c(\tau)\theta_{\bar{\sigma}} = 0 \tag{6.16}$$

Summing all of these equations yields an equation that allows us to solve for θ_τ.

$$\sum_{\sigma \in \Sigma_\sharp/\{\tau\}} \big(c(\sigma)\theta_\tau - c(\tau)\theta_\sigma\big) = 0 \tag{6.28}$$

$$\sum_{\sigma \in \Sigma_\sharp/\{\tau\}} c(\sigma)\theta_\tau - \sum_{\sigma \in \Sigma_\sharp/\{\tau\}} c(\tau)\theta_\sigma = 0 \tag{6.29}$$

$$\theta_\tau \sum_{\sigma \in \Sigma_\sharp/\{\tau\}} c(\sigma) - c(\tau) \sum_{\sigma \in \Sigma_\sharp/\{\tau\}} \theta_\sigma = 0 \tag{6.30}$$

$$\theta_\tau \sum_{\sigma \in \Sigma_\sharp/\{\tau\}} c(\sigma) - c(\tau)(1 - \theta_\tau) = 0 \tag{6.31}$$

$$\tag{6.32}$$

Solving for θ_τ yields

$$\theta_\tau \sum_{\sigma \in \Sigma_\sharp/\{\tau\}} c(\sigma) - c(\tau)(1 - \theta_\tau) = 0 \tag{6.33}$$

$$\theta_\tau \sum_{\sigma \in \Sigma_\sharp/\{\tau\}} c(\sigma) - c(\tau) + c(\tau)\theta_\tau = 0 \tag{6.34}$$

$$\theta_\tau \left(\sum_{\sigma \in \Sigma_\sharp/\{\tau\}} c(\sigma) + c(\tau) \right) - c(\tau) = 0 \tag{6.35}$$

$$\theta_\tau \sum_{\sigma \in \Sigma_\sharp} c(\sigma) - c(\tau) = 0 \tag{6.36}$$

$$\theta_\tau = \frac{c(\tau)}{\sum_{\sigma \in \Sigma_\sharp} c(\sigma)} \tag{6.37}$$

Since τ was aribitrary, it follows this true for all $\sigma \in \Sigma_\sharp$.

Rational deterministic stochastic languages

Consider any *deterministic* finite-state automata $\mathcal{A}$. Let Q refer to the states of this automata, and let $q_0 \in Q$ be the intial state. Let the transition function be a partial function $\delta : Q \times \Sigma \to Q$. This is extended to strings in Σ^* in the usual way. So $\delta(q, \lambda) = q$ etc.

$\mathcal{A}$ describes a family of stochastic languages with $|Q|(|\Sigma|+1)$ parameters. These parameters are, for all $q \in Q$ and $\sigma \in \Sigma_\sharp$,

$$Pr(\sigma \mid q) = \theta_{q\sigma}$$

In other words, the probabilities at each state of following path labeled σ or ending. The probabilities of words can be determined as follows.

$$Pr(\sigma_1 \cdots \sigma_n) = \left(\prod_{i=0}^{n} \theta_{\delta(q_0,\sigma_i)\sigma_{i+1})}\right) \theta_{\delta(q_0,\sigma_1\cdots\sigma_n)\sharp} \tag{6.38}$$

By convention let $\delta(q_0, \sigma_0) = \delta(q_0, \lambda)$. For example, let $w = abcde$. Then

$$Pr(w) = \theta_{\delta(q_0,\lambda)a}\, \theta_{\delta(q_0,a)b}\, \theta_{\delta(q_0,ab)c}\, \theta_{\delta(q_0,abc)d}\, \theta_{\delta(q_0,abcd)e}\, \theta_{\delta(q_0,abcde)\sharp}$$

For all $q \in Q$, $\sigma \in \Sigma$, and $w \in \Sigma^*$, define the counting function $c_w(q\sigma)$ as follows

$$c_w(q\sigma) = |\{u\tau \in \text{Pfx}(w) : \delta(q_0, u) = q \text{ and } \tau = \sigma\}|$$

When w is clear from context, it is omitted. We extend the domain of the count function from words to multisets of words in the normal way.

Then we can rewrite Equation 6.38 as follows.

$$Pr(w) = \left(\prod_{\substack{\sigma\in\Sigma \\ q\in Q}} {\theta_{q\sigma}}^{c(q\sigma)}\right) \theta_{\delta(q_0,q)\sharp} \tag{6.39}$$

It follows for some (multi)set D that the likelihood function is

$$Pr(D) = \left(\prod_{\substack{\sigma\in\Sigma \\ q\in Q}} {\theta_{q\sigma}}^{c(q\sigma)}\right) \left(\prod_{q\in Q} {\theta_{q\sharp}}^{c(q\sharp)}\right) \tag{6.40}$$

$$= \left(\prod_{\substack{\sigma\in\Sigma_\sharp \\ q\in Q}} {\theta_{q\sigma}}^{c(q\sigma)}\right) \tag{6.41}$$

where for convenience we let $c(q\sharp)$ denote the number of words in D which ended at state q.

For each $q \in Q$, it is the case (by definition) that

$$\sum_{\sigma\in\Sigma_\sharp} \theta_{q\sigma} = 1$$

Thus for any $\tau \in \Sigma_\sharp$, it is the case that

$$\theta_{q\tau} = 1 - \sum_{\sigma\in\Sigma_\sharp/\{\tau\}} \theta_{q\sigma} \tag{6.42}$$

And thus the likelihood function in Equation 6.41, for each $\bar{q} \in Q$, can be rewritten as follows.

$$Pr(D) = \left(\prod_{\substack{\sigma \in \Sigma_\sharp / \{\tau\} \\ q \in Q}} {\theta_{q\sigma}}^{c(q\sigma)} \right) \left(1 - \sum_{\sigma \in \Sigma_\sharp / \{\tau\}} \theta_{\bar{q}\sigma} \right)^{c(\bar{q}\tau)} \tag{6.43}$$

And now, for each $\bar{\sigma} \in \Sigma_\sharp / \{\tau\}$ and $\bar{q} \in Q$, we take the partial derivative with respect to each $\theta_{\overline{q\sigma}}$.

$$\begin{aligned} \frac{\partial L}{\partial \theta_{\overline{q\sigma}}} = {} & c(\overline{q\sigma}) {\theta_{\overline{q\sigma}}}^{c(\overline{q\sigma})-1} \left(\prod_{\substack{\sigma \in \Sigma_\sharp / \{\tau, \bar{\sigma}\} \\ q \in Q}} {\theta_{q\sigma}}^{c(q\sigma)} \right) \left(1 - \sum_{\sigma \in \Sigma_\sharp / \{\tau\}} \theta_{\bar{q}\sigma} \right)^{c(\bar{q}\tau)} \\ & - c(\bar{q}\tau) \left(\prod_{\substack{\sigma \in \Sigma_\sharp / \{\tau\} \\ q \in \pi}} {\theta_{q\sigma}}^{c(q\sigma)} \right) \left(1 - \sum_{\sigma \in \Sigma_\sharp / \{\tau\}} \theta_{\bar{q}\sigma} \right)^{c(\bar{q}\tau)-1} \end{aligned} \tag{6.44}$$

This is virtually identical to Equation 6.12. The same treatment used in the unigram case will yield that, For any $\bar{q} \in \pi$ and $\bar{\sigma} \in \Sigma$, the maximum likelihood estimate of $\theta_{\overline{q\sigma}}$ is

$$\theta_{\overline{q\sigma}} = \frac{c(\overline{q\sigma})}{\sum_{\sigma \in \Sigma_\sharp} c(\bar{q}\sigma)} \tag{6.45}$$

Bibliography

Ghahramani, Zoubin, and Michael I. Jordan. 1997. Factorial hidden markov models. *Machine Learning* 29:245–273.

Heinz, Jeffrey, and William Idsardi. 2017. Computational phonology today. *Phonology* 34:211–219.

Hsu, Daniel, Sham M. Kakade, and Tong Zhang. 2012. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences* 78:1460–1480. JCSS Special Issue: Cloud Computing 2011.
URL https://www.sciencedirect.com/science/article/pii/S0022000012000244

Jurafsky, Daniel, and James Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. 2nd ed. Upper Saddle River, NJ: Prentice-Hall.

Shibata, Chihiro, and Jeffrey Heinz. 2019. Maximum likelihood estimation of factored regular deterministic stochastic languages. In *Proceedings of the 16th Meeting on the Mathematics of Language*, 102–113. Toronto, Canada: Association for Computational Linguistics.