Sequence Models

John Thickstun

Suppose we have data $\mathbf{x} \in \mathbb{R}^{T \times d}$, which we will interpret as a sequence of d-dimensional vectors x_t indexed by a discrete temporal $0 \leq t < T$. As usual, we are interested in modeling the unknown distribution p over a collection of observed sequences $\mathbf{x}^1, \ldots, \mathbf{x}^n \sim p$ (for time series, we will index sequences with a superscript to distinguish this index from the temporal index). One way to approach time series modeling is to ignore the temporal structure, and simply treat sequences as vectors in $\mathbb{R}^{Td}$. This is technically valid, but there are a couple of reasons we might want to incorporate temporal structure into our models:

1. The length of the sequences can be variable: each data point $\mathbf{x}^i$ can have a unique length T_i.
2. The collapsed vectors in $\mathbb{R}^{Td}$ can be very high dimensional, and we may want to exploit temporal structure to simply our modeling task.

We can address both concerns with autoregressive modeling. The idea is to to model the conditional distributions $p(x_t|x_{<t})$, i.e. the distributions over x_t given preceding observations $x_{<t} = \{x_0, \ldots, x_{t-1}\}$. The point is simply that we can use the chain rule for probabilities to factor the joint distribution over $\mathbf{x}$ into conditional distributions:

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t|x_{<t}).$$

Given a collection of learned conditional distributions $\hat{p}(x_t|x_{<t})$, we can sample from model by iteratively sampling $\hat{x}_t \sim \hat{p}(\cdot|\hat{x}_{<t})$. We can control the length of an autoregressive model's output by choosing the number of sampling iterations. If we want to learn the distribution over sequence lengths, we can train an additional family of conditional models $\hat{p}(\texttt{stop}_t|x_{<t})$, where $\texttt{stop} \in \{0, 1\}$ and stop sampling when $\texttt{stop} = 1$. As we'll see in the next two sections, we can control the dimensionality of high-dimensional sequences by imposing structure on the conditional probability distributions $p(x_t|x_{<t})$.

Linear Autoregressive Models

For a simple example of sequential data, let $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ (i.i.d.) and consider a scalar sequence of observations $\mathbf{x} \in \mathbb{R}^{T \times 1}$ defined by the linear dynamics

$$\begin{aligned} x_t &= \rho(x_{t-1} - \mu) + \mu + \varepsilon_t, \\ x_1 &= \mu + \varepsilon_0. \end{aligned}$$

These dynamics are referred to as a (linear) autoregressive process of order 1, or AR(1) process [Rao, 2020]; order 1 refers to the fact that x_t is a function of only one previous observation x_{t-1}

(this is also known as the Markov property):

$$p(x_t|x_{<t}) = p(x_t|x_{t-1}) = \mathcal{N}(\rho(x_{t-1} - \mu) + \mu, \sigma^2).$$

By induction, we see that $\mathbb{E}[x_t] = \mu$. Because the mean of the process is not a function of time, we say that the mean is stationary.

Definition. *A time series* $\boldsymbol{x}$ *has* ***stationary p'th moment*** *if* $\mathbb{E}[x_t^p]$ *is constant, for all* $t \in \mathbb{N}$.

To simplify things a lot, suppose we know ρ and σ^2. Then to construct an estimator $\hat{p}$, given a sequence of observations $\mathbf{x}$ of length T, all we need is an estimate of the mean $\hat{\mu}$. It is tempting to try the estimator $\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} x_t$. This is an unbiased estimator of μ. However, our observations x_t are not i.i.d. so the law of large numbers does not ensure consistency of $\hat{\mu}$. If we examine the proof of the (weak) law of large numbers, we can recover a result for time series. Chebyshev's inequality tells us that

$$P(|\hat{\mu} - \mu| \geq \delta) \leq \frac{\mathrm{Var}(\hat{\mu})}{\delta^2}.$$

Unlike the i.i.d. case, we must track covariances when analyzing $\mathrm{Var}(\hat{\mu})$:

$$\mathrm{Var}(\hat{\mu}) = \frac{1}{T^2}\left(\sum_{t=1}^{T}\mathrm{Var}(x_t) + 2\sum_{1\leq s<t\leq T}\mathrm{Cov}(x_s, x_t)\right).$$

And, again in contrast to the i.i.d. case, it is not at all clear that $\mathrm{Var}(\hat{\mu}) \to 0$ as $T \to \infty$.

Definition. *A time series* X *is* ***wide-sense stationary*** *if it has stationary mean and*

$$\mathrm{Cov}(x_t, x_{t-k}) = \mathrm{Cov}(x_s, x_{s-k}),\ \forall s, t, k \in \mathbb{N}.$$

In this case, we define the ***autocovariance*** *function* $\gamma : \mathcal{N} \to \mathbb{R}$ *by*

$$\gamma(k) = \mathrm{Cov}(x_{k+1}, x_1).$$

Wide-sense stationarity allows us to complete our analysis of the sample mean:

$$\mathrm{Var}(\hat{\mu}) = \frac{1}{T^2}\left(T\gamma(0) + 2\sum_{k=1}^{T}(T-k)\gamma(k)\right) = \frac{\gamma(0)}{T} + \frac{2}{T}\sum_{k=1}^{T}\left(1 - \frac{k}{T}\right)\gamma(k) \leq \frac{\gamma(0)}{T} + \frac{2}{T}\sum_{k=1}^{T}\gamma(k).$$

We must have $\gamma(k) \to 0$ as $k \to \infty$ and if it decays quickly enough then the estimator will converge in probability. If γ is summable, i.e. $\sum_{k=1}^{\infty}\gamma(k)$ is finite, then $\mathrm{Var}(\hat{\mu}) = O(1/T)$. Intuitively, the faster a process forgets its history–as measured by the rate at which the autocovariance function decays–the faster the sample mean $\hat{\mu}$ converges to the distribution mean μ.

For an AR(1) process, $\mathrm{Var}[x_t] = \rho^2\,\mathrm{Var}[x_{t-1}] + \sigma^2$, from which we deduce that $\mathbf{x}$ has stationary variance iff $|\rho| < 1$ and in particular, $\mathrm{Var}[\mathbf{x}] = \frac{\sigma^2}{1-\rho^2}$. A straightforward induction shows that an AR(1) process is wide-sense stationary, and its autocovariance function[1] is given by

$$\gamma(k) = \frac{\rho^k\sigma^2}{1-\rho^2}.$$

[1]The natural analog to a correlation coefficient for stochastic processes is the autocorrelation function defined by $f(k) = \gamma(k)/\gamma(0)$. In the case of an AR(1) process, $f(k) = \rho^k$ and in particular $f(1) = \rho$, mirroring the Pearson correlation coefficient and motivating the choice of the symbol ρ as a regression coefficient.

This function is clearly summable when $|\rho| < 1$, which justifies use of the estimator $\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} x_t$.

All the preceding discussion of scalar time series generalizes to vector sequences $\mathbf{x} \in \mathbb{R}^{T \times d}$. Another way to generalize the AR(1) model involves conditioning on more items in the history, i.e. $p > 1$ items. Specifically, a vector-valued linear autoregressive process of order p, an AR(p) process, is defined by the linear dynamics

$$x_t = \sum_{k=1}^{p} \rho_k (x_{t-1} - \mu) + \mu + \varepsilon_t,$$
$$x_1 = \mu + \varepsilon_0.$$

In this case $\rho \in \mathbb{R}^m, \mu \in \mathbb{R}^d$, and $\varepsilon_t \sim \mathcal{N}(0, \sigma^2 I)$. A similar analysis to the one we did for AR(1) processes generalizes to AR(p) processes.

Previously, we focused on estimating the mean of an autoregressive process because it is easy to analyze. A more interesting question is: how do we estimate the regression coefficients ρ? Drawing inspiration from the i.i.d. setting, it is tempting to try the maximum likelihood estimator. There is a little subtlety to learning the distribution over the first few steps of the process[2] (before we've observed p items) but ignoring that we can proceed exactly as we do in the i.i.d. setting to fit the conditional maximum likelihood estimator of $p(x_t | x_{t-p}, \ldots, x_{t-1})$. In this case, the MLE is simply the least squares estimator and can be solved in closed form.

Statisticians have thought deeply about how best to fit the regression coefficients of an AR(p) model, and it is not clear that conditional maximum likelihood estimation is the best approach. Other contenders include Yule-Walker estimation, based on the method of moments, and the Burg Algorithm [Rao, 2020]. Nevertheless, when we generalize these autoregressive models to the neural autoregressive setting, we will focus exclusively on the conditional maximum likelihood estimator. It is unclear to me whether or not any empirical gains could be achieved by developing these alternative parameter estimators in the neural setting, or if reasonable generalizations of these estimators even exist.

n-gram Models

In the previous section, we considered estimation of the mean of scalar-valued process governed by parametric dynamics. We now turn our attention to discrete-valued processes. We will use language as a running example of such time series. By language, we mean sequences $\mathbf{w} \in \mathcal{V}^T$ (e.g. sentences or documents) consisting of discrete tokens $w_t \in \mathcal{V}$ (e.g. words or characters) taken from some finite vocabulary $\mathcal{V}$. The most direct way to estimate a finite distribution is to simply tabulate the probability of each element of the space. But for sequence models, this is a difficult task. First, there is the theoretical nuance that, although $\mathcal{V}$ is finite, the space of sequences may not be (this isn't a concern in practice of course). But even for modest and fixed sequence lengths T, the space $\mathcal{V}^T$ is far too large even to enumerate; constructing probability estimates with reasonable variance is a hopeless task.

Autoregressive modeling can help us here. Unless $|\mathcal{V}|$ is very large, we can tabulate $p(w_t | w_{<t})$ for a fixed history $w_{<t}$ (this is just a distribution over $\mathcal{V}$). The catch is that there are $O(\mathcal{V}^T)$ conditional distributions to model: we need to construct a probability table of size $\mathcal{V}$ for each possible history sequence $w_{<t}$ at each time index t. The n-gram model [Jurafsky and Martin, 2008]

[2] For details, see Section 1 of Chapter 9 in Rao [2020].

makes two simplifying assumptions that drastically reduce the number of conditional distributions that we need to model.

Definition. *A process is* ***strictly stationary*** *if its joint distribution is invariant to time shifts:*

$$p(w_t, \ldots, w_{t+n}) = p(w_s, \ldots, w_{s+n}), \text{ for all } s, t, n \in \mathbb{N}.$$

Assuming that a process is stationarity means that we only have to construct one set of probability tables, rather than a table at each time index. This is a much stronger stationarity assumption than the moment conditions or wide-sense stationarity introduced in the previous section.

Definition. *A process is* ***n'th order Markov*** *if it has limited-horizon temporal dependencies:*

$$p(w_t|w_{<t}) = p(w_t|w_{t-n+1}, \ldots, w_{t-1}), \text{ for all } t \in \mathbb{N}.$$

Assuming that a process is Markov means that we only have to construct $\mathcal{V}^n$ of these tables, rather than $O(\mathcal{V}^T)$ tables.

These assumptions still don't quite get us to where we need to be. A typical vocabulary might consist of $50,000$ words, and for this vocabulary size there are about as many 5-grams as there are stars in the universe. But we are saved by sparsity: most of these 5-grams will never occur in real-world data. So instead of storing a full 5-gram table with $O(\mathcal{V}^5)$ elements, we can use e.g. a hash table of counts. While this sparsity saves us computationally, it should give you pause from a learning perspective: if our observed data doesn't come close to covering the space, how can trust our probability estimates?

The final piece of the n-gram puzzle is smoothing [Chen and Goodman, 1999]. Instead of using the MLE count statistics to fill in the n-gram table, we regularize the problem by stealing probability mass from the observed n-grams and re-allocating it to the rest of the n-grams in some more-or-less clever way. Abstractly, this smoothing is analogous to non-parametric methods like kernel density estimation [Tsybakov, 2008] for continuous density estimation which perform a similar function by reallocating probability mass away from the observations in order to better cover the space. This is a preview of things to come, where we will continue to see the distinctions between continuous and finite modeling blur as the size of our finite spaces become large.

References

Stanley F Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 1999. (document)

Daniel Jurafsky and James H Martin. Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing. *Upper Saddle River, NJ: Prentice Hall*, 2008. (document)

Suhasini Subba Rao. A course in time series analysis. *Technical Report, Texas A&M University*, 2020. (document), 2

Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008. (document)