Statistics Review

Ricson

July 12, 2017

Contents

1 Probability Theory

1.1 Conjugate Priors

1.2 Basic manipulations of Multivariate Gaussians

1.3 Extremal Value Distributions

2 Information Theory

2.1 Basics

2.2 KL Divergence

2.3 Fisher Information Matrix

3 Linear Algebra

3.1 Identities

3.2 Singular Value Decomposition

3.3 Principle Component Analysis

4 Probabilistic Graphical Models

4.1 Viterbi and Forward Backward Algorithm

The key to deriving the forward backward algorithm for the linear-chain CRF is to realize that there is no fundamental difference between HMM and CRF. The likelihood of a linear-chain CRF is defined to be the product of all the potentials. But before we get mixed up in math here: o_i denotes observation and x_i denotes the hidden state.

$$P(\mathbf{x}) = \prod_i \phi(x_i) \prod_{i,i+1} \phi(x_i, x_{i+1})$$

On the other hand, the likelihood of the equivalent HMM model is the following:

$$P(x) = \prod_i P(o_i|x_i)P(x_i|x_{i+1})$$

Note this product is incorrect when $i = n$, but let's ignore that.

Now if you squint really closely, you're realize that these two equations are actually the same thing. We can just replace $P(o_i|x_i)$ with $\phi(x_i)$ and $P(x_i|x_{i+1})$ with $\phi(x_i, x_{i+1})$. So it suffices to solve inference on the general setting of CRFs. However, we'll start on HMMs simply to avoid a bit of headaches.

We want to solve for $P(x_k|\mathbf{o})$. The key is the following manipulation:

$$\begin{aligned}
P(x_k|\mathbf{o}) &= \frac{P(\mathbf{o}|x_k)p(x_k)}{P(\mathbf{o})} \\
&\propto P(\mathbf{o}|x_k)p(x_k) \\
&= P(\mathbf{o}_{1:k}|x_k)P(\mathbf{o}_{k+1:n}|x_k)P(x_k)
\end{aligned}$$

Let's take that second term and apply Bayes' Theorem again

$$\begin{aligned}
P(\mathbf{o}_{k+1:n}|x_k) &= \frac{P(x_k|\mathbf{o}_{k+1:n})P(\mathbf{o}_{k+1:n})}{P(x_k)} \\
&\propto \frac{P(x_k|\mathbf{o}_{k+1:n})}{P(x_k)}
\end{aligned}$$

And substitute it back in...

$$\begin{aligned}
P(\mathbf{o}_{1:k}|x_k)P(\mathbf{o}_{k+1:n}|x_k)P(x_k) &= P(\mathbf{o}_{1:k}|x_k)\frac{P(x_k|\mathbf{o}_{k+1:n})}{P(x_k)}P(x_k) \\
&= P(\mathbf{o}_{1:k}|x_k)P(x_k|\mathbf{o}_{k+1:n})
\end{aligned}$$

So now we just need to figure out how to compute $P(\mathbf{o}_{1:k}|x_k)$ and $P(x_k|o_{k+1:n})$. We can do this inductively. To do this, we specify that each x_i is a categorical variables in one of m classes indexed by j. In the base case, we are given $P(o_1|x_1)$. Since x_k is between $o_{1:k-1}$ and o_k it renders o_k independent from the rest, conditioned on x_k. We will take advantage of this.

$$
\begin{aligned}
P(o_{1:k}|x_k) &= P(o_{1:k-1}|x_k)P(o_k|x_k) \\
&= P(o_k|x_k)\sum_j P(o_{1:k-1}|x_{k-1}=j)p(x_{k-1}=j|x_k)
\end{aligned}
$$

Notice that we can retrieve $P(o_{1:k-1}|x_{k-1})$ from a recursive case, so we have solved the problem.

Computing $P(x_k|o_{k+1:n})$ is very similar. We start with the base case of $P(x_{n-1}|o_n)$. This can be computed simply by "rolling out" x_n and then o_n

Question: what if we need to know $P(x_n)$? Well we won't, because if you look at our final equation $P(\mathbf{o}_{1:k}|x_k)P(x_k|\mathbf{o}_{k+1:n})$, if we have $k=n$, then the first term will contain everything we need and the second term will just be 1.

Normalization is needed, because we did a lot of trickery with only dealing with $\propto$

4.2 Message Passing and Loopy Belief Propagation

4.3 Efficient Inference in Dense Gaussian CRFS

5 Linear and Nonlinear Bayesian Models

5.1 Bayesian Linear Regression

5.1.1 Evidence Approximation

5.2 Bayesian Logistic Regression

5.2.1 Laplace Approximation

5.3 Spline and Wavelet Bases

5.4 Bayesian Neural Networks

6 Approximate Inference

6.1 Expectation Maximization

6.2 Variational Inference

7 Sampling Algorithms

7.1 Importance Sampling

7.2 Monte Carlo Markov Chain

7.2.1 Metropolis Hastings

7.2.2 Reversible Jump MCMC

7.3 Gibbs Sampling

8 Optimization

8.1 Lagrange Multipliers

8.2 Lagrangian Duality

8.3 KKT Conditions

8.4 SGD Variants

8.5 BADMM Algorithm

9 Reinforcement Learning