Sampling Methods

Guy Lebanon

November 28, 2006

A large part of statistics and machine learning depends on computing expectations. As a few examples consider: (i) summarizing the posterior distribution $\mathsf{E}\left(g(\theta)|D\right) = \int_\theta g(\theta)p(\theta|D)\,d\theta$ (ii) computing the gradient of the normalization term for maximizing likelihood of exponential models $\mathsf{E}\left(f_i(x)\right) = \int f_i(x)Z^{-1}(\theta)\exp(\sum \theta_i f_i(x))\,dx = \frac{\partial}{\partial\theta_i}\log Z(\theta)$ (iii) marginalizing over missing data $p(x) = \int p(x,z)\,dz = \mathsf{E}_{p(z)}(p(x|z))$ and (iv) computing probabilities $p(X \in C) = E_p(1_C)$.

When the expectation (integral or summation) does not have a closed form we need to resort to approximation techniques. One class of approximation methods is numerical integration techniques such as the trapezoid or Simpson's method. A second class of approximation methods, which we will concentrate on, are based on sampling and the law of large numbers

$$\frac{1}{m}\sum_{i=1}^{m} g(x_i) \approx \mathsf{E}_{p(x)}(g) \quad \text{if} \quad x_1,\ldots,x_m \sim p(x).$$

The above estimator is unbiased and has variance $m^{-1}\mathsf{Var}\left(g(X)\right)$. The convergence above is quite stable and rapid (as indicated by the uniform law of large numbers and large deviation theory) and does not depend on the dimensionality of X. Slow convergence may occur, however, if g is high were p is low and vice verse.

The benefit of sampling methods over numerical integration methods is that they work better in high dimensional cases. Some high dimensional models such as Bayesian networks $p(x) = \prod p(x_i|\text{pa}(x_i))$ are easy to sample from (sample from the conditional distributions - starting from the parents and progressing to the leaves). In other high dimensional models such as exponential family models or Markov random fields, sampling is not straightforward. In general, we will assume that we can sample from a uniform $U([0,1])$ distribution. Sampling from a uniform distribution has been widely studied and many efficient methods for doing so exist.

Histogram Method

To sample from a discrete one dimensional RV X, we can just generate a $r \sim U([0,1])$ random number and compare it with cdf F_X and sample x_i for which $r_i \in [F_X(x_i), F_X(x_{i+1})]$. The above method can be applied to continuous RVs by discretizing them (approximating a continuous RV by its discrete histogram). The method works well for one dimensional RVs but suffers greatly in high dimensional cases.

Transformation Method

We focus on the case of one dimensional distribution. Extensions to high dimensionality models are straightforward. Assume we can sample from a uniform RV in $[0,1]$ and we wish to sample from a RV X. The RV transformation $X \mapsto F_X(X)$ results in a uniform RV

$$P(F_X(X) \le r) = P(F_X^{-1}(F_X(X)) \le F_X^{-1}(r)) = P(X \le F_X^{-1}(r)) = F_X(F_X^{-1}(r)) = r.$$

As a result transforming the uniform samples by $r \mapsto F^{-1}(r)$ results in samples from X. Technically, there is a problem with the method as stated above if the pdf or pmf of X is not strictly positive (F_X is not invertible). A more careful method statement should resolve that difficulty. In low dimensional cases, the above transformation works well. The basic problem of computing F_X and inverting it become difficult in high dimensions and other methods are necessary.

Rejection Sampling

Again, we start with a one dimensional formulation. Assume that we want to sample from p, but we can sample from q instead, and we know further that $p(x) \leq kq(x)$ everywhere for some constant k. Sampling $x_i \sim q$ and then $r_i \sim U([0, kq(x_i)])$ would give a pair (x_i, r_i) which would be uniformly distributed over the graph of the function (area under the function curve) kq. By rejecting the pair if $r_i \geq p(x)$ we ensure that the remaining sample pairs are uniformly distributed over the graph of $p(x)$. We can then discard the r_i and keep the x_i samples which constitute a sample from p.

Rejection sampling can be modified for use if p is known up to a constant $p = c\tilde{p}$ (its normalization term is not easily computable). In this case we find q such that $\tilde{p} \leq kq$ and proceed as before.

Adaptive rejection sampling is way of adaptively computing q and k for distributions p whose logarithm is concave. In this case, we can upper bound the $\log p$ with a piecewise linear function (envelope) computed based on the derivative $\nabla \log p$ at different points. The distribution itself p is then upper bounded by a piecewise exponential function which constitutes the proposal q. As samples get rejected, they are added to the computation of the envelope and the quality of the upper bound improves.

The difficulty here is as before in cases of high dimensionality. It is not clear how to find k and moreover the probability of rejection (grows with k) grows exponentially with the dimensionality. Bishop [1] shows how rejection sampling from $N(\mu, \sigma_p^2 I)$ through the distribution $N(\mu, \sigma_q^2 I)$ would necessitates $k = (\sigma_q/\sigma_p)^d$. The acceptance rate would be 1/20,000 for $d = 1000$ if σ_q exceeds σ_p by just one percent.

Importance Sampling

Importance sampling directly estimates the expectation $\mathsf{E}_p(f)$ by noticing that

$$\mathsf{E}_p(g) = \int g(x) \frac{p(x)}{q(x)} q(x)\, dx = \mathsf{E}_q(g\, p/q)$$

which is approximated by averaging $g(x)\, p(x)/q(x)$ over samples from $x_1, \ldots, x_m \sim q$.

A useful trick is performing importance sampling when we can't evaluate the normalization terms of p and q [1]. In this case $p = \tilde{p}/Z_p$ and $q = \tilde{q}/Z_q$ and

$$\mathsf{E}_p(g) = \frac{Z_q}{Z_p} \int g(x) \frac{\tilde{p}(x)}{\tilde{q}(x)} q(x)\, dx \approx \frac{Z_q}{Z_p} \frac{1}{m} \sum_{i=1}^{m} \frac{\tilde{p}(x_i)}{\tilde{q}(x_i)} g(x_i)$$

where x_i are samples from q. The factor Z_q/Z_p may be approximated as follows

$$\frac{Z_q}{Z_p} = \frac{1}{Z_q} \int \tilde{p}(x)\, dx = \int \frac{\tilde{p}(x)}{\tilde{q}(x)} q(x)\, dx \approx \frac{1}{m} \sum_{i=1}^{m} \frac{\tilde{p}(x_i)}{\tilde{q}(x_i)}$$

where $x_i \sim q$. Putting all this together gives

$$\mathsf{E}_p(g) \approx \sum_{i=1}^{m} w_i g(x_i) \qquad w_i = \frac{\tilde{p}(x_i)/\tilde{q}(x_i)}{\sum_{i=1}^{m} \tilde{p}(x_i)/\tilde{q}(x_i)} \qquad x_i \sim q.$$

As before, the main problem is high dimensions. If p, q are high dimensional, weights $p(x_i)/q(x_i)$ become smaller rapidly. If q is low where pg is high, the estimator will be highly inaccurate since it may take a long time to obtain a meaningful sample.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.