The Natural Gradient – Building Intelligent Probabilistic Systems

The Natural Gradient

A common activity in statistics and machine learning is optimization. For instance, finding maximum likelihood and maximum a posteriori estimates require maximizing the likilihood function and posterior distribution respectively. Another example, and the motivating example for this post, is using variational inference to approximate a posterior distribution. Suppose we are interested in a posterior distribution, p, that we cannot compute analytically. We will approximate p with the variational distribution $q(\phi)$ that is parameterized by the variational parameters ϕ. Variational inference then proceeds to minimize the KL divergence from q to p, $KL(q||p)$. The dominant assumption in machine learning for the form of q is a product distribution, that is $q = \prod_k q_k(\phi_k)$ (where we assume there are K variational parameters). It can be shown that minimizing $KL(q||p)$ is equivalent to maximizing the evidence lower bound [2], denoted $L(\phi)$.

In this post we will consider optimizing the parameters of a probability distribution and will see that using the gradient as in basic calculus can follow suboptimal directions. This can cause slow convergence or convergence to inferior local modes in non-convex problems. Along the way we will introduce some concepts from differential geometry and derive more efficient gradient directions to follow.

For the rest of this post we will assume the setting of variational inference so that the function to optimize is $L(\phi)$ and the components of ϕ are the variational parameters.

Basic calculus tells us that to optimize a continuous function (the objective function) we set the derivative of the function to zero and solve. In all but the simplest cases an analytic solution will not exist and we will need to resort to numerical methods. The simplest numerical optimization algorithm to maximize a function is gradient ascent (descent if one is minimizing a function) which updates the current value of the parameters, $\phi^{(j)}$, to a new value, $\phi^{(j+1)}$, such that $L(\phi^{(j)}) \leq L(\phi^{(j+1)})$ by following the gradient. The update rule is of the form

$$\phi^{(j+1)} = \phi^{(j)} + \gamma \nabla L(\phi)$$

where $\nabla L(\phi) = (\frac{\partial L}{\partial \phi_1}, \ldots, \frac{\partial L}{\partial \phi_K})$ is the gradient vector of $L(\phi)$ and γ determines how for to move along the gradient. The gradient vector points in the direction that the function increases most quickly, where changes in the function are measured with respect to Euclidean distance. In other words, the gradient indicates the direction such that the smallest change in parameter values results in the largest change in function value, where changes in the parameters and function values are measured by the Euclidean distance. If the Euclidean distance between the variables being optimized (the variational parameters in our running example) is not a good measure of variation in the objective function then gradient ascent will move suboptimaly through the parameter values.

The Euclidean distance between parameter values of distributions turns out to not be a good

The Euclidean distance between parameter values of distributions turns out to not be a good measure of the variation between the two distributions. As a simple example [2] consider the two distributions $N(0, 1000)$ and $N(10, 1000)$ as well as the two distributions $N(0, 0.1)$ and $N(0.1, 0.1)$. The first pair of distributions are essentially the same whereas the second two do not share any significant support. Yet, the Euclidean distance between the parameters in the first pair is 10, while in the second the distance is only 0.1. The reason for this is that probability distributions do not naturally live in Euclidean space but rather on a manifold (a statistical manifold to be precise). There are better ways of defining the distance between distributions, one of the simplest being the symmetrized Kullback-Leibler divergence

$$KL_{sym}(p_1, p_2) = \frac{1}{2}(KL(p_1||p_2) + KL(p_2||p_1))$$

which will arise later.

To understand how to improve the gradient ascent algorithm when optimizing the parameters of probability distributions we must understand a bit of differential geometry. Assume that $\phi \in \Phi \subset R^n$ where Φ is a parameter space of interest. When Φ is Euclidean space with an orthonormal basis the squared distance between two points ϕ and $\phi + d\phi$ is given by the usual Euclidean inner product

$$||d\phi||^2 = \langle d\phi, d\phi \rangle$$

When Φ is a manifold the previous distance is given by the bilinear form

$$||d\phi||^2 = \langle d\phi, G(\phi) d\phi \rangle = \sum_{ij} g_{ij}(\phi) d\phi_i d\phi_j$$

The matrix $G(\phi) = [g_{ij}(\phi)]$ is called the Riemannian metric tensor and is an inner product on the tangent space of the manifold Φ at the point ϕ. This means that G allows us to define a norm, and thus distances on the manifold (the tangent space at a point can be thought of as the space of directional derivatives at the point). Note that G depends on the location ϕ, a consequence of the fact that G is defined with respect to the tangent space at ϕ and so is location dependent.

As a concrete example, imagine two people standing on two different mountain tops. If one of the people is Superman (or anybody else who can fly) then the distance they would fly directly to the other person is the Euclidean distance (in R^3). If both people were normal and needed to walk on the surface of the Earth the Riemannian metric tensor tells us what this distance is from the Euclidean distance. Don't take this illustration too seriously since technically all of this needs to take place in a differential patch (a small rectangle whose side lengths go to 0).

Intuitively, the Riemannian metric tensor describes how the geometry of a manifold affects a differential patch, $d\phi$, at the point ϕ. The length of a line between two points on $d\phi$ is the distance between them. The Riemannian metric tensor either stretches or shrinks that line and the resulting length is the distance between the two points on the manifold.

In Euclidean space with an orthonormal basis $G(\phi)$ is simply the identity matrix. When Φ is a space of parameters of probability distributions and the symmetrized KL divergence is used to measure the distance between distributions then $G(\phi)$ turns out to be the Fisher information matrix. This arises from placing an inner product on the statistical manifold of log-probability densities [3].

As stated above, the gradient vector of a function $L(\phi)$ points in the direction of steepest ascent. We can formally write this as maximize $L(\phi + d\phi)$ such that $||d\phi||^2 < \epsilon^2$ for ϵ small. To solve this optimization problem we set $d\phi = \epsilon v$ for some vector v. We can then rewrite the optimization problem as maximizing

$$L(\phi + d\phi) = L(\phi) + \epsilon \nabla L(\phi)^T v$$

under the constraint that $||v||^2 =< v, G(\phi)v >= 1$, that is v must be unit length under the inner product defined by $G(\phi)$ and so we are searching for the direction that causes the function to increase the most where distances arise from the inner product defined by G. We can rewrite this constrained optimization problem as an unconstrained problem using Lagrange mulitpliers

$$L(\phi) + \nabla L(\phi)^T v - \lambda v^T G(\phi) v$$

and after taking the derivative with respect to v and setting it to zero we obtain $\nabla L(\phi) = 2\lambda G(\phi) v$ and so $v = \frac{1}{2\lambda} G^{-1} \nabla L(\phi)$. We can then solve for λ using the unit-vector constraint on v. We see that when Φ is Euclidean space and G is the identity matrix then the usual gradient vector of L is the solution. When G is non-trivial the vector $G(\phi)^{-1} \nabla L(\phi)$ is called the natural gradient and as we have just seen is the direction that L increases most quickly when the parameter space is a manifold with Riemannian metric G. This tells us that a more efficient variational inference algorithm is to follow the natural gradient of the variational parameters, where the Riemannian metric tensor is just the Fisher information matrix of the variational distribution.

The derivations in this paper followed that in [1] which goes on to show what the natural gradient looks like for neural networks and other models and proves that gradient descent with the natural gradient is efficient in a certain sense. The application of differential geometry to statistics is a deep subject. See the book “Differential Geometry and Statistics” [3] for a nice introduction. Perhaps in a future post we can delve into why the Fisher information matrix appears as the Riemannian metric tensor for statistical manifolds.

References:

[1] Amari, S., Natural Gradient Works Efficiently in Learning. In Neural Computation, Vol. 10, No. 2, 1998.

[2] Hoffman, M., Blei, D. M., Wang, C., Paisley, J., Stochastic Variational Inference. arXiv: 1206.7051.

[3] Murray, M.K., Rice, J.W., Differential Geometry and Statistics. Monographs on Statistics and Applied Probability, No. 48, 1993.

Posted in Computation, Machine Learning, Statistics.

By Nick Foti – January 25, 2013 No comments