CS/CNS/EE 253: Advanced Topics in Machine Learning
Topic: GP Classification & Active Nonparametric Learning **Lecturer:** Andreas Krause
Scribe: Daniel Erenrich **Date:** March 3, 2010

16.1 Review

Recall from the previous lectures that non-parametric learning is a process which accepts a data and returns a continuous valued response with indications of uncertainty about these responses. That is, we can predict the form of any function and these predictions are paired with confidence bands concerning the predictions. In particular we discussed the method of Gaussian processes. We take a Bayesian approach by looking at functions of the form

$$P(f) = GP(f; \mu, k) \tag{16.1.1}$$

where μ is the mean (assumed for the moment to be zero) and k is our kernel (covariance) function. Where we let k be

$$k = c \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^{\mathbf{2}}/\sigma^{\mathbf{2}}\right) \tag{16.1.2}$$

where c is the magnitude and σ is a measure of the scale of the change. In this way we solve general problems with the uncertainty bounds we desired.

Note that we assumed that the observations we see are determined by an underlying Gaussian distribution.

$$P(y(x)|f(x)) = \mathcal{N}(y(x), f(x), \sigma^2) \tag{16.1.3}$$

With which we want $P(x|y_A = y') = GP$ which should give us a posterior mean and confidence bounds.

16.2 GP Classification

We would like to use similar techniques in order to solve a (non-linear) classification problem. That is, we would like it to provide uncertainty estimates in the label-space. The process would be similar to the result of an SVM. The key difference is that SVMs do not give a clear indication of confidence bounds they only give us a some notion of a distance from a bounding plane. We would like our classification method when run over the data in Figure 16.2 to give us a result like of Figure 16.2.

The naive approach to doing this with minimal changes to our GP would be to learn on the above curve directly as if we were not performing classification. The problem with this is that the result may not match our expected boundary conditions. We expect the resultant distribution to integrate to 1 and to always be in the range $(0, 1)$.

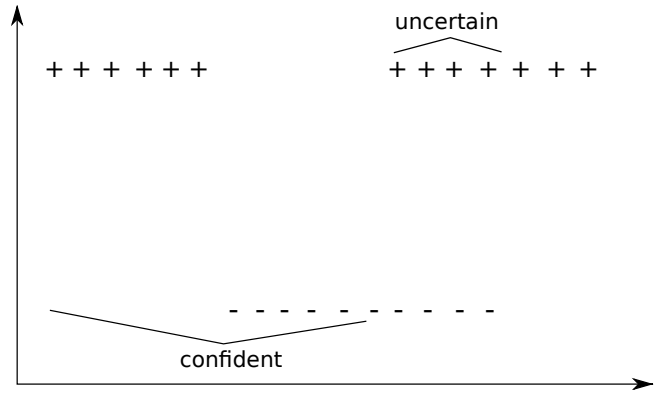

Figure 16.2.1: A classification problem

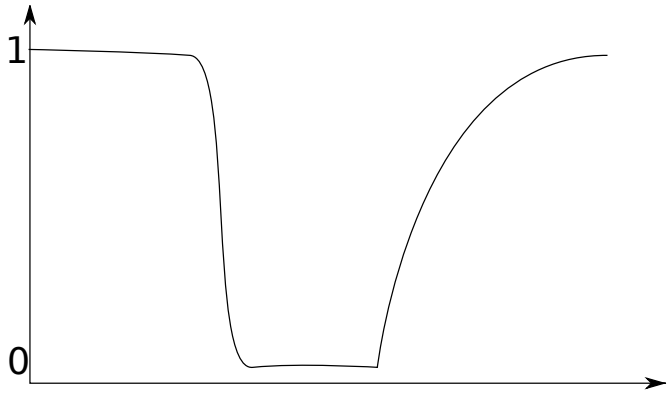

Figure 16.2.2: The desired result of $P(y = 1|x)$

The more advanced way would be to apply a squashing function Φ which will ensure that the above properties hold. We now look at the probability function of the form

$$P(y(x) = 1|\Phi(f(x|D))) \tag{16.2.4}$$

Our choice of the squashing function is the sigmoid defined as

$$\Phi(f) = \frac{1}{1 + \exp(-f)} \tag{16.2.5}$$

which is graphed in Figure 16.2.1. To predict this probability distribution we assume that there is an underlying GP that models dependencies. The actual values that we input though are determined the by squashing function. This is technique is related to the method known as logistic regression.

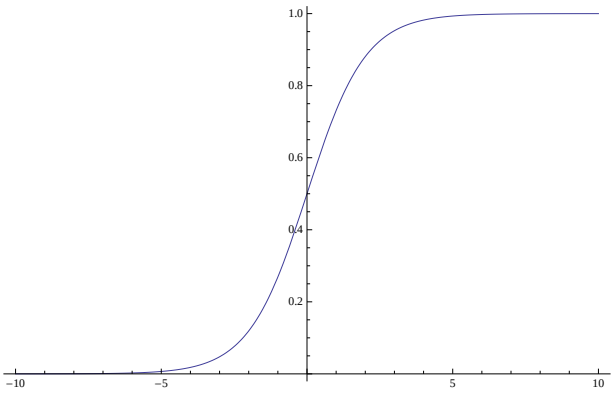

Figure 16.2.3: The plot of the sigmoid function.

16.2.1 Non-Gaussian Observations

Up to now we have been assuming that observations were drawn from an underlying Gaussian distribution. What if the observations are Bernoulli observations where the following holds

$$P(y(x) = 1|D) = \Phi(f(x|D)) \tag{16.2.6}$$

We would like to compute $P(y(x) = 1|D) = P(y(x) = 1|y_A = y_1)$. We can do this by integrating out these functions

$$\int P(y(x) = 1, f|y_A = y')df = \int P(y(x) = 1|f, y_A = y')P(f|y_A = y')df \tag{16.2.7}$$

Intuitively here we are summing over all values that agree with our data. This is easy to solve if our posterior is a Gaussian distribution, but what about when it is not?

The idea to solve this problem is to approximate $p(F|y_A = y)$ by a Gaussian of the form $\tilde{p}(f) = \mathcal{N}(f_i, \hat{\mu}, \hat{\Sigma})$. We can determine the proper value for $\hat{\mu}$ by computing the following

$$\hat{\mu} = \arg\max_f P(f|y_A) \tag{16.2.8}$$

where we can compute the maximization using convex optimization. Intuitively we are centering our distribution around the max of the skewed distribution.

Computing the optimal value of $\hat{\Sigma}$ is more difficult. We compute it as

$$\hat{\Sigma} = (-\nabla\nabla \log(f|y_A))^{-1} \tag{16.2.9}$$

or in words we take the inverse of the negative of the Hessian of the log probability. We rationalize this by noting that $P(f|y_A) = \mathcal{N}(f, \mu, \Sigma)$ which implies that $p \propto \exp(-.5f^T\Sigma f)$ (assuming μ is zero) and so in the simplified case $\frac{d^2}{dx^2} = .5\sigma^{-1}f^2$.

This technique is known as Laplace Approximation. It returns a non-skewed distribution centered at the maximal point of the original skewed distribution (or if it is run on a normal distribution it returns same distribution).

There are some issues to using the Laplace Approximation. It is a poor approximation in cases where a distribution has multiple modes and in cases where the distribution is very skewed. In the distributions graphed, the maximum is noted. This is where the normal distribution under the Laplace Approximation will be centered. Note that the normal is not a good approximation in these cases. Fortunately, when we have large amounts of data the distribution is about Gaussian which is not skewed and with only a single mode.

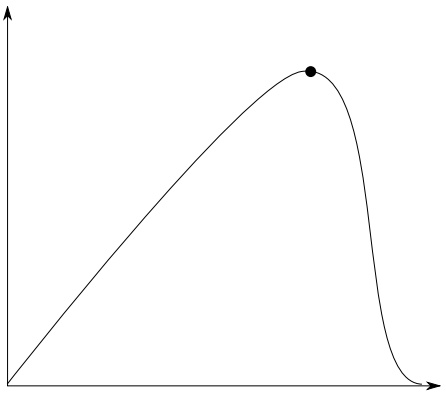

Figure 16.2.4: A skewed distribution.

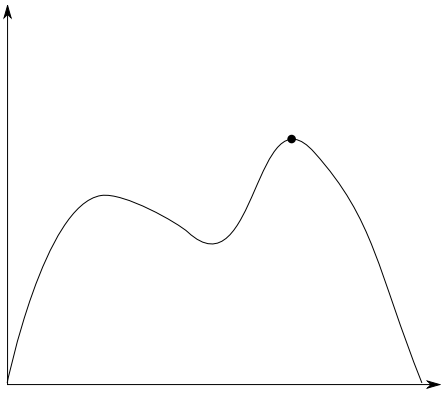

Figure 16.2.5: A distribution with multiple nodes.

An alternative to Laplace Approximation is KL approximation. This method is not covered in detail. Intuitively, we find a distribution such that

$$\hat{P} = \arg\min_{p'} = KL(p||p') \tag{16.2.10}$$

Sadly solving this minimization is difficult and so we approximate the solution using expectation propagation (EP). The end result though is a more diffused and more cautious distribution which matches the purpose of GP classification (giving us more meaningful probabilities of being wrong).

We can also avoid approximating in this manner by instead modeling all together. For example, rain fall is often modeled as a Poisson distribution where the mean varies as a Gaussian process. We can use this method with any density.

16.3 Active Non-Parametric Learning

In the rest of the course we will consider combining methods discussed in the course. These topics are online, active and non-parametric learning. Here we will cover active non-parametric learning. Consider the problem of non-parametric learning where we can choose our next input to evaluate. Which inputs do we ask for if we want to best estimate f?

The simplest approach is called uncertainty sampling or maximum entropy sampling (MES). We ask for our next point in the region where our current confidence is lowest. We find the current confidence bands and pick a point where the width is greatest.

MES

1 $\mu_t(x) =$ posterior mean after t observations
2 $\sigma_t(x) =$ posterior variance after t observations
3 **return** $\arg\max_x \sigma_t(x)$

MES is the greedy algorithm for active non-parametric learning.

16.3.1 Aside: Differential Entropy

If $y \in \mathbb{R}$ and we have $p(y)$ we define the (differential) entropy to be $H(y) = -\int_{-\infty}^{\infty} p(y) \lg p(y) dy$ which in the discrete case is $-\sum p(y) \lg p(y)$. The discrete case has interesting code theoretic meanings. The main idea is that entropy conveys some notion of uncertainty.

For Gaussians with mean μ and variance σ^2 we have the entropy being $H(y) = \frac{1}{2} \lg (2\pi e \sigma^2)$ which is a constant plus the log of the standard deviation. Note that there is no dependence on the mean. In the higher dimensional case we have the entropy being $0.5 \lg [(2\pi e)^n |\Sigma|]$ where n is the dimension.

Some properties of entropy are the flowing. The chain rule tells us that if $z = (x, y)$ then $H(z) = H(x) + H(y|x)$. The information never hurts rule tells us that if $z = (x, y, w)$ then $H(x|y) \geq H(x|y, w)$. The information gain $I(x, y)$ is defined to be $H(x) - H(x|y)$ which the same as $H(y) - H(y|x)$. If y tells us a lot about x the the difference here will be large (the information gain is large).

16.3.2 Back To GP Active Learning Solution

We want to select a set of inputs that are maximally informative. This means that we want a set A such that $I(f, y_a) = H(f) - H(f|y_A) = H(y_A) - H(y_A|f)$ is maximized. We note for clarity that $y_A = f_A + \epsilon_A$ and $f_A \sim \mathcal{N}(0, \Sigma_{AA})$ and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ which implies that $y_A \sim \mathcal{N}(0, \Sigma_{AA} + \sigma^2 I)$. We use this to compute $H(y_A) = 0.5 \lg ((2\pi e)^k |[\Sigma_{AA} + \sigma^2 I]|)$ where $A = \{x_1, \ldots, x_k\}$. For convenience

we define the function F to be

$$F(A) = I(y_A; f) \tag{16.3.11}$$

Brute forcing over the set A is infeasible. In fact computing

$$A^* = \arg\max_A F(A) \tag{16.3.12}$$

where $|A| \leq k$ is NP-hard. The question now is, how well does our greedy algorithm approximate this optimal but intractable solution? This question will be answered more rigorously in the next lecture, but the answer is that it the greedy solution is no worse than a constant multiple of the optimal solution.