CS/CNS/EE 253: Advanced Topics in Machine Learning
Topic: Nonparametric learning and Gaussian processes **Lecturer:** Andreas Krause
Scribe: Nathan Watson **Date:** Feb 24, 2010

14.1 Review

From the last lecture, we have the following general formulation for learning problems:

$$f^* = \min_{f \in \mathcal{H}_k} ||f||^2 + \sum_i l(y_i, f(x_i)) \tag{14.1.1}$$

We have already seen one specific selection for the loss function l: the hinge loss function, as used by support vector machines (SVMs). In general, the abstraction of loss functions is a very powerful mechanism, allowing the same general optimization problem to be used in various learning algorithms for different purposes.

14.2 Loss functions

14.2.1 Hinge loss

The hinge loss function is the following:

$$l(y, f(x)) = \max(0, 1 - y \cdot f(x)) \tag{14.2.2}$$

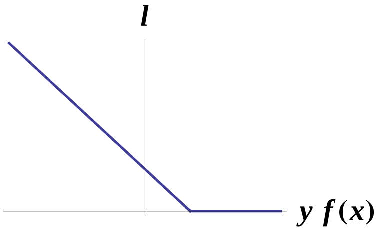

Figure 14.2.1: A plot of a typical hinge loss function.

Hinge loss works well for its purposes in SVM as a classifier, since the more you violate the margin, the higher the penalty is. However, hinge loss is not well-suited for regression-based problems as a result of its one-sided error. Luckily, various other loss functions are more suitable for regression.

14.2.2 Square loss

The square loss function is the following:

$$l(y, f(x)) = (y - f(x))^2 \tag{14.2.3}$$

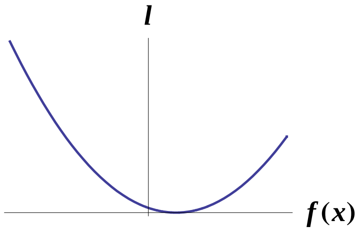

Figure 14.2.2: A plot of a typical square loss function.

Square loss is one such function that is well-suited for the purpose of regression problems. However, it suffers from one critical flaw: outliers in the data (isolated points that are far from the desired target function) are punished very heavily by the squaring of the error. As a result, data must be filtered for outliers first, or else the fit from this loss function may not be desirable.

14.2.3 Absolute loss

The absolute loss function is the following:

$$l(y, f(x)) = |y - f(x)| \tag{14.2.4}$$

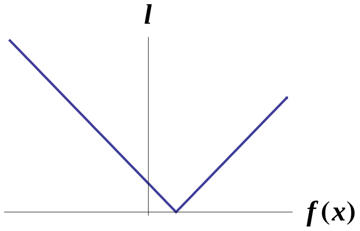

Figure 14.2.3: A plot of a typical absolute loss function.

Absolute loss is applicable to regression problems just like square loss, and it avoids the problem of weighting outliers too strongly by scaling the loss only linearly instead of quadratically by the error amount.

14.2.4 ϵ-insensitive loss

The ϵ-insensitive loss function is the following:

$$l(y, f(x)) = |y - f(x)| \tag{14.2.5}$$

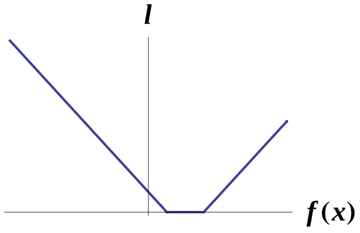

Figure 14.2.4: A plot of a typical ϵ-insensitive loss function.

This loss function is ideal when small amounts of error (for example, in noisy data) are acceptable. It is identical in behavior to the absolute loss function, except that any points within some selected range ϵ incur no error at all. This error-free margin makes the loss function an ideal candidate for support vector regression. (With SVMs, we had $f = \sum_i \alpha_i \mathrm{k}(x_i, \cdot)$, and solutions tended to be sparse; i.e., most $\alpha_i = 0$, and the support vectors were those points for which $\alpha_i \neq 0$. With a suitable selection of ϵ, similar sparsity of solutions tend to result from the usage of the ϵ-insensitive loss function in regression-based learning algorithms.)

There are many more loss functions other than those listed above that are used in practice in machine learning, so it is recommended to remember the general framework for learning problems presented in Equation 14.1.1.

14.3 Reproducing kernel Hilbert spaces

A natural question from the above formulation of learning problems would be: When can we apply this framework? That is, what exactly is $\mathcal{H}_k$, and what functions does it encompass?

Formally, $\mathcal{H}_k$ is known as a "reproducing kernel Hilbert space" (RKHS). This means that $\mathcal{H}_k$ is a Hilbert space with some inner product $\langle \cdot, \cdot \rangle$ and some positive definite kernel function $k : X \times X \to \mathbb{R}$ with the following pair of properties:

$$\mathcal{H}_k = \{f : f = \sum_{i=1}^{\infty} \alpha_i \mathrm{k}(x_i, \cdot)\} \tag{14.3.6}$$

In plain English, this means that the space consists of all functions resulting from a linear combination of kernel evaluations.

$$\langle f, k(x_i, \cdot) \rangle = f(x_i) \tag{14.3.7}$$

In an intuitive sense, this means that the kernel functions can be thought of as a kind of basis for the space.

To illustrate these concepts, consider the example of the square exponential kernel. Let $X \subset \mathbb{R}^n$, and $k(x, x') = \exp\left(\frac{-||x-x'||^2}{h}\right)$. Evaluating this kernel at specific points results in Gaussian distributions.

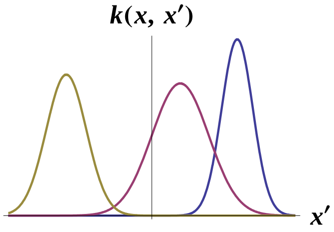

Figure 14.3.5: A plot of the square exponential evaluated at various points.

We can also use functions that are linear combinations of these bell curves (sums of Gaussians). (As a side note, if we consider superpositions of infinitely many Gaussians, we get a dense set that is capable of approximating any continuous function.)

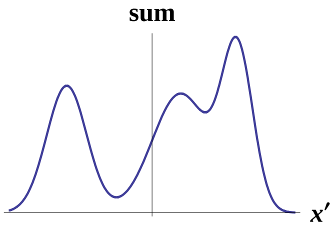

Figure 14.3.6: A plot of the sum of curves in Figure 14.3.5.

14.4 The Representer Theorem

Theorem 14.4.1 *For any data set* $\{(x_1, y_1), \ldots, (x_n, y_n)\}, \exists \alpha_1, \ldots, \alpha_n$ *such that* $f^* \in \textit{argmin}\left(\frac{1}{2}||f||^2 + \sum_i l(y_i, f(x_i))\right)$ *can be instead written as* $f^* = \sum_i \alpha_i k(x_i, \cdot)$.

What follows is a relatively straightforward proof of the above theorem, along with a geometric representation to develop intuition for the theorem.

Lemma 14.4.2 *Let* $\mathcal{H}_k$ *be an RKHS, with* $\mathcal{H}'$ *as a subspace of* $\mathcal{H}_k$. *We can write* $\mathcal{H}_k = \mathcal{H}' \oplus \mathcal{H}_\perp$ *such that any* $f \in \mathcal{H}_k$ *can be uniquely represented as* $f = f_{||} + f_\perp, f_{||} \in \mathcal{H}', f_\perp \in \mathcal{H}_\perp$. *Furthermore,* $\forall f_{||} \in \mathcal{H}', f_\perp \in \mathcal{H}_\perp : \langle f_{||}, f_\perp \rangle = 0$. *Lastly,* $||f||^2 = ||f_{||}||^2 + ||f_\perp||^2$.

Proof: Let D be the data set, and define $\mathcal{H}' = \{f : f = \sum_{i=1}^{n} \alpha_i \mathrm{k}(x_i, \cdot)\}$, and let $\mathcal{H}_\perp$ be the orthogonal complement of $\mathcal{H}'$. Now, pick any $f \in \mathcal{H}_k$, with $f = f_{||} + f_\perp$. For any data point $x_j \in D$. From the definition of RKHS, it follows that $\langle f, k(x_j, \cdot)\rangle = f(x_j)$. Additionally, splitting f into $f_{||} + f_\perp$ gives:

$$\langle f, k(x_j, \cdot)\rangle = \langle f_{||} + f_\perp, k(x_j, \cdot)\rangle = \langle f_{||}, k(x_j, \cdot)\rangle + \langle f_\perp, k(x_j, \cdot)\rangle$$

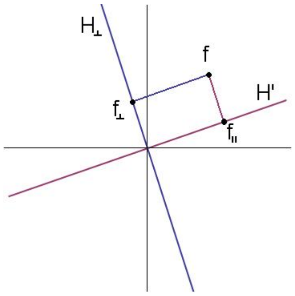

Figure 14.4.7: A diagram of geometric projections to parallel and perpendicular components.

However, from the fact that $f_\perp$ lies strictly in the orthogonal component $\mathcal{H}_\perp$, it follows that $\langle f_\perp, k(x_j, \cdot)\rangle = 0$.

Now, we let $L(f)$ be the total loss, $L(f) = \sum_i l(y_i, f(x_i))$. Varying the orthogonal component does not change its contribution to the loss at all; the contribution remains zero. As a result, it follows that $L(f) = L(f_{||})$. Since $||f||^2 = ||f_{||}||^2 + ||f_\perp||^2$ and varying $f_\perp$ cannot reduce $L(f)$, the minimum of $(\frac{1}{2}||f||^2 + L(f))$ must necessarily come only when $f_\perp = 0$, since this minimizes the contribution of $f_\perp$ to $||f||^2$. Thus, f^* is composed only of $f_{||}$, lying solely in $\mathcal{H}'$. This suffices to prove the representer theorem. ■

14.5 Nonparametric regression

Suppose we want to learn some function $f : X \to \mathbb{R}$, but we do not have any prior knowledge about f, so f might be any arbitrary function. We could solve such a regression problem by using, for example, the square loss function described earlier.

$$\min_{f \in \mathcal{H}_k} \left(\frac{1}{2}||f||^2 + \sum_i (y_i - f(x_i))^2 \right) \tag{14.5.8}$$

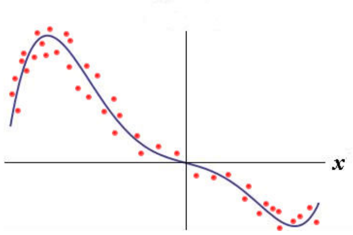

Figure 14.5.8: An example of using regression to determine a suitable function f.

This method will give us a single function that has a good fit to the data we have. However, consider a case in which the given data set contains many points in particular areas, but with large gaps of little or no data between the clusters. The absence of data would not disrupt the process of finding a suitable function to fit the given data, but the resulting function could be very different from the target function in the areas where the given data set has gaps.

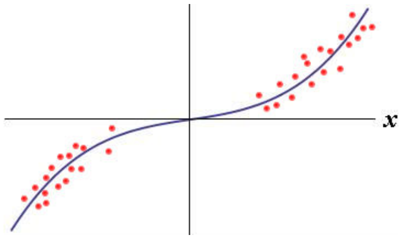

Figure 14.5.9: Using regression on data with a gap.

A natural problem that arises from this is to determine a way to quantify how certain we are of the quality of the fit at various points on the function derived from regression on the data. It would be beneficial to devise a way to place confidence intervals around the function to reflect the areas of uncertainty. To accomplish this task, we frame the problem of regression slightly differently, thinking in terms of fitting a distribution $P(f)$ over the target function f rather than the target function itself.

Intuitively, we desire the properties that low values of $||f||$ yield high values of $P(f)$, and high values of $||f||$ yield low values of $P(f)$ (highly erratic functions are less likely to match the target function than relatively simple functions). If we have the prior distribution $P(f)$, and the likelihood $P(y|f,x)$, we can compute the posterior distribution $P(f|y,x)$ via application of Bayes' theorem: $P(f|y,x) = \frac{P(f)P(y|f,x)}{P(y|x)}$.

Two questions arise from this setup of the problem: What might be a suitable prior distribution $P(f)$, and how can we compute $P(f|D)$? To answer these questions, we turn to the simplest distribution available: the Gaussian distribution.

14.6 Gaussian processes

As a brief review, the following two equations are for the one-dimensional and n-dimensional Gaussian distributions, respectively.

$$f \in \mathbb{R}.\ P(f) = \mathcal{N}(f;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(f-\mu)^2}{2\sigma^2}\right) \tag{14.6.9}$$

$$f \in \mathbb{R}^n.\ P(f) = \mathcal{N}(f;\mu,\Sigma) = (2\pi|\Sigma|)^{(-n/2)} \exp\left(-\frac{1}{2}(f-\mu)^\top \Sigma^{-1}(f-\mu)\right) \tag{14.6.10}$$

For the n-dimensional case, $\mu \in \mathbb{R}^n$ is the mean vector, and $\Sigma \in \mathbb{R}^n \times \mathbb{R}^n$ is the positive definite covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & & \vdots \\ \vdots & & \ddots & \vdots \\ \sigma_{n1} & \dots & \dots & \sigma_n^2 \end{pmatrix}$$

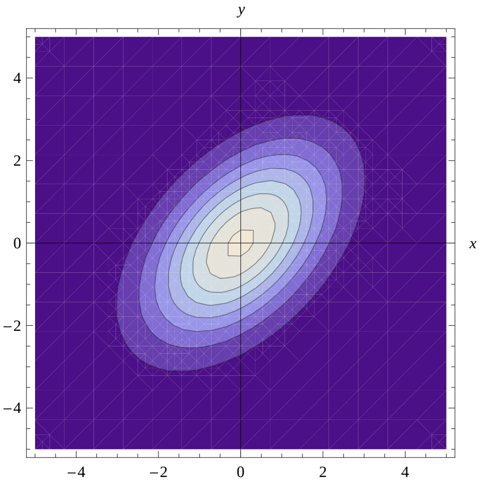

Figure 14.6.10: A sample plot of two Gaussian distributions. Fixing the value of x gives $P(y|x)$.

We can write $P(f_1) = \int\int\int \dots \int p(f_1 \dots f_n) df_2 \dots df_n$. At first, this looks very daunting, but the expression evaluates to a Gaussian, $\mathcal{N}(f_1; \mu_1, \sigma_1^2)$. In general, if we take some subset $A = \{i_1, \dots, i_k\} \subset \{1, \dots, n\}$, then for all of the functions $f_A = (f_{i_1}, \dots, f_{i_k})$, the distribution of this is also Gaussian: $P(f_A) = \mathcal{N}(f_A; \mu_A, \Sigma_{AA})$. This is an example of marginalization.

If we take two subsets $A, B \subset \{1, \dots, n\}$, we can say that $P(f_A|f_B = f') = \mathcal{N}(f_A; \mu_{A|B}, \Sigma_{A|B})$, where $\mu_{A|B} = \mu_A + (\Sigma_{AB}\Sigma_{BB}^{-1})(f' - \mu_B)$ and $\Sigma_{A|B} = \begin{pmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{pmatrix}$. This is an example of conditioning.

A Gaussian Process (GP) is a collection of random variables with index set X for the function $f(x)$; for all $x \in X$, there exists a positive definite kernel (or "covariance function") $k : X \times X \to \mathbb{R}$ and a mean function $\mu : X \to \mathbb{R}$. If $A \subset X, |A| < \infty$, then $P(f_A) = \mathcal{N}(f_A; \mu_A, \Sigma_{AA})$ with $\mu_A = (\mu(x_1) \dots \mu(x_n))$ and Σ_{AA} defined as the kernel matrix:

$$\Sigma_{AA} = \begin{pmatrix} k(x_1, x_1) & \dots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) \end{pmatrix}$$

In plain English, with Gaussian processes, any finite subset of indices selected from the index set form a Gaussian distribution.