9.520: Statistical Learning Theory and Applications March 3rd, 2010

Manifold Regularization

Lecturer: Lorenzo Rosasco *Scribe: Hooyoung Chung*

1 Introduction

In this lecture we introduce a class of learning algorithms, collectively called *manifold regularization* algorithms, suited for predicting/classifying data embedded in high-dimensional spaces. We introduce manifold regularization in the framework of *semi-supervised learning*, a generalization of the supervised learning setting in which our training set may consist of unlabeled as well as labeled examples.

2 Semi-supervised learning

Recall that in supervised learning, the case with which we have dealt up to this point, we are given a training set S consisting of labeled examples (x_i, y_i) drawn from a space $X \times Y$ according to some probability distribution $p(x, y)$. In semi-supervised learning, we are given u examples

$$x_1, \ldots, x_u$$

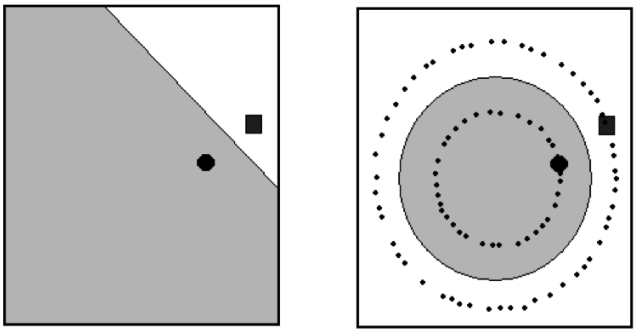
drawn from the marginal distribution $p(x)$. The first n points $x_1, \ldots, x_n$ (with $n \leq u$) are given labels

$$y_1, \ldots, y_n$$

drawn from the conditional distributions $p(y \mid x)$. The value of n is typically smaller than u by orders of magnitude.

An intuitive example for semi-supervised learning is human learning. For example, a child may learn to classify members of a category of objects by being provided with a small number of labeled examples and observing many more unlabeled examples in its environment. In this problem setting (image labeling) and others, it can be preferable to use a semi-supervised learning algorithm due to the high expense of labeling examples.

It is clear from our characterization of the semi-supervised learning problem that the role of the $u-n$ unlabeled training examples can only be to provide additional information about the *distribution* of the input space X. The following diagram may help to intuitively justify the assertion that this additional information is useful. In the two graphs, the same two (+)- and (−)-labeled examples (represented by a square and circle) have been given. On the right, additional unlabeled examples have been provided. The curves separating the shaded and unshaded regions in both graphs are plausible decision boundaries.



The classifier depicted in the second graph seems intuitively better since the boundary cuts through a region of low density in data points. On the other hand, if the distribution on the input space is highly uniform, it is unclear whether unlabeled data points drawn randomly from that distribution can provide much benefit. In general, learning in a uniform input space of high dimension is difficult.

3 Learning in high dimension

Intuitively, it seems that—provided the training set is sufficiently large, and that the conditional distribution is not pathologically discontinuous—one can always approximate the conditional distribution $p(y \mid x)$ by interpolating from the labels of "nearby" points—say the k nearest. Perhaps it is reasonable to select $k = rn$ for some fixed fraction r, say $r = 0.01$. If we are in some bounded space, such as the d-dimensional hypercube $[0,1]^d \subset \mathbb{R}^d$, intuition may also suggest that we can collect $\geq k$ nearby points by taking some small rectangle around the point x whose label we wish to predict.

The edge length of the d-dimensional hypercube required to capture a fraction r of the sampled points in $[0,1]^d$, assuming uniform distribution, is $r^{1/d}$. For instance, if $r = 0.01$ and $d = 10$, then $r^{1/d} = 0.63$. This means that finding the 1% of the training set closest to x requires examining 63% of the range of each input dimension. This behavior is a manifestation of the so-called "curse of dimensionality"; roughly speaking, local methods—looking at small neighborhoods of a point in the input space—are not likely to be effective at estimating outputs if the input space is truly high-dimensional.

Fortunately, it is often enough the case that the data we wish to analyze possess some sort of "intrinsic geometry" and are of effectively lower dimension than the entire putative input space. For instance, this is true if the data are the result of some process involving a relatively small number of degrees of freedom. Some examples:

1. Pose variations: the positions of arbitrarily many points on the arm and hand are governed by a small number of joint angles;
2. Facial expressions: these are controlled by the tone of a small number of facial muscles.

4 Manifold regularization

In this lecture, we make the more particular assumption that our data lie in some *Riemannian manifold* of smaller dimension than the input space, which has been embedded into the input space. We refer to the input space as the *ambient space*.

A Riemannian manifold $\mathcal{M}$ is a set of points which is everywhere "locally similar" to $\mathbb{R}^d$. Formally, for each point $x \in \mathcal{M}$, some neighborhood of x can be associated with a smooth bijective map α from the neighborhood into $\mathbb{R}^d$. The manifold can be expressed as a union of these neighborhoods:

$$\mathcal{M} = \bigcup_{\alpha} U_\alpha$$

where every point is in some U_α. The map $\alpha \colon U_\alpha \to \mathbb{R}^d$ is called a *system of coordinates* for U_α. Further, if two neighborhoods intersect ($U_\alpha \cap U_\beta \neq \emptyset$) we require that the *transition function* $\beta \circ \alpha^{-1} \colon \alpha(U_\alpha \cap U_\beta) \to \mathbb{R}^d$ and its inverse be smooth. Using the systems of coordinates, it is possible to define analogues of most Euclidean geometric concepts, including angles, distances, volumes, and so forth.

In particular, given a manifold $\mathcal{M}$ of dimension d, it is possible to define an analogue of the gradient operator ∇ which applies to functions on $\mathcal{M}$. Recall that in $\mathbb{R}^d$, the gradient of a function

f is given by

$$\nabla f(x) \equiv \left(\frac{\partial f}{\partial x_1}(x), \ldots \frac{\partial f}{\partial x_d}(x) \right).$$

The gradient points in a direction in $\mathbb{R}^d$ along which change in $f(x)$ is maximized, and has magnitude representative of the amount of variation. The *gradient of f with respect to $\mathcal{M}$*, denoted $\nabla_{\mathcal{M}} f(x)$, likewise represents the magnitude and direction of maximal variation at x within $\mathcal{M}$.

Suppose $p(x)$ is the marginal distribution of input points to a function we wish to learn. If f is some element of the hypothesis space, then

$$S(f) := \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f(x)\|^2 \, dp(x)$$

is a measure of the smoothness of f with respect to $\mathcal{M}$. If f varies widely near points in $\mathcal{M}$ of high probability density, S will be large. Hence S is a natural penalty function for regularization over functions on $\mathcal{M}$. Extending Tikhonov regularization with this additional term, an optimal $f \in \mathcal{H}$ is given by

$$f^* = \operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i) + \lambda_A \|f\|_K^2 + \lambda_I \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f(x)\|^2 \, dp(x).$$

Here the parameter λ_A regularizes with respect to the ambient space, whereas λ_I regularizes with respect to the intrinsic geometry $\mathcal{M}$. The term λ_A is necessary since the manifold $\mathcal{M}$ is a strict subset of the input space X; among many $f \in \mathcal{H}$ which give the same value of $S(f)$—perhaps because they are identical on $\mathcal{M}$—we prefer a solution which is smooth in the ambient space.

Using Stokes' theorem, we can rewrite $S(f)$ in terms of the *Laplacian* with respect to $\mathcal{M}$ giving the following equivalent form of the optimum:

$$f^* = \operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i) + \lambda_A \|f\|_K^2 + \lambda_I \int_{\mathcal{M}} f(x) \Delta_{\mathcal{M}} f(x) \, dp(x). \tag{1}$$

(Recall that in $\mathbb{R}^d$, the Laplacian of a function f is given by

$$\Delta f(x) \equiv -\frac{\partial^2 f}{\partial x_1^2}(x) - \cdots - \frac{\partial^2 f}{\partial x_d^2}(x);$$

$\Delta_{\mathcal{M}} f$ is the analogue on $\mathcal{M}$.)

5 The graph Laplacian

Since we are not given $\mathcal{M}$ and the embedding $\boldsymbol{\phi}\colon \mathcal{M} \to \mathbb{R}^d$, we cannot precisely compute the smoothness penalty $S(f) = \int_{\mathcal{M}} f(x) \Delta_{\mathcal{M}} f(x) \, dp(x)$. Instead, we use an empirical proxy for S based on the assumption that the input points are drawn i.i.d. from the uniform distribution on $\mathcal{M}$.[1]

Consider the *weighted neighborhood graph* G given by taking the graph on vertex set $V = \{x_1, \ldots, x_u\}$ (the labeled and unlabeled input points) with edges (x_i, x_j) if and only if $\|x_i - x_j\|^2 \leq \varepsilon$, and assigning to edge (x_i, x_j) the weight

$$W_{ij} = \exp\left(-\frac{1}{\varepsilon} \|x_i - x_j\|^2 \right).$$

[1] Note that this trivially implies each $x \in \boldsymbol{\phi}(\mathcal{M})$, which is not satisfied if the training set also incorporates noise, e.g. $x = \phi(p) + \mathbf{N}(0, \varepsilon)$, with $\phi(p) \in \mathcal{M}$.

The *graph Laplacian* of G is the matrix $\mathbf{L}$ given by

$$L_{ij} = D_{ij} - W_{ij}, \quad \text{where } \mathbf{D} = \operatorname{diag}\Big(\sum_{j=1}^{u} W_{ij}\Big)_{i=1}^{u}.$$

(i.e., D is the diagonal matrix whose ith entry is the sum of the weights of edges leaving x_i.) The graph Laplacian is a discrete analogue of the manifold Laplacian: one can show that if $\mathbf{f} = (f(x_1), \ldots, f(x_u))$ is the vector given by evaluating an arbitrary f at each input point, then we can write

$$\sum_{i=1}^{u}\sum_{j=1}^{u} W_{ij}(f_i - f_j)^2 = \mathbf{f}^T\mathbf{L}\mathbf{f} \approx u^2 \int_{\mathcal{M}} f(x)\Delta_{\mathcal{M}} f(x)\, dp(x).$$

Using this approximation, the minimization problem (1) becomes

$$f^* = \operatorname*{arg\,min}_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i) + \lambda_A\|f\|_K^2 + \frac{\lambda_I}{u^2}\mathbf{f}^T\mathbf{L}\mathbf{f}. \tag{2}$$

This problem depends on the training set points, the regularization parameters λ_A, λ_I, and the parameter ε used to construct the neighborhood graph. Choosing V to be the square loss or the hinge loss, respectively, we recover natural generalizations of regularized least squares and SVM to semi-supervised learning. As in the case of Tikhonov regularization, we have a *representer theorem*: it is possible to write the solution to (2) as a linear combination of representers of the training set; that is,

$$f^*(x) = \sum_{i=1}^{u} c_i K(x_i, x)$$

for some u-tuple $\mathbf{c} = (c_1, \ldots, c_u)$. (The proof is similar to the one sketched in lecture 3.)

6 Manifold RLS and SVM

As examples, we show how to generalize regularized least squares and SVM to the semi-supervised setting. Taking $V(f(x_i), y_i) = (f(x_i) - y_i)^2$, using the representer theorem in (1) gives

$$\mathbf{c}^* = \operatorname*{arg\,min}_{\mathbf{c}\in\mathbb{R}^u} \frac{1}{n}(\mathbf{y} - \mathbf{J}\mathbf{K}\mathbf{c})^T(\mathbf{y} - \mathbf{J}\mathbf{K}\mathbf{c}) + \lambda_A\mathbf{c}^T\mathbf{K}\mathbf{c} + \frac{\lambda_I}{u^2}\mathbf{c}^T\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{c},$$

where $\mathbf{c}^*$ is such that $f^*(x) = \sum_{i=1}^{u} c_i^* K(x_i, x)$. Here $\mathbf{y} = (y_1, \ldots, y_n, 0, \ldots, 0) \in \mathbb{R}^u$, and $\mathbf{J}$ is the $n \times n$ diagonal matrix with ones in the first n diagonal entries and zeroes in the remaining $u - n$ entries; the effect of $\mathbf{J}$ is to disregard the value of f predicted for the unlabeled training points.

Since the functional is strictly convex and differentiable, we can solve for $\mathbf{c}^*$ by setting the derivative equal to zero. This yields the solution

$$\mathbf{c}^* = \mathbf{M}^{-1}\mathbf{y},$$

where

$$\mathbf{M} = \mathbf{J}\mathbf{K} + \lambda_A n\mathbf{I} + \frac{\lambda_I n^2}{u^2}\mathbf{L}\mathbf{K}.$$

Similarly, we can use the hinge loss $V(f(x_i), y_i) = (1 - y_i f(x_i))_+$ in (1), apply the representer

theorem, create slack variables ξ_i and add an unpenalized bias b, yielding

$$\begin{aligned} \mathbf{c}^* = \underset{\mathbf{c}\in\mathbb{R}^u, \boldsymbol{\xi}\in\mathbb{R}^n}{\arg\min} \; & \frac{1}{n}\sum_{i=1}^{n}\xi_i + \lambda_A \mathbf{c}^T\mathbf{K}\mathbf{c} + \frac{\lambda_I}{u^2}\mathbf{c}^T\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{c} \\ \text{subject to} \quad & y_i\Big(\sum_{j=1}^{u} c_j K_{ij} + b\Big) \geq 1 - \xi_i \quad \text{for } i = 1, \ldots, n; \\ & \xi_i \geq 0 \quad \text{for } i = 1, \ldots, n. \end{aligned}$$

This can be transformed to the dual problem

$$\begin{aligned} \boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}\in\mathbb{R}^n}{\arg\max} \; & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} \\ \text{subject to} \quad & \sum_{i=1}^{n} y_i\alpha_i = 0 \\ & 0 \leq \alpha_i \leq \frac{1}{n} \quad \text{for } i = 1, \ldots, n, \end{aligned}$$

where

$$\mathbf{Q} = \mathbf{y}^T\mathbf{J}\mathbf{K}\Big(2\lambda_A\mathbf{I} + 2\frac{\lambda_I}{u^2}\mathbf{L}\mathbf{K}\Big)^{-1}\mathbf{J}^T\mathbf{y}.$$

7 Remarks

We present some final remarks.

Convergence of the smoothness penalty. A result on convergence of the empirical estimate of the smoothness penalty is as follows. For a point x_i in the training set, note $(\mathbf{L}\mathbf{f})_i = \sum_j \big(f(x_i) - f(x_j)\big)\exp\big(-\frac{1}{\varepsilon}\|x_i - x_j\|^2\big)$. We extend this to an operator $\mathcal{L}$ on $\mathcal{H}$, given by

$$(\mathcal{L}f)(x) = \sum_j (f(x) - f(x_j))\exp\Big(-\frac{1}{\varepsilon}\|x - x_j\|^2\Big).$$

We then have the following convergence theorem[2] due to Belkin and Niyogi [3]:

Theorem 1 *Let the training points $\{x_1, \ldots, x_u\}$ be sampled from the uniform distribution over a manifold $\mathcal{M}$ (of dimension d) embedded in X. Put $\varepsilon = u^{-\alpha}$, where $0 < \alpha < \frac{1}{2+d}$. Then for all $f \in C^\infty$ and $x \in X$, there exists a constant C such that*

$$C\frac{\varepsilon^{-\frac{d+2}{2}}}{u}(\mathcal{L}f)(x) \to \Delta_{\mathcal{M}} f(x) \quad \textit{in probability as } u \to \infty.$$

Spectral properties of $\Delta_{\mathcal{M}}$ and L. Significant information about the space $\mathcal{M}$ can be gleaned from the spectrum of $\Delta_{\mathcal{M}}$. It can be shown that for $\mathcal{M}$ topologically compact, the eigenfunctions of $\Delta_{\mathcal{M}}$ form a countable basis for $L_2(\mathcal{M})$. If $\mathcal{M}$ is a connected space, $f(x) \equiv 1$ is the unique eigenfunction of $\Delta_{\mathcal{M}}$ having eigenvalue 0. The spectrum of $\mathbf{L}$ likewise encodes useful information; for instance, the smallest nonzero eigenvalue of $\mathbf{L}$ is the size of the minimum cut on G. In the *Laplacian eigenmap algorithm* [2], the training data $\{x_1, \ldots, x_n\}$ are projected onto the eigenvectors of the Laplacian; this map preserves local distances, allowing dimensionality reduction in the case that the input space has dimension $\gg n$.

[2]The version presented is from the class slides [1].

Computational complexity. The graph Laplacian is a dense $u \times u$ matrix. We can achieve better performance and overcome some space limitations by sparsifying $\mathbf{L}$, e.g. by adding an edge between x_i and x_j in the weighted neighborhood graph only if x_j is a kth-nearest-neighbor of x_i or vice versa.

References

[1] Lorenzo Rosasco, "Manifold Regularization." `http://web.mit.edu/9.520/www/Classes/class07_stability_2010.pdf`

[2] M. Belkin and P. Niyogi, *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation.* Neural Computation, 15 (6):1373-1396, June 2003.

[3] M. Belkin and P. Niyogi, *Towards a Theoretical Foundation for Laplacian-Based Manifold Methods.* Proc. of Computational Learning Theory, 2005.