Manifold Regularization

Lorenzo Rosasco

9.520 Class 08

March 2, 2008

About this class

Goal To analyze the limits of learning from examples in high dimensional spaces. To introduce the semi-supervised setting and the use of unlabeled data to learn the intrinsic geometry of a problem. To define Riemannian Manifolds, Manifold Laplacians, Graph Laplacians. To introduce a new class of algorithms based on Manifold Regularization (LapRLS, LapSVM).

Unlabeled data

Why using unlabeled data?

- labeling is often an “expensive” process
- semi-supervised learning is the natural setting for human learning

Semi-supervised Setting

u i.i.d. samples drawn on X from the marginal distribution $p(x)$

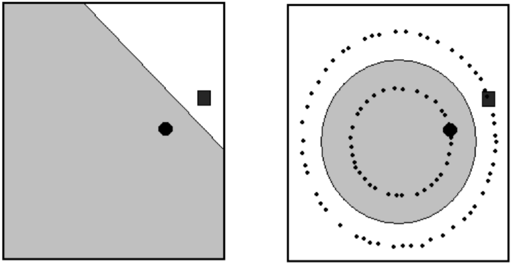
$$\{x_1, x_2, \dots, x_u\},$$

only n of which endowed with labels drawn from the conditional distributions $p(y|x)$

$$\{y_1, y_2, \dots, y_n\}.$$

The extra $u - n$ unlabeled samples give additional information about the marginal distribution $p(x)$.

The importance of unlabeled data



Curse of dimensionality and $p(x)$

Assume X is the D-dimensional hypercube $[0,1]^D$. The worst case scenario corresponds to uniform marginal distribution $p(x)$.

Local Methods

A prototype example of the effect of high dimentionality can be seen in nearest methods techniques. As d increases, local techniques (eg nearest neighbors) become rapidly ineffective.

Curse of dimensionality and k-NN

- It would seem that with a reasonably large set of training data, we could always approximate the conditional expectation by k-nearest-neighbor averaging.
- We should be able to find a fairly large set of observations close to any $x \in [0, 1]^D$ and average them.
- This approach and our **intuition break down in high dimensions.**

Sparse sampling in high dimension

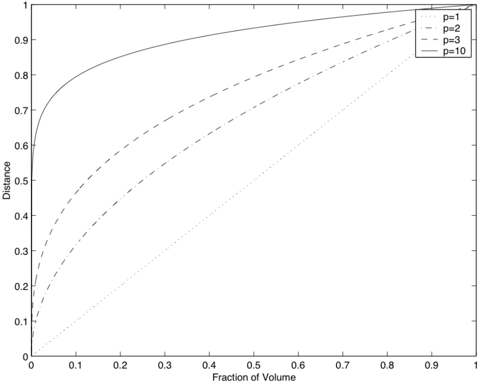
Suppose we send out a cubical neighborhood about one vertex to capture a fraction r of the observations. Since this corresponds to a fraction r of the unit volume, the expected edge length will be

$$e_D(r) = r^{\frac{1}{D}}.$$

Already in ten dimensions $e_{10}(0.01) = 0.63$, that is to capture 1% of the data, we must cover 63% of the range of each input variable!

No more ”local” neighborhoods!

Distance vs volume in high dimensions



Intrinsic dimensionality

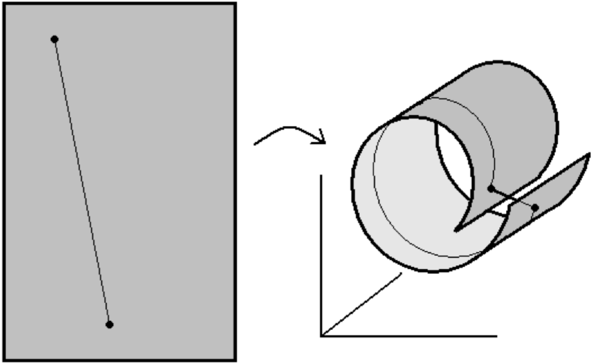
Raw format of natural data is often high dimensional, but in many cases it is the outcome of some process involving only *few degrees of freedom*.

Examples:

- Acoustic Phonetics ⇒ vocal tract can be modelled as a sequence of few tubes.
- Facial Expressions ⇒ tonus of several facial muscles control facial expression.
- Pose Variations ⇒ several joint angles control the combined pose of the elbow-wrist-finger system.

Smoothness assumption: y's are “smooth” relative to natural degrees of freedom, **not** relative to the raw format.

Manifold embedding



Riemannian Manifolds

A *d*-dimensional manifold

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

is a mathematical object that generalizes domains in $\mathbb{R}^d$.
Each one of the “patches” U_α which cover $\mathcal{M}$ is endowed with a *system of coordinates*

$$\alpha : U_\alpha \to \mathbb{R}^d.$$

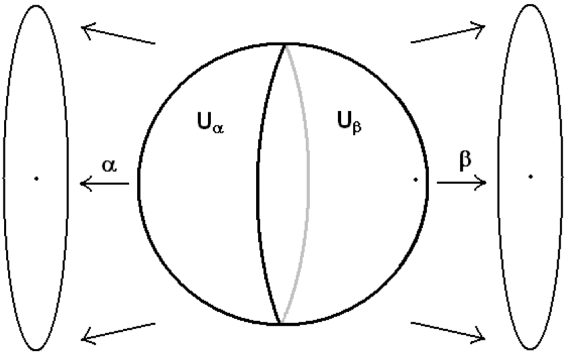
If two patches U_α and U_β, overlap, the *transition functions*

$$\beta \circ \alpha^{-1} : \alpha(U_\alpha \bigcap U_\beta) \to \mathbb{R}^d$$

must be smooth (eg. infinitely differentiable).

- The Riemannian Manifold inherits from its local system of coordinates, most geometrical notions available on $\mathbb{R}^d$: **metrics, angles, volumes, etc.**

Manifold's charts



Differentiation over manifolds

Since each point x over $\mathcal{M}$ is equipped with a local system of coordinates in $\mathbb{R}^d$ (its *tangent space*), all **differential operators** defined on functions over $\mathbb{R}^d$, can be extended to analogous operators on functions over $\mathcal{M}$.

Gradient: $\nabla f(\mathbf{x}) = (\frac{\partial}{\partial x_1} f(\mathbf{x}), \dots, \frac{\partial}{\partial x_d} f(\mathbf{x})) \Rightarrow \nabla_{\mathcal{M}} f(x)$

Laplacian: $\triangle f(\mathbf{x}) = -\frac{\partial^2}{\partial x_1^2} f(\mathbf{x}) - \cdots - \frac{\partial^2}{\partial x_d^2} f(\mathbf{x}) \Rightarrow \triangle_{\mathcal{M}} f(x)$

Measuring smoothness over $\mathcal{M}$

Given $f : \mathcal{M} \to \mathbb{R}$

- $\nabla_{\mathcal{M}} f(x)$ represents amplitude and direction of variation around x
- $S(f) = \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f(x)\|^2 dp(x)$ is a global measure of smoothness for f
- Stokes' theorem (generalization of integration by parts) links gradient and Laplacian

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

Manifold regularization Belkin, Niyogi,Sindhwani, 04

A new class of techniques which extend standard Tikhonov regularization over RKHS, introducing the additional regularizer $\|f\|_I^2 = \int_{\mathcal{M}} f(x)\triangle_{\mathcal{M}} f(x)dp(x)$ to enforce smoothness of solutions relative to the underlying manifold

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

- λ_I controls the complexity of the solution in the **intrinsic** geometry of $\mathcal{M}$.
- λ_A controls the complexity of the solution in the **ambient** space.

Manifold regularization (cont.)

Other natural choices of $\|\cdot\|_I^2$ exist

- Iterated Laplacians $\int_{\mathcal{M}} f \triangle_{\mathcal{M}}^s f$ and their linear combinations. These smoothness penalties are related to Sobolev spaces

$$\int f(x) \triangle_{\mathcal{M}}^s f(x) dp(x) \approx \sum_{\omega \in Z^d} \|\omega\|^{2s} |\hat{f}(\omega)|^2$$

- Frobenius norm of the Hessian (the matrix of second derivatives of f) Hessian Eigenmaps; Donoho, Grimes 03
- Diffusion regularizers $\int_{\mathcal{M}} f e^{t\triangle}(f)$. The semigroup of smoothing operators $G = \{e^{-t\triangle_{\mathcal{M}}} | t > 0\}$ corresponds to the process of diffusion (Brownian motion) on the manifold.

An empirical proxy of the manifold

We cannot compute the intrinsic smoothness penalty

$$\|f\|_I^2 = \int_{\mathcal{M}} f(x) \triangle_{\mathcal{M}} f(x) dp(x)$$

because we don't know the manifold $\mathcal{M}$ and the embedding

$$\Phi : \mathcal{M} \to \mathbb{R}^D.$$

But we assume that the unlabeled samples are drawn i.i.d. from the uniform probability distribution over $\mathcal{M}$ and then mapped into $\mathbb{R}^D$ by Φ

Neighborhood graph

Our proxy of the manifold is a *weighted neighborhood graph* $G = (V, E, W)$, with **vertices** V given by the points $\{x_1, x_2, \dots, x_u\}$, **edges** E defined by one of the two following adjacency rules

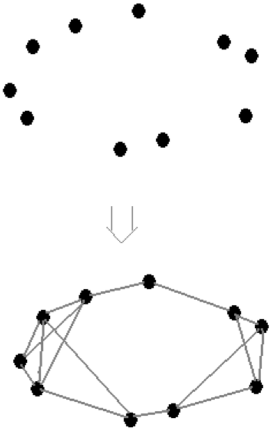
- connect x_i to its k nearest neighborhoods
- connect x_i to ϵ-close points

and **weights** W_{ij} associated to two connected vertices

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\epsilon}}$$

Note: computational complexity $O(u^2)$

Neighborhood graph (cont.)



The graph Laplacian

The *graph Laplacian* over the weighted neighborhood graph (G, E, W) is the matrix

$$\mathbf{L}_{ij} = \mathbf{D}_{ii} - \mathbf{W}_{ij}, \qquad \mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}.$$

L is the discrete counterpart of the manifold Laplacian $\triangle_{\mathcal{M}}$

$$\mathbf{f}^T\mathbf{L}\mathbf{f} = \sum_{i,j=1}^{n} \mathbf{W}_{ij}(\mathbf{f}_i - \mathbf{f}_j)^2 \approx \int_{\mathcal{M}} \|\nabla f(x)\|^2 dp(x).$$

Analogous properties of the *eigensystem*: nonnegative spectrum, null space

Looking for rigorous convergence results

A convergence theorem *Belkin, Niyogi, 05*

Operator $\mathcal{L}$: “out-of-sample extension” of the graph Laplacian **L**

$$\mathcal{L}(f)(x) = \sum_i (f(x) - f(x_i)) e^{-\frac{\|x - x_i\|^2}{\epsilon}} \quad x \in X, \ \ f : X \to \mathbb{R}$$

Theorem: Let the u data points $\{x_1, \ldots, x_u\}$ be sampled from the uniform distribution over the embedded d-dimensional manifold $\mathcal{M}$. Put $\epsilon = u^{-\alpha}$, with $0 < \alpha < \frac{1}{2+d}$. Then for all $f \in C^\infty$ and $x \in X$, there is a constant C, s.t. in probability,

$$\lim_{u \to \infty} C \frac{\epsilon^{-\frac{d+2}{2}}}{u} \mathcal{L}(f)(x) = \triangle_{\mathcal{M}} f(x).$$

Laplacian-based regularization algorithms (Belkin et al. 04)

Replacing the unknown manifold Laplacian with the graph Laplacian $\|f\|_I^2 = \frac{1}{u^2}\mathbf{f}^T\mathbf{L}\mathbf{f}$, where $\mathbf{f}$ is the vector $[f(x_1), \ldots, f(x_u)]$, we get the minimization problem

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

- $\lambda_I = 0$: standard regularization (RLS and SVM)
- $\lambda_A \to 0$: out-of-sample extension for Graph Regularization
- $n = 0$: unsupervised learning, Spectral Clustering

The Representer Theorem

Using the same type of reasoning used in Class 3, a Representer Theorem can be easily proved for the solutions of Manifold Regularization algorithms.
The expansion range over all the **supervised and unsupervised** data points

$$f(x) = \sum_{j=1}^{u} c_j K(x, x_j).$$

LapRLS

Generalizes the usual RLS algorithm to the semi-supervised setting.
Set $V(w, y) = (w - y)^2$ in the general functional.
By the representer theorem, the minimization problem can be restated as follows

$$\mathbf{c}^* = \arg\min_{\mathbf{c}\in\mathbb{R}^u} \frac{1}{n}(\mathbf{y} - \mathbf{JKc})^T(\mathbf{y} - \mathbf{JKc}) + \lambda_A \mathbf{c}^T\mathbf{Kc} + \frac{\lambda_I}{u^2}\mathbf{c}^T\mathbf{KLKc},$$

where $\mathbf{y}$ is the u-dimensional vector $(y_1, \ldots, y_n, 0, \ldots, 0)$, and $\mathbf{J}$ is the $u \times u$ matrix $diag(1, \ldots, 1, 0, \ldots, 0)$.

LapRLS (cont.)

The functional is differentiable, strictly convex and coercive. The derivative of the object function vanishes at the minimizer $\mathbf{c}^*$

$$\frac{1}{n}\mathbf{KJ}(\mathbf{y} - \mathbf{JKc}^*) + (\lambda_A \mathbf{K} + \frac{\lambda_I n}{u^2}\mathbf{KLK})\mathbf{c}^* = 0.$$

From the relation above and noticing that due to the positivity of λ_A, the matrix $\mathbf{M}$ defined below, is invertible, we get

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

where

$$\mathbf{M} = \mathbf{JK} + \lambda_A n\mathbf{I} + \frac{\lambda_I n^2}{u^2}\mathbf{LK}.$$

LapSVM

Generalizes the usual SVM algorithm to the semi-supervised setting.
Set $V(w, y) = (1 - yw)_+$ in the general functional above.
Applying the representer theorem, introducing *slack variables* and adding the unpenalized *bias term b*, we easily get the primal problem

$$
\begin{aligned}
\mathbf{c}^* = \arg \min_{\mathbf{c} \in \mathbb{R}^u, \xi \in \mathbb{R}^n} \quad & \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda_A \mathbf{c}^T \mathbf{K} \mathbf{c} + \frac{\lambda_I}{u^2} \mathbf{c}^T \mathbf{K} \mathbf{L} \mathbf{K} \mathbf{c} \\
\text{subject to :} \quad & y_i \Big(\sum_{j=1}^{u} c_j K(x_i, x_j) + b \Big) \geq 1 - \xi_i \qquad i = 1, \ldots, n \\
& \xi_i \geq 0 \qquad i = 1, \ldots, n
\end{aligned}
$$

LapSVM: the dual program

Substituting in our expression for **c**, we are left with the following "dual" program:

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

Here, vQ is the matrix defined by

$$\mathbf{Q} = \mathbf{YJK} \left(2\lambda_A \mathbf{I} + 2\frac{\lambda_I}{u^2} \mathbf{LK} \right)^{-1} \mathbf{J}^T \mathbf{Y}.$$

One can use a standard SVM solver with the matrix Q above, hence compute c solving a linear system.

Numerical experiments

http://manifold.cs.uchicago.edu/manifold_regularization

- Two Moons Dataset
- Handwritten Digit Recognition
- Spoken Letter Recognition

Spectral Properties of the Laplacian

Ideas similar to those described in this class can be used in other learning tasks. The spectral properties of the (graph-) Laplacian turns out to be useful:

If M is *compact*, the operator $\triangle_{\mathcal{M}}$ has a *countable* sequence of eigenvectors ϕ_k (with *non-negative* eigenvalues λ_k), which is a complete system of $L_2(\mathcal{M})$. If M is *connected*, the constant function is the only eigenvector corresponding to null eigenvalue.

Manifold Learning

The Laplacian allows to exploit some geometric features of the manifold.

- **Dimensionality reduction.** If we project the data on the eigenvectors of the graph Laplacian we obtain the so called Laplacian eigenmap algorithm. It can be shown that such a feature map preserves local distances.
- **Spectral clustering.** The smallest non-null eigenvalue of the Laplacian is the value of the minimum cut on the graph and the associated eigenvector is the cut.