Semi-Supervised Learning on Riemannian Manifolds

University of Vienna

Peter Mühlbacher

October 1, 2014

Abstract

Contents

Introduction

The contentual layout follows that of Belkin (2003). Some definitions, theorems and examples will be taken from this work as well.

Chapter 1

Mapping Graphs to the Real Line

1.1 Classification

Given a space X and a probability distribution τ on $X \times \mathbb{R}$ [1], a *regression problem* is the problem of finding a function $f : X \to \mathbb{R}$, $f(x) = \mathrm{E}_{\tau_x}$, where E_{τ_x} is the expected value of a random variable Y which has the distribution τ_x (the marginal distribution of x on $\mathbb{R}$).

In real-world applications we have a finite set of samples $(x_1, y_1), \ldots, (x_n, y_n)$ from which we reconstruct a function that should

1. describe the given examples well
2. make the probability of making large errors for new examples small

This problem is closely related to the idea of interpolation and as we already know fulfilling the first requirement perfectly hardly ever goes hand in hand with the second one. In practice one wants to balance the complexity [2] of a function and its ability to describe given data.

Classification may be seen as a special case of regression where we aim to find a function $f : X \to \{c_1, \ldots, c_n\}$.

[1] another way to put this would be that for every $x \in X$ there exists a probability distribution τ_x specifying how likely this x goes together with some real number r

[2] Depending on the concrete implementation there are different definitions of complexity. In the theory of splines (or kernel methods in machine learning) $||f||_{\mathbb{H}}$, where $\mathbb{H}$ is a reproducing kernel Hilbert space, may be seen as one; there also exist combinatorial approaches and in this work we will introduce a smoothness functional to be minimized.

1.2 An Implicit Definition of the Classifier

1.2.1 The Discrete Case

To develop a better understanding the discrete case will be dealt with in the first place.

Given a set of points $\{x_i\}_{i=1,\dots,n}$, $x_i \in \mathbb{R}^l$ a weighted graph G can be constructed. For determining which nodes are *adjacent* we will resort to either of two major common definitions:

1. **k nearest neighbours**: find the k nearest nodes
2. **ϵ-neighbourhoods**: two nodes x, y are connected if and only if $||x - y|| < \epsilon$

Nearby nodes x_i, x_j are connected with weights $W_{ij} = e^{-\frac{||x_i - x_j||^2}{t}}$, the weight between unconnected nodes is set to 0.

depending on the length of the work either remark that an explanation will be given later on (heat diffusion resembles this (=Gaussian heat kernel) or not at all (in this case set weights to 1 (which may also be seen as $t \to \infty$, all mathematical properties should stay the same))

Consider the problem of mapping the weighted graph to the real line so that connected points stay together as close as possible.[3] As the above defined weights of adjacent nodes are defined in such a way that they are indirectly proportional to their (euclidian) distance it seems natural that $(y_1, \dots, y_n)^T$ is a good map if

$$\sum_{ij} (y_i - y_j)^2 W_{ij} \tag{1.1}$$

is minimal.

Let D be the degree matrix of the graph G, then its Laplacian matrix $\mathcal{L}$ is defined as

Definition 1.1. $L := D - W$

and (1.1) may be written as

$$\begin{aligned} \textstyle\sum_{ij}(y_i - y_j)^2 W_{ij} &= \textstyle\sum_{ij}(y_i^2 + y_j^2 - 2y_i y_j) W_{ij} \\ &= \textstyle\sum_i y_i^2 D_{ii} + \sum_j y_j^2 D_{jj} - 2\sum_{ij} y_i y_j W_{ij} \\ &= 2y^T L y \end{aligned}$$

Not only do we see that the minimization problem reduces to finding

$$\arg\min_y \; y^T L y$$

[3] For simplicity's sake assume that the graph is connected.

but also that L is positive semidefinite.

However, an additional constraint ($\langle y, L\mathbf{1}\rangle = 0$, where $\mathbf{1}$ is the constant function mapping all nodes to 1; it is easy to see that this is an eigenfunction to the eigenvalue 0) is needed as the trivial solution is not excluded.

Additionaly, we want to assign some kind of importance to each of the nodes, based on how many adjacent nodes it has; we do this by adding the additional constraint $y^T Dy = 1$ which removes an arbitrary scaling factor on the real line.

To solve $\arg\min_{y^T Dy=1} y^L y$ we make use of Lagrange multipliers:

$$\frac{\partial}{\partial y}(y^L y - \lambda(y^T Dy - 1)) = 2Ly - 2\lambda Dy = (L - \lambda D)y \stackrel{!}{=} 0$$

Thus we arrive at the generalized Eigenvalue problem $\mathcal{L}y = \lambda Dy$. (Note that the first constraint was not taken into account, so we have to ignore the trivial solution.)

explain why eigenvalues are important here

1.2.2 The Continuous Case

Definition 1.2. $\mathcal{L} := -\text{div} \circ \nabla$

In analogy to the discrete case let $f : \mathcal{M} \to \mathbb{R}$ be a function that maps points with "small" geodesic distance "close" together on the real line and assume $f \in C^2$.

Appendix A

Implementations

Bibliography

Belkin, M. (2003), *Problems of Learning on Manifolds*. Ph.D. thesis, The University of Chicago.