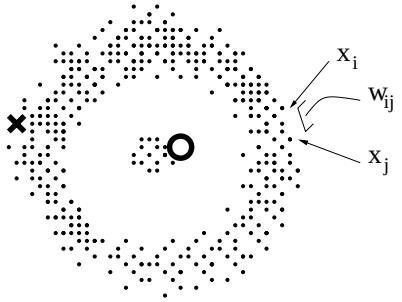
Lecture 12 - *Semi-Supervised Learning with Kernels*

OBJECTIVE: In this lecture, we show how placing kernels on data points allows us to project them to nonlinear manifolds where it becomes easier to do semi-supervised learning. The lecture is based on the paper of Zhu, Lafferty and Ghahramani: "Semi-Supervised Learning Using Gaussian Fields" and the paper of Smola and Kondor: "Kernels and Regularization on Graphs".

Semi-Supervised Learning

We are given N feature vectors $\mathbf{x} \in \mathbb{R}^d$ as shown for $d = 2$ in Fig. 2. Some of the points have labels. In the figure, two points have the labels $y^l = 1$ and $y^l = 0$. The rest of the points have unknown labels y^u. The goal is to compute the unknown labels.



Input data: Two points $(\times)$ *and* (o) *have labels* $y^l = 1$ *and* $y^l = 0$ *respectively. The remaining points are unlabelled* $y^u =?$*, but their topology is essential to the construction of a good classifier.*

A human would classify all the points in the outer ring as 1 and the points in the inner circle as 0. We want an algorithm that reproduces this perceptual grouping.

We do this by considering each point $\mathbf{x}_i$ as a node in a fully connected undirected graph. The edges of the graph have weights corresponding to a similarity kernel. In our case, the weight between points $\mathbf{x}_i$ and $\mathbf{x}_j$ is

$$w_{ij} = \exp\left(-\frac{1}{\sigma}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right),$$

where $\|\cdot\|$ denotes the L_2 norm. Hence, points that are close together will have high similarity (high w), whereas points that are far apart will have low similarity.

It is sensible to minimize the following error function to compute the unknown labels y^u:

$$E(y^u) = \frac{1}{2}\left(\sum_{i\in L, j\in L} w_{ij}(y_i^l - y_j^l)^2 + 2\sum_{i\in U, j\in L} w_{ij}(y_i^u - y_j^l)^2 + \sum_{i\in U, j\in U} w_{ij}(y_i^u - y_j^u)^2\right),$$

where L is the set of labelled points and U is the set of unlabelled points. If two points are close then w will be large. Hence, the only way to minimize the error function is to make these two nearby points have the same label y.

Let us introduce the adjacency matrix $\mathbf{W}$ with entries w_{ij} and the following block structure:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_{ll} & \mathbf{W}_{lu} \\ \mathbf{W}_{ul} & \mathbf{W}_{uu} \end{pmatrix}.$$

where $\mathbf{W}_{uu}$ denotes the entries w_{ij} with $i, j \in U$.

Let $\mathbf{D} = diag(d_i)$ where $d_i = \sum_j w_{ij}$. That is,

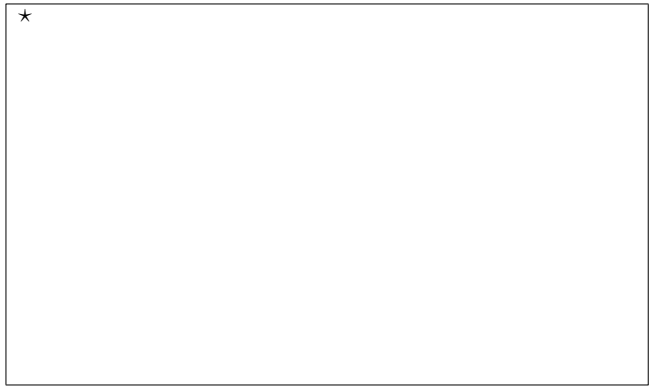
$$\mathbf{D} = \begin{pmatrix} \sum_j w_{1j} & 0 & 0 & \cdots & 0 \\ 0 & \sum_j w_{2j} & 0 & \cdots & 0 \\ & & \vdots & & \\ 0 & \cdots & 0 & 0 & \sum_j w_{Nj} \end{pmatrix}$$

$\mathbf{D}$ is a diagonal matrix whose i-th diagonal entry is the sum of the entries of row i of $\mathbf{W}$. Let the vector $\mathbf{y}_u$ contain all the unknown labels y^u and similarly let $\mathbf{y}_l$ contain all the labels y^l. Then, the error function can be written in matrix notation as follows:

$$E(\mathbf{y}_u) = \mathbf{y}_u^T(\mathbf{D}_{uu} - \mathbf{W}_{uu})\mathbf{y}_u - 2\mathbf{y}_l^T\mathbf{W}_{ul}\mathbf{y}_u + \mathbf{y}_l^T(\mathbf{D}_{ll} - \mathbf{W}_{ll})\mathbf{y}_l,$$

$$E(\mathbf{y}_u) = \begin{pmatrix} \mathbf{y}_l & \mathbf{y}_u \end{pmatrix} \begin{pmatrix} \mathbf{D}_{ll} - \mathbf{W}_{ll} & -\mathbf{W}_{lu} \\ -\mathbf{W}_{ul} & \mathbf{D}_{uu} - \mathbf{W}_{uu} \end{pmatrix} \begin{pmatrix} \mathbf{y}_l \\ \mathbf{y}_u \end{pmatrix}$$

Differentiating this error function and equating to zero, gives us our solution in terms of a linear system of equations:



where $0 \leq y^u \leq 1$. A naive solution would cost $O(M^3)$, where $M = |U|$ is the number of unlabelled points, since $|L|$ is significantly smaller than $|U|$. However, using fast n-body methods in conjunction with the conjugate gradients method for solving linear systems, it is possible to reduce the computation to $O(M \log M)$, and even $O(M)$ in some cases.

The solution that we have just obtained is actually not very different from kernel ridge regression. Assume that the regulariser is set to zero; $\delta = 0$. Then the equations for kernel ridge regression are as follows:

$$\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{y} \quad \text{and} \quad \widehat{\mathbf{y}}(\mathbf{x}^\star) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}^\star, \mathbf{x}_i)$$

or in our current language:

$$\mathbf{y}_u = \mathbf{K}_{ul}\mathbf{K}_{ll}^{-1}\mathbf{y}_l$$

We have simply made the choice $\mathbf{K} = \mathbf{L}^{-1} = (\mathbf{D} - \mathbf{W})^{-1}$

⋆ Proof sketch:

If we have many classes, we can use the same equation with voting

⋆ Proof sketch:

Once we have labels for all N points in the training data, a new point $\mathbf{x}_k$ in the test set data can be classified using the following classical kernel discriminant (Nadaraya-Watson estimate):

$$y_k = \frac{\sum_{i=1}^{N} w_{ik} y_i}{\sum_{i=1}^{N} w_{ik}}.$$

EXAMPLE: Aibo is an interactive robot that learns to recognize objects using semi-supervised input from a portable computer.