1 Kernel methods

Kernel methods help in using linear decision boundaries on non-linearly separable data by morphing the feature space. They can also help in disentangling data for clustering. Kernels can incorporate domain knowledge into your model. It is possible to create vectors for objects that have no direct vector space representation.

1.1 Introduction to Kernels

Kernels can be seen in two ways: As similarity measures between data-points or transformations of the data-points into a higher dimensional space. As a similarity measure a kernel can be understood as how much an object has to be transformed to be identical to the other. For two points x and x' a Kernel is given by:

$$K(x,x') = \sum_{i=1}^{n} h_i(x)h_i(x') = \langle h(x), h(x') \rangle, \tag{1}$$

where $h(x)$ is a transformation-function and $\langle \cdot \rangle$ is the inner product.

Building Kernels Kernels can be constructed by combining known kernels:

$$k = k_1 + k_2$$
$$k = k_1 * k_2$$
$$k = \lambda * k_1$$

$$k(x,x') = \begin{cases} k_0(x,x'), & \text{if } x, x' \in D \\ 0, & \text{else} \end{cases}$$

1.2 Kernels for Real-Valued Data

Real-valued data contains of sample represented by vectors of a given dimensionality.

Polynomial Kernel $K(x,x') = (\langle x, x' \rangle + \lambda)^d$
The degree (d) of the polynomial also determines the flexibility (e.g. for classification).

Gaussian Kernel $K(x,x') = \exp(\frac{1}{\sigma}||x-x'||^2)$
σ is a parameter that determines the width of the Gaussian (and thereby its "stiffness"). The Gaussian Kernel is essentially zero if the squared distance $||x-x'||$ is much larger than σ. Small σ-values lead to very local kernels with large curvatures of decision boundaries. Outside regions, where the data is concentrated, the kernel is basically constant.

1.3 Kernels for other Data

Spectrum Kernel $K_l^{spectrum}(\hat{x}, \hat{x}') = \langle \phi_{spec}(\hat{x}), \phi_{spec}(\hat{x}') \rangle$
Spectrum kernels count the occurrence of subsequences of a certain length (l). Since they are used in kernel functions only the subsequences that actually occur in a sequence ($\hat{x}$) are compared to each other. For long subsequences, the chance of observing common subsequences is very low. *Mixed spectrum*

kernels alleviate this problem by using a weighting factor (β): $K_l^{m\_spectrum}(\hat{x}, \hat{x}') = \sum_{d=1}^{l} \beta_d K_d(\hat{x}, \hat{x}')$

Weighted-Degree (WD) Kernel $K_l^{WD}(\hat{x}, \hat{x}') = \sum_{d=1}^{l} \sum_{l=1}^{L-d+1} \beta_d K_d^{spectrum}(\hat{x}_{[l:l+d]}, \hat{x}'_{[l:l+d]})$
Weighted degree kernels also incorporates the position of the substrings, by analyzing sequences of length (L) and looking at the substrings at each position (l) of separately. XXXX suggest to employ $\beta_d = 2\frac{l-d+1}{l^2+1}$. Variations are the *WD kernel with shifts*, the *locality improved kernel* and the *oligo kernel* (allow for positional flexibility).

1.4 Kernel SVM

! → Since only the relative position of points to each other is important for the quality of the solution, we can focus solely on similarity measures between points → Kernels. Instead of inner products, we now use Kernels as different similarity measures between points.

! → The discriminant function for Gaussian Kernel-SVMs are a sum of bumps around the support vectors .

1.5 Unsupervised Multiple Kernel Learning (UMKL)

Goal: Integrate different data sets with different data types into one analysis [**?**].