Informal notes that I wanted to discuss about kernels and the kernel trick during the lab session. Please feel free to contact me with any questions or comments. I highly recommend looking at the tutorial posted on the class website for a more complete picture: http://phd.gccis.rit.edu/justindomke/courses/sml/08kernels.pdf

1 Features

A feature mapping \( \phi(x) \) is any function that takes as input \( x \) and outputs some features of \( x \) which are not necessarily in the same dimension of \( x \). Examples:

- \( \phi(x) = x \): the identity feature
- \( \phi(x) = [x_1, ..., x_n, x_1^2, ..., x_n^2]^T \): a feature mapping that appends an elementwise squared version of \( x \) on to the original \( x \) vector.
- \( \phi(x) = [\text{len}(x), \text{mean}(x)] \): if \( x \) is not a fixed length, we may want to describe it by recording its length and its mean.

Any of these \( \phi \) functions are valid as feature representations of the original data \( x \).

Q: Why are feature representations more interesting than the original \( x \) vector itself? Isn’t all of the information in \( \phi(x) \) already contained in \( x \)?

A: A dataset represented in a different feature space may be linearly separable while in the original feature space it was not.

Toy example that you can draw for yourself. Consider points in the 2-D plane, \( x = [x_1, x_2]^T \) which belong to class \(-1\) if \( \text{sign}(x_1) = \text{sign}(x_2) \) and class \(1\) otherwise. In the original feature space, these points are not linearly separable, but if \( \phi(x) = [x_1, x_2, x_1x_2]^T \) then they would be separable.

2 The SVM Dual

There are two equivalent formulations of the SVM problem, the primal:

\[
\min_{w, b, \xi} ||w||^2 + C \sum \xi_i
\]  

(1)
subject to: \[ y_i(w \cdot x_i + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \quad \forall i. \]

and the dual:

\[
\max_{\alpha} \sum_{i=1}^{N} \alpha_i + \frac{1}{2} \sum_{i,j} \alpha_i\alpha_j y_iy_j (x_i \cdot x_j) \tag{2}
\]

subject to: \[ \sum_i \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C \quad \forall i. \]

In this section we’re more interested in the dual formulation. Say we wanted to use some feature representation, \( \phi(x) \) instead of \( x \) itself for learning. In the dual, we never need to represent \( \phi(x) \) directly, it only ever appears with the dot product \( \phi(x_1) \cdot \phi(x_2) \).

### 3 Feature mappings and Kernels

We can use this to our advantage for reasons of computational and space efficiency. There are some functions \( \phi \) for which computing and storing \( \phi(x) \) requires a lot of time and space, but computing and storing \( \phi(x_1) \cdot \phi(x_2) \) does not! A striking example of this is a high order polynomial feature mapping:

\[
\phi\left( \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right) = \begin{pmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_1^3 \\ x_2^2 \\ x_1 x_2 \\ x_1 x_2^2 \\ \cdots \end{pmatrix}.
\]

Representing the entire feature vector may be extremely resource consuming, but calculating \( \phi(u) \cdot \phi(v) \) for 2 vectors \( u, v \) is much easier (see class slides and readings for details on the derivation). This is referred to as the kernel trick and exists for a number of different feature mappings.

This calculation of the dot product is referred to as the kernel function \( K_{\phi}(u, v) = \phi(u) \cdot \phi(v) \). Here I have made the \( \phi \) explicit in the kernel function, \( K_{\phi} \), to emphasize that every kernel as an associated feature function \( \phi \). In fact, the definition of a valid kernel is a function that can be written as \( K(u, v) = \phi(u) \cdot \phi(v) \) for some feature representation \( \phi \) which takes \( u \) and \( v \) as arguments.

(aside: As a result of this definition, all kernel functions \( K \) must be symmetric \( K(u, v) = K(v, u) \). Proof: \( K \) must have some feature mapping \( \phi \) such that: \( K(u, v) = \phi(u)\phi(v) = \phi(v)\phi(u) = K(v, u) \).)

### 4 Kernels and Similarity Measures

One can think of a kernel function \( K_{\phi} \) as a similarity measure in the feature space of \( \phi \).

**Intuition:** dot-products can be thought of as similarity measures. \( u \cdot v \) is maximized when \( u = v \) and minimized when the two vectors point in opposite directions. When \( u, v \) are orthogonal, the dot product is 0.