Dual representations

Dual representations

Many linear models can be reformulated in terms of a *dual representation*, where the kernel occurs naturally.

Consider regression:

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \approx t$$

We want to minimize

$$J(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) - t_n\}^2 + \lambda \mathbf{w}^T \mathbf{w} \qquad \lambda \geq 0$$

Dual representations

This can be rewritten with a matrix equation

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{w} - \mathbf{w}^T\mathbf{\Phi}^T\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

where defined

$$\mathbf{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1), & \phi_1(\mathbf{x}_1), & \dots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2), & \phi_1(\mathbf{x}_2), & \dots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N), & \phi_1(\mathbf{x}_N), & \dots & \phi_{M-1}(\mathbf{x}_N) \end{bmatrix}$$

$$\mathbf{t} = [t_1, t_2, \dots, t_N]^T$$

Dual representations

Now we set $\nabla_{\mathbf{w}} J = 0$ and get

$$\mathbf{w} = -\frac{1}{\lambda}\sum_{n=1}^{N}\{\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - t_n\}\boldsymbol{\phi}(\mathbf{x}_n) = \sum_{n=1}^{N} a_n\boldsymbol{\phi}(\mathbf{x}_n) = \mathbf{\Phi}^T\mathbf{a} \quad (1)$$

where $$a_n = -\frac{1}{\lambda}\{\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - t_n\} \quad (2)$$

Now, instead of working with **w**, the optimization is reformulated in terms of **a** using *dual representation*.

Dual representations

We substitute $\mathbf{w} = \mathbf{\Phi}^T \mathbf{a}$ into

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T \mathbf{\Phi}^T \mathbf{\Phi} \mathbf{w} - \mathbf{w}^T \mathbf{\Phi}^T \mathbf{t} + \frac{1}{2}\mathbf{t}^T \mathbf{t} + \frac{\lambda}{2}\mathbf{w}^T \mathbf{w}$$

and get

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T \mathbf{\Phi}\mathbf{\Phi}^T \mathbf{\Phi}\mathbf{\Phi}^T \mathbf{a} - \mathbf{a}^T \mathbf{\Phi}\mathbf{\Phi}^T \mathbf{t} + \frac{1}{2}\mathbf{t}^T \mathbf{t} + \frac{\lambda}{2}\mathbf{a}^T \mathbf{\Phi}\mathbf{\Phi}^T \mathbf{a}$$

Dual representations

We define the *Gram matrix* $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$

which is an NxN matrix with elements

$$K_{nm} = \boldsymbol{\phi}^T(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m)$$

where $k(\mathbf{x}_n, \mathbf{x}_m)$ is the kernel function.

Notice that the Gram matrix is a representation of the training data.

Dual representations

We can rewrite

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}^T\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^T\mathbf{K}\mathbf{a}$$

we also have (using (1) and (2) on slide 4)

$$\mathbf{a} = (\mathbf{K} + \lambda\mathbf{I}_N)^{-1}\mathbf{t}$$

Dual representations

We can now rewrite the regression formula

$$y(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) = \mathbf{a}^T \boldsymbol{\Phi} \boldsymbol{\phi}(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{t}$$

where

$$\mathbf{k}(\mathbf{x}) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}) \\ k(\mathbf{x}_2, \mathbf{x}) \\ \vdots \\ k(\mathbf{x}_N, \mathbf{x}) \end{bmatrix}$$

The dual representation allows for the solution to the least-squares problem to be expressed entirely in terms of the kernel function $k(\mathbf{x}_n, \mathbf{x}_m)$.

Dual representations

- Pros: Solution only in terms of a kernel function. Allows the use a very high dimensional feature space.
- Cons: Need to invert an *NxN* matrix instead of *MxM* . Normally *M*<< *N* (but not always)
- We can use kernels in Support Vector Machines
- Or anywhere where the data enters the algorithm in the form of scalar products

$$\mathbf{x}_n^T \mathbf{x}_m \rightarrow \boldsymbol{\phi}_n^T \boldsymbol{\phi}_m = k(\mathbf{x}_n, \mathbf{x}_m)$$

This is called *kernel trick* or *kernel substitution*