

Kernel Methods

Introduction (I)

- ▶ Parametric models for classification/regression:
 - ▶ The objective is to learn a set of adaptive parameters \mathbf{w} which determine the mapping from the input \mathbf{x} to the target y :

$$\hat{y} = f(\mathbf{x}, \mathbf{w})$$

- ▶ Need of a training phase
 - ▶ One example is linear regression:

$$\hat{y} = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x})$$

- ▶ Non-parametric models:
 - ▶ A subset of the input data points are used during classification
 - ▶ No training is needed, but methods are slow at making predictions
 - ▶ Examples: Parzen windows, Nearest Neighbors, etc.

Introduction (II)

- ▶ Some linear parametric models can be reformulated using a *dual representation*
- ▶ The predictions on the test data are based only on a linear combination of a *kernel* function which is evaluated on a subset of the training data
- ▶ Somehow we manage to express the parametric model as a non-parametric one
- ▶ One of such models is regularized linear regression

Regularized linear regression

- ▶ The attribute vectors are \mathbf{x}_i , $i = 1, \dots, N$
- ▶ The targets are y_i , $i = 1, \dots, N$
- ▶ The model (estimation of the target) is:

$$\hat{y}_i = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}_i)$$

- ▶ The base functions $\boldsymbol{\phi}(\mathbf{x}_i) = (\phi_0(\mathbf{x}_i), \phi_1(\mathbf{x}_i), \dots, \phi_{M-1}(\mathbf{x}_i))^T$ represent the attributes
- ▶ We assume $\phi_0(\mathbf{x}_i) = 1$
- ▶ We minimize the following error function:

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N \{y_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i)\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

Minimization of $J(\mathbf{w})$

- We want to minimize:

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{2} \sum_{i=1}^N \{y_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i)\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \\ &= \frac{1}{2} \sum_{i=1}^N \{y_i - \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}_i)\}^2 + \frac{\lambda}{2} \sum_{j=0}^{M-1} w_j^2 \end{aligned}$$

- With respect to the parameters w_k :

$$\frac{\partial J(\mathbf{w})}{\partial w_k} = 0$$

Minimization of $J(\mathbf{w})$

► Operating:

$$\begin{aligned}\frac{\partial J(\mathbf{w})}{\partial w_k} &= -\sum_{i=1}^N y_i \phi_k(\mathbf{x}_i) + \sum_{i=1}^N \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}_i) \phi_k(\mathbf{x}_i) + \lambda w_k \\&= -\sum_{i=1}^N y_i \Phi_{ik} + \sum_{i=1}^N \sum_{j=0}^{M-1} w_j \Phi_{ij} \Phi_{ik} + \lambda w_k \\&= -\sum_{i=1}^N \Phi_{ki}^T y_i + \sum_{j=0}^{M-1} \sum_{i=1}^N \Phi_{ki}^T \Phi_{ij} w_j + \lambda w_k \\&= -(\Phi^T \mathbf{y})_k + \sum_{j=0}^{M-1} (\Phi^T \Phi)_{kj} w_j + \lambda w_k \\&= -(\Phi^T \mathbf{y})_k + (\Phi^T \Phi \mathbf{w})_k + \lambda w_k = 0\end{aligned}$$

Minimization of $J(\mathbf{w})$

- Finally:

$$-\Phi^T \mathbf{y} + \Phi^T \Phi \mathbf{w} + \lambda \mathbf{w} = \mathbf{0}$$

$$\mathbf{w} = (\mathbf{A} + \lambda I)^{-1} \Phi^T \mathbf{y}$$

- Where Φ is the *design matrix*, $\Phi_{ij} = \phi_j(\mathbf{x}_i)$:

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

- And $\mathbf{A} = \Phi^T \Phi$ is a $M \times M$ matrix

An example

- ▶ Four pairs $(x; y)$: $\{(1; 0, 8), (4; 4, 1), (6; 6, 2), (9; 8, 5)\}$
- ▶ The vector of attributes is $\phi(x) = (1, x)^T$
- ▶ And the design matrix is:

$$\Phi = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 4 \\ 1 & 6 \\ 1 & 9 \end{pmatrix}$$

- ▶ Let us assume $\lambda = 1$
- ▶ Full solution in the IPython Notebook

Reformulation of regularized linear regression

- Let's do it a different way:

$$\frac{\partial J(\mathbf{w})}{\partial w_k} = - \sum_{i=1}^N \Phi_{ki}^T y_i + \sum_{j=0}^{M-1} \sum_{i=1}^N \Phi_{ki}^T \Phi_{ij} w_j + \lambda w_k = 0$$

- So:

$$\begin{aligned} w_k &= -\frac{1}{\lambda} \sum_{i=1}^N \left\{ \sum_{j=0}^{M-1} \Phi_{ki}^T \Phi_{ij} w_j - \Phi_{ki}^T y_i \right\} \\ &= -\frac{1}{\lambda} \sum_{i=1}^N \Phi_{ki}^T \left\{ \sum_{j=0}^{M-1} \Phi_{ij} w_j - y_i \right\} \\ &= -\frac{1}{\lambda} \sum_{i=1}^N \Phi_{ki}^T \left\{ \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}_i) - y_i \right\} \end{aligned}$$

Reformulation of regularized linear regression

- This leads to:

$$\begin{aligned}w_k &= -\frac{1}{\lambda} \sum_{i=1}^N \Phi_{ki}^T \left\{ \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}_i) - y_i \right\} \\&= -\frac{1}{\lambda} \sum_{i=1}^N \Phi_{ki}^T \{ \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - y_i \} \\&= \sum_{i=1}^N \Phi_{ki}^T a_i\end{aligned}$$

- Where:

$$a_i = -\frac{1}{\lambda} \{ \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - y_i \}$$

Reformulation of regularized linear regression

- In matrix form:

$$\mathbf{w} = \Phi^T \mathbf{a}$$

- Now we can substitute w_k into $J(\mathbf{w})$ to obtain an expression that depends only on \mathbf{a} :

$$J(\mathbf{a}) = \frac{1}{2} \sum_{i=1}^N \{y_i - \sum_{j=0}^{M-1} \sum_{k=1}^N \Phi_{jk}^T a_k \Phi_{ij}\}^2 + \frac{\lambda}{2} \sum_{j=0}^{M-1} (\sum_{k=1}^N \Phi_{jk}^T a_k)^2$$

- The *dual problem* consists of minimizing $J(\mathbf{a})$ with respect to a_k :

$$\frac{\partial J(\mathbf{a})}{\partial a_k} = 0$$

Reformulation of regularized linear regression

- After some algebra we obtain:

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

- Where $\mathbf{K} = \Phi \Phi^T$ is a $N \times N$ matrix that satisfies that:

$$\mathbf{K}_{ij} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \equiv k(\mathbf{x}_i, \mathbf{x}_j)$$

- The function $k(\mathbf{x}_i, \mathbf{x}_j)$ is known as a *kernel* function

Conclusion

- In the primal formulation:

$$\hat{y} = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x})$$

To obtain \mathbf{w} we have to invert the matrix $\mathbf{A} + \lambda I$, which is $M \times M$

- In the dual formulation:

$$\hat{y} = \mathbf{a}^T \boldsymbol{\Phi} \boldsymbol{\phi}(\mathbf{x}) = \sum_{i=1}^N a_i k(\mathbf{x}_i, \mathbf{x})$$

To obtain \mathbf{a} we have to invert the matrix $\mathbf{K} + \lambda I$, which is $N \times N$

- This technique is generalizable to other problems