Large Margin Nearest Neighbours in Shogun

Fernando J. IGLESIAS GARCÍA

SHOGUN WORKSHOP 2014. JULY 28. BERLIN

Mahalanobis distance

- Euclidean distance.

$$\mathcal{D}\left(\vec{x_i}, \vec{x_j}\right) = \|\vec{x_i} - \vec{x_j}\|_2^2 = \left(\vec{x_i} - \vec{x_j}\right)^T \left(\vec{x_i} - \vec{x_j}\right) \quad \vec{x_i}, \vec{x_j} \in \mathbb{R}^n$$

- Mahalanobis distance.

$$\mathcal{D}_{\mathsf{M}}\left(\vec{x_i}, \vec{x_j}\right) = \left(\vec{x_i} - \vec{x_j}\right)^T \mathsf{M} \left(\vec{x_i} - \vec{x_j}\right) \quad \vec{x_i}, \vec{x_j} \in \mathbb{R}^n, \mathsf{M} \in \mathbb{R}^{n \times n}$$

$$\vec{x}^T \mathsf{M} \vec{x} \geq 0 \ \forall \ \vec{x} \in \mathbb{R}^n$$

k-nearest neighbours classification

- What distance to use?
  - Amplify the most informative dimensions,
  - while shrinking the least informative ones.
- Distance adapted to the data.
- LMNN: find a Mahalanobis distance that maximises k-NN classification.

LMNN

1. Choose *target neighbours* (TN).

LMNN

1. Choose *target neighbours* (TN).
   - Notation: $j \rightsquigarrow i$ means that $\vec{x_j}$ is a target neighbour of $\vec{x_i}$.

LMNN

1. Choose *target neighbours* (TN).
   - Notation: $j \rightsquigarrow i$ means that $\vec{x_j}$ is a target neighbour of $\vec{x_i}$.
   - Remark: $j \rightsquigarrow i \not\Rightarrow i \rightsquigarrow j$.

LMNN

1. Choose *target neighbours* (TN).
   - Notation: $j \rightsquigarrow i$ means that $\vec{x_j}$ is a target neighbour of $\vec{x_i}$.
   - Remark: $j \rightsquigarrow i \not\Rightarrow i \rightsquigarrow j$.
2. Learn a distance such that NN = TN.

LMNN

1. Choose *target neighbours* (TN).
   - Notation: $j \rightsquigarrow i$ means that $\vec{x_j}$ is a target neighbour of $\vec{x_i}$.
   - Remark: $j \rightsquigarrow i \nRightarrow i \rightsquigarrow j$.
2. Learn a distance such that NN $=$ TN.
   - *Impostors* are avoided, that is

$$\nexists\ \vec{x_l} \mid \mathcal{D}_{\mathsf{M}}\left(\vec{x_i}, \vec{x_l}\right) \leq \mathcal{D}_{\mathsf{M}}\left(\vec{x_i}, \vec{x_j}\right)\ \forall\ i, j, l \mid j \rightsquigarrow i, y_l \neq y_i.$$

LMNN

1. Choose *target neighbours* (TN).
   - Notation: $j \rightsquigarrow i$ means that $\vec{x_j}$ is a target neighbour of $\vec{x_i}$.
   - Remark: $j \rightsquigarrow i \not\Rightarrow i \rightsquigarrow j$.
2. Learn a distance such that NN = TN.
   - *Impostors* are avoided, that is

$$\nexists\ \vec{x_l} \mid \mathcal{D}_{\mathsf{M}}\left(\vec{x_i}, \vec{x_l}\right) \leq \mathcal{D}_{\mathsf{M}}\left(\vec{x_i}, \vec{x_j}\right)\ \forall\ i, j, l \mid j \rightsquigarrow i, y_l \neq y_i.$$

   - Introducing a *margin*,

$$\nexists\ \vec{x_l} \mid \mathcal{D}_{\mathsf{M}}\left(\vec{x_i}, \vec{x_l}\right) \leq \mathcal{D}_{\mathsf{M}}\left(\vec{x_i}, \vec{x_j}\right) + 1\ \forall\ i, j, l \mid j \rightsquigarrow i, y_l \neq y_i.$$

LMNN – SDP formulation

$$\forall\ i, j, l \mid j \rightsquigarrow i, y_l \neq y_i$$

$$\mathcal{D}_{\mathsf{M}}\left(\vec{x}_i, \vec{x}_j\right) \leq \mathcal{D}_{\mathsf{M}}\left(\vec{x}_i, \vec{x}_l\right)$$

$$\mathcal{D}_{\mathsf{M}}\left(\vec{x}_i, \vec{x}_j\right) + 1 \leq \mathcal{D}_{\mathsf{M}}\left(\vec{x}_i, \vec{x}_l\right)$$

$$\mathcal{D}_{\mathsf{M}}\left(\vec{x}_i, \vec{x}_j\right) + 1 \leq \mathcal{D}_{\mathsf{M}}\left(\vec{x}_i, \vec{x}_l\right) + \xi_{ijl}$$

$$\min_{\mathsf{M}} \sum_{i, j \rightsquigarrow i} \mathcal{D}_{\mathsf{M}}\left(\vec{x}_i, \vec{x}_j\right) + \mu \sum_{i, j \rightsquigarrow i, l} \xi_{ijl}$$

subject to:

$$\mathcal{D}_{\mathsf{M}}\left(\vec{x}_i, \vec{x}_j\right) + 1 \leq \mathcal{D}_{\mathsf{M}}\left(\vec{x}_i, \vec{x}_l\right) + \xi_{ijl}$$

$$\xi_{ijl} \geq 0$$

$$\vec{x}^T \mathsf{M} \vec{x} \geq 0$$

Weinberger, K. Q., Saul, L. K. *Distance Metric Learning for Large Margin Nearest Neighbor Classification*. Journal of Machine Learning Research (JMLR). 2009.

LMNN – Linear metric learning interpretation

$$\vec{x}^T \mathsf{M} \vec{x} \geq 0 \ \forall \vec{x} \in \mathbb{R}^n \Rightarrow \mathsf{M} = \mathsf{L}^T \mathsf{L}$$

LMNN – Linear metric learning interpretation

$$\vec{x}^T \mathsf{M} \vec{x} \geq 0 \ \forall \vec{x} \in \mathbb{R}^n \Rightarrow \mathsf{M} = \mathsf{L}^T \mathsf{L}$$

$$\begin{aligned}\mathcal{D}_{\mathsf{M}} \left(\vec{x_i}, \vec{x_j}\right) &= \left(\vec{x_i} - \vec{x_j}\right)^T \mathsf{M} \left(\vec{x_i} - \vec{x_j}\right) = \\ &= \left(\vec{x_i} - \vec{x_j}\right)^T \mathsf{L}^T \mathsf{L} \left(\vec{x_i} - \vec{x_j}\right) = \mathcal{D}\left(\mathsf{L}\vec{x_i}, \mathsf{L}\vec{x_j}\right)\end{aligned}$$

LMNN – Linear metric learning interpretation

$$\vec{x}^T \mathsf{M} \vec{x} \geq 0 \ \forall \vec{x} \in \mathbb{R}^n \Rightarrow \mathsf{M} = \mathsf{L}^T \mathsf{L}$$

$$\begin{aligned}\mathcal{D}_{\mathsf{M}}\left(\vec{x_i}, \vec{x_j}\right) &= \left(\vec{x_i} - \vec{x_j}\right)^T \mathsf{M} \left(\vec{x_i} - \vec{x_j}\right) = \\ &= \left(\vec{x_i} - \vec{x_j}\right)^T \mathsf{L}^T \mathsf{L} \left(\vec{x_i} - \vec{x_j}\right) = \mathcal{D}\left(\mathsf{L}\vec{x_i}, \mathsf{L}\vec{x_j}\right)\end{aligned}$$

- Beyond classification:
  - Dimension reduction with rectangular L (i.e. $\mathsf{L} \in \mathbb{R}^{r \times n} | r < n$).
  - Feature selection with diagonal L.