Learning from Data, Tutorial Sheet for Week 3: Answers

School of Informatics, University of Edinburgh

Instructor: Amos Storkey

1. **Principal Components Analysis in *MATLAB***

   See *w3ans.m* for the code

2. **Principal Components Analysis**

   $\vec{x}^\mu \approx \vec{m} + \sum_{i=1}^{M} a_i \, \vec{e_i}$

   a) Optimal coefficients a_i

   $\frac{d}{d\,a_j}(\vec{x}^\mu - \vec{m} - \sum_{i=1}^{M} a_i \, \vec{e_i})^2 = 0 \Leftrightarrow$

   $\Leftrightarrow 2(\vec{x}^\mu - \vec{m} - \sum_{i=1}^{M} a_i \, \vec{e_i}) \frac{d}{d\,a_j}(-\sum_{i=1}^{M} a_i \, \vec{e_i}) = 2(\vec{x}^\mu - \vec{m} - \sum_{i=1}^{M} a_i \, \vec{e_i})(-\vec{e_j}) = 0 \Leftrightarrow$

   $\Leftrightarrow (\vec{x}^\mu - \vec{m}).\vec{e_j} = \sum_{i=1}^{M} a_i \, \vec{e_i}.\vec{e_j}$

   Since the basis set $\vec{e_i}$ is orthonormal (orthogonal[1] and normalised[2]), we have[3]:
   $\vec{e_i}.\vec{e_j} = 1, i = j$ and $\vec{e_i}.\vec{e_j} = 0, i \neq j$.

   Taking this into account, we get for our optimisation expression:

   $a_j = (\vec{x}^\mu - \vec{m}).\vec{e_j}$

   b) Calculating distances

   $(\vec{x}^a - \vec{x}^b)^2 \approx (\vec{m} + \sum_{i=1}^{M} a_i \, \vec{e_i} - \vec{m} - \sum_{i=1}^{M} b_i \, \vec{e_i})^2$

   $(\sum_{i=1}^{M} a_i \, \vec{e_i} - \sum_{i=1}^{M} b_i \, \vec{e_i})^2 = \sum_{i=1}^{M} a_i \, \vec{e_i} \sum_{j=1}^{M} a_j \, \vec{e_j} + \sum_{i=1}^{M} b_i \, \vec{e_i} \sum_{j=1}^{M} b_j \, \vec{e_j} - 2\sum_{i=1}^{M} a_i \, \vec{e_i} \sum_{j=1}^{M} b_j \, \vec{e_j} =$

   $= \sum_{i=1}^{M} a_i \sum_{j=1}^{M} a_j \, \vec{e_i}.\vec{e_j} + \sum_{i=1}^{M} b_i \sum_{j=1}^{M} b_j \, \vec{e_i}.\vec{e_j} - 2\sum_{i=1}^{M} a_i \sum_{j=1}^{M} b_j \, \vec{e_i}.\vec{e_j}$

   Again, since $\vec{e_i}.\vec{e_j} = 1, i = j$ and $\vec{e_i}.\vec{e_j} = 0, i \neq j$,

   $(\sum_{i=1}^{M} a_i \, \vec{e_i} - \sum_{i=1}^{M} b_i \, \vec{e_i})^2 = \sum_{i=1}^{M} a_i^2 + \sum_{i=1}^{M} b_i^2 - 2\sum_{i=1}^{M} a_i \, b_i = \vec{a}.\vec{a} + \vec{b}.\vec{b} - 2\vec{a}.\vec{b}$

   $(\vec{x}^a - \vec{x}^b)^2 \approx (\vec{a} - \vec{b})^2$

3. **Calculating distances**

   Instead of calculating the distances using the original high-dimensional representation of the data points (considerable computational effort), we can calculate the approximated distances using the lower-dimensional representation.

---

[1] The dot product of any two different vectors of the set is 0
[2] Length, or norm, equal to 1
[3] Simply $\vec{e_i}.\vec{e_j} = \delta_{ij}$