~~Name~~: ML2 Sheet 1

Lecture:

Dimensionality reduction - why?

- fewer parameter to describe data
- visualisation in 2D or 3D
- compress data and save space ~~speedup algorithm~~
- less overfitting
- speedup algorithm

Why not use PCA always?

→ only works if data lays on ~~a~~ linear manifold

Nonlinear dimension reduction:

- kernel PCA
- ISOMAP
- local linear embedding (LLE)
- ...

LLE Algorithm:

1. assign ~~K~~ K nearest neighbors to each datapoint $\vec{X}_i$
2. calc weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors by solving

n: datapoints N_i = k neighbors

$$R(W) = \sum_{i=1}^{n} \| X_i - \sum_{j \in N_i} W_{ij} X_j \|^2$$ with the constraint $\sum_{j \in N_i} W_{ij} = 1$

3. calc low-dim embedding vectors Y_i by minimizing

$$\phi(Y) = \sum_{i=1}^{n} \| Y_i - \sum_{j \in N_i} W_{ij} Y_j \|^2$$

Limitation of LLE:

- sensitive to noise
- sensitive to non-uniform sampling of the manifold
- does not provide a mapping
- quadratic complexity on training set size
- no info about intrinsic dimensionality (unlike ISOMAP)
- no good method to define hyperparameter of ~~··~~ K neighbors

t-SNE

p: discrete input space

q: embedding space

~~SNE~~ Kullback-Leibler divergence:

→ measure of how much two probability distributions vary from each other

t-SNE: minimizes KL divergence between q and p