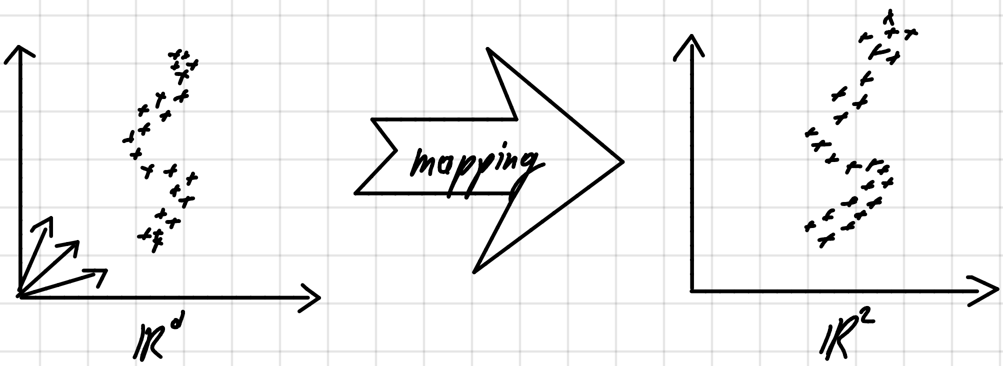
CS4786

Data Visualization

Problem setup: Input data $\vec{x}_1, \dots, \vec{x}_n \in \mathbb{R}^d$ $(d \gg 0)$

We want a mapping $\vec{x}_i \to \vec{z}_i$ with $\vec{z}_1, \dots, \vec{z}_n \in \mathbb{R}^r$ with $r \ll d$.

To visualize the data we want $r=2$ or $r=3$.



Finding a mapping from $\mathbb{R}^d \to \mathbb{R}^r$ is easy if there are no constraints (just map points randomly). But we want the low-dimensional representation to tell us something about the high dimensional data.

We need some guarantee that tells us what properties are preserved in the low dimensional space.

Common guarantee: preserve distances

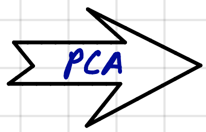
3 minutes Quiz: How many points can you place in $\mathbb{R}^d$ such that each one is equidistant from all the others?

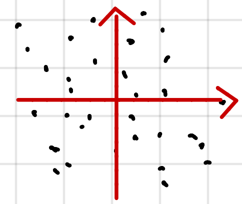
$\Rightarrow$ You cannot preserve all pairwise distances exactly.

But maybe we can make assumptions on the data?

Multidimensional Scaling / PCA

Easy case: Data lies in a low dimensional subspace within $\mathbb{R}^d$





$$\vec{z}_i = U^T(\vec{x}_i - \vec{\mu})$$

$$\vec{\mu} = \frac{1}{n}\sum_{i=1}^{n}\vec{x}_i$$

mean input vector

$$C = \frac{1}{n-1}\sum_{i=1}^{n}(\vec{x}_i - \vec{\mu})(\vec{x}_i - \vec{\mu})^T \longrightarrow C\vec{u}_i = \lambda_i \longrightarrow U = \begin{bmatrix} \uparrow & \uparrow \\ \vec{u}_1 & \vec{u}_2 \\ \downarrow & \downarrow \end{bmatrix}$$

covariance matrix — eigenvectors

$$\begin{pmatrix} u_i^T u_i = 1 \\ j \neq i:\ u_i^T u_j = 0 \end{pmatrix}$$

PCA maximizes spread. Variance after projection: $\sum_{i=1}^{n}(\vec{x}_i^T\vec{u} - \vec{\mu}^T\vec{u})^2 = \sum_{i=1}^{n}(x_i^T u)^2 = \sum_{i=1}^{n} u^T x_i x_i^T u = u^T\left(\sum_{i=1}^{n} x_i x_i^T\right)u = u^T C u$

(Assume data is centered, i.e. $\vec{\mu} = 0$)

maximize $u^T C u$ s.t. $u^T u = 1$

$\Rightarrow \mathcal{L}(u,\lambda) = u^T C u - \lambda(u^T u - 1)$

$\frac{\partial \mathcal{L}}{\partial u} = 2Cu - 2\lambda u = 0$

$\Rightarrow Cu = \lambda u \Rightarrow$ u is eigenvector of C!

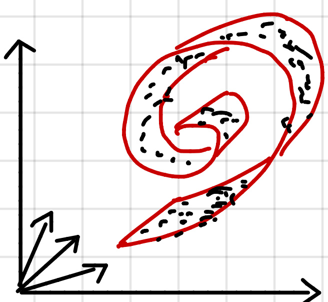
Preserves pairwise squared distances.

Limitations of PCA: - Focus on large distances ← Not always what we want

Manifold Learning:

Assume data lies on a low-dimensional sub-manifold.

PCA wouldn't work! Why?



$\mathbb{R}^d$

Solution:

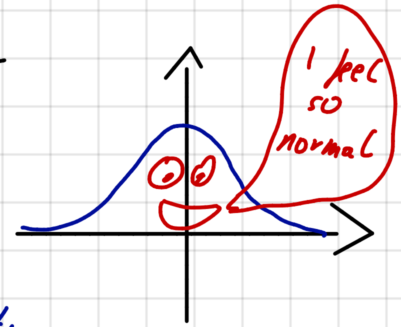
Approximate manifold with nearest neighbor graph. Embed graph in low dimensions.

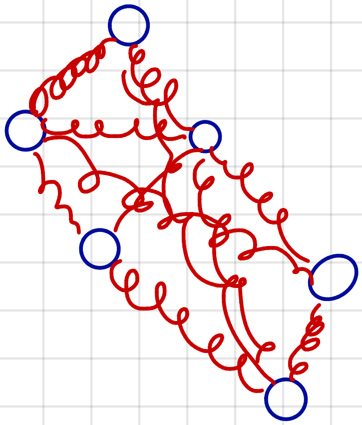
Algorithms: ISOMAP, MVU, LLE, Laplacian Eigenmaps

$\mathbb{R}^d$

Stochastic Neighbour Embedding

(SNE)





Place springs between any two data points, so that it is relaxed in the original space. Then place the data into low dimensions. Points that should be close pull each other close, others repell each other.

SNE:

Preserve local neighborhoods. Point z_i should have similar neighbors as point x_i.

But neighbors are discrete, which makes optimization hard.

Stochastic Neighborhood:

Place a Gaussian around each input data point and pretend you are drawing neighbors from that distribution.

Probability of drawing neighbor j for input i.

$$p_{ij} = \frac{e^{-\frac{(\bar{x}_i - \bar{x}_j)^2}{2\sigma^2}}}{\sum_k e^{-\frac{(\bar{x}_i - \bar{x}_k)^2}{2\sigma^2}}} \qquad \text{set } p_{ii} = 0$$

Similarly define:

$$q_{ij} = \frac{e^{-(z_i - z_j)^2}}{\sum_k e^{-(z_i - z_k)^2}} \qquad q_{ii} = 0$$

Loss function:

$$\min_{z_1, \dots, z_n} \underbrace{\sum_{i=1}^{n} \sum_{j \neq i}^{n} p_{ij} \log \frac{p_{ij}}{q_{ij}}}_{KL(P_i \| Q_i)}$$

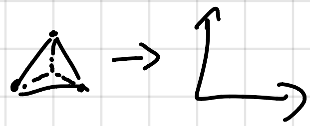
Kullback Leibler Divergence of the two neighbor distributions.
(is always non-negative)

$\boxed{\begin{array}{l}\text{How large is the penalty if:} \\ p_{ij} \text{ is large and } q_{ij} \text{ small?} \\ p_{ij} \text{ is small and } q_{ij} \text{ large?}\end{array}}$

What can you conclude about t-SNE's focus?

Problems with SNE:

Crowding: If data is intrinsicly high dimensional there is no way to map local neighborhood into low dimensional space.



We must move dissimilar points artificially too far away.
But SNE doesn't do this, because Gaussian tails drop of too fast $e^{-(z_i-z_j)^2}$.

t-SNE

Solution: Use the student t-distribution in the low dimensional space instead. The heavier tails can accomodate points that are shoved further away.

$$q_{ij} = \frac{(1+\|z_i - z_j\|_2^2)^{-1}}{\sum_k (1+\|z_i - z_k\|_2^2)^{-1}} \qquad q_{ii} = 0$$