Deep Learning (BEV033DLE)
Lecture 11
KL Divergence, t-SNE, Unsupervised RL

Czech Technical University in Prague

- KL Divergence
- Stochastic Neighbor Embedding (t-SNE)
- Unsupervised Representation Learning
  - Latent Variable Models
  - ELBO, Variational Inference
  - Stochastic EM,
  - Multi-sense word vectors

KL Divergence

KL Divergence

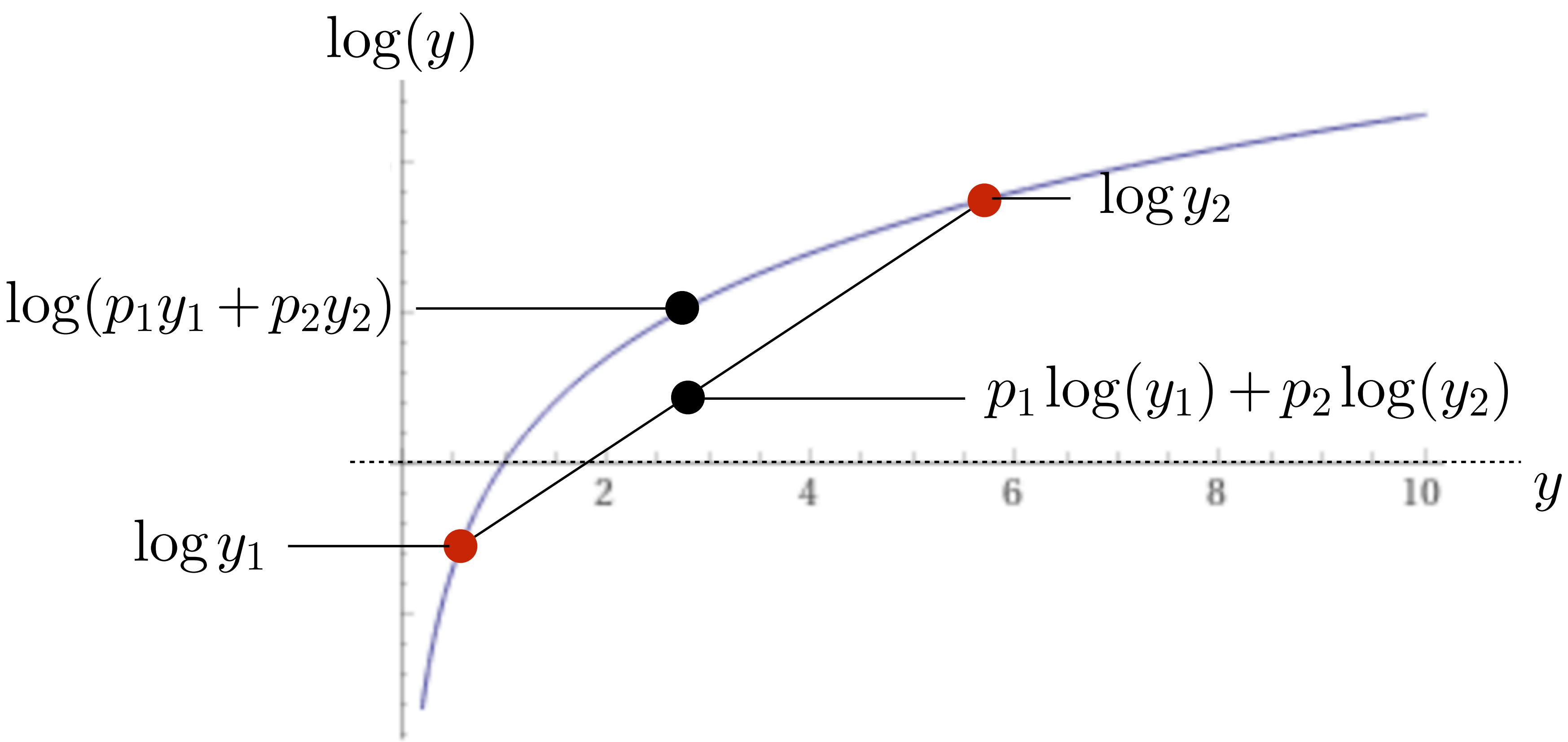
- Let $p(x)$ and $q(x)$ be two probability distributions.
- Kullback–Leibler divergence of p and q is

$$D_{\mathrm{KL}}(p \,\|\, q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

  - Definition allows $p(x) = 0$ by the extension $\lim_{p \to 0} p \log p = 0$
  - Defined when $\mathrm{supp(p)} \subseteq \mathrm{supp(q)}$, i.e. $q(x) = 0 \Rightarrow p(x) = 0$
- Properties:
  - D_{KL} is a *divergence*: $D_{\mathrm{KL}} \geq 0$ with equality iff $q = p$
  - Non-symmetric
  - (Invariant under change of variables)
  - Information-theoretic properties (Amount of information lost when q is used to approximate p)

Non-negativity

- Non-negativity: $D_{\mathrm{KL}}(p\|q) \geq 0$
  - let $y(x) = \frac{q(x)}{p(x)}$
  - The inequality $\sum_x p(x) \log \frac{p(x)}{q(x)} \geq 0$ is equivalent to $\sum_x p(x) \log y(x) \leq 0$
  - Observe that $\log$ is concave, apply Jensen's inequality:
  - $\sum_x p(x) \log y(x) \leq \log \sum_x p(x) y(x) = \log \sum_x q(x) = \log 1 = 0.$
- From strict concavity follows that $D_{\mathrm{KL}}(p\|q) = 0$ iff $p = q$



Maximum Likelihood, Cross-Entropy and (Forward) KL

- Maximum Likelihood Learning for Classification:
  - (x_i, y_i) – training data. Assume it is given by the true distribution $p(x, y)$
  - Model: $q(y|x;\theta)$
  - Negative Log-Likelihood (NLL) minimization:

$$\min_\theta \mathbb{E}_{(x,y)\sim p}\Big[-\log q(y|x;\theta)\Big]$$

$$= \min_\theta \mathbb{E}_{x\sim p(x)}\Big[\underbrace{\sum_y p(y|x)(-\log q(y|x;\theta))}_{\text{Crossentropy of } p(y|x) \text{ and } q(y|x;\theta)}\Big]$$

$$= \min_\theta \mathbb{E}_{x\sim p(x)}\Big[D_{\mathrm{KL}}(p(y|x)\,\|\,q(y|x;\theta))\Big] \underbrace{-\sum_y p(y|x)\log p(y|x)}_{\text{Entropy of } p(y|x)}$$

  - For minimization in θ, the NLL, Cross-entropy and KL divergence are equivalent
  - Can apply SGD

Asymmetry

Minimizing **forward KL** divergence:

$$\min_q D_{\mathrm{KL}}(p\|q)$$

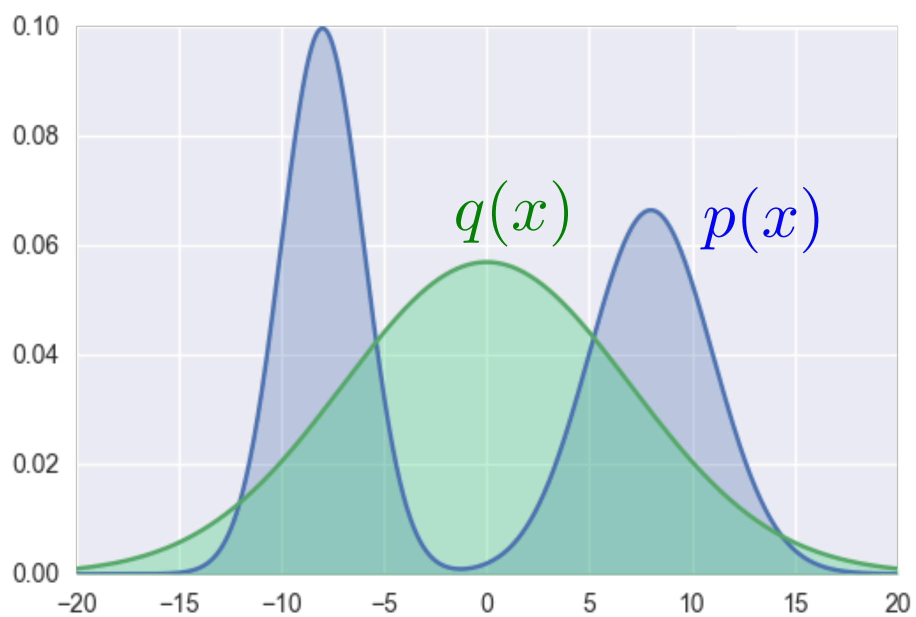
$$\min_q \int p(x)(\log p(x) - \log q(x))dx$$

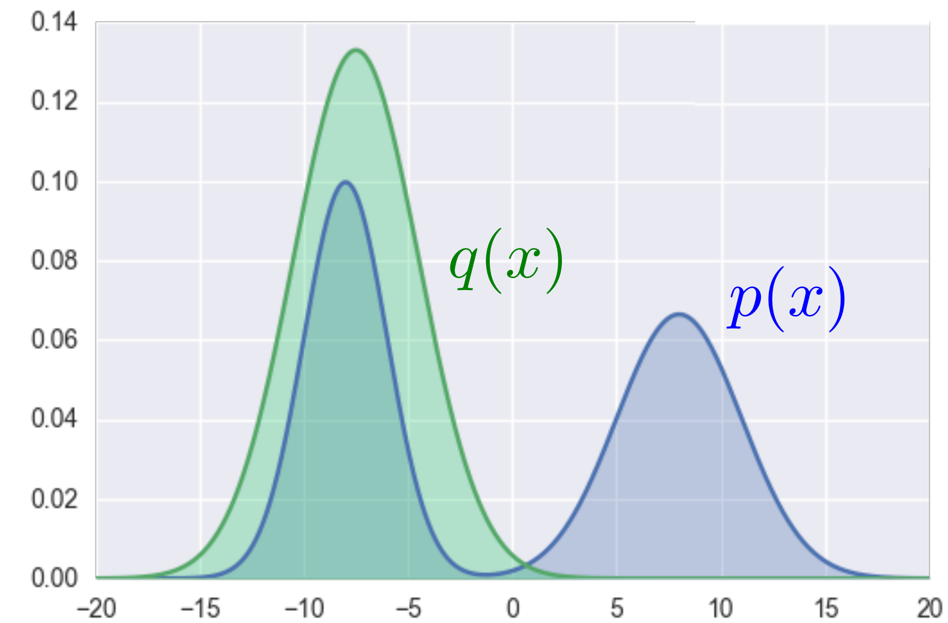
Minimizing **reverse KL** divergence:

$$\min_q D_{\mathrm{KL}}(q\|p)$$

$$\min_q \int q(x)(\log q(x) - \log p(x))dx$$

Example: q is Gaussian





- Approximates well on average in p
- Matches moments for q in EF (e.g. Gaussian)
- Suffices to sample from $p(x)$

- Approximates well on average in q
- Selects a mode
- Requires $\log(p)$

Stochastic Neighbor Embedding

Motivation

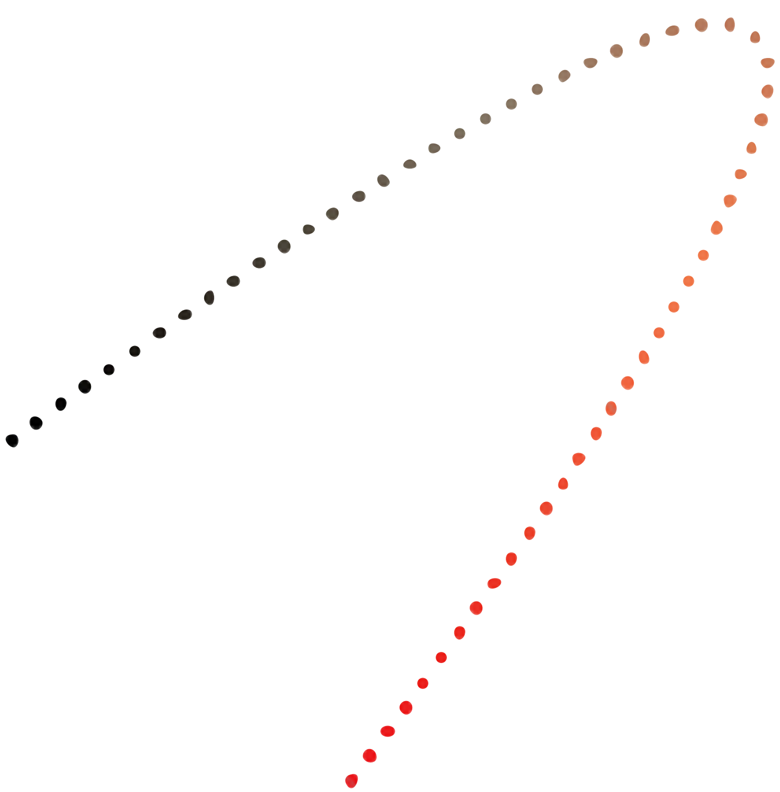
✦ Tool of representational geometry:

- dimensionality reduction for data visualization

✦ Goals:

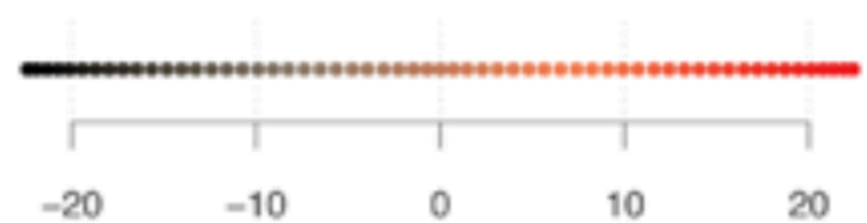
- Data often lies on a lower-dimensional manifold
- Preserve small distances accurately
- Large distances can be increased more

Data in $\mathbb{R}^n$

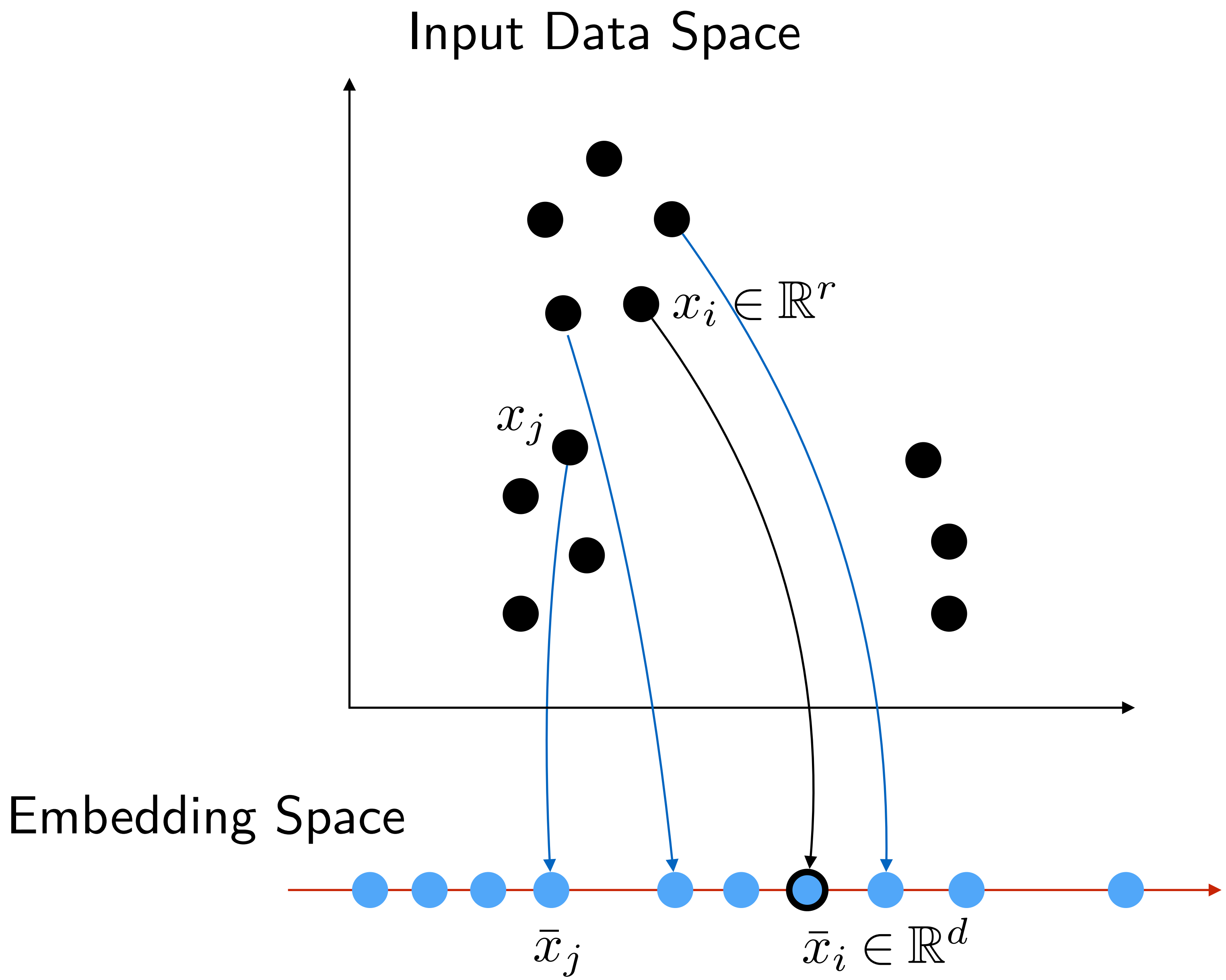


Non-linear embedding

Representation in $\mathbb{R}^d$

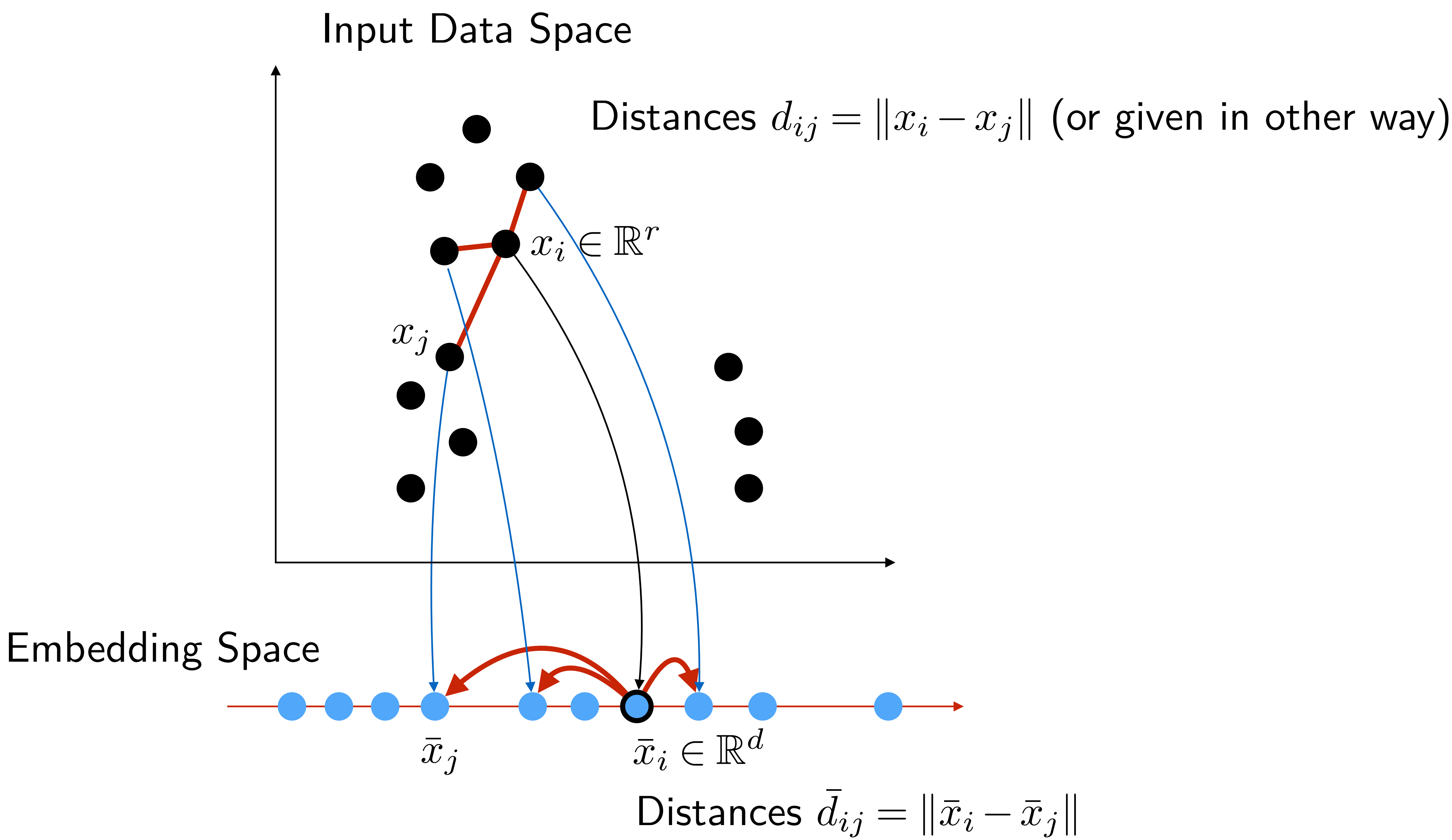


Multidimensional Scaling (MSD)

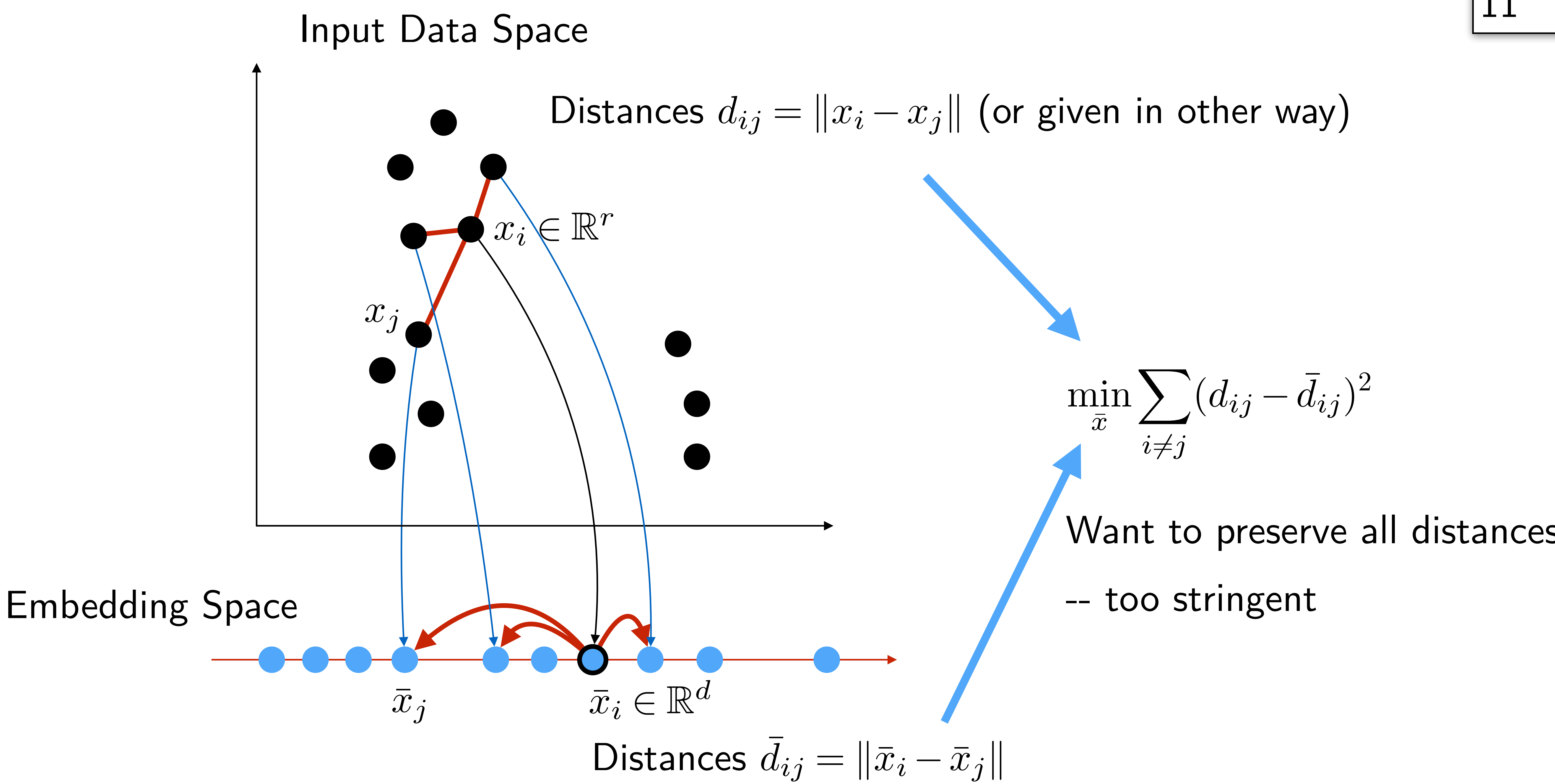


- Non-parametric model: for each data point x_t we find a corresponding embedding $\bar{x}_t$

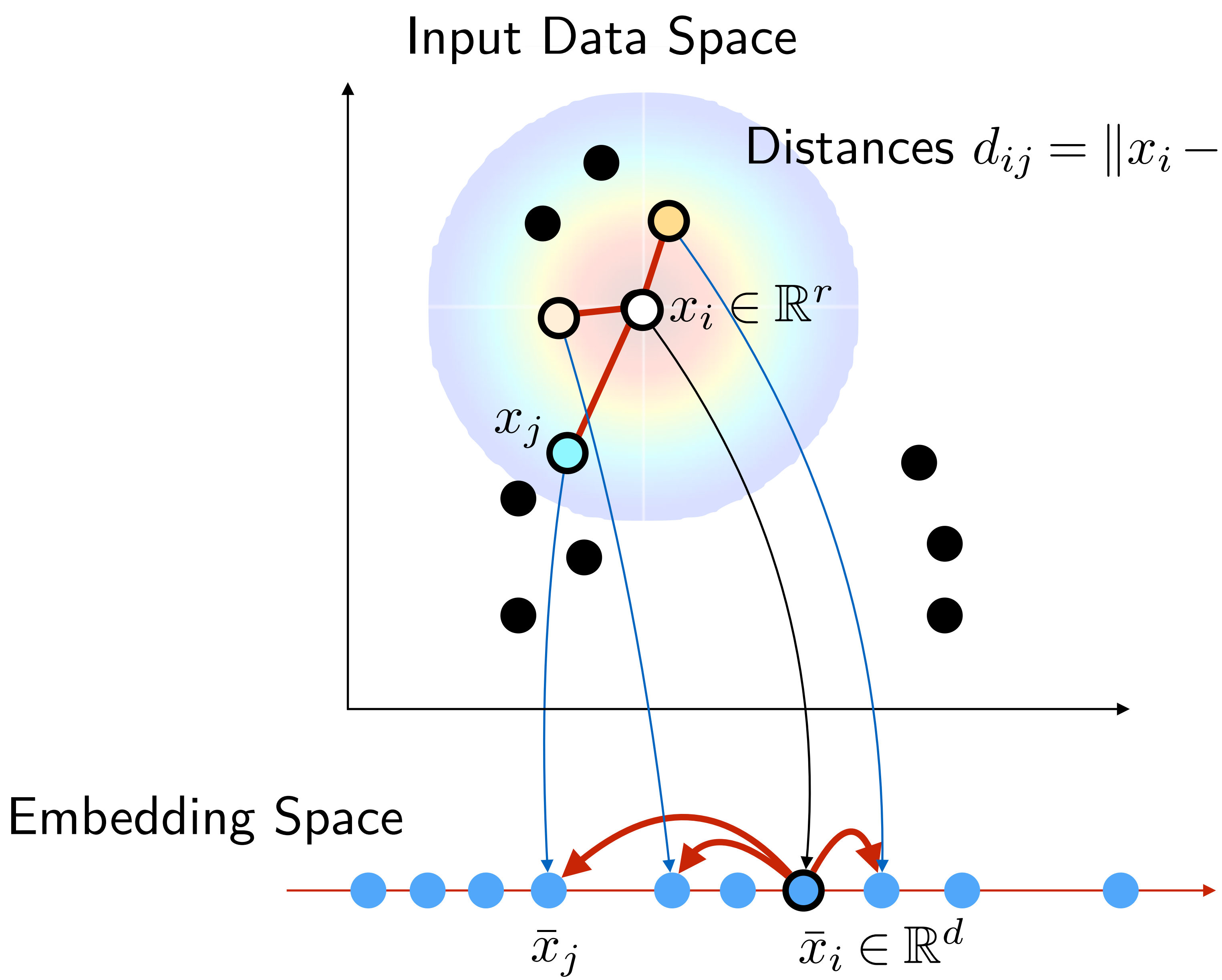
Multidimensional Scaling (MSD)



Multidimensional Scaling (MSD)



Stochastic Neighbor Embedding (SNE)



Distances $d_{ij} = \|x_i - x_j\|$ (or given in other way)

$$p_i(j) = \frac{e^{-d_{ij}^2/2\sigma_i^2}}{\sum_{j'\neq i} e^{-d_{ij'}^2/2\sigma_i^2}}, \quad \forall j \neq i$$

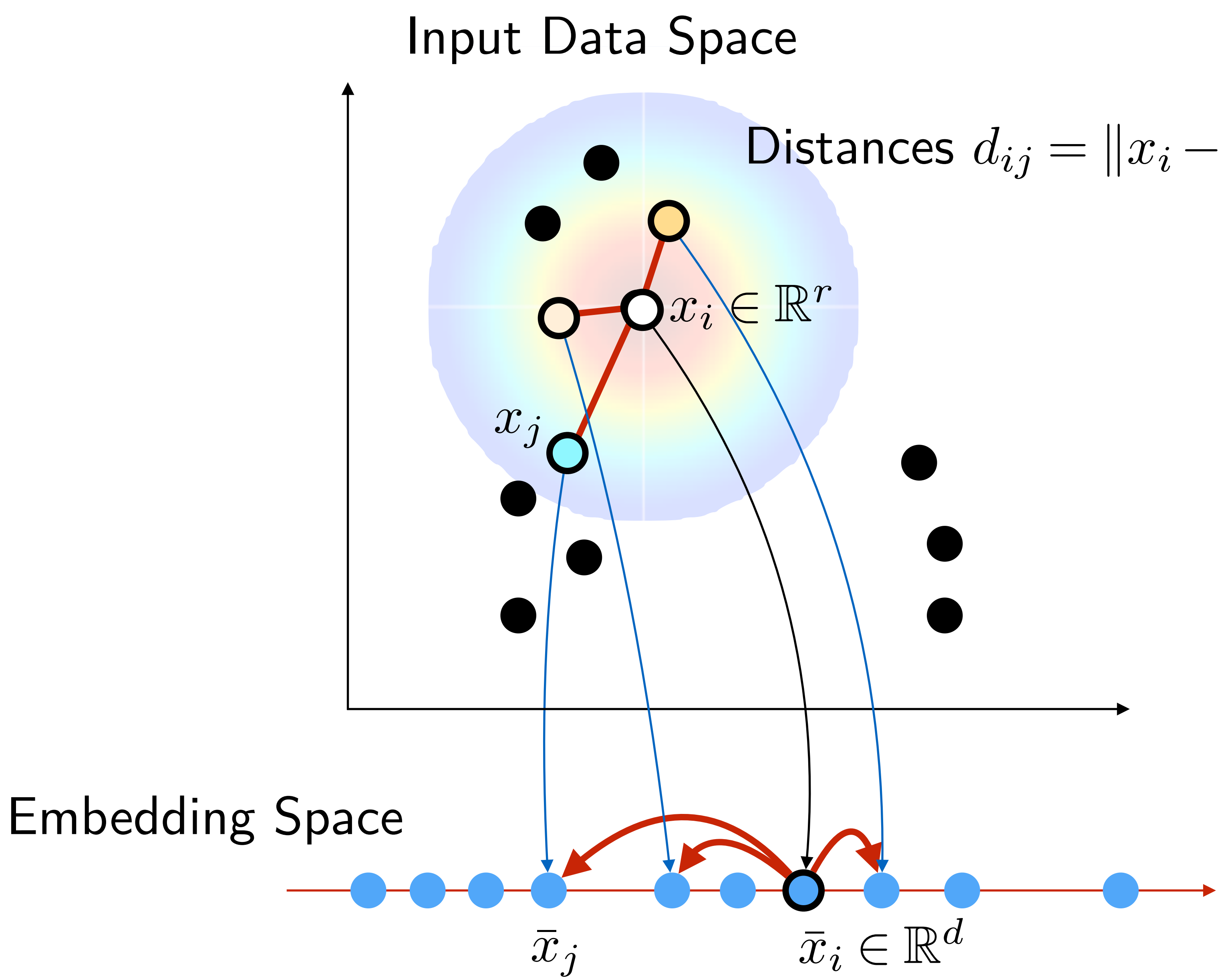
$$\min_{\bar{x}} \sum_i \mathrm{KL}(p_i \,\|\, q_i)$$

$$q_i(j) = \frac{e^{-\bar{d}_{ij}^2}}{\sum_{j\neq i} e^{-\bar{d}_{ij}^2}}, \quad \forall j \neq i$$

Distances $\bar{d}_{ij} = \|\bar{x}_i - \bar{x}_j\|$

- $\operatorname{argmin}_{\bar{x}} \sum_i \mathrm{KL}(p_i \,\|\, q_i) = \operatorname{argmax}_{\bar{x}} \sum_i \sum_j p_i(j) \log q_i(j)$
- Maximum likelihood learning to predict the "nearest neighbor" by q
- In comparison to MDS: normalization, distant neighbors are down-weighted
- In comparison to "Contrastive Learning": distribution $p_i(j)$ instead of a known "positive"

t-Distributed SNE (t-SNE)



Distances $d_{ij} = \|x_i - x_j\|$ (or given in other way)

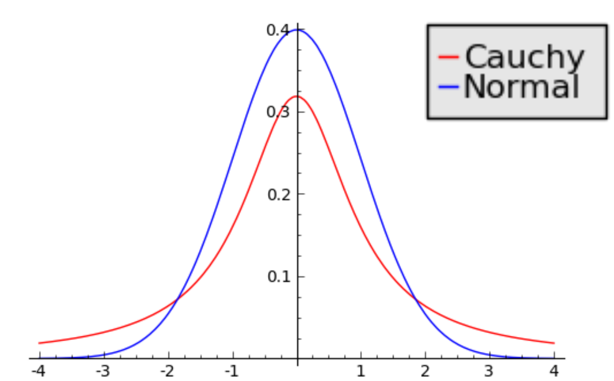
$$p(j|i) \propto e^{-d_{ij}^2/2\sigma^2}, \quad \forall j \neq i$$

$$\min_{\bar{x}} \sum_i \mathrm{KL}\big(p(\cdot|i) \,\|\, q(\cdot|i)\big)$$

$$q(j|i) \propto \left(1 + \bar{d}_{ij}^2\right)^{-1}, \quad \forall j \neq i$$

Distances $\bar{d}_{ij} = \|\bar{x}_i - \bar{x}_j\|$

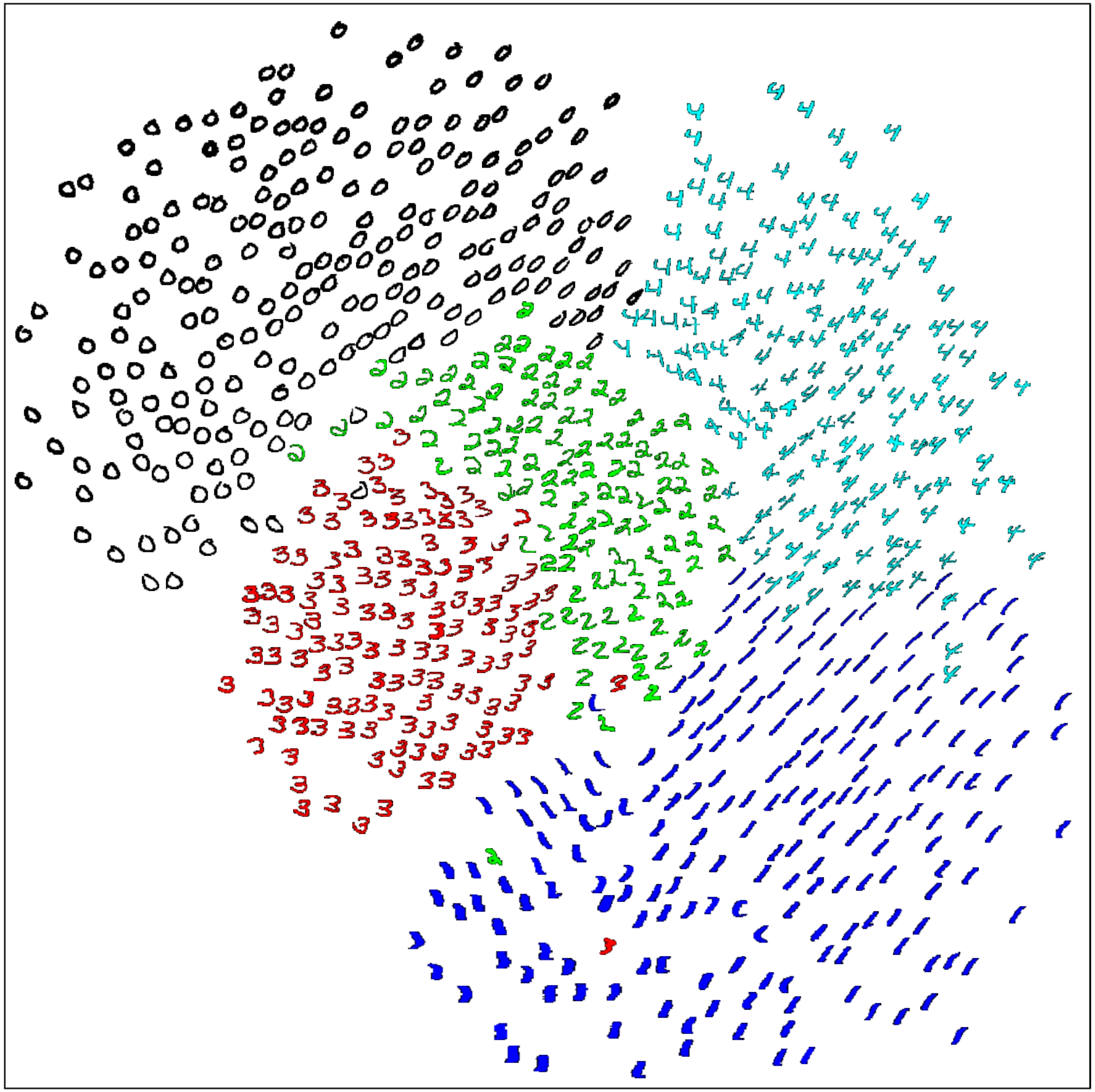
(Student t with 1 degree of freedom is Cauchy)



- Improves clustering of the data (sometimes too much)
- Omitted: symmetrization, initialization, adaptive sigma

[Maaten & Hinton (2008): Visualizing Data using t-SNE]

Examples

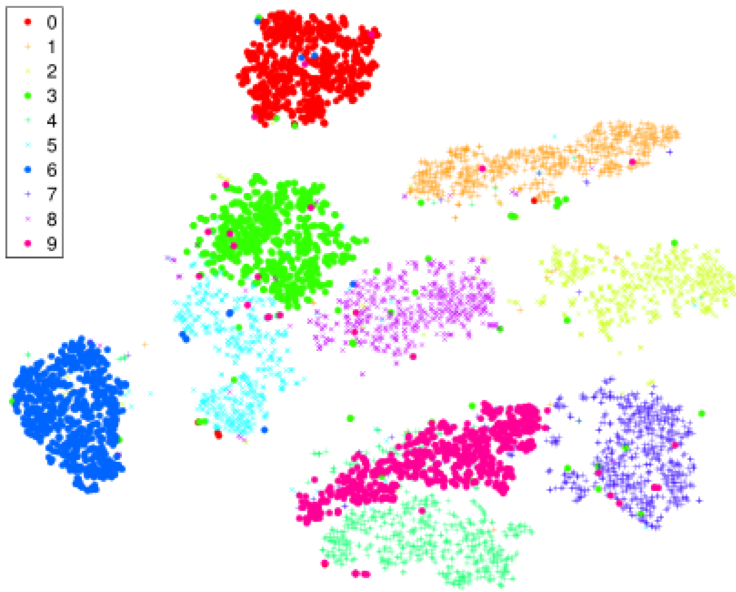


SNE algorithm on 256-dimensional grayscale images of handwritten digits

[Hinton & Roweis (2002): Stochastic Neighbor Embedding]

Examples

MNIST data

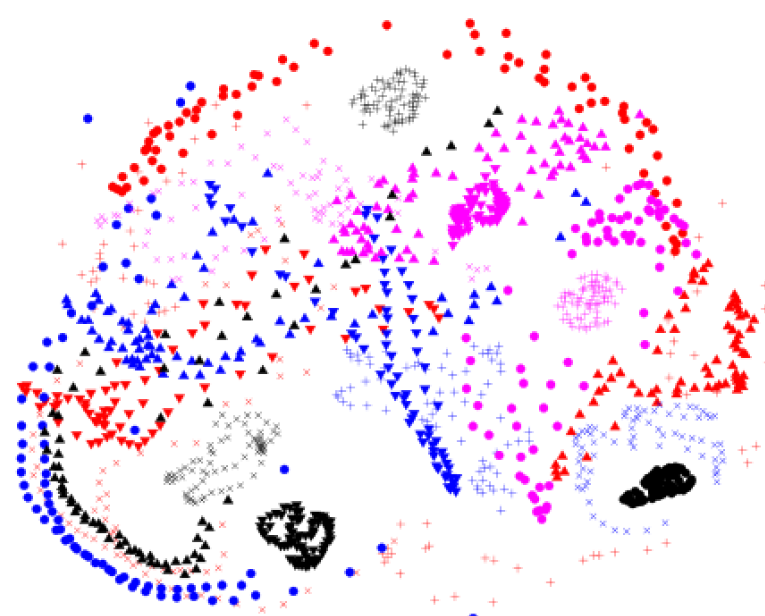


t-SNE

Sammon Mapping: $\mathcal{L} = \sum_{i \neq j} \frac{(d_{ij} - \bar{d}_{ij})^2}{d_{ij}}$

COIL data

t-SNE



Sammon Mapping

[Maaten & Hinton (2008): Visualizing Data using t-SNE]

Unsupervised Representation Learning

Latent Variable Models

- We explicitly model that multiple observations have some common causes (common factors) that are not directly observed or, *latent*
- Examples:
  - The true class labels for classification are not observed, only labels given by several experts, which may be error-prone. The true label is latent.
  - A text document has a particular topic that we do not know. The frequency of word occurrence and their meaning depend on this common latent topic.
  - In a handwritten note, the style and appearance of letters follow a particular style, unique for each writer and the writer is latent.
  - In our word vector example, words may have multiple meanings.

Unsupervised Learning

- Model:

  x – observed, z – latent, c – conditioning (side information)

  $p_\theta(x|z,c)$ – model of observations knowing the latent state

  $p_\theta(z|c)$ – model of latent states

  Generative model: $p_\theta(x,z|c) = p_\theta(x|z,c)p_\theta(z,c)$

Observation x

Conditioning word, $\cdot$

Latent meaning per conditioning word, z

Latent embedding $u(x)$, latent embedding $v(z,c)$

- Maximum likelihood learning (omitting conditioning on c):

  Observations $\{x_i\}_{i=1}^n$

  Likelihood of x_i: $p_\theta(x_i) = \sum_z p_\theta(x_i,z) = \sum_z p_\theta(x_i|z)p_\theta(z)$

  Log-likelihood: $L(\theta) = \sum_i \log \sum_z p_\theta(x_i|z;\theta)p_\theta(z)$

Learning by EM

- Need to maximize the log-likelihood of the **data evidence**:

$$\underbrace{\sum_i \log p(x_i)}_{\text{Evidence}} = \sum_i \log \underbrace{\sum_z p(x_i|z)p(z)}_{\text{difficult in general}}$$

$$= \sum_i \log \sum_z q(z|x_i) \frac{p(x_i|z)p(z)}{q(z|x_i)} \geq \underbrace{\sum_i \sum_z q(z|x_i) \log \frac{p(x_i|z)p(z)}{q(z|x_i)}}_{\text{Evidence Lower Bound (ELBO)}}$$

Holds for any distribution $q(z|x_i)$ by Jensen inequality

- Proof using KL (omitting the outer sum in i):

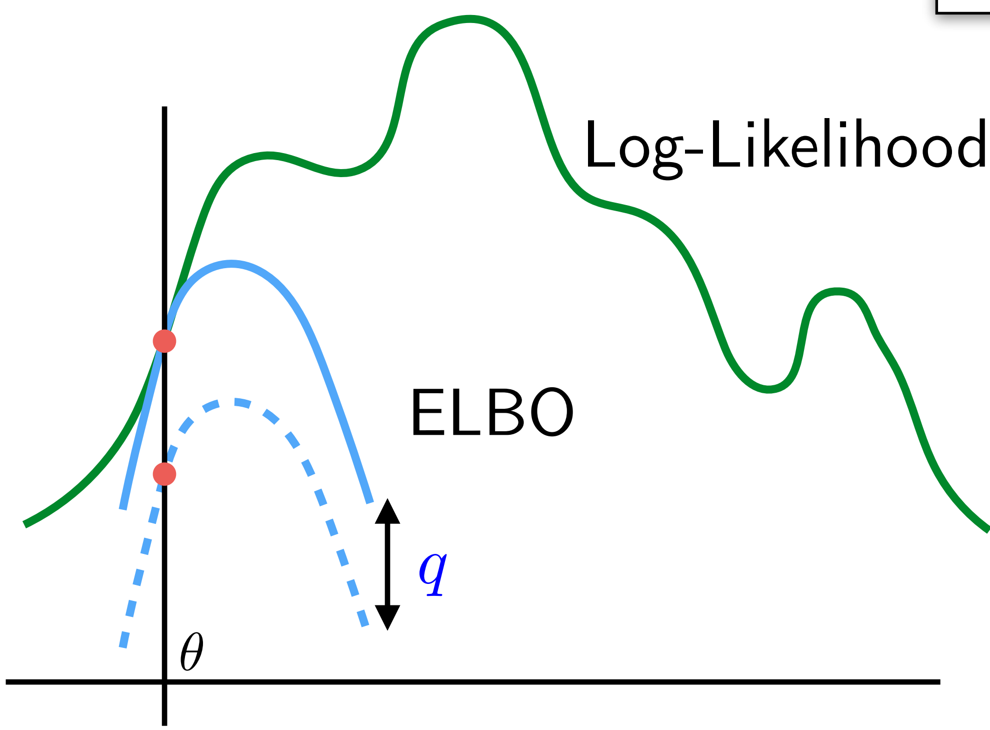
$$\underbrace{\log p(x)}_{\text{Evidence}} - \underbrace{\sum_z q(z|x) \log \frac{p(x,z)}{q(z|x)}}_{\text{ELBO}} = \sum_z q(z|x) \Big(\log p(x) - \log \frac{p(x,z)}{q(z|x)} \Big)$$

$$= \sum_z q(z|x) \Big(-\log \frac{p(x,z)}{p(x)q(z|x)} \Big)$$

$$= \sum_z q(z|x) \log \frac{q(z|x)}{p(z|x)} = D_{\mathrm{KL}}(q(z|x) \,\|\, p(z|x)) \geq 0.$$

EM Algorithm

$$\text{ELBO}(\theta, q) = \sum_i \sum_z q(z|x_i) \log \frac{p_\theta(x_i|z) p_\theta(z)}{q(z|x_i)}$$



- EM Algorithm:
  - **E**-step: For current θ maximize ELBO in q
  - **M**-step: For current q maximize ELBO in θ

- **E**-step:

  $\text{ELBO}(\theta, q) = \text{Evidence}(\theta) - \sum_i D_{\text{KL}}(q(z|x_i) \,\|\, p_\theta(z|x_i))$

  Optimal q minimizes the reverse KL divergence to the *posterior* $p(z|x_i)$!

  When q is general enough, the optimizer is $q(z|x_i) = p_\theta(z|x_i)$ (Bayesian posterior expectation). *I.e.*, $q(z|Y_t, x_t)$ learns to perform inference: it predicts distribution over hidden representations given observation x.

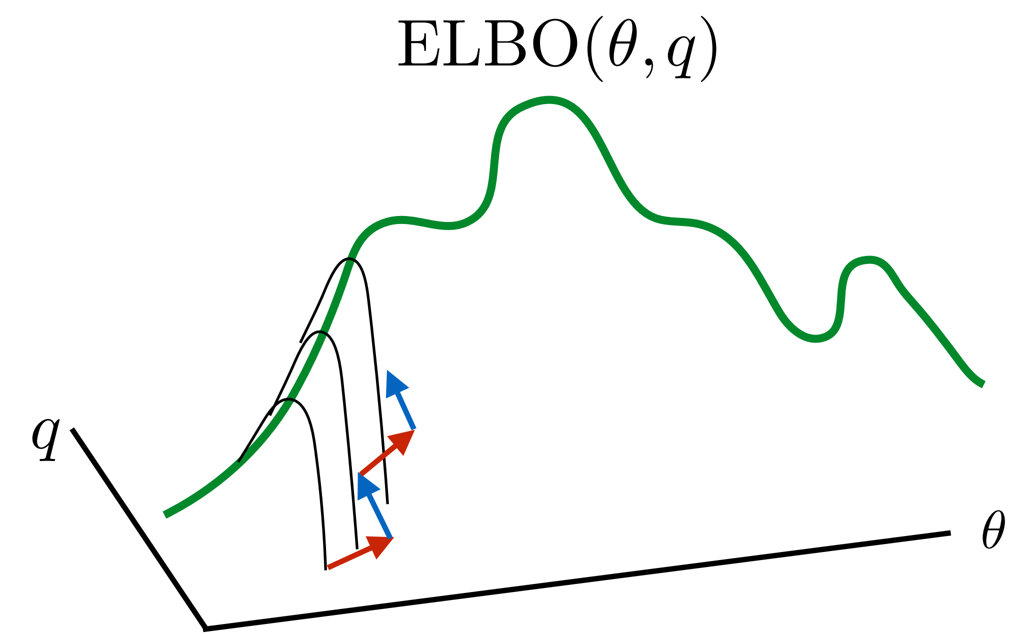
- **M**-step:

  $$\underset{\theta}{\text{argmax}} \sum_i \sum_z q(z|x_i) \log p_\theta(x_i|z)$$

  Supervised learning problem (*maximum* likelihood), assuming that $q(z|x_i)$ is the true data conditional distribution.

Stochastic EM Algorithm

$$\text{ELBO}(\theta, q) = \sum_i \sum_z q(z|x_i) \log \frac{p_\theta(x_i|z)p_\theta(z)}{q(z|x_i)}$$



- EM Algorithm:
  - **E**-step: For current θ maximize ELBO in q
  - **M**-step: For current q maximize ELBO in θ

- **E**-step:
  - Perform one step of SGD for improving $q \to$ Stochastic Variational Inference
  - Need to differentiate expectation in $q(z|x_i)$
- **M**-step:
  - Perform one step of SGD $\to$ Stochastic EM
  - Like supervised learning with $z \sim q(z|x_i)$

Multi-Sense Word Vectors

- Learned prior distribution

WORD	$p(z\|c)$	NEAREST NEIGHBOURS
python	0.33	monty, spamalot, cantsin
	0.42	perl, php, java, c++
	0.25	molurus, pythons
apple	0.34	almond, cherry, plum
	0.66	macintosh, iifx, iigs

- Inference $q(z|x,c)$

`Our train has departed from Waterloo at 1100pm`

Closest word:

Probabilities of meanings
0.948032
0.00427984
0.000470485
0.0422029
0.0050148

"paddington"
"euston"
"victoria"
"liverpool"
"moorgate"
"via"
"london"

`Who won the Battle of Waterloo?`

Probabilities of meanings
0.0000098
0.997716
0.0000309
0.00207717
0.00016605

"sheriffmuir"
"agincourt"
"austerlitz"
"jena-auerstedt"
"malplaquet"
"königgrätz"
"mollwitz"
"albuera"