Deep Learning (BEV033DLE)
Lecture 11
t-SNE, Stochastic Expectation Maximization

Czech Technical University in Prague

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

KL Divergence

KL Divergence

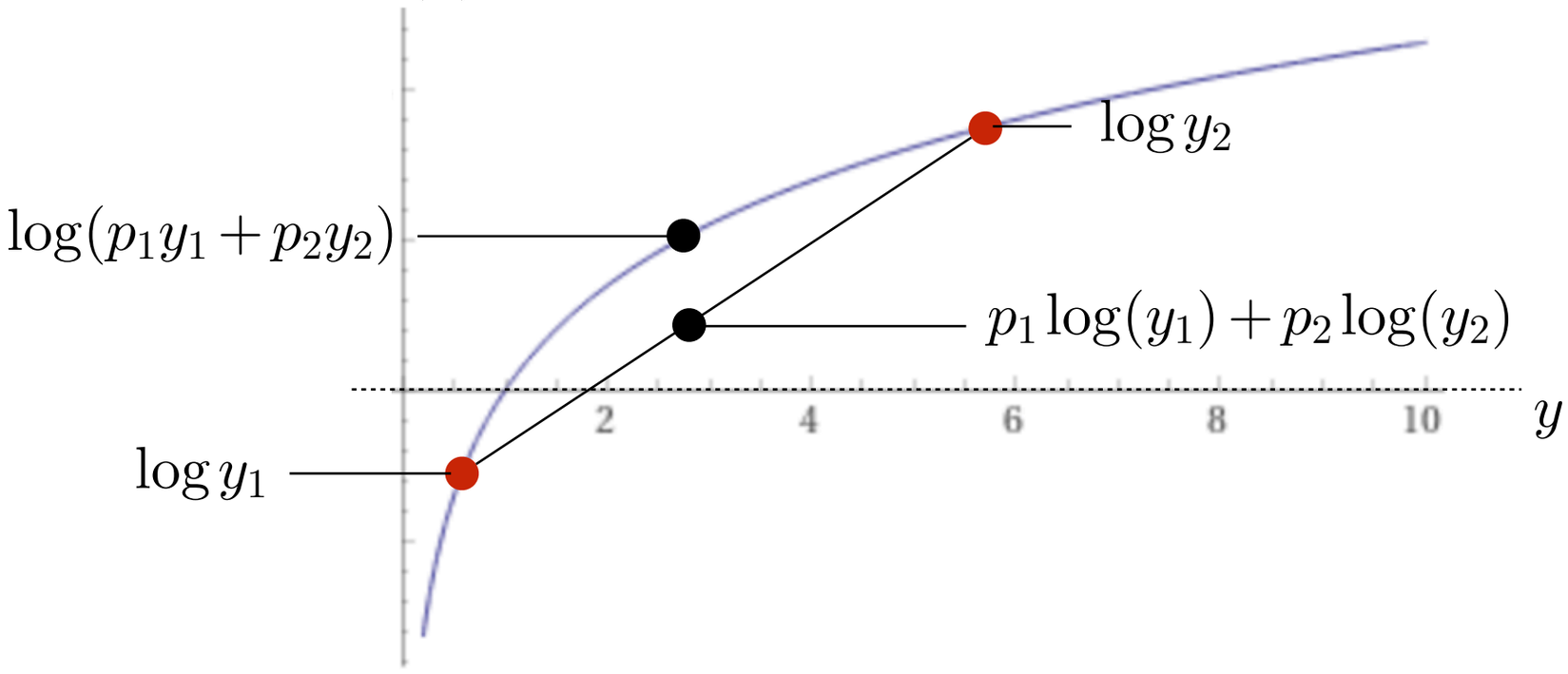
- Let $p(x)$ and $q(x)$ be two probability distributions.
- Kullback–Leibler divergence of p and q is

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

  - 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:

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

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

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

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

Asymmetry

Minimizing **forward KL** divergence:

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

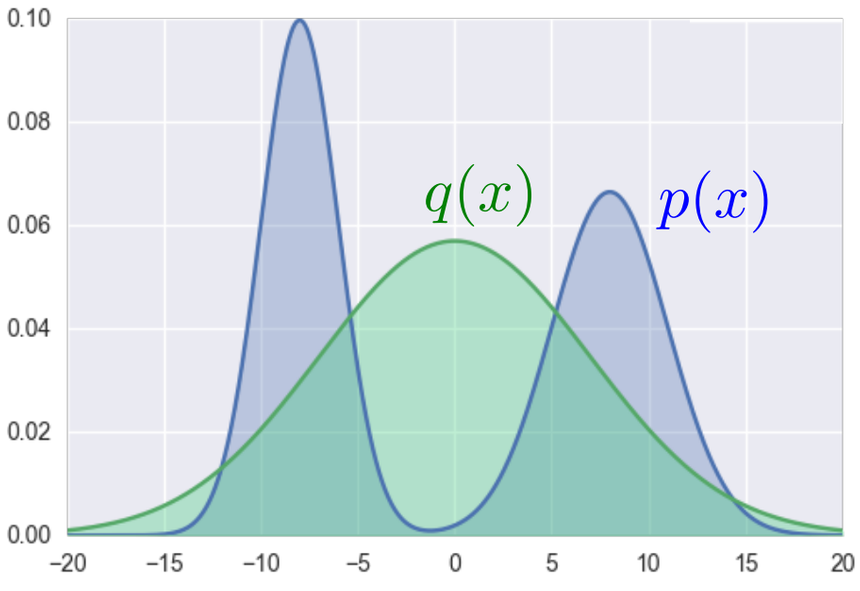
$$\min_q \int p(x)(\log p(x) - \log q(x))dx$$

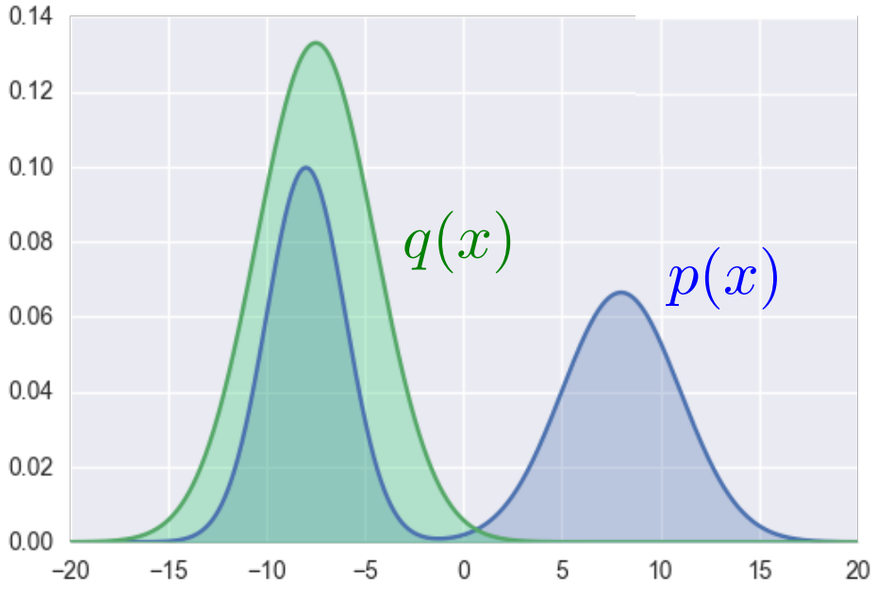
Minimizing **reverse KL** divergence:

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

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

Example: q is Gaussian





- Approximates well on average under p
- Matches moments
- Suffices to sample from $p(x)$

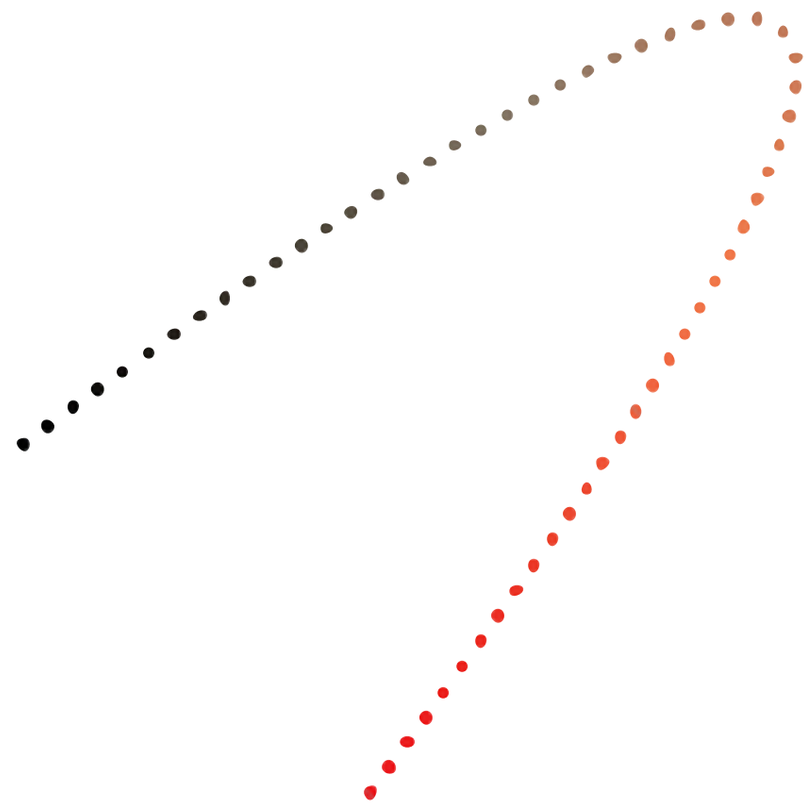
- Approximates well on average under q
- Selects a mode
- Requires $\log(p)$

Stochastic Neighbor Embedding

Motivation

- Goals:
  - Reduce dimensionality of the data for visualization
  - Preserve as much of the significant structure of the data as possible

Data in 2D

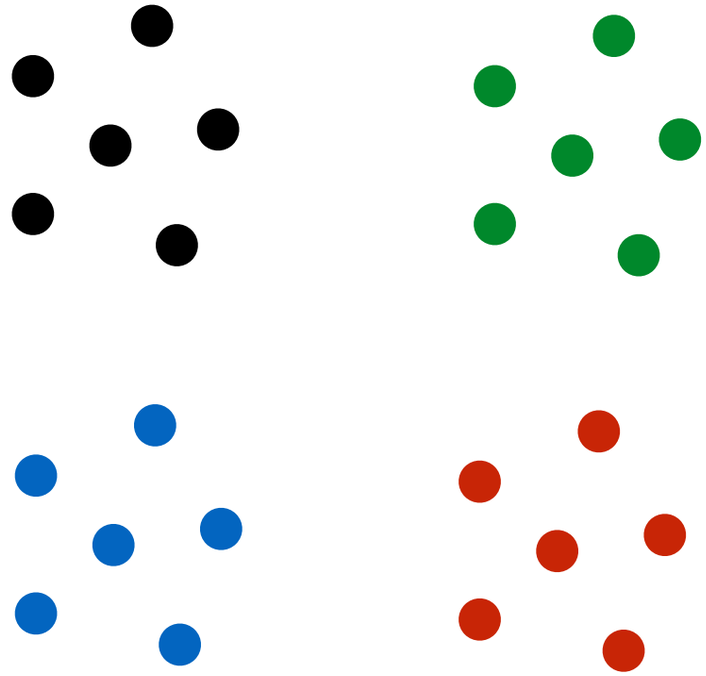


Draw its PCA embedding in 1D

Motivation

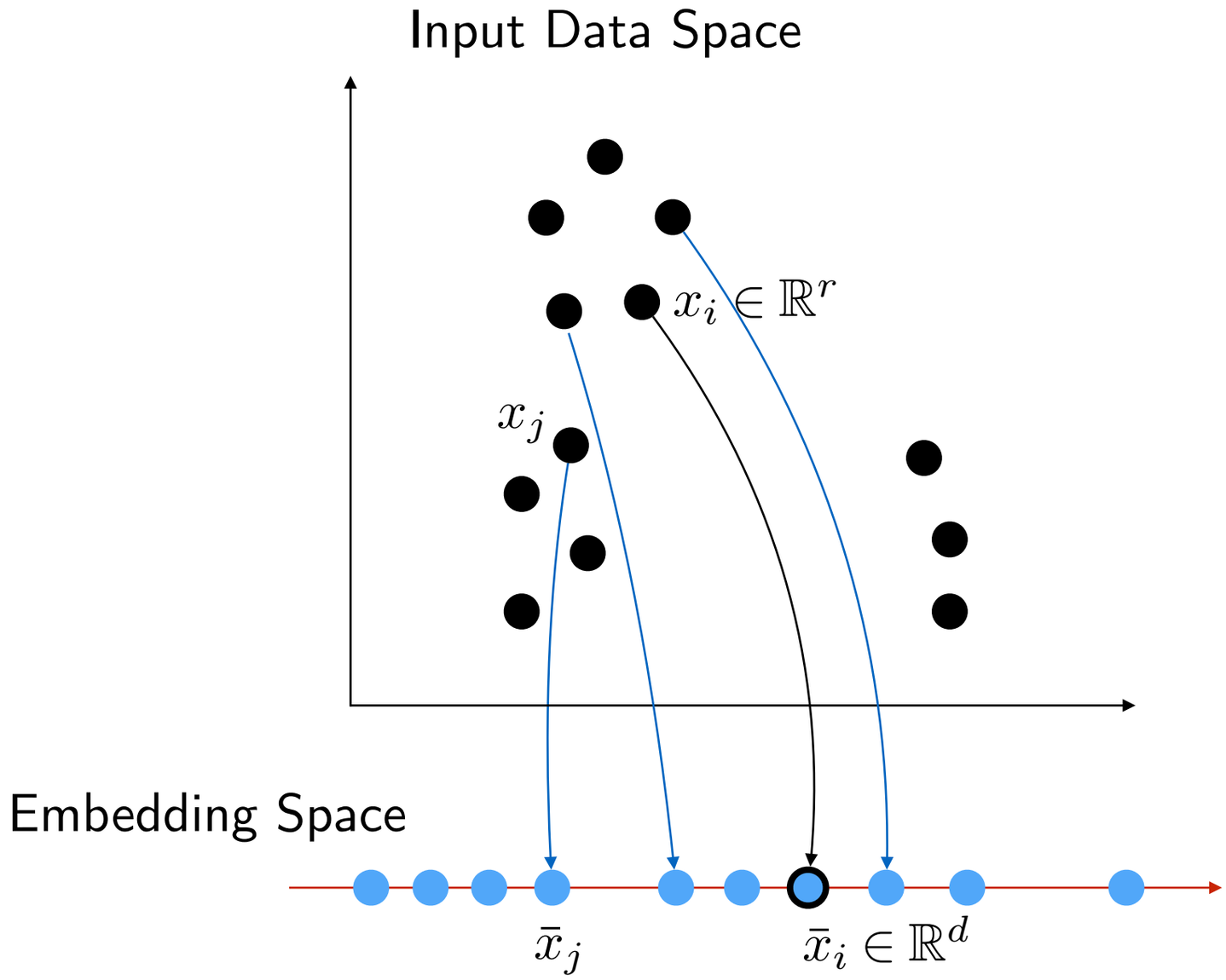
- Goal:
  - Reduce dimensionality,
  - Preserve as much of the significant structure of the data as possible

Data in 2D



No linear embedding would be good

Multidimensional Scaling (MSD)

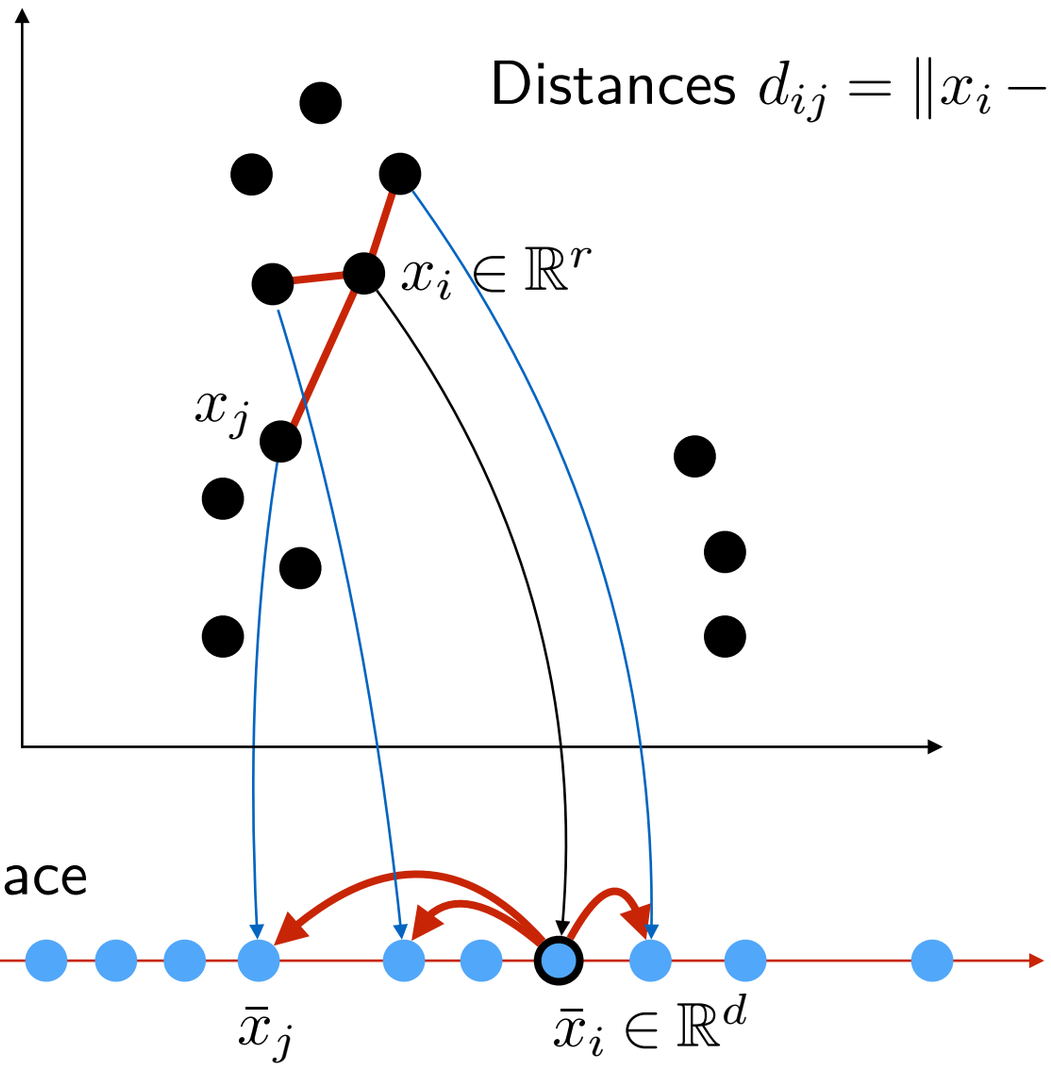


- Non-parametric model: for each data point x_t we find a corresponding embedding $\bar{x}_t$

Multidimensional Scaling (MSD)

Input Data Space

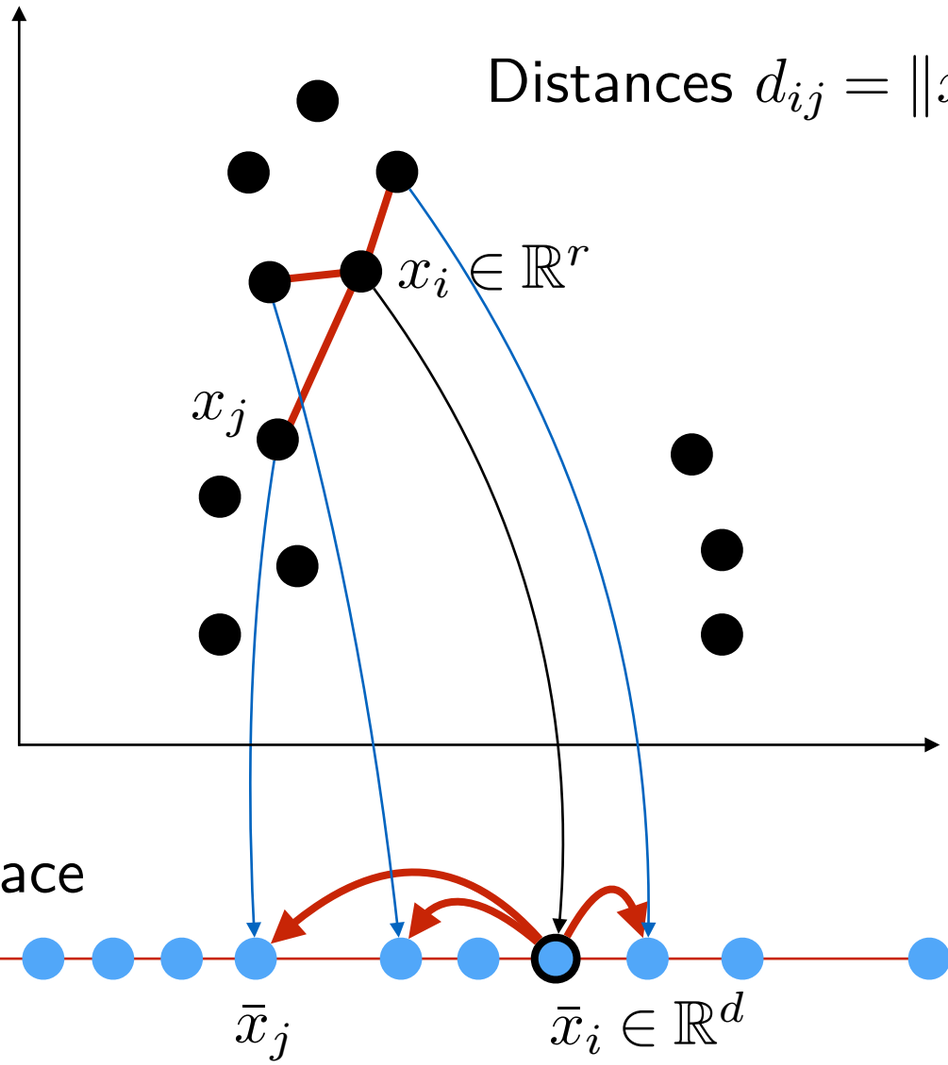
Distances $d_{ij} = \|x_i - x_j\|$ (or given in other way)



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

Multidimensional Scaling (MSD)

Input Data Space



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

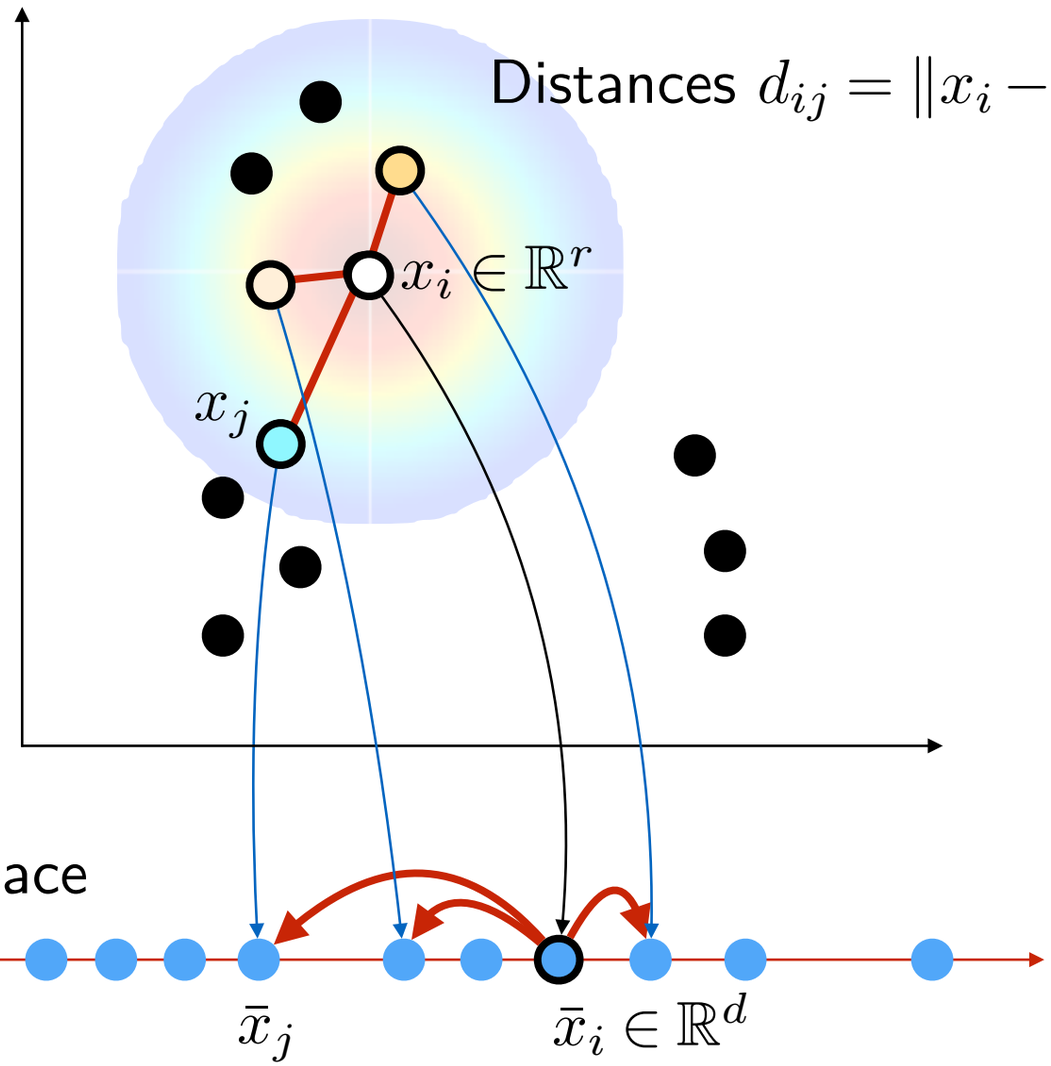
$$\min_{\bar{x}} \sum_{i \neq j} (d_{ij} - \bar{d}_{ij})^2$$

Can be too stringent

Embedding Space

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

Stochastic Neighbor Embedding



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

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

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

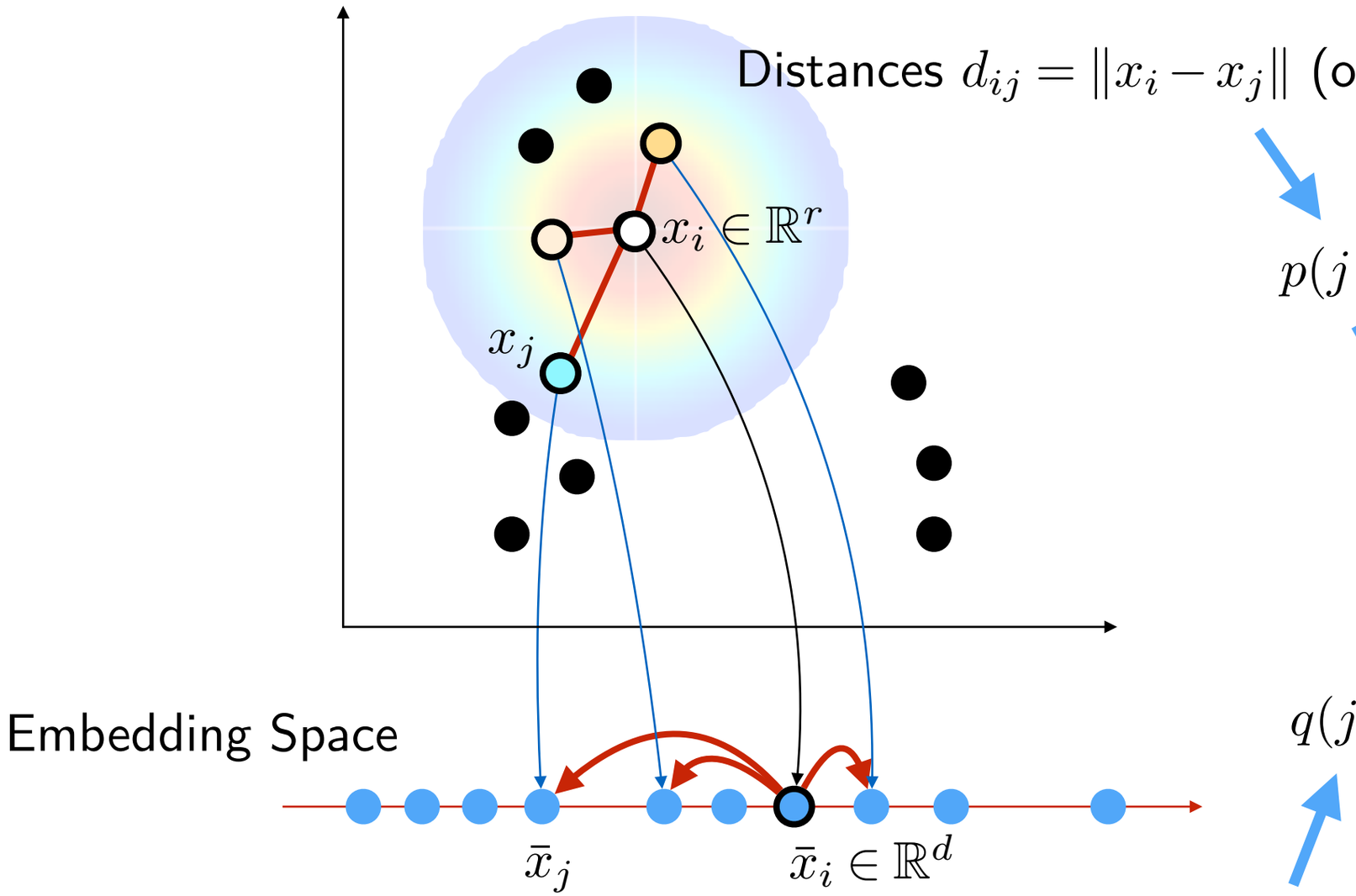
$$q(j|i) = \frac{e^{-\bar{d}_{ij}^2}}{\sum_{j \neq i} e^{-\bar{d}_{ij}^2}}, \quad \forall j \neq i$$

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

- $p(j|i)$ – probability that i picks j as its neighbor
- More distant neighbors have small probabilities – down-weighted
- Can be extended to mixture model (multi-sense embeddings)

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

t-Distributed SNE (t-SNE)



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

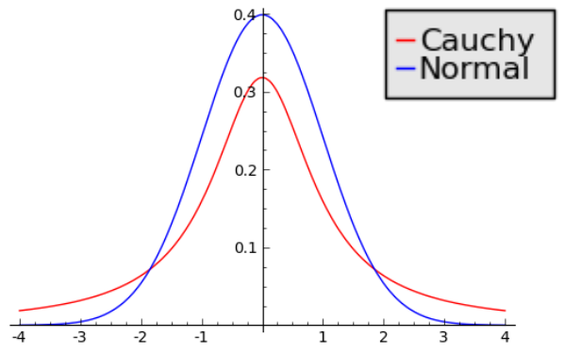
$$p(j|i) \propto e^{-d_{ij}^2/2\sigma^2}, \quad \forall j \neq i$$

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

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

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

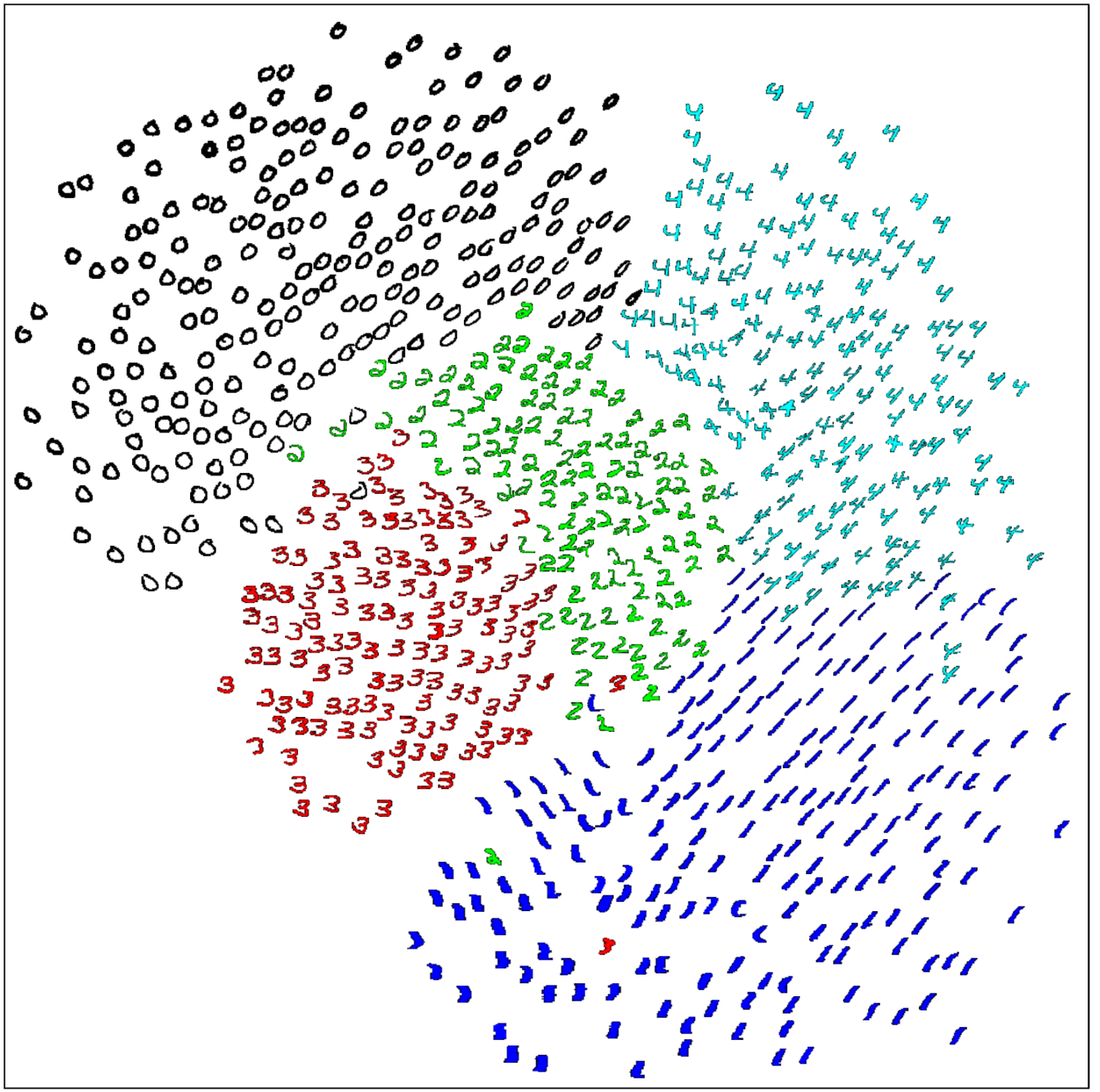
(Student t with 1 degree of freedom is Cauchy)



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

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

Examples

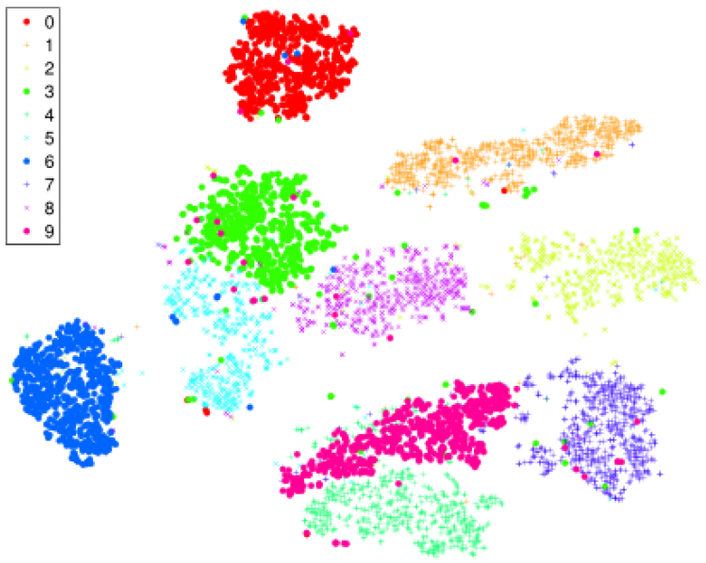


SNE algorithm on 256-dimensional grayscale images of handwritten digits

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

Examples

MNIST data

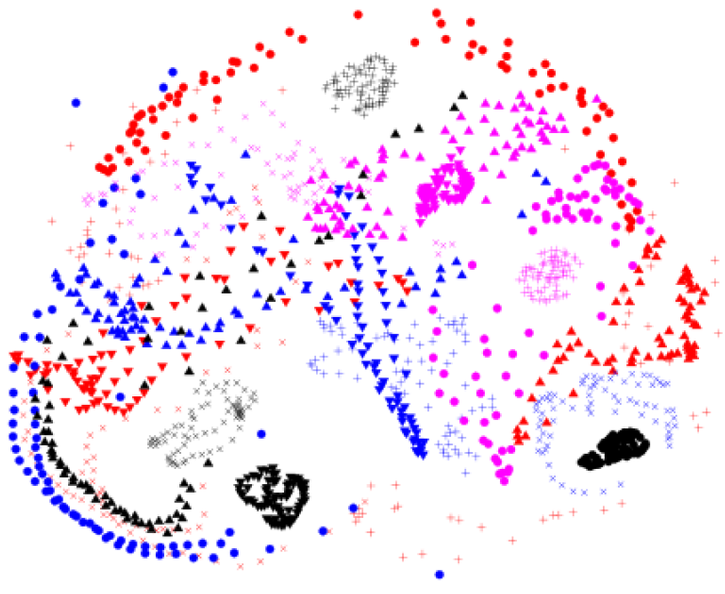


t-SNE

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]

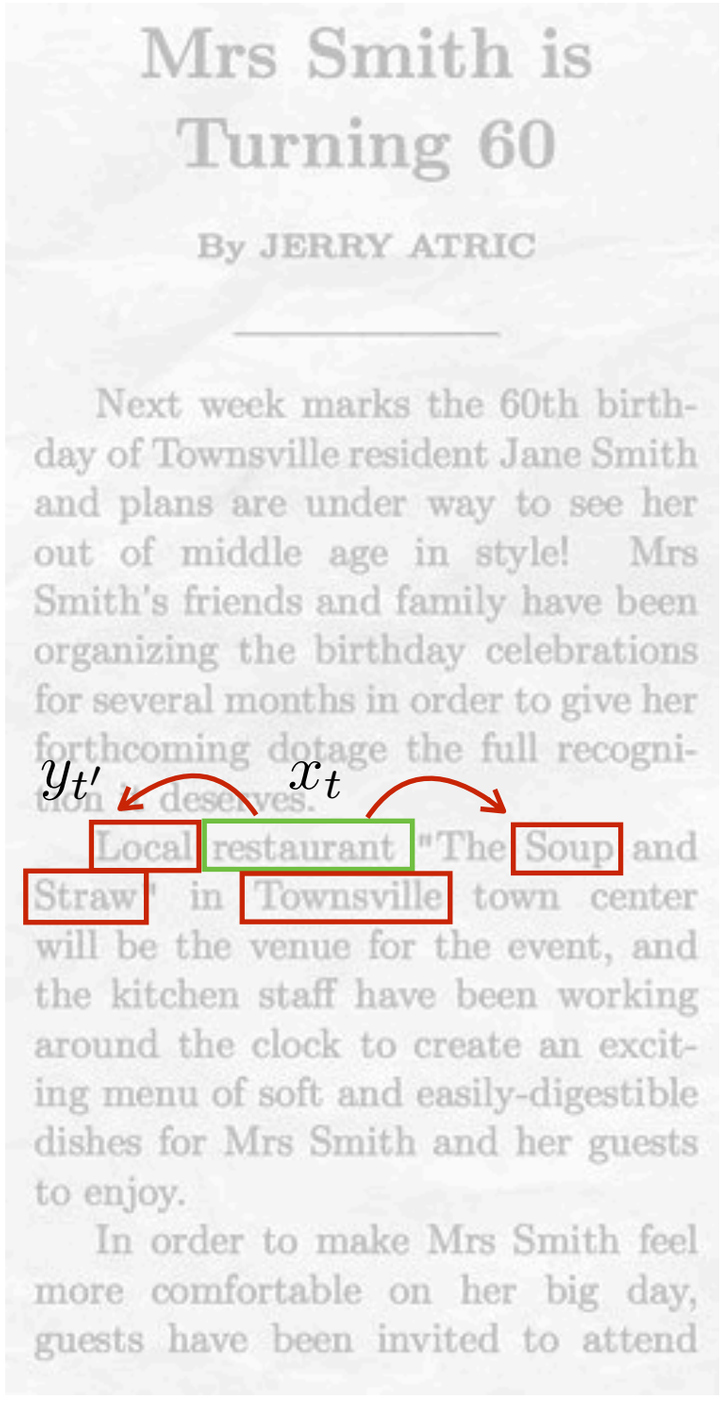
Stochastic EM / Multi-sense WV

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.

Base Word Vectors Model

- Finite vocabulary of words
- Position t
- Neighbors $t' \in \mathcal{N}(t)$
- Word vectors: $V_{x_t,:}$; $U_{y_{t'},:}$
- Model: $p(y_{t'}|x_t) \propto \exp(\text{sim}(x_t, y_{t'}))$



Multi-sense Word Vectors Model

✦ Often, words have multiple meanings (homographs):

I eat grape **jam**.

I was in a traffic **jam**.

- Word vectors: $V_{x_t,z,:}$, $U_{y_t,:}$, $z \in \{1, \ldots \text{max meanings}\}$ (simplifiaction)
- All words in the context commonly depend on the latent meaning of the current word:

$$\underbrace{\prod_{t' \in \mathcal{N}(t)} p(y_{t'}|z, x_t)\, p(z|x_t)}_{p(Y_t|z,x_t)}$$

$p(z|x_t)$ – e.g. a discrete distribution per word to be learned

Y_t – context words

- Do not observe z, the probability of the observed context is given by *marginalization*:

$$p(Y_t|x_t) = \sum_z \prod_{t' \in \mathcal{N}(t)} p(y_{t'}|z, x_t) p(z|x_t)$$

- **Learning** (ML):

$$\max \sum_t \log \sum_z \prod_{t' \in \mathcal{N}(t)} p(y_{t'}|z, x_t) p(z|x_t)$$

[Burtanov et al. (2016): Breaking Sticks and Ambiguities with Adaptive Skip-gram]

Learning by EM

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

$$\underbrace{\sum_t \log p(Y_t|x_t)}_{\text{Evidence}} = \sum_t \log \underbrace{\sum_z}_{\text{difficult}} p(Y_t|z,x_t)p(z|x_t)$$

$$\geq \underbrace{\sum_t \sum_z q(z|x_t,Y_t) \log \frac{p(Y_t|z,x_t)p(z|x_t)}{q(z|x_t,Y_t)}}_{\text{Evidence Lower Bound (ELBO)}}$$

Holds for any distribution $q(z|x_t,Y_t)$ by Jensen inequality

- Proof using KL (omitting dependence on x_t everywhere and the outer sum in t):

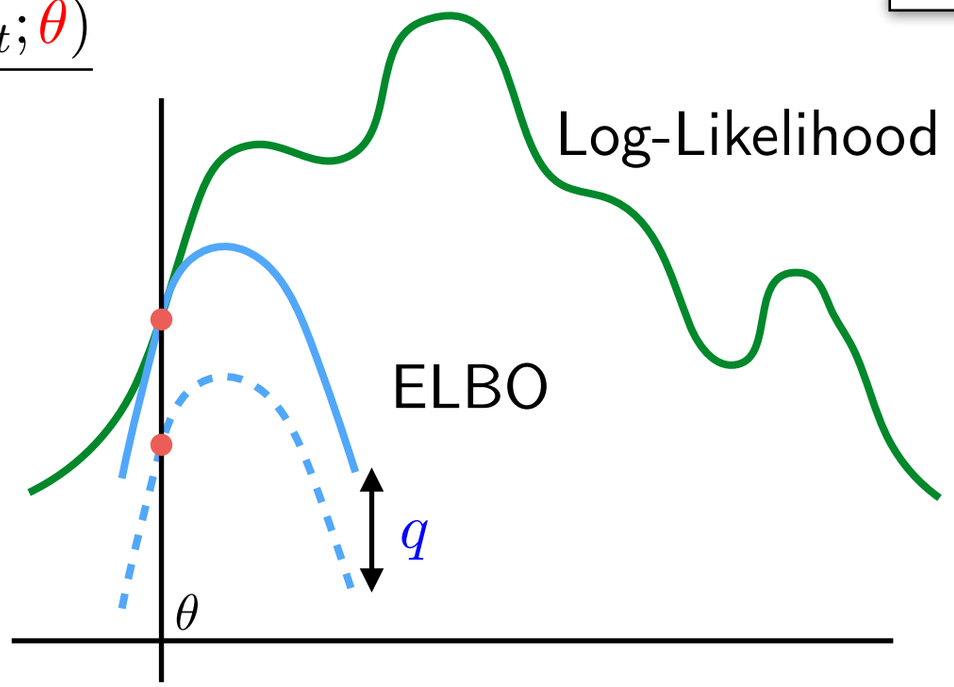
$$\underbrace{\log p(Y)}_{\text{Evidence}} - \underbrace{\sum_z q(z|Y) \log \frac{p(Y,z)}{q(z|y)}}_{\text{ELBO}} = \sum_z q(z|Y) \Big(\log p(Y) - \log \frac{p(Y,z)}{q(z|y)} \Big)$$

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

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

EM Algorithm

$$\text{ELBO}(\theta, q) = \sum_t \sum_z q(z|x_t, Y_t) \log \frac{p(Y_t|z, x_t; \theta) p(z|x_t; \theta)}{q(z|x_t, Y_t)}$$



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

- **E**-step:

  $\text{ELBO}(\theta, q) = \text{Evidence}(\theta) - \sum_t D_{\text{KL}}(q(z|Y_t, x_t) \,\|\, p(z|Y_t, x_t; \theta))$

  Optimal q minimizes the reverse KL divergence!

  When q is general enough, the optimizer is $q(z|Y_t, x_t) = p(z|Y_t, x_t, \theta)$ (estimate with Bayes theorem). $q(z|Y_t, x_t)$ learns to perform inference.

- **M**-step:

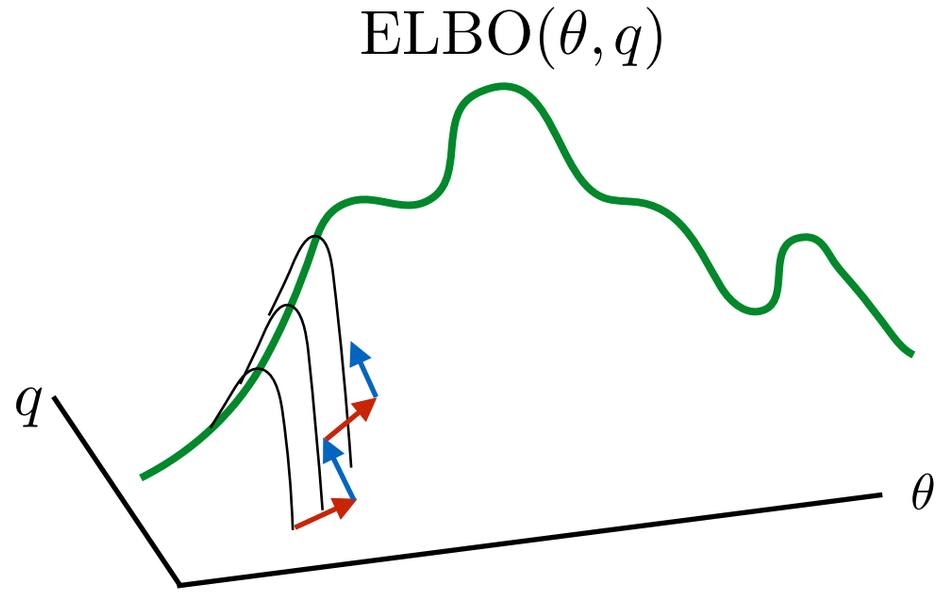
  $\text{argmax}_\theta \sum_t \sum_z q(z|x_t, Y_t) \log p(Y_t|z, x_t; \theta)$

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

Stochastic EM Algorithm

$$\text{ELBO}(\theta, q) = \sum_t \sum_z q(z|x_t, Y_t) \log \frac{p(Y_t|z, x_t; \theta) p(z|x_t; \theta)}{q(z|x_t, Y_t)}$$

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



- **E**-step:

  $$\operatorname*{argmax}_q \text{ELBO}(\theta, q) = \operatorname*{argmax}_q \sum_t \sum_z q(z|x_t, Y_t)(\log p(Y_t, z|x_t; \theta) - \log q(Y_t|z, x_t))$$

  Perform one step of SGD for improving $q \rightarrow$ Stochastic Variational Inference

- **M**-step:

  $$\operatorname*{argmax}_\theta \text{ELBO}(\theta, q) \operatorname*{argmax}_\theta \sum_t \sum_z q(z|x_t, Y_t) \log p(Y_t|z, x_t; \theta)$$

  Perform one step of SGD $\rightarrow$ Stochastic EM

Multi-Sense Word Vectors

- Learned prior distribution $p(z|x)$

WORD	$p(z)$	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
date	0.10	unknown, birth, birthdate
	0.28	dating, dates, dated
	0.31	to-date, stateside
	0.31	deadline, expiry, dates
bow	0.46	stern, amidships, bowsprit
	0.38	spear, bows, wow, sword
	0.16	teign, coxs, evenlode

Discovers semantic clusters

Closest words to "platform"		
fwd	stabling	software
sedan	turnback	ios
fastback	pebblemix	freeware
chrysler	citybound	netfront
hatchback	metcard	linux
notchback	underpass	microsoft
rivieraoldsmobile	sidings	browser
liftback	tram	desktop
superoldsmobile	cityrail	interface
sheetmetal	trams	newlib

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

`Our train has departed from Waterloo at 1100pm`

Probabilities of meanings
0.948032
0.00427984
0.000470485
0.0422029
0.0050148

Closest words:
"paddington"
"euston"
"victoria"
"liverpool"
"moorgate"
"via"
"london"
"street"
"central"
"bridge"

Multi-Sense Word Vectors

- Learned prior distribution $p(z|x)$

WORD	$p(z)$	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
date	0.10	unknown, birth, birthdate
	0.28	dating, dates, dated
	0.31	to-date, stateside
	0.31	deadline, expiry, dates
bow	0.46	stern, amidships, bowsprit
	0.38	spear, bows, wow, sword
	0.16	teign, coxs, evenlode

Discovers semantic clusters

Closest words to "platform"		
fwd	stabling	software
sedan	turnback	ios
fastback	pebblemix	freeware
chrysler	citybound	netfront
hatchback	metcard	linux
notchback	underpass	microsoft
rivieraoldsmobile	sidings	browser
liftback	tram	desktop
superoldsmobile	cityrail	interface
sheetmetal	trams	newlib

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

Who won the Battle of Waterloo?

Probabilities of meanings
0.0000098
0.997716
0.0000309
0.00207717
0.00016605

Closest words:
"sheriffmuir"
"agincourt"
"austerlitz"
"jena-auerstedt"
"malplaquet"
"königgrätz"
"mollwitz"
"albuera"
"toba-fushimi"
"hastenbeck"