Deep Learning (BEV033DLE)
Lecture 10
Learning Representations, Stochastic EM

Alexander Shekhovtsov

Czech Technical University in Prague

Block Overview

- Lecture 10:
  - Examples of Learning Representation
    - Embedding of words, tSNE
  - KL Divergence
    - Forward & Reverse, KL & Cross-entropy
  - Latent Variable Models
    - Multi-sense word vectors
    - Stochastic EM, Variational inference

- Lecture 11: Variational Autoencoders

- Lecture 12: Supervised Representation and Similarity Learning

Two Examples of Learning Representation

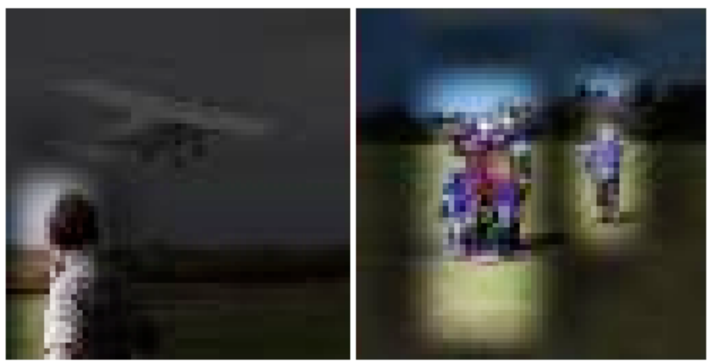
Motivation: Networks Learn Useful Representations

- In networks trained for different complex problems some intermediate layers activations correspond object parts

lamps in places net

wheels in object net

people in video net



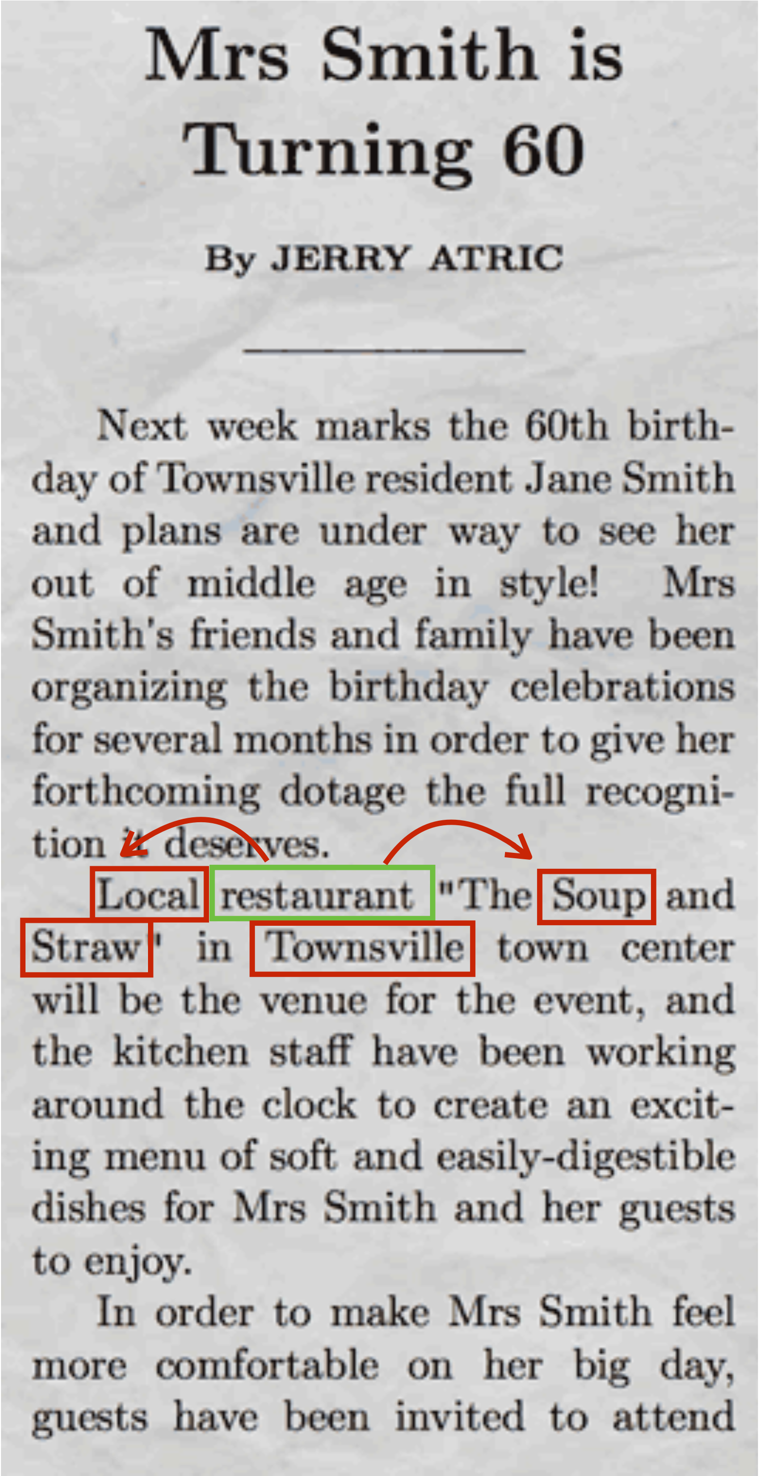
Word Vectors

- Example: Simple model for predicting context words:
  - Assume a finite vocabulary I, $|I| = n$
  - For every word x in the text, try to predict all nearby words y

- A completely general model:
  - $$p(y|x) = P_{y,x} = \frac{\exp(W_{y,x})}{\sum_{y'} \exp(W_{y',x})}$$
  - $P \in \mathbb{R}_+^{n \times n}$ — conditional probability matrix
  - $W \in \mathbb{R}^{n \times n}$ — unconstrained
  - Learn by maximum likelihood:

$$\max_W \prod_t \underbrace{\prod_{t' \in \mathcal{N}(t)} p(y_{t'}|x_t)}_{\text{Naive Bayes model}},$$

  t – position in the text, $\mathcal{N}(t)$ – nearby positions
  - Learning is inefficient: matrix P is too large



Mrs Smith is Turning 60

By JERRY ATRIC

Next week marks the 60th birthday of Townsville resident Jane Smith and plans are under way to see her out of middle age in style! Mrs Smith's friends and family have been organizing the birthday celebrations for several months in order to give her forthcoming dotage the full recognition it deserves.

Local restaurant "The Soup and Straw" in Townsville town center will be the venue for the event, and the kitchen staff have been working around the clock to create an exciting menu of soft and easily-digestible dishes for Mrs Smith and her guests to enjoy.

In order to make Mrs Smith feel more comfortable on her big day, guests have been invited to attend

Word Vectors

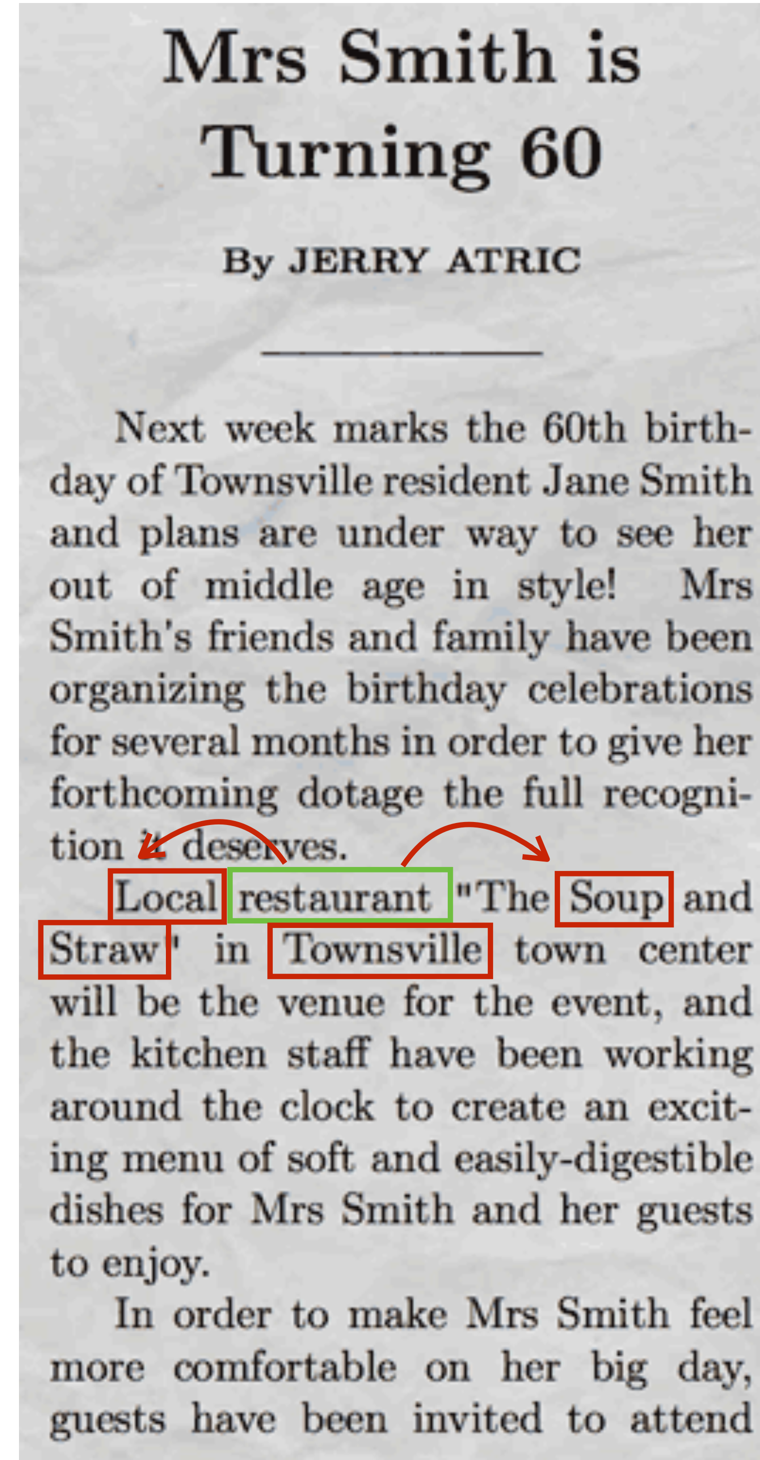
- More refined model:

$$p(y|x) = \frac{\exp W_{y,x}}{\sum_{y'} \exp(W_{y,x})}; \qquad W = U^\top V$$

  - $U, V \in \mathbb{R}^{n \times d}$ are *embedding matrices*
  - Learn by maximum likelihood:

$$\max_{U,V} \prod_t \prod_{t' \in \mathcal{N}(t)} p(y_{t'}|x_t),$$

- $V_{x,:} \in \mathbb{R}^d$ is the *embedding* (prototype) of x
- $U_{y,:} \in \mathbb{R}^d$ is another embedding of y



Mrs Smith is Turning 60

By JERRY ATRIC

Next week marks the 60th birthday of Townsville resident Jane Smith and plans are under way to see her out of middle age in style! Mrs Smith's friends and family have been organizing the birthday celebrations for several months in order to give her forthcoming dotage the full recognition it deserves.

Local restaurant "The Soup and Straw" in Townsville town center will be the venue for the event, and the kitchen staff have been working around the clock to create an exciting menu of soft and easily-digestible dishes for Mrs Smith and her guests to enjoy.

In order to make Mrs Smith feel more comfortable on her big day, guests have been invited to attend

Word Vectors

- More refined model:

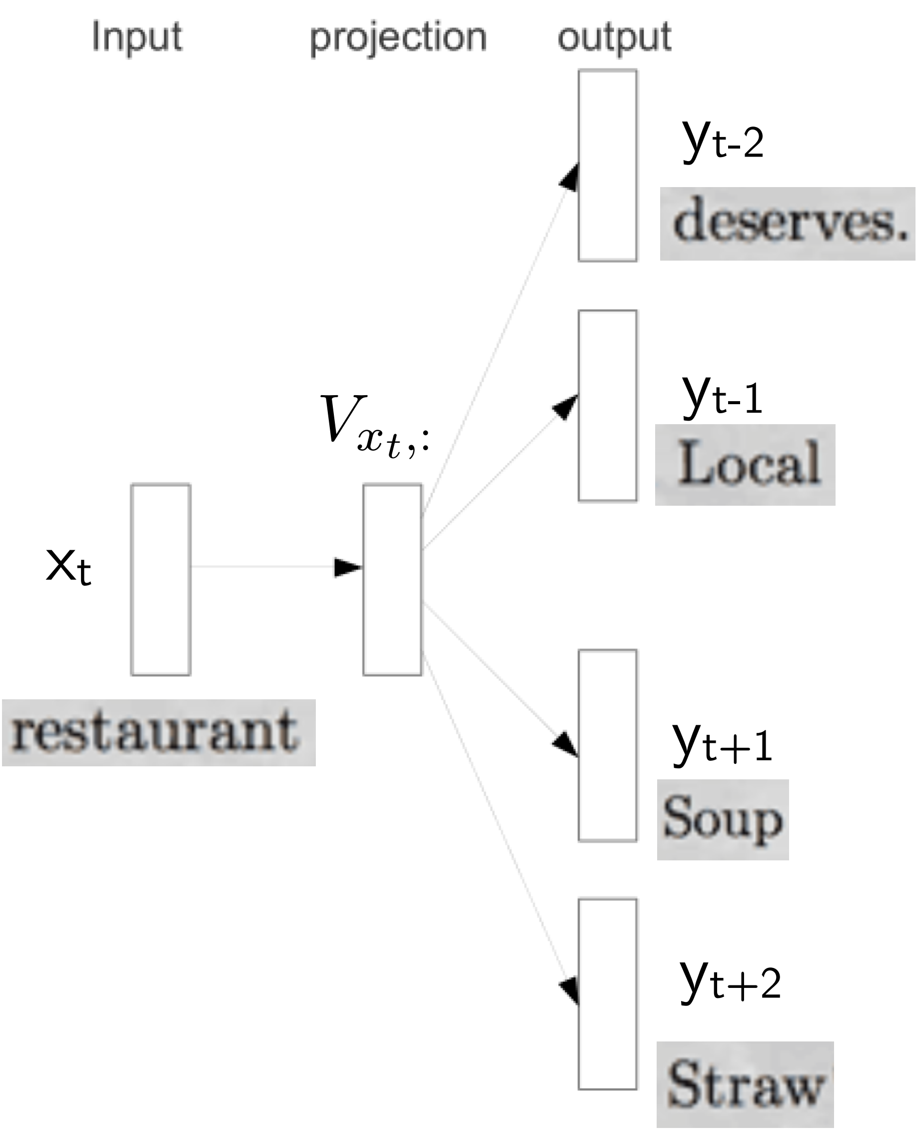
$$p(y|x) = \frac{\exp W_{y,x}}{\sum_{y'} \exp(W_{y,x})}; \qquad W = U^{\mathsf{T}} V$$

  - $U, V \in \mathbb{R}^{n \times d}$ are *embedding matrices*
  - Learn by maximum likelihood:

$$\max_{U,V} \prod_t \prod_{t' \in \mathcal{N}(t)} p(y_{t'}|x_t),$$

- $V_{x,:} \in \mathbb{R}^d$ is the *embedding* (prototype) of x
- $U_{y,:} \in \mathbb{R}^d$ is another embedding of y

Skip-gram model



- What problems we can solve using this model?

[Mikolov et al. (2013) Distributed Representations of Words and Phrases and their Compositionality]

Word Vectors

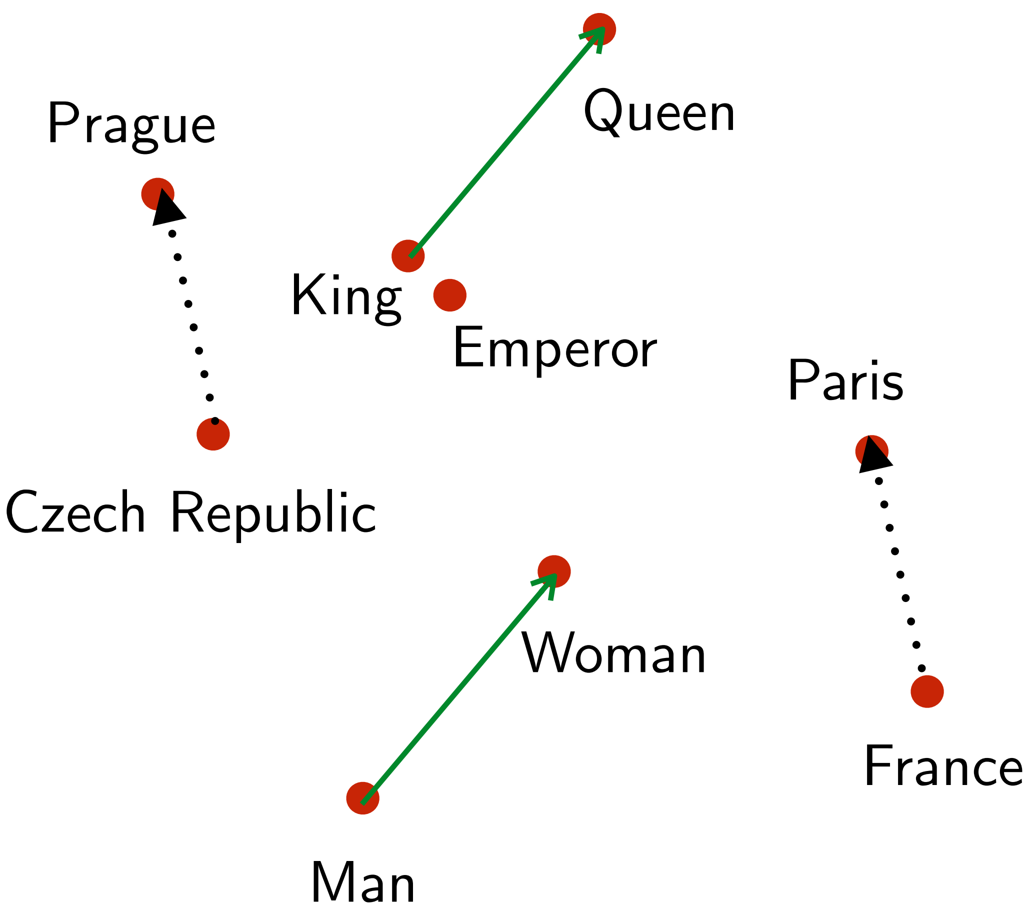
- Learned a **representation** of x as the embedding $V_{x,:} \in \mathbb{R}^d$:
- The directions of learned vectors turn out to capture abstract relations:
  - Semantic:

    "King" - "Man" + "Woman" ≈ "Queen"

    "Prague" - "Czech Republic" + "France" ≈ "Paris"

    "Czech" + "currency" ≈ "koruna"
  - Syntactic:

    "quick" - "quickly" ≈ "slow" - "slowly"
- Evaluated on a corpus of such relation predictions
- More complex (supervised) learning tasks are easier when using such vector representation



- What kind of learning is it: supervised or unsupervised?
  - We want to learn embeddings, no one ever supervises the embedding
  - Mathematically however we maximize supervised classification likelihood

Word Vectors

- Back to the learning formulation:
  - Maximize in $\theta = (U, V)$ the objective:

$$\sum_t \sum_{t' \in \mathcal{N}(t)} \log p(y_{t'} | x_t; \theta),$$

  - Let us define weights: $w_{t',t} \geq 0$: $\sum_{t'} w_{t',t} = 1$

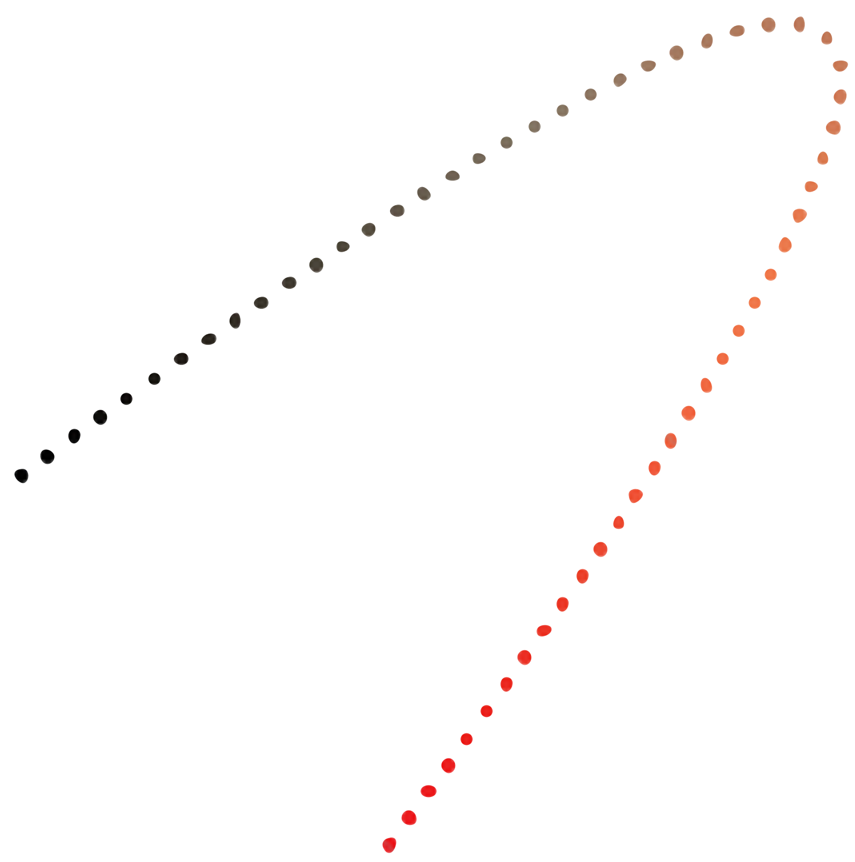
$$\sum_t \sum_{t'} w_{t',t} \log p(y_{t'} | x_t; \theta),$$

- What loss function it resembles?
  - Cross-entropy between discrete distributions on word indices t
  - This will establish an analogy with the t-SNE embedding (next)

Stochastic Neighbor Embedding

- Task: to represent high-dimensional vectors in low dimension, preserving the neighborhood relations

Data in 2D

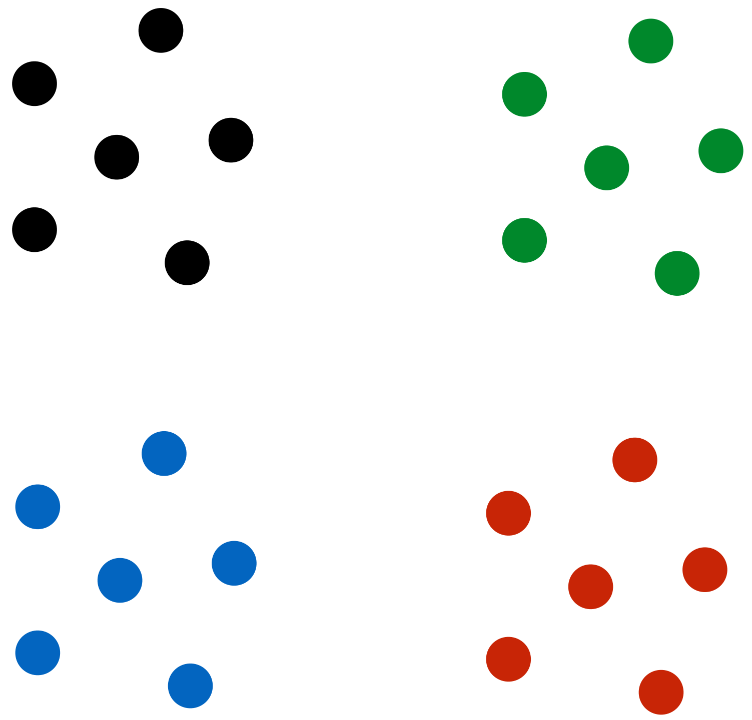


Draw its PCA embedding in 1D

Stochastic Neighbor Embedding

- Task: to represent high-dimensional vectors in low dimension, preserving the neighborhood relations

Data in 2D

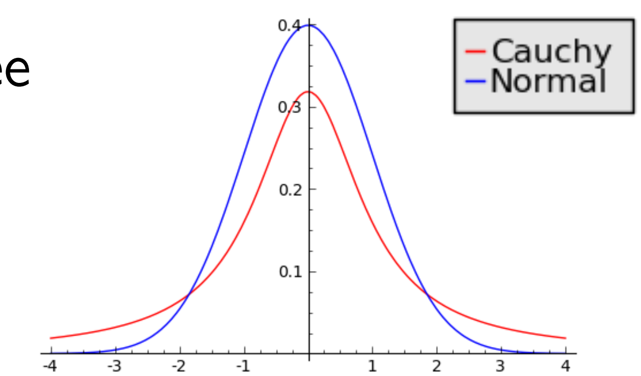


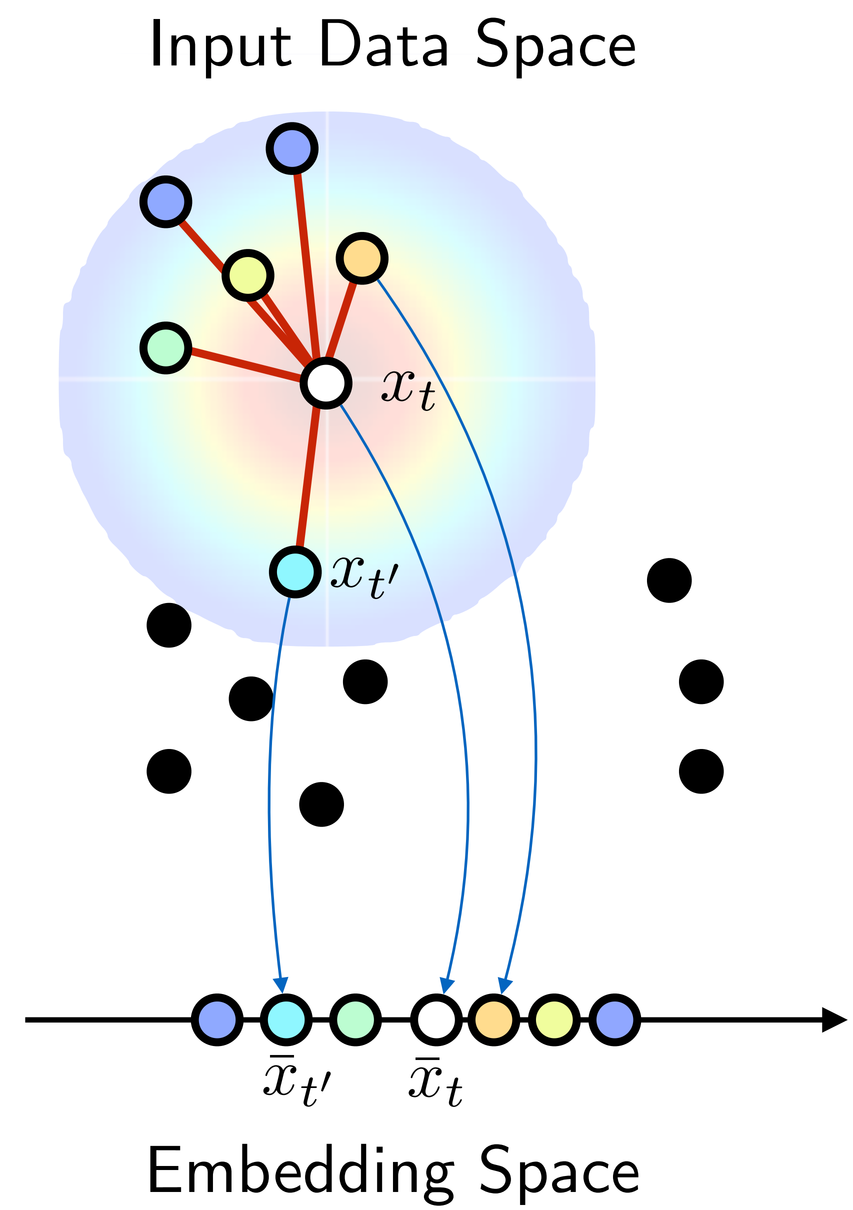
No linear embedding would be good

Stochastic Neighbor Embedding

- Denote $\bar{x}_t \in \mathbb{R}^d$ the embedding of $x_t \in \mathbb{R}^r$, where t is the point index
- Target distribution: $p^*(t'|t) \propto \mathcal{N}(x_t - x_{t'}; 0, \sigma_t^2)$
  - For each t, $p^*(t'|t)$ is a discrete distribution over data points (like RBF kernel)
  - Need only distances between data points (Euclidean here)
  - Variance σ_t^2 may be selected adaptively
- Model distribution: $p(t'|t) \propto \mathcal{N}(\bar{x}_t - \bar{x}_{t'}; 0, 1)$.
  - A better choice is Student t-distribution

(Student t with 1 degree of freedom is Cauchy)



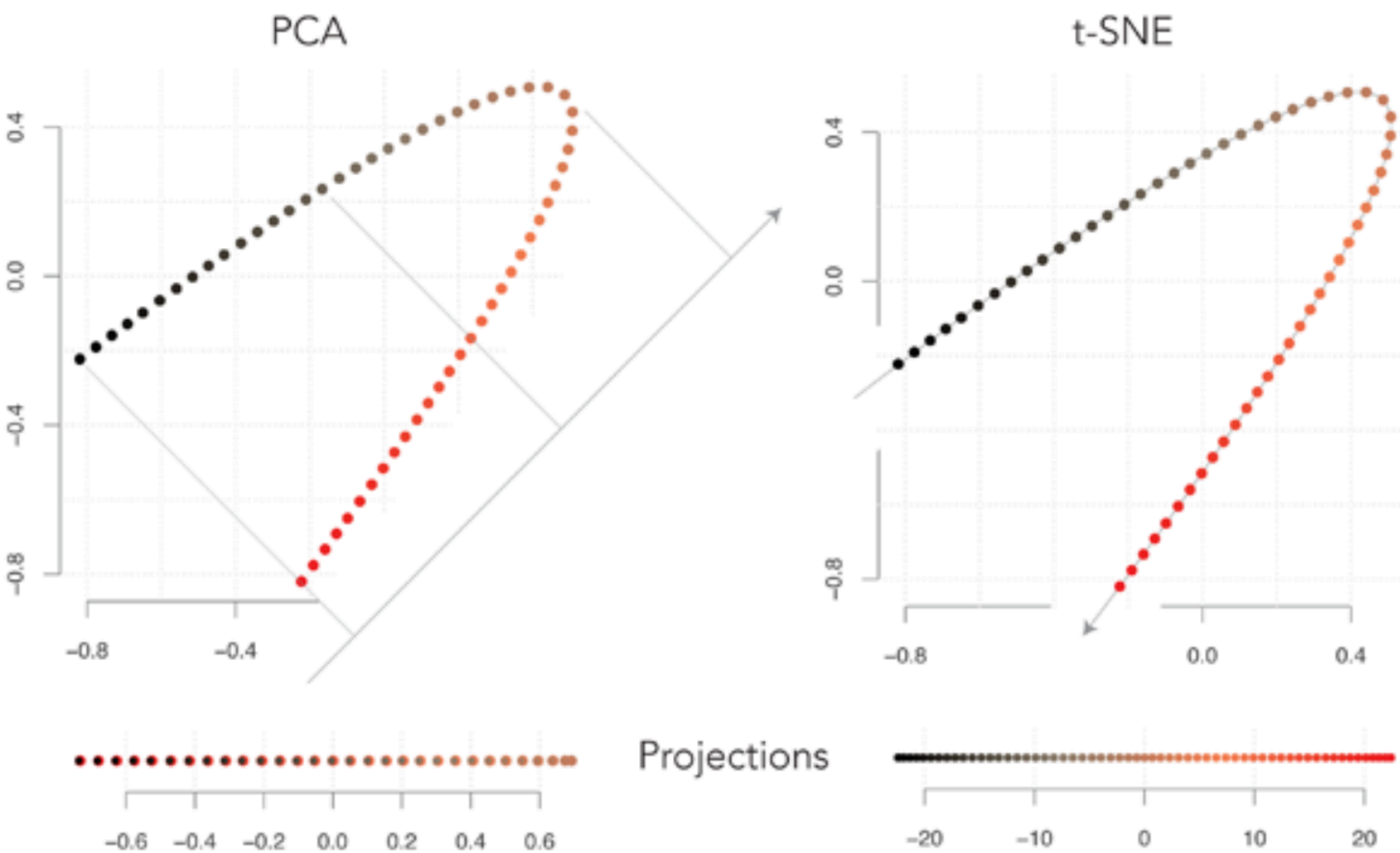


- Learning formulation:

$$\theta = (\bar{x}_t | t \in T) \qquad \max_\theta \sum_t \sum_{t'} p^*(t'|t) \log p(t'|t;\theta)$$

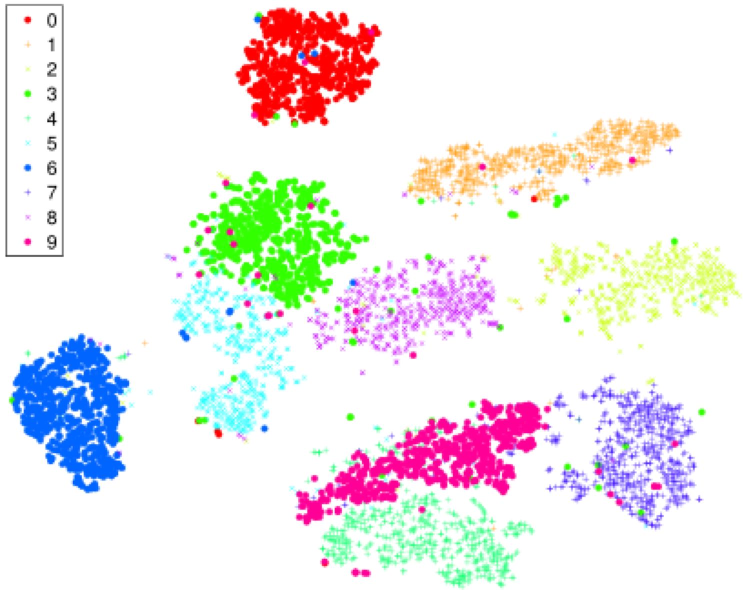
[Hinton & Roweis (2002): Stochastic Neighbor Embedding; Maaten & Hinton (2008): Visualizing Data using t-SNE]

Stochastic Neighbor Embedding



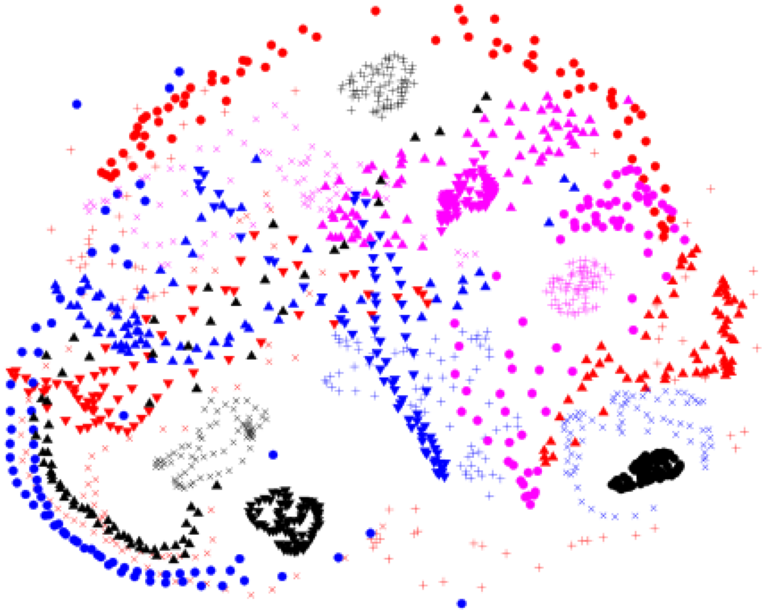
[Guillaume Filion (2018): A tutorial on t-SNE]

Stochastic Neighbor Embedding



t-SNE of MNIST data

t-SNE of COIL data



Sammon Mapping, COIL data

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

KL Divergence

KL Divergence

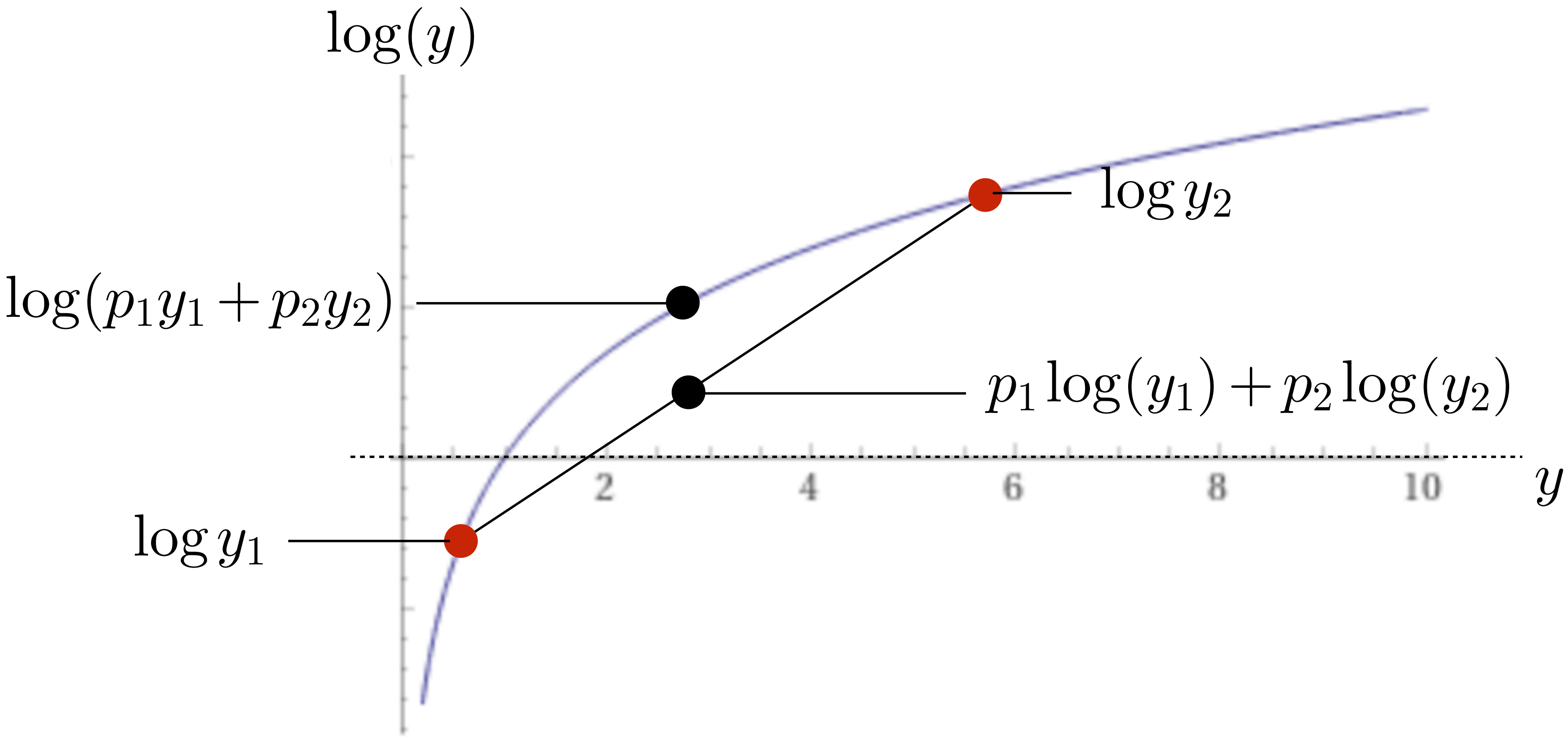
- Let $p(x)$ and $q(x)$ be two probability distributions.
- Kullback–Leibler divergence of p and q is

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

  - Amount of information lost when q is used to approximate p
  - Measured in *nats* ($\log$ is the natural logarithm)
  - Defined only if $q(x) = 0 \Rightarrow p(x) = 0$
  - $\lim_{x \to 0} x \log x = 0$
- 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)

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 convexity follows that $D_{\mathrm{KL}}(p\|q) = 0$ iff $p = q$



Asymmetry

Minimizing **forward KL** divergence:

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

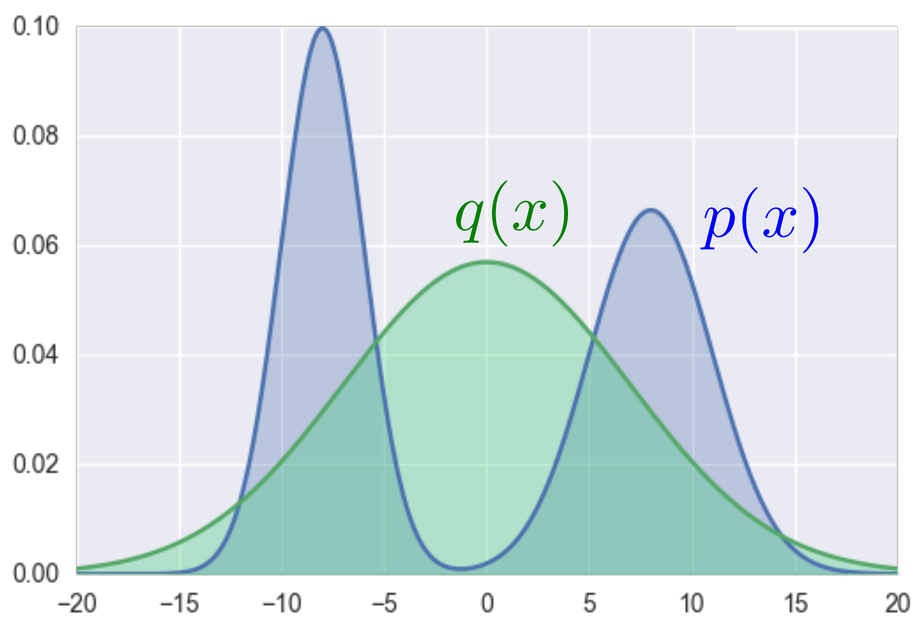
$$\min_q \int p(x)(\log p(x) - \log q(x))dx$$

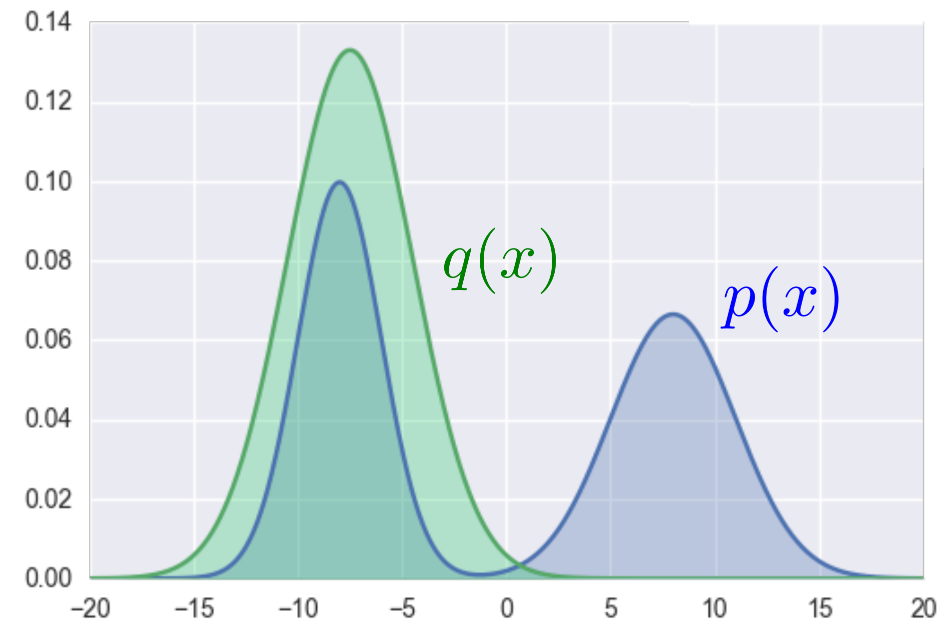
Minimizing **reverse KL** divergence:

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

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

Example: q is constrained to be a Gaussian





Well on average in the expectation over p

Matching moments

Well on average in the expectation over q

Selects a mode

Maximum Likelihood, Cross-Entropy and KL

- Common ML Learning for Classification:
  - (x_i, y_i) – training data. Assume it is given by the generator distribution $p^*(x, y)$
  - Model: $p(y|x;\theta)$
  - Conditional ML:

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

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

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

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

Latent Variable Models, Stochastic EM

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 (seminar).
  - 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 (next slide).

Multi-sense Word Vectors

✦ Often, words have multiple meanings (homographs):

I eat grape **jam**.

I was in a traffic **jam**.

Be careful not to **jam** your finger in the door.

- All words in the context commonly depend on the latent meaning of the current word:

$$\underbrace{\prod_{t' \in \mathcal{N}(t)} p(y_{t'} | z, x_t)}_{p(Y_t | z, x_t)} p(z | x_t), \quad z \in \{1 \ldots \text{max meanings}\} \qquad \text{(assume for simplicity)}$$

Y_t – context words

- Do not know 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):

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

- **Inference**:

Compute $p(z | x_t, Y_t)$ (then maximize in z, use the word vector $W_{x_t, z, :}$, *etc*.)

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

Learning

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

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

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

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):

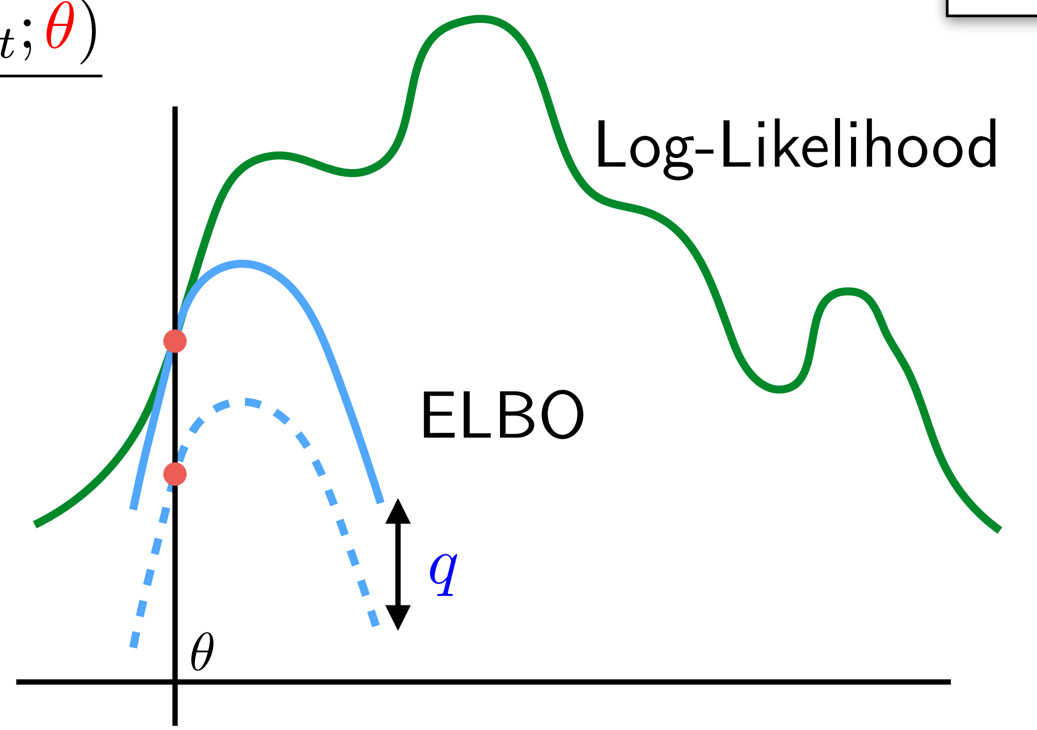
$$\underbrace{\log p(Y)}_{\text{Evidence}} - \underbrace{\sum_z q(z|Y) \log \frac{p(Y,z)}{q(z|y)}}_{\text{ELBO}} = \sum_z q(z|Y)\Big(\log p(Y) - \log \frac{p(Y,z)}{q(z|y)}\Big)$$

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

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

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).

- **M**-step:

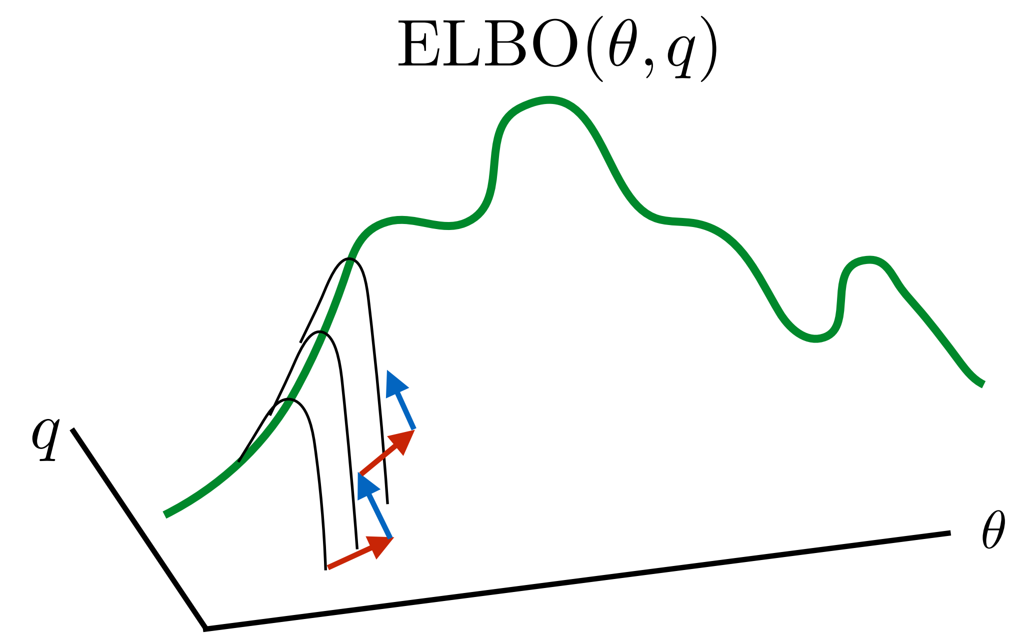
  $\text{argmax}_\theta \sum_t \sum_z q(z|x_t, Y_t) \log p(Y_t|z, x_t; \theta)$

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

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:

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

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

- **M**-step:

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

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"