Inference and Representation
Lecture 9

Joan Bruna
Courant Institute, NYU

Variational Autoencoders

[Kingma & Welling'14, Rezende et al.'14]

- Recall the variational lower bound:

$$\log p(X \mid \theta) = \mathbb{E}_{q(z|\beta)}\{\log(p(X, Z \mid \theta)\} + H(q(z \mid \beta)) + D_{KL}(q(z|\beta)||p(z|x, \theta)$$

$$\log p(X \mid \theta) = \mathcal{L}(\theta, \beta, X) + D_{KL}(q(z|\beta)||p(z|X, \theta))$$

Variational Autoencoders

[Kingma & Welling'14, Rezende et al.'14]

- Recall the variational lower bound:

$$\log p(X \mid \theta) = \mathbb{E}_{q(z|\beta)}\{\log(p(X, Z \mid \theta)\} + H(q(z \mid \beta)) + D_{KL}(q(z|\beta)||p(z|x, \theta)$$

$$\log p(X \mid \theta) = \mathcal{L}(\theta, \beta, X) + D_{KL}(q(z|\beta)||p(z|X, \theta))$$

- Can we optimize jointly both generative and variational parameters efficiently?

Variational Autoencoders

[Kingma & Welling'14, Rezende et al.'14]

- Recall the variational lower bound:

$$\log p(X \mid \theta) = \mathbb{E}_{q(z|\beta)}\{\log(p(X, Z \mid \theta)\} + H(q(z \mid \beta)) + D_{KL}(q(z|\beta)||p(z|x, \theta)$$

$$\log p(X \mid \theta) = \mathcal{L}(\theta, \beta, X) + D_{KL}(q(z|\beta)||p(z|X, \theta))$$

- Can we optimize jointly both generative and variational parameters efficiently?
- For appropriate posterior approximations, we can reparametrize samples as

$$Z \sim q(z|x, \beta) \Rightarrow Z \stackrel{d}{=} g_\beta(\epsilon, x) \ , \ \epsilon \sim p_0$$

$$\left(\text{e.g. } \ q(z|x, \beta) = \mathcal{N}(z; \mu(x), \Sigma(x)) \leftrightarrow z = \mu(x) + \Sigma(x)^{1/2}\epsilon \ , \ \epsilon \sim \mathcal{N}(0, \mathbf{1})\right)$$

Variational Autoencoders

- It results that

$$\mathcal{L}(\theta, \beta, X) = -D_{KL}(q_\beta(z|X)||p_\theta(z)) + \mathbb{E}_{q_\beta(z|X)}\{\log p(X|z, \theta)\}$$

can be estimated via Monte-Carlo by

$$\widehat{\mathcal{L}(\theta, \beta, X)} = -D_{KL}(q_\beta(z|X)||p_\theta(z)) + \frac{1}{S}\sum_{s \leq S} \log p(X|z^{(s)}, \theta)$$

$$z^{(s)} = g_\beta(X, \epsilon^{(s)}) \text{ and } \epsilon^{(s)} \sim p_0 \ .$$

Variational Autoencoders

- It results that

$$\mathcal{L}(\theta, \beta, X) = -D_{KL}(q_\beta(z|X)||p_\theta(z)) + \mathbb{E}_{q_\beta(z|X)}\{\log p(X|z, \theta)\}$$

can be estimated via Monte-Carlo by

$$\widehat{\mathcal{L}(\theta, \beta, X)} = -D_{KL}(q_\beta(z|X)||p_\theta(z)) + \frac{1}{S}\sum_{s \leq S} \log p(X|z^{(s)}, \theta)$$

$$z^{(s)} = g_\beta(X, \epsilon^{(s)}) \text{ and } \epsilon^{(s)} \sim p_0 \ .$$

- First term acts as a regularizer: limits the capacity of the encoder
- Second term is a reconstruction error.

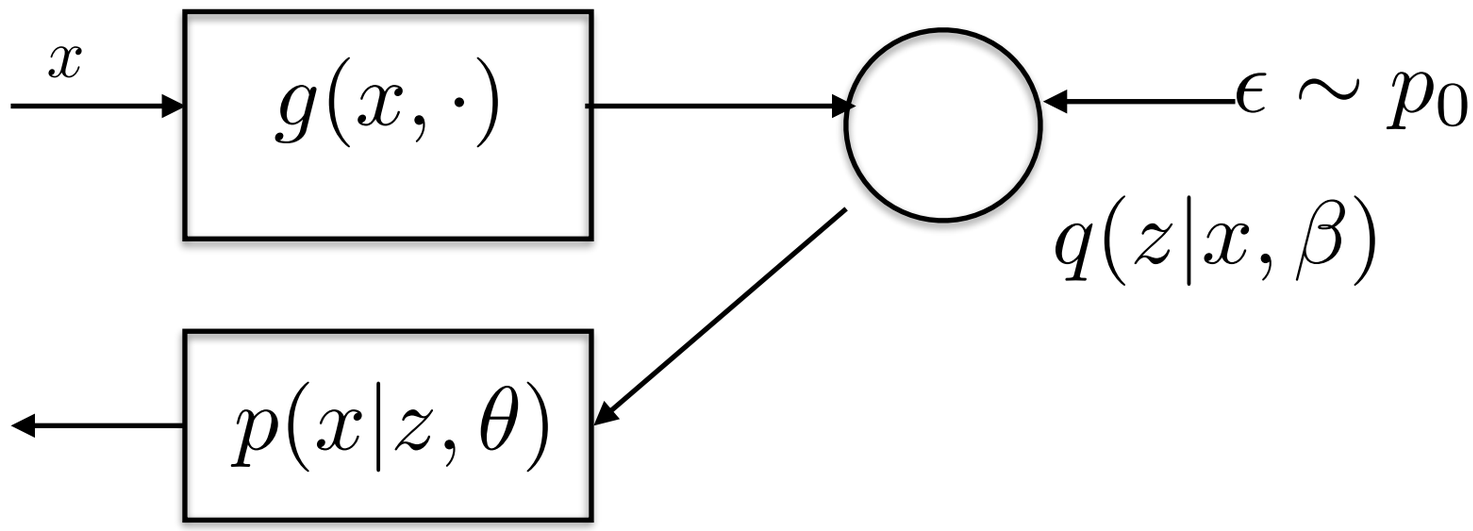
Variational Autoencoders

- How to model $x \mapsto g_\beta(x, \cdot)$ and $z \mapsto p_\theta(\cdot, z)$?

Variational Autoencoders

- How to model $x \mapsto g_\beta(x, \cdot)$ and $z \mapsto p_\theta(\cdot, z)$?

- VAE idea: use neural networks to approximate variational and generative parameters.



Variational Autoencoder

- Example: Let the prior over latent variables be Gaussian isotropic:

$$p(z) = \mathcal{N}(z; 0, \mathbf{I})$$

Variational Autoencoder

- Example: Let the prior over latent variables be Gaussian isotropic:

$$p(z) = \mathcal{N}(z; 0, \mathbf{I})$$

- Let the conditional likelihood be also Gaussian:

$$p(x|z) = (x; \mu(z), \Sigma(z)) \qquad \mu(z), \Sigma(z) \ : \ \text{Neural networks}$$

Variational Autoencoder

- Example: Let the prior over latent variables be Gaussian isotropic:

$$p(z) = \mathcal{N}(z; 0, \mathbf{I})$$

- Let the conditional likelihood be also Gaussian:

$$p(x|z) = (x; \mu(z), \Sigma(z)) \qquad \mu(z), \Sigma(z) \ : \ \text{Neural networks}$$

- Variational approximate posterior also Gaussian:

$$q_\beta(z|x) = \mathcal{N}(z; \overline{\mu}(x), \overline{\Sigma}(x))$$

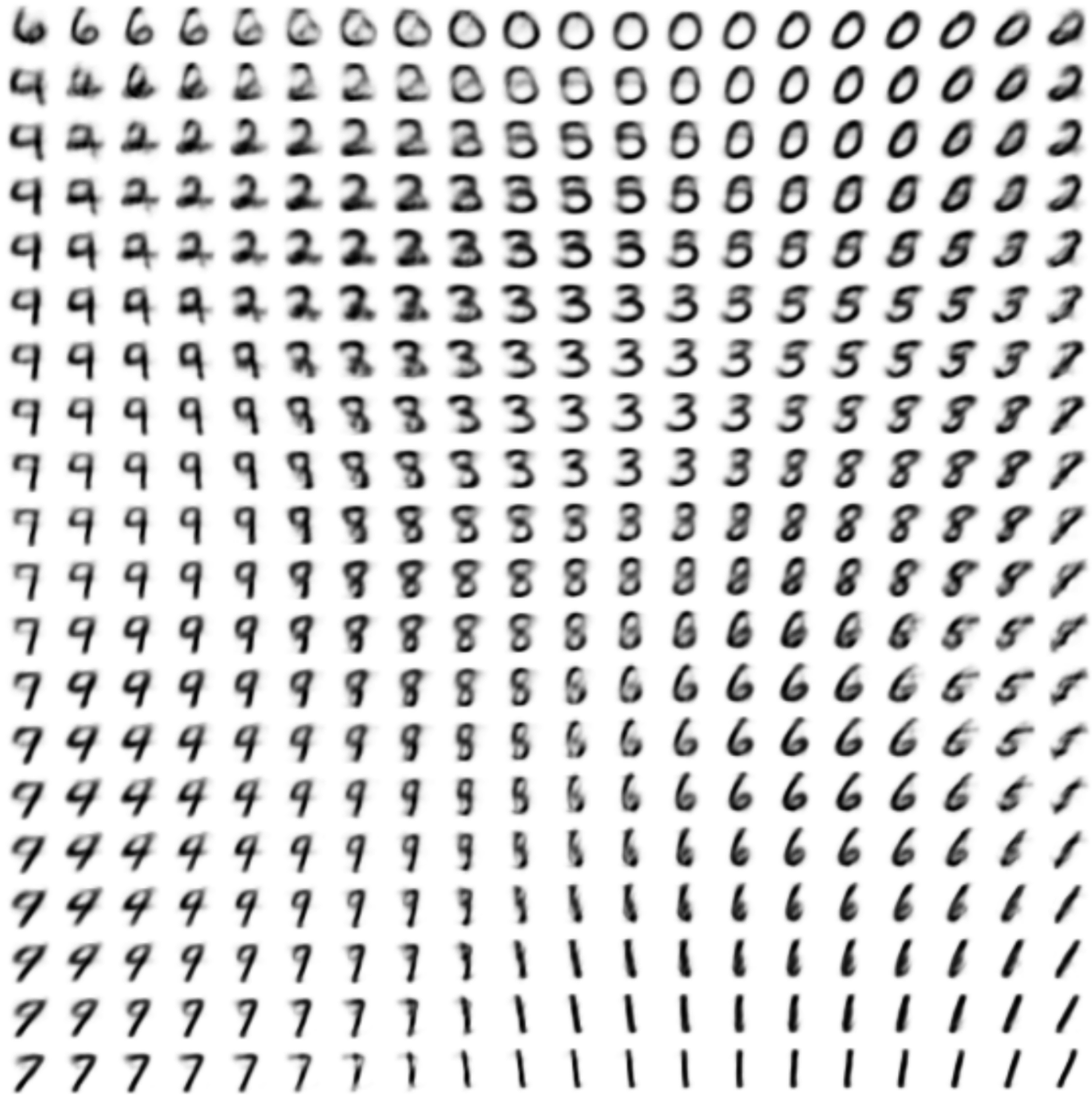
$$\overline{\mu}(z), \overline{\Sigma}(z) \ : \ \text{Neural networks}, (\overline{\Sigma} \text{ diagonal})$$

$$Z \sim q_\beta(z|x) \Leftrightarrow Z = \overline{\mu}(x) + \overline{\Sigma}(x)^{1/2}\epsilon \ , \ \epsilon \sim \mathcal{N}(0, 1)$$

Variational Autoencoder

- Examples using a two-dimensional latent space:

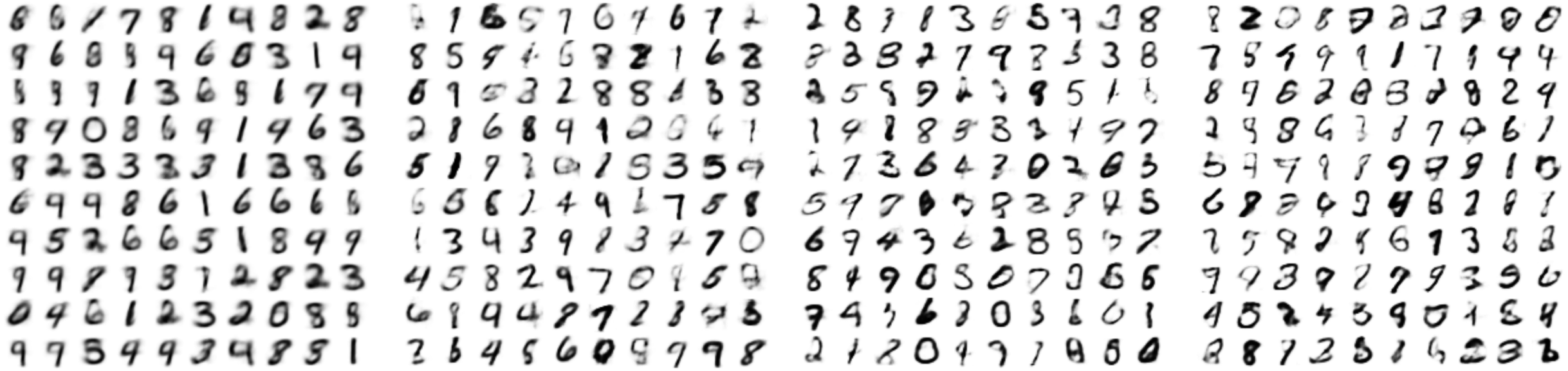
(a) Learned Frey Face manifold



(b) Learned MNIST manifold

Examples

- Increasing latent dimensionality:



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Extensions to semi-supervised learning

- Semi-supervised learning:

We observe $\{x_i\}_{i \leq L_1}$ and $\{x_j, y_j\}_{j \leq L_2}$, with $x_i \sim p(x)$, $x_j \sim p(x)$.

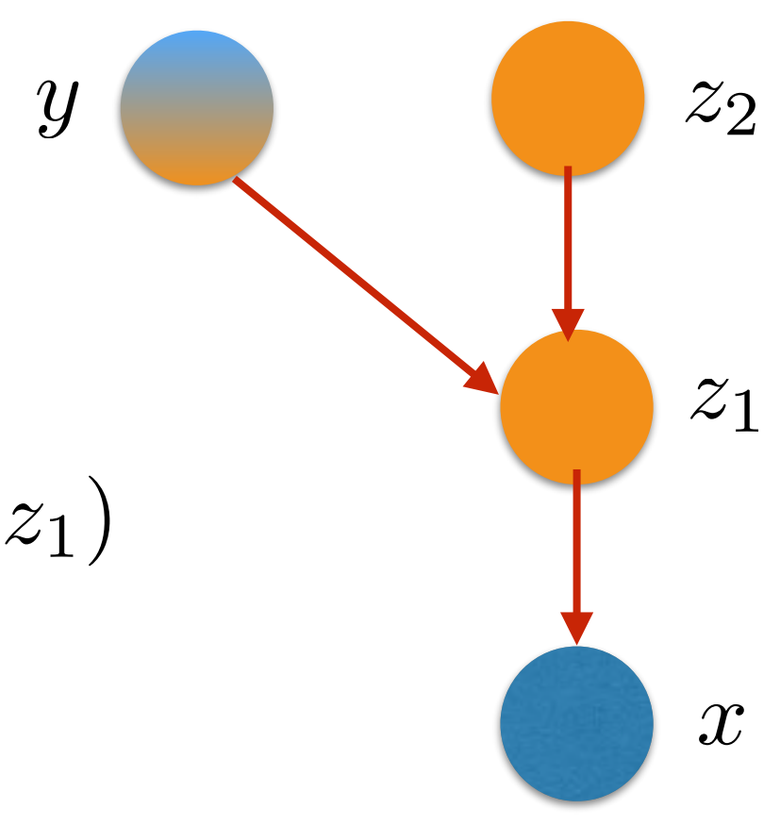
$L_1 \gg L_2$

Extensions to semi-supervised learning

- "Semi-supervised Learning with Deep Generative Networks", Kingma et al,'14.

- Labels are treated as either observed or hidden.

$$p(x, y, z_1, z_2) = p(y)p(z_2)p(z_1|z_2, y)p(x|z_1)$$



Extension to Semi-Supervised Learning

- "Semi-supervised Learning with Deep Generative Networks", Kingma et al,'14.

- For datapoint with labels:

$$\log p_\theta(x, y) \geq \mathbb{E}_{q_\beta(z|x,y)} \left(\log p_\theta(x|y, z) + \log p_\theta(y) + \log p(z) - \log q_\beta(z|x, y) \right)$$

- For datapoint with no labels:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\beta(y,z|x)} \left(\log p_\theta(x|y, z) + \log p_\theta(y) + \log p(z) - \log q_\beta(z, y|x) \right)$$

Extension to Semi-Supervised Learning

- *"Semi-supervised Learning with Deep Generative Networks", Kingma et al,'14.*
- Classification results on MNIST:

Table 1: Benchmark results of semi-supervised classification on MNIST with few labels.

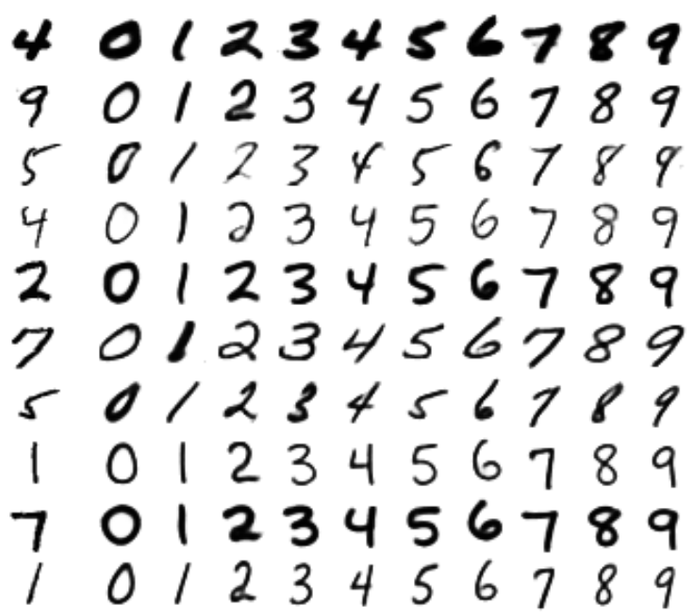
N	NN	CNN	TSVM	CAE	MTC	AtlasRBF	M1+TSVM	M2	M1+M2
100	25.81	22.98	16.81	13.47	12.03	8.10 ($\pm$ 0.95)	11.82 ($\pm$ 0.25)	11.97 ($\pm$ 1.71)	**3.33** ($\pm$ 0.14)
600	11.44	7.68	6.16	6.3	5.13	–	5.72 ($\pm$ 0.049)	4.94 ($\pm$ 0.13)	**2.59** ($\pm$ 0.05)
1000	10.7	6.45	5.38	4.77	3.64	3.68 ($\pm$ 0.12)	4.24 ($\pm$ 0.07)	3.60 ($\pm$ 0.56)	**2.40** ($\pm$ 0.02)
3000	6.04	3.35	3.45	3.22	2.57	–	3.49 ($\pm$ 0.04)	3.92 ($\pm$ 0.63)	**2.18** ($\pm$ 0.04)

- Now there are stronger models on that task.
  - Ladder-Networks [Rasmus et al'15]
  - GANs [Salimans et al'16]
  - Graph Neural Networks [Kipf & Welling, '17].

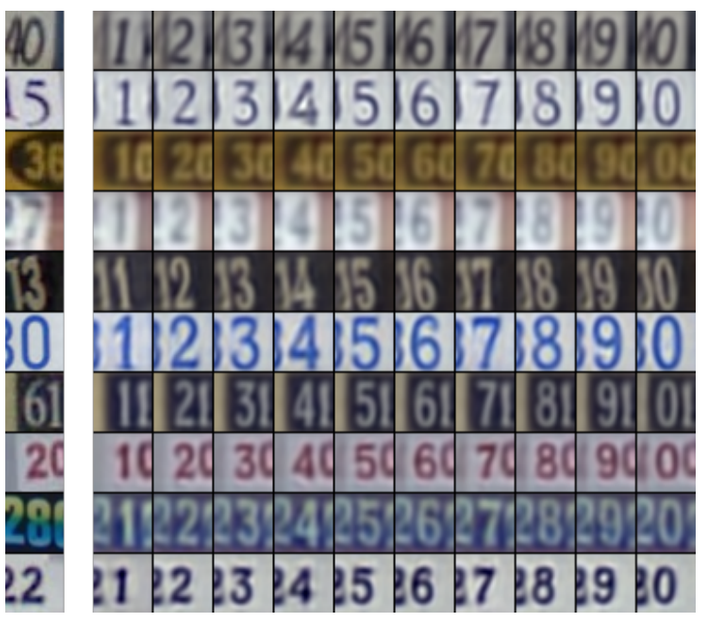
Extension to Semi-Supervised Learning

- *"Semi-supervised Learning with Deep Generative Networks", Kingma et al,'14.*
- Disentangling label and "style":

(a) Handwriting styles for MNIST obtained by fixing the class label and varying the 2D latent variable **z**



(b) MNIST analogies



(c) SVHN analogies

Incorporate MCMC to posterior approx.

"Markov Chain Monte Carlo and Variational Inference: Bridging the Gap", Salimans et al'15

- We saw in Lecture 7 how to use Markov Chains to approximate intractable posteriors.

$$p(z \mid x) \stackrel{d}{=} \lim_{T \to \infty} q_0(z_0 \mid x) \prod_{t \leq T} q(z_t \mid z_{t-1}, x) \ .$$

Incorporate MCMC to posterior approx.

"Markov Chain Monte Carlo and Variational Inference: Bridging the Gap", Salimans et al'15

- We saw in Lecture 7 how to use Markov Chains to approximate intractable posteriors.

$$p(z \mid x) \stackrel{d}{=} \lim_{T \to \infty} q_0(z_0 \mid x) \prod_{t<T} q(z_t \mid z_{t-1}, x) \ .$$

- For fixed T, this can be seen as another variational approximation, by considering $y = z_1, \ldots, z_{T-1}$ as extra hidden variables.

Incorporate MCMC to posterior approx.

"Markov Chain Monte Carlo and Variational Inference: Bridging the Gap", Salimans et al'15

- We saw in Lecture 7 how to use Markov Chains to approximate intractable posteriors.

$$p(z \mid x) \stackrel{d}{=} \lim_{T \to \infty} q_0(z_0 \mid x) \prod_{t \leq T} q(z_t \mid z_{t-1}, x) \ .$$

- For fixed T, this can be seen as another variational approximation, by considering $y = z_1, \ldots, z_{T-1}$ as extra hidden variables.
- The resulting Variational Lower bound becomes

$$\begin{aligned} \mathcal{L}_{MCMC} &= \mathcal{L} - \mathbb{E}_{q(z_T|x)}\{D_{KL}(r(y|z_T, x)||q(y \mid z_T, x)\} \\ &\leq \mathcal{L} \leq \log p(x) \ . \end{aligned}$$

$r(y|x, z_T)$: auxiliary variational approximation

Incorporate MCMC to posterior approx.

"Markov Chain Monte Carlo and Variational Inference: Bridging the Gap", Salimans et al'15

- We saw in Lecture 7 how to use Markov Chains to approximate intractable posteriors.

$$p(z \mid x) \stackrel{d}{=} \lim_{T \to \infty} q_0(z_0 \mid x) \prod_{t<T} q(z_t \mid z_{t-1}, x) \ .$$

- For fixed T, this can be seen as another variational approximation, by considering $y = z_1, \ldots, z_{T-1}$ as extra hidden variables.
- The resulting Variational Lower bound becomes

$$\begin{aligned} \mathcal{L}_{aux} =& \mathcal{L} - \mathbb{E}_{q(z_T|x)}\{D_{KL}[q(y|z_T, x)||r(y|z_T, x)]\} \\ \leq& \mathcal{L} \leq \log p(x) \ . \end{aligned}$$

$r(y|x, z_T)$: auxiliary variational approximation

- If we choose *r* to be an inverse Markov chain, we obtain

$$\mathcal{L}_{aux} = \mathbb{E}_q \left\{\log p(x, z_T) - \log q(z_0|x)\right\} + \sum_{t=1}^{T} (\log r_t(z_{t-1}|x, z_t) - \log q_t(z_t|x, z_{t-1}))$$

Incorporate MCMC to posterior approx.

"Markov Chain Monte Carlo and Variational Inference: Bridging the Gap", Salimans et al'15

$$\mathcal{L}_{aux} = \mathbb{E}_q \left\{ \log p(x, z_T) - \log q(z_0|x) \right\} + \sum_{t=1}^{T} \left(\log r_t(z_{t-1}|x, z_t) - \log q_t(z_t|x, z_{t-1}) \right)$$

- The authors consider Hamilton Monte-Carlo as MCMC choice, resulting in Hamiltonian Variational Inference.
- It provides a flexible (albeit more computationally demanding) variational approximation that can be adjusted with the number *T* of MCMC steps.

Variational inference with Importance Sampling

"Importance Weighted Autoencoders"
Burda et al'16

- Another mechanism to improve the variational lower bound is to use importance sampling.

Variational inference with Importance Sampling

"Importance Weighted Autoencoders"
Burda et al'16

- Another mechanism to improve the variational lower bound is to use importance sampling.
- For each k, we define

$$\mathcal{L}_k(x) = \mathbb{E}_{z_1,\ldots,z_k \sim q(z|x)} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p(x, z_i)}{q(z_i|x)} \right] .$$

Variational inference with Importance Sampling

"Importance Weighted Autoencoders"
Burda et al'16

- Another mechanism to improve the variational lower bound is to use importance sampling.
- For each k, we define

$$\mathcal{L}_k(x) = \mathbb{E}_{z_1,\ldots,z_k \sim q(z|x)} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p(x, z_i)}{q(z_i|x)} \right] .$$

- It follows that

$$\forall\ k\ ,\ \ \log p(x) \geq \mathcal{L}_{k+1}(x) \geq \mathcal{L}_k(x)\ ,\quad \text{and}$$

$$\lim_{k\to\infty} \mathcal{L}_k(x) = \log p(x) \text{ if } \frac{p(x,z)}{q(z|x)} \text{ is bounded .}$$

Structured Output Prediction

- Conditional Random Fields (CRF) [Lec 9]

- Parameter Learning in the Exponential Family [Lec 9]
  - ML / Maximum Entropy
  - Monte-Carlo
- Pseudo-likelihood [Lec 9]

- MAP Inference [Lec 10]
  - Max-Product

- Sequence-to-Sequence Models [Lec 10]

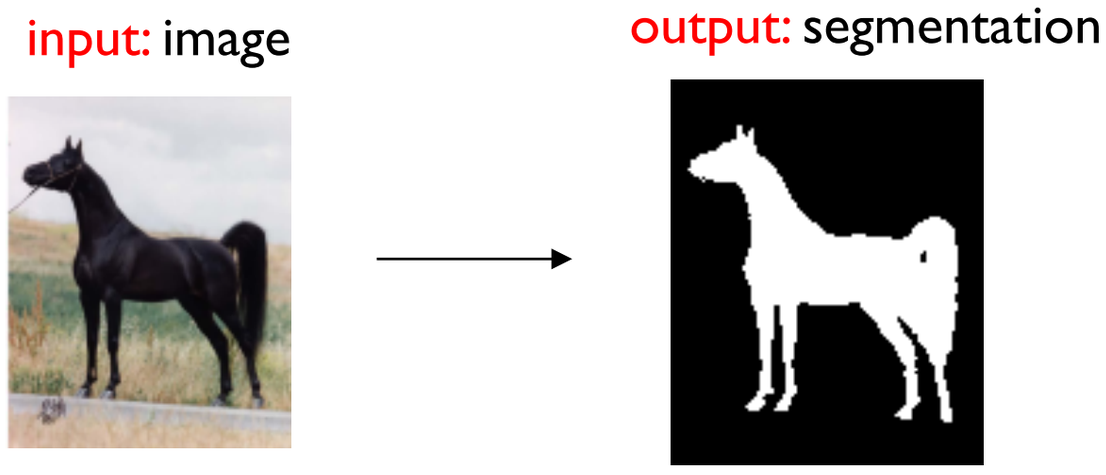
- Unrolling BP with Graph Neural Networks [Lec 10].

Conditional Random Fields (CRF)

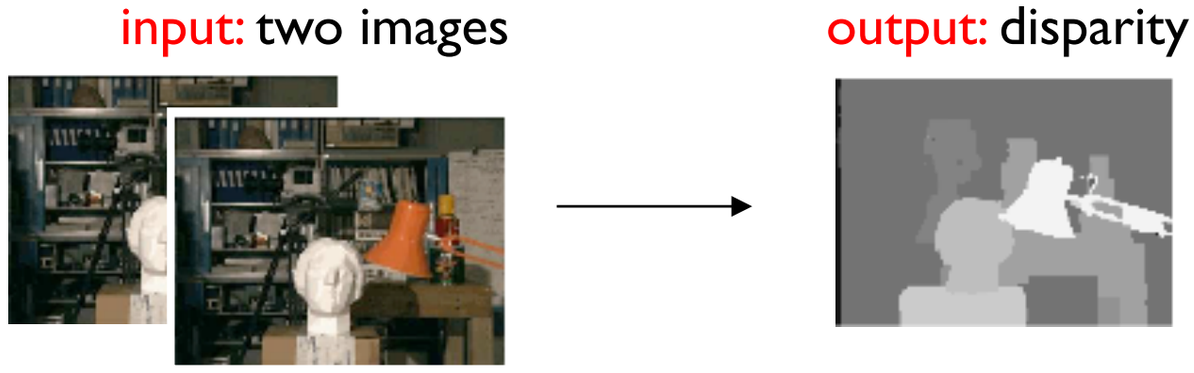
- Goal: Model *structured* outputs Y from observations X

Computer vision

Image segmentation

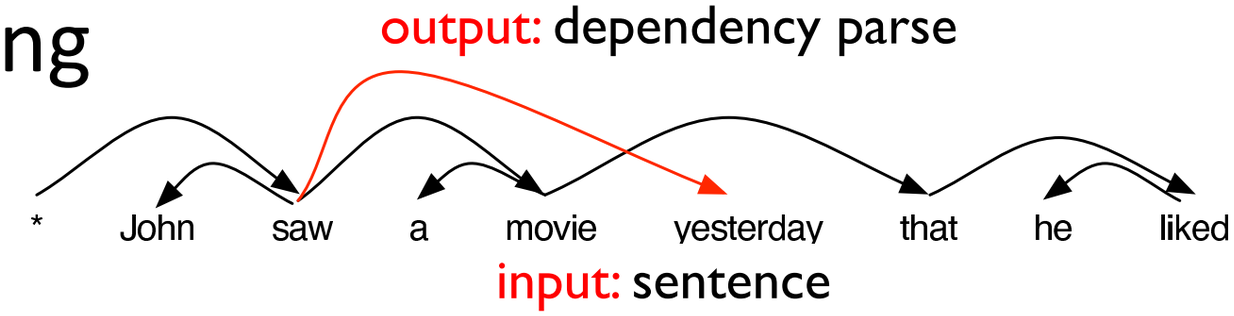


Stereopsis



Natural language processing

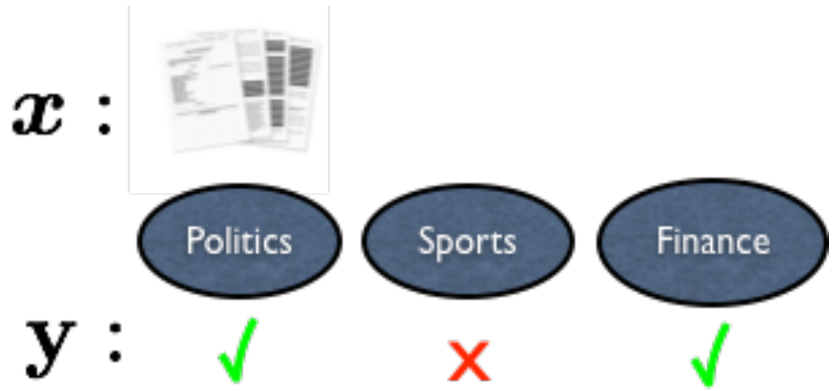
Parsing



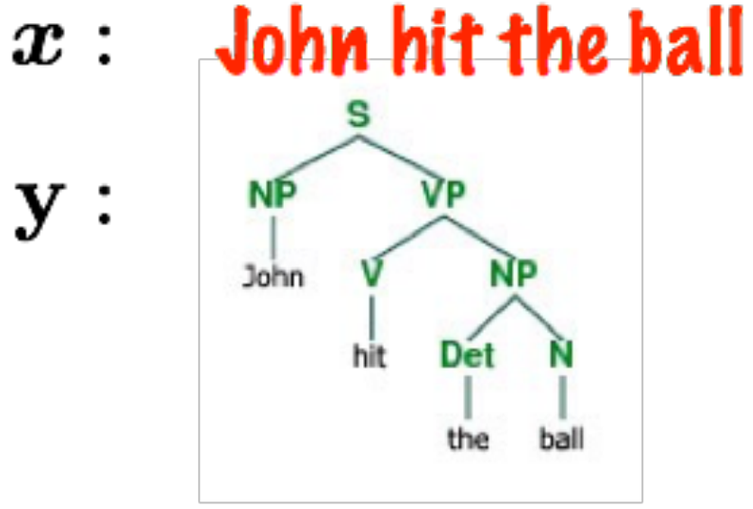
Conditional Random Fields (CRF)

- Goal: Model *structured* outputs Y from observations X

- **Examples:**
  - **Multi-label prediction:**



  - Natural language parsing:



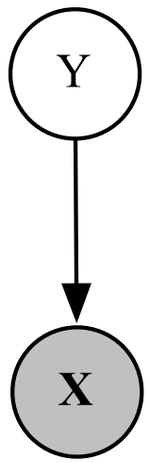
  - Protein side-chain placement:

Discriminative vs Generative Models

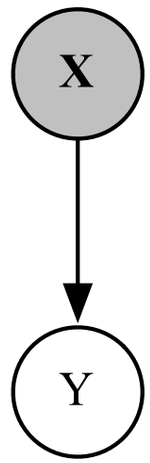
- Two equivalent models for $p(X, Y)$:

Generative

$$p(X, Y) = p(Y)p(X \mid Y)$$



Discriminative



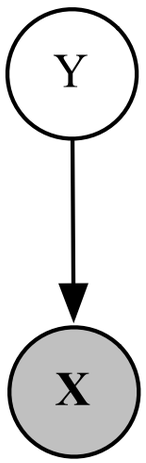
$$p(X, Y) = p(X)p(Y \mid X)$$

Discriminative vs Generative Models

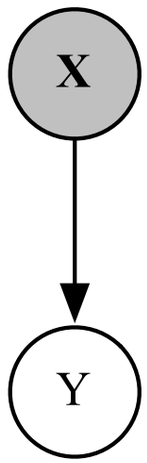
- Two equivalent models for $p(X, Y)$:

Generative

$$p(X, Y) = p(Y)p(X \mid Y)$$



Discriminative



$$p(X, Y) = p(X)p(Y \mid X)$$

- However, in some setups, we only care about prediction $p(Y \mid X)$
- In the generative setup, we need to estimate both $p(Y)$ and $p(X|Y)$
  - If $\mathbf{dim}(X) \gg \mathbf{dim}(Y)$, this is exponentially harder than directly estimating $p(Y|X)$!!

Conditional Random Fields (CRF)

- CRFs are undirected graphical models for the conditional distribution $p(Y|X)$ (Y = target variables, X = observed variables).
- A particular case of what we have seen so far:

$$p(Y|X=x) = \frac{1}{Z(x)} \prod_{c \in \mathcal{C}} \phi_c(x_c, y_c) \ ,$$

$$Z(x) = \sum_{y} \prod_{c \in \mathcal{C}} \phi_c(x_c, y_c) \ .$$

Conditional Random Fields (CRF)

- CRFs are undirected graphical models for the conditional distribution $p(Y|X)$ (Y = target variables, X = observed variables).
- A particular case of what we have seen so far:

$$p(Y|X=x) = \frac{1}{Z(x)} \prod_{c \in \mathcal{C}} \phi_c(x_c, y_c) \ ,$$

$$Z(x) = \sum_{y} \prod_{c \in \mathcal{C}} \phi_c(x_c, y_c) \ .$$

- The only difference with unconditional MRF is the normalization:
  - In a CRF, we marginalize only over Y.
  - In a MRF, we marginalize over both X, Y.

CRFs

- We will consider the exponential family parametrization:

$$\phi_c(x_c, y_c) = \exp(\langle w_c, f(x_c, y_c)\rangle)$$

  - adjustable weights: w_c
  - feature vector: $f(x_c, y_c)$
- Special, important case: $w_c = w \;\forall\; c \in \mathcal{C}.$ (weight sharing).

CRFs

- We will consider the exponential family parametrization:

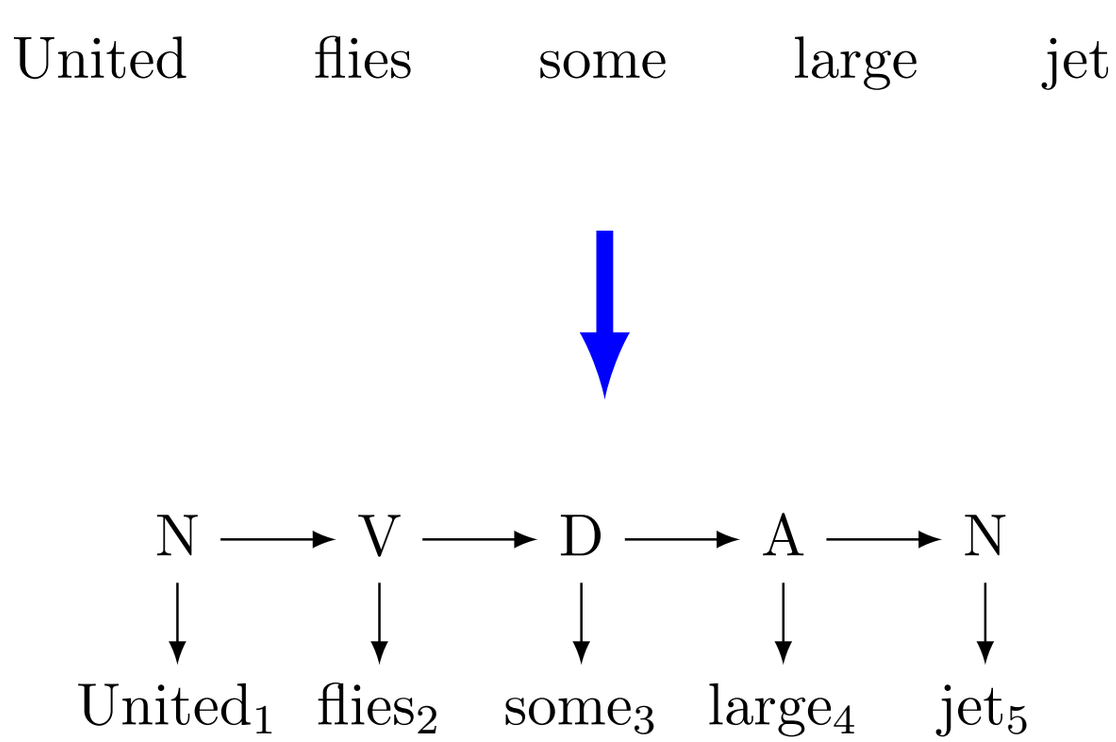
$$\phi_c(x_c, y_c) = \exp(\langle w_c, f(x_c, y_c)\rangle)$$

  - adjustable weights: w_c
  - feature vector: $f(x_c, y_c)$
- Special, important case: $w_c = w \;\forall\; c \in \mathcal{C}.$ (weight sharing).

- We can always reduce to the weight sharing case by *lifting* the sufficient statistics (how?)

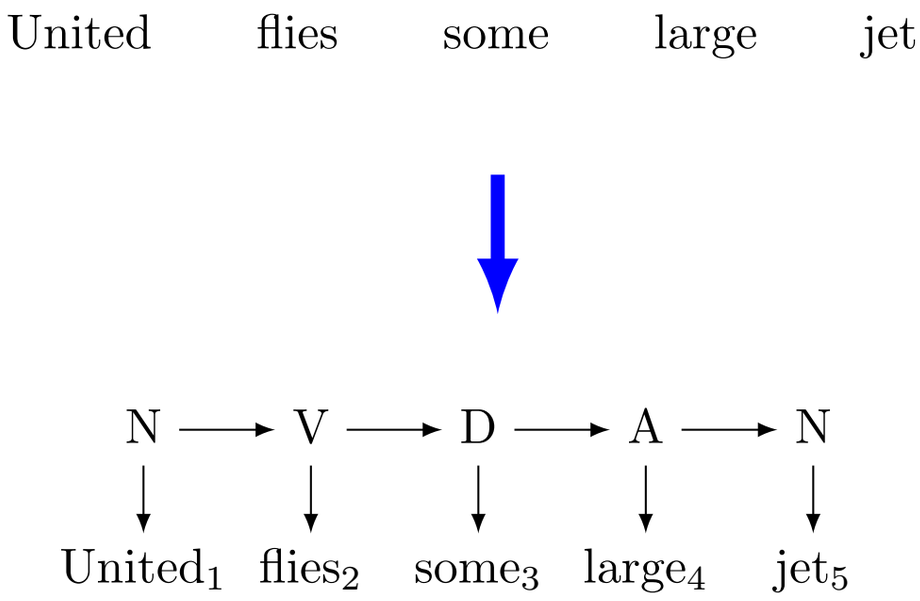
CRF Examples

- Part-of-Speech Tagging:



CRF Examples

- Part-of-Speech Tagging:



- Model:
  - A sentence of length n and a tag set $\mathcal{T}$.
  - One variable taking values in $\mathcal{T}$ for each word.
  - Edge potentials $\theta(i-1, i, t', t)$, $i \in 2, n$, $t, t' \in \mathcal{T}$.
  - Ex: $\mathcal{T} = \{A, D, N, V\}$.

CRF Examples

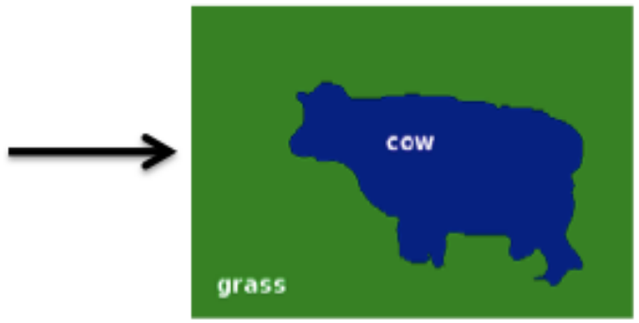
- Parametrization using exponential family:
  - Weights: $w \in \mathbb{R}^d$
  - Feature vectors $f_c(x, y_c) \in \mathbb{R}^d$.
    - presence or absence of certain attributes of each word (e.g. capitalization, "ing").
    - conditional on the whole input sequence x
- Unary potentials: $\phi_c(x, y_c; w) = \exp(\langle w, f_c(x, y_c)\rangle)$

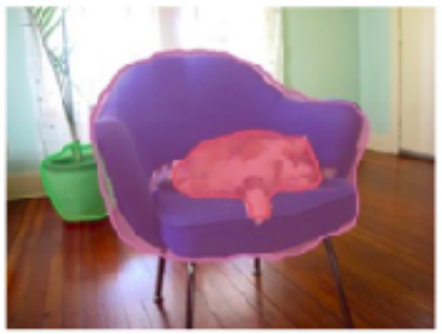
- Edge potentials: Fully parametrized $\mathcal{T} \times \mathcal{T}$ and stationary, i.e.

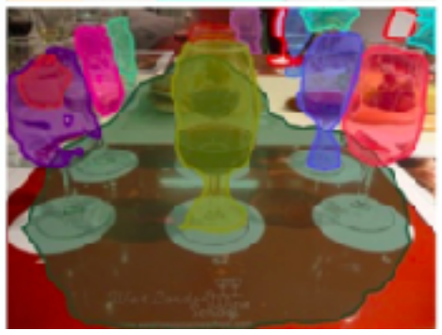
$$\theta(i-1, i, t', t) = \bar{w}_{t',t}$$

- Weights are *shared* for all nodes/edges.

CRF Examples: Semantic Segmentation



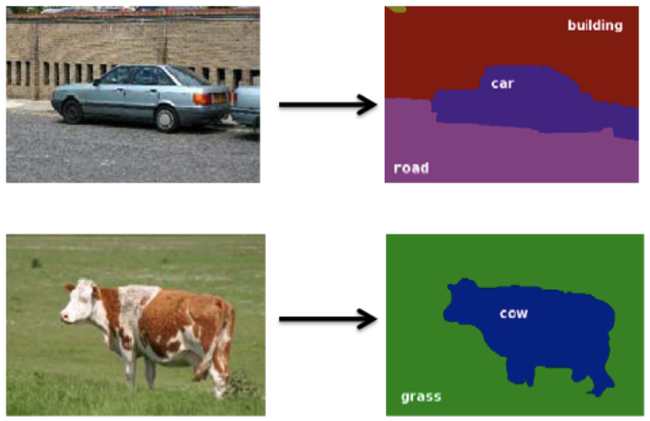


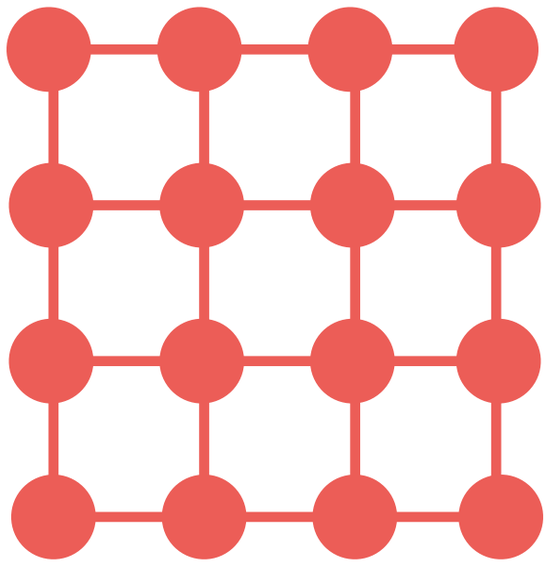


- Given an image $I(u)$, $u \in \Omega \subset \mathbb{R}^2$, produce a label map

$$Y(u) \in \{1, K\}, u \in \Omega$$

CRF Examples: Semantic Segmentation

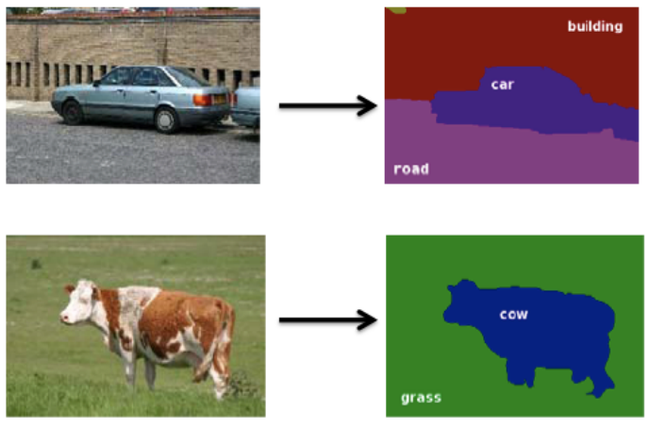


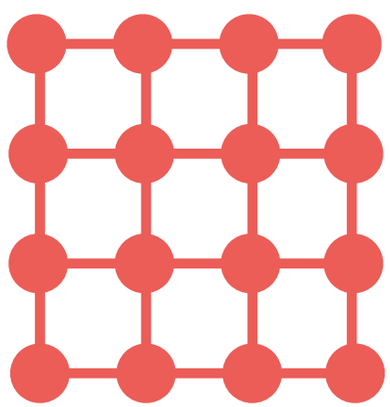


- Define a CRF on the graph defined by the square grid.
  - Neighboring pixels with similar colors are likely to have the same label.
  - E.g.

$$\phi_{i,j} = \exp\{(-\mathbf{1}(y_i = y_j) + \mathbf{1}(y_i \neq y_j))|x_i - x_j|\}$$

CRF Examples: Semantic Segmentation





- Define a CRF on the graph defined by the square grid.
  - Neighboring pixels with similar colors are likely to have the same label.
  - E.g. $\phi_{i,j} = \exp\{(-\mathbf{1}(y_i = y_j) + \mathbf{1}(y_i \neq y_j))|x_i - x_j|\}$
- Single node potentials over labels, e.g. a CNN which sees a local patch around the target pixel.
- MAP estimation over Y:

$$Y^* = \arg\max_Y \log P(Y \mid X) \ .$$

CRF Examples: Named-Entity Recognition

- Given a sentence, determine the people/entities involved and the relevant locations:

```
"Mrs Green spoke today in New York. Green chairs the
                    finance committee"
```

- Entities sometimes span multiple words. Require the context to disambiguate.

CRF Examples: Named-Entity Recognition

- Given a sentence, determine the people/entities involved and the relevant locations:

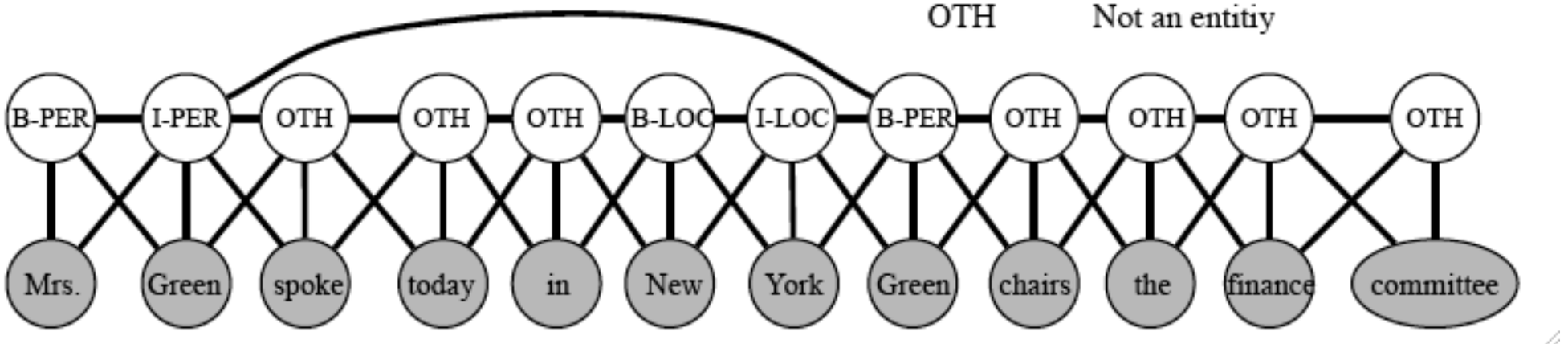
"Mrs Green spoke today in New York. Green chairs the finance committee"

- Entities sometimes span multiple words. Require the context to disambiguate.

- CRF has a variable Y_i for each word, encoding the possible labels of that word.
  - E.g. "B-person, I-person, B-location, I-location, B-organization, I-organization"
    - Beginning (B) and within (I) allows the model to segment adjacent entities.

CRF Examples: named-entity recognition

- "Skip-Chain" CRF:



- Three types of potentials:
  - $\phi^A(Y_i, Y_{i+1})$ models dependencies between neighboring target variables (such as in a HMM).
  - $\phi^B(Y_i, Y_j)$ for all pairs (i, j) such that $x_i = x_j$, since it is likely the same word represents the same entity.
  - $\phi^C(Y_i, X_1, \ldots, X_T)$ for dependencies between entity and the word input sequence.

Reminder: Exponential Families

- The exponential family associated with ϕ is defined as the parametric family

$$p_\theta(x) = \exp\{\langle \theta, \phi(x) \rangle - A(\theta)\} \ , \ \text{with}$$

$$A(\theta) = \log \int \exp\{\langle \theta, \phi(x) \} dx \qquad \text{log-partition function}$$

- It is well defined for the family of parameters

$$\Omega = \{\theta \ ; \ A(\theta) < \infty\}$$

- Several well-known models belong to the exponential family:
  - Energy based models
  - Gaussian Mixtures
  - Latent Dirichlet Allocation

Reminder: Exponential Families

- **Proposition:** The log-partition function $A(\theta)$ satisfies

  - $$\frac{\partial A}{\partial \theta_k}(\theta) = \mathbb{E}_\theta\{\phi_k(X)\} = \int \phi_k(x) p_\theta(x) dx \ .$$

  - $A(\theta)$ is convex in its domain Ω.

$$\frac{\partial^2 A}{\partial \theta_k \theta_l}(\theta) = \mathbb{E}_\theta\{[\phi_k(X) - \mathbb{E}(\phi_k(X))][\phi_l(X) - \mathbb{E}(\phi_l(X))]\} \ .$$

- Higher order derivatives always exist.

Learning under the Exponential Family

- Let us start with Maximum-Likelihood.
- On a MRF,

$$p(x;\theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{c \in \mathcal{C}} \log \phi_c(x_c;\theta)\right)$$

Learning under the Exponential Family

- Let us start with Maximum-Likelihood.
- On a MRF,

$$p(x;\theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{c\in\mathcal{C}} \log \phi_c(x_c;\theta)\right)$$

- Thus, given data $x^1, \ldots, x^L$, maximum likelihood estimate is

$$\max_\theta \frac{1}{L} \sum_{l=1}^{L} \left(\sum_{c\in\mathcal{C}} \log \phi_c(x_c^l;\theta) - \log Z(\theta)\right)$$

$$= \max_\theta \left(\sum_{c\in\mathcal{C}} L^{-1} \sum_{l\leq L} \log \phi_c(x_c^l;\theta) - A(\theta)\right) \quad .$$

Learning under the Exponential Family

- Let us start with Maximum-Likelihood.
- On a MRF,

$$p(x;\theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{c\in\mathcal{C}} \log \phi_c(x_c;\theta)\right)$$

- Thus, given data $x^1, \ldots, x^L$, maximum likelihood estimate is

$$\max_\theta \frac{1}{L}\sum_{l=1}^{L}\left(\sum_{c\in\mathcal{C}} \log\phi_c(x_c^l;\theta) - \log Z(\theta)\right)$$

$$= \max_\theta \left(\sum_{c\in\mathcal{C}} L^{-1}\sum_{l\leq L} \log\phi_c(x_c^l;\theta) - A(\theta)\right) \quad .$$

- The optimization is NOT separable into cliques.

Learning under the Exponential Family

- Suppose $p(x)$ is in the exponential family with shared parameters across cliques, ie

$$\log \phi_c(x_c; \theta) := \langle w, f(x_c) \rangle \ .$$

Learning under the Exponential Family

- Suppose $p(x)$ is in the exponential family with shared parameters across cliques, ie

$$\log \phi_c(x_c; \theta) := \langle w, f(x_c) \rangle \ .$$

- Then

$$\log p(x^1, \ldots, x^L; \theta) = \left\langle \theta \ , \ \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f(x_c^l) \right\rangle - A(\theta) \ .$$

- With

$$A(\theta) = \log \int \exp \left(\langle \theta, \sum_{c \in \mathcal{C}} f_c(x_c) \rangle \right) dx \ .$$

- No closed form solution for MLE!

Learning under the Exponential Family

$$\mathcal{L}(\theta) = \log p(x^1, \ldots, x^L; \theta) = \left\langle \theta \ , \ \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(x_c^l) \right\rangle - A(\theta) \ .$$

- How about local ascent methods?

Learning under the Exponential Family

$$\mathcal{L}(\theta) = \log p(x^1, \ldots, x^L; \theta) = \left\langle \theta \ , \ \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(x_c^l) \right\rangle - A(\theta) \ .$$

- How about local ascent methods?
- MLE is a convex (concave) optimization problem (why?).

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(x_c^l) - \frac{\partial A(\theta)}{\partial \theta}$$

- Denoting $f(x) := \sum_{c \in \mathcal{C}} f_c(x_c)$, we just saw that $\frac{\partial A(\theta)}{\partial \theta} = \mathbb{E}_\theta f(x)$

Learning under the Exponential Family

$$\mathcal{L}(\theta) = \log p(x^1, \ldots, x^L; \theta) = \left\langle \theta \ , \ \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(x_c^l) \right\rangle - A(\theta) \ .$$

- How about local ascent methods?
- MLE is a convex (concave) optimization problem (why?).

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(x_c^l) - \frac{\partial A(\theta)}{\partial \theta}$$

- Denoting $f(x) := \sum_{c \in \mathcal{C}} f_c(x_c)$, we just saw that $\frac{\partial A(\theta)}{\partial \theta} = \mathbb{E}_\theta f(x)$
- Thus $\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \overline{f} - \mathbb{E}_\theta\{f\}$

Empirical average Population average

MLE under Exponential Family

$$\text{MLE: } \sup_{\theta} \langle \theta, f(x) \rangle - A(\theta)$$

- Q1: Have we seen this function before?

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \overline{f} - \mathbb{E}_{\theta}\{f\}$$

- Q2: How to implement this ascent algorithm in practice?

MLE and Maximum Entropy

- Recall that

$$A^*(\mu) = \sup_{\theta} \langle \theta, f(x) \rangle - A(\theta)$$

  is the convex conjugate of the log-partition function.

- $A^*(\mu) =$ (negative) maximum entropy of any distribution p such that $\mathbb{E}_{X \sim p} f(X) = \overline{f}$. $(\mu = \overline{f})$

MLE and Maximum Entropy

- Recall that

$$A^*(\mu) = \sup_{\theta} \langle \theta, f(x) \rangle - A(\theta)$$

  is the convex conjugate of the log-partition function.

- $A^*(\mu) =$ (negative) maximum entropy of any distribution p such that $\mathbb{E}_{X \sim p} f(X) = \overline{f}$. $(\mu = \overline{f})$

- MLE solution satisfies $\mathbb{E}_{\theta} f = \overline{f}$. (why?)
  - This is denoted as *Moment Matching:* Empirical and model moments match at optimality.

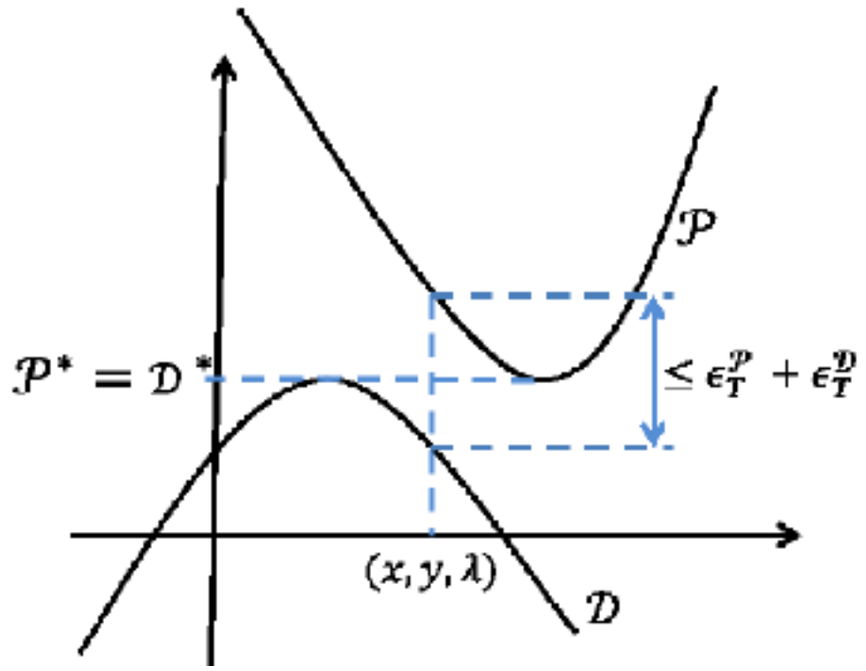
MLE and Maximum Entropy

- This is an instance of convex duality:
  - On the one hand, we assume exponential family and have shown that MLE implies model moments = empirical moments.
  - On the other hand, we assume model moments match empirical moments and shown that Max Ent implies exponential family.

MLE and Maximum Entropy

- This is an instance of convex duality:
  - On the one hand, we assume exponential family and have shown that MLE implies model moments = empirical moments.
  - On the other hand, we assume model moments match empirical moments and shown that Max Ent implies exponential family.
- Moreover, both objectives reach the same value at optimality:

$$A^*(\mu) = \sup_\theta \langle \theta, \mu \rangle - A(\theta) \qquad A(\theta) = \sup_\mu \langle \theta, \mu \rangle - A^*(\mu) \ .$$



Monte-Carlo

- Q2: How to efficiently implement such ascent method?

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \overline{f} - \mathbb{E}_\theta\{f\}$$

Monte-Carlo

- Q2: How to efficiently implement such ascent method?

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \overline{f} - \mathbb{E}_\theta\{f\}$$

- Use MCMC and estimate the gradient from

$$\widehat{\mathbb{E}_\theta f} = \frac{1}{S} \sum_{x^{(s)} \sim p_\theta} f(x^{(s)})$$

- We never need to estimate $A(\theta)$.

- Q: Other alternatives?

Approximating log-partition function

- The original learning objective function was

$$\mathcal{L}(\theta) = \log p(x^1, \ldots, x^L; \theta) = \left\langle \theta \ , \ \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(x_c^l) \right\rangle - A(\theta) \ .$$

- We can consider a tractable approximation of the log-partition function

$$\widetilde{\mathcal{L}}(\theta) = \left\langle \theta \ , \ \sum_{c \in \mathcal{C}} L^{-1} \sum_{l \leq L} f_c(x_c^l) \right\rangle - \tilde{A}(\theta) \ .$$

- For instance using convex relaxations, ie variational methods.

  - tree-reweighed belief propagation [Wainwright, Wilski & Jakkola] (based on "convexifying" Bethe approximation).
  - log-determinant relaxation [Wainwright & Jordan].

Pseudo-Likelihood

- An alternative route is to replace the objective function $\mathcal{L}(\theta)$ by another one $\bar{\mathcal{L}}(\theta)$, such that they are asymptotically maximized at the same parameter value as $L \to \infty$

Pseudo-Likelihood

- An alternative route is to replace the objective function $\mathcal{L}(\theta)$ by another one $\bar{\mathcal{L}}(\theta)$, such that they are asymptotically maximized at the same parameter value as $L \to \infty$

- The Pseudo-likelihood method (Besag'71) is a consistent estimator if the data is generated by a model p_{θ_0} in our family.

- From Bayes rule, we have

$$p(x;\theta) = \prod_i p(x_i \mid x_1 \ldots, x_{i-1};\theta) \ .$$

Pseudo-Likelihood

- An alternative route is to replace the objective function $\mathcal{L}(\theta)$ by another one $\bar{\mathcal{L}}(\theta)$, such that they are asymptotically maximized at the same parameter value as $L \to \infty$

- The Pseudo-likelihood method (Besag'71) is a consistent estimator if the data is generated by a model p_{θ_0} in our family.

- From Bayes rule, we have

$$p(x;\theta) = \prod_i p(x_i \mid x_1 \ldots, x_{i-1};\theta) \ .$$

- We consider the approximation

$$p(x;\theta) \approx \prod_i p(x_i \mid x_1 \ldots, x_{i-1}, x_{i+1}, \ldots, x_n;\theta) = \prod_i p(x_i \mid x_{-i};\theta)$$

– conditioning over all the other variables (similarly as in Gibbs sampling).

Pseudo-Likelihood

- We replace the likelihood $\mathcal{L}(\theta)$ by the "pseudo-likelihood"

$$\mathcal{L}_P(\theta) = \frac{1}{L} \sum_{l \leq L} \sum_{i=1}^{n} \log p(x_i^l \mid x_{N(i)}^l; \theta) \ .$$

$$N(i) = \text{ Markov Blanket of node } i \ .$$

- Example: pair-wise MRF. Then

$$p(x_i \mid x_{N(i)}; \theta) = \frac{1}{Z(x_{N(i)}; \theta)} \exp \left(\sum_{j \in N(i)} \theta_{ij}(x_i, x_j) \right) \ ,$$

$$Z(x_{N(i)}; \theta) = \sum_{x_i} \exp \left(\sum_{j \in N(i)} \theta_{ij}(x_i, x_j) \right) \ .$$

Pseudo-Likelihood

- This approximation only involves "local" partition functions
  - We need to compute n univariate integrals instead of a single n-dimensional one.
- In the exponential family, the pseudo-likelihood is still concave.
- Although it is asymptotically consistent as $L \to \infty$, this does not mean it is optimal for a fixed L.
- Again, we have a comptutational/statistical tradeoff.