Learning in undirected models:
Given structure and completely observed data

*Probabilistic Graphical Models*
*Sharif University of Technology*
*Spring 2016*

*Soleymani*
Parameter learning in undirected models with complete data

- We assume that the structure of the model is known
  - consider learning parameters for an undirected model with a given structure

- Goal: estimate parameters of an undirected models from a dataset $\mathcal{D} = \{x^{(1)}, \ldots, x^{(N)}\}$ of $N$ independent, identically distributed (i.i.d.) training samples.
  - Each training sample $x^{(n)}$ is a vector that every element $x_i^{(n)}$ is known
UGM: distribution

\[ P(X; \theta) = \frac{1}{Z(\theta)} \prod_c \phi(X_c) \]

\[ Z(\theta) = \sum_{X_1, \ldots, X_{|\mathcal{V}|}} \prod_c \phi(X_c) \]

- Log-linear form:

\[ P(x; \theta) = \frac{1}{Z(\theta)} \exp \left\{ \sum_{i=1}^{M} \theta_i^T f_i(x_i) \right\} \]

\( f_i \) shows the feature vector defined on the i-th clique (\( x_i \) denotes the nodes in the i-the clique)

Thus, we can have more than one feature on a clique
Likelihood function

\[
\ln P(\mathcal{D}|\theta) = \ln \prod_{n=1}^{N} P(x^n|\theta) = \sum_{n=1}^{N} \left[ \sum_{i=1}^{M} \theta_i^T f_i(x_i^{(n)}) - \ln Z(\theta) \right]
\]

- Likelihood decomposes for DGMs but it does not decompose for UGMs because of the use of a global normalization constant \(Z(\theta)\).
  - \(Z(\theta)\) couples all of the parameters across the network
  - preventing us from estimating local groups of parameters separately

\[
\nabla_{\theta_i} \ln P(\mathcal{D}|\theta) = \sum_{n=1}^{N} \left[ f_i(x_i^{(n)}) - \nabla_{\theta_i} \ln Z(\theta) \right]
\]

\[
\nabla_{\theta_i} \ln Z(\theta) = E_{P(x|\theta)}[f_i(x_i)]
\]
**MLE properties**

\[
\nabla_{\theta_i} \ln P(D|\theta) = \sum_{n=1}^{N} f_i(x_i^{(n)}) - NE_{P(x|\theta)}[f_i(x_i)]
\]

\[
= N \left[ \frac{1}{N} \sum_{n=1}^{N} f_i(x_i^{(n)}) - E_{P(x|\theta)}[f_i(x_i)] \right]
\]

\[
E_{P_{emp}}[f_i(x_i)]
\]

Then, \( \hat{\theta} \) is a maximum-likelihood parameter assignment iff

\[
E_{P_{emp}}[f_i(x_i)] = E_{P(x|\hat{\theta})}[f_i(x_i)]
\]

for all features.

The expected sufficient statistics in the learned distribution must match the empirical expectations (moment matching).

\( x_i \): shows the scope of the i-th clique

\( f_i(x_i) \): denotes a feature vector on the i-the clique
MLE properties

- The criterion $E_{P_{emp}}[f_i(x_i)] = E_{P(x|\theta)}[f_i(x_i)]$ does not provide a constructive definition of the maximum likelihood parameters $\hat{\theta}$
  - A system of equations that constrains ML estimates
  - In general, no closed form solution for it
  - We need to use gradient-based optimizers.

- The likelihood function is concave.
  - Likelihood is unimodal (has no local optima).

- Gradient ascent methods can be used
MLE: gradient-based optimization

\[ \theta_{i}^{t+1} = \theta_{i}^{t} + \alpha \nabla_{\theta_{i}} \ln P(D|\theta) \]

\[ \theta_{i}^{t+1} = \theta_{i}^{t} + \alpha \left( \frac{1}{N} \sum_{n=1}^{N} f_{i}(x_{i}^{(n)}) - E_{P(x|\theta_{t})}[f_{i}(x_{i})] \right) \]

- Computing \( E_{P(x|\theta_{t})}[f_{i}(x_{i})] = E_{P(x_{i}|\theta_{t})}[f_{i}(x_{i})] \) requires inference
  - Thus, a full inference step is needed in each gradient step
  - Therefore, UGM training much slower than DGM training

\( x_{i} \): shows the scope of the i-th clique
\( f_{i}(x_{i}) \): denotes a feature vector on the i-the clique
Gradient-based optimization: inference

- To compute $E_{P(x_i|\theta^t)}[f_i(x_i)]$ for all features, a single inference pass is sufficient.
  - An inference pass calibrates an entire tree suffices to compute all of the marginal distributions $P(x_i|\theta^t)$
    - all of the variables involved in a feature occurs together in a clique

- In models where inference is intractable, learning also becomes intractable.
  - Inference is almost always costly in time and space

- Approaches to speed up the learning process for UGMs:
  - Using approximate inference
  - Various computationally faster alternative objectives to ML estimation
    - formulation of alternative more tractable objectives (e.g., pseudo-likelihood)
We often want to use a UGM to perform a particular inference task.

A common use of UGMs is for settings where

- Variables include two sets $X$ and $Y$
  - $X$ is a known set of observed variables, or features,
  - $Y$ is a predetermined set of variables that we want to query

We may prefer to use discriminative training

- train the network as a CRF that encodes a conditional distribution $P(Y|X, \theta)$. 
Conditional trained model

- Training data: $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$

- Conditional likelihood as an appropriate objective function:
  \[ P(y^{(1)}, \ldots, y^{(N)} | x^{(1)}, \ldots, x^{(N)}, \theta) = \prod_{n=1}^{N} P(y^{(n)} | x^{(n)}, \theta) \]
  i.i.d. assumption

- The conditional likelihood is a concave function and its gradient is obtained as:
  \[ \nabla_{\theta_i} \ln \prod_{n=1}^{N} P(y^{(n)} | x^{(n)}, \theta) \]
  \[ = \sum_{n=1}^{N} \left( f_i (x^{(n)}, y^{(n)}) - E_{P(y|x^{(n)}, \theta)} [f_i (x^{(n)}, y)] \right) \]
CRF training vs. MRF training

- CRF training requires inference for each $x^{(n)}$ at each gradient step
  - We must (in general) execute inference for every single data instance in each iteration of gradient ascent
    - As opposed to learning of MRFs where each gradient step requires only a single execution of inference

- However, the inference is performed on a simpler model in CRF training
  - Learning $P(y|x)$ can be much simpler than learning $P(x, y)$
    - conditioning on evidence reduces the computational cost.
Leaning in undirected models

- Each step in the iterative algorithm requires inference on the network
  - making even simple parameter estimation a fairly expensive, or even intractable process.

- Bayesian estimation, which requires integration over the space of parameters, is even harder
  - there is no closed-form expression for the parameter posterior.
  - Thus, the integration associated with Bayesian estimation must be performed using approximate inference
Reference

- Koller & Friedman, Chapter 20.1-20.3.2.