Probabilistic Graphical Models

Conditional & Local Probability Models

Siamak Ravanbakhsh

Fall 2019
Learning Objective

- conditional random fields
- local probability models:
  - deterministic CPDs
  - noisy-OR model
  - generalized linear model
Conditional Random Fields: Motivation

**structured prediction:** output labels are structured

X is always observed
Y is structured

**Examples:**
- image segmentation
- part of speech tagging
- optical character recognition
Conditional Random Fields (CRF)

- a conditional graphical model $P(Y \mid X)$
- first attempt:  $P(Y \mid X) = \frac{P(X,Y)}{P(X)}$
  - for prediction, no need to model $P(X)$
    - may not have enough data
    - $X$ could be high-dim and $P(X)$ may be complex
Conditional Random Fields (CRF)

second attempt:

\[ P(Y \mid X) = \frac{1}{Z(X)} \tilde{P}(X, Y) = \frac{1}{Z(X)} \prod_k \phi_k(D_k) \]
Conditional Random Fields (CRF)

second attempt:

\[ P(Y \mid X) = \frac{1}{Z(X)} \tilde{P}(X, Y) = \frac{1}{Z(X)} \prod_k \phi_k(D_k) \]

- differs from MRF in the partition function
  - input-dependent \[ Z(X) = \sum_Y \tilde{P}(Y, X) \]
Conditional Random Fields: a running example

\[
P(Y \mid X) = \frac{1}{Z(X)} \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1})
\]

\[
Z(X) = \sum_{Y} \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1})
\]
Conditional Random Fields: a running example

\[ P(Y \mid X) = \frac{1}{Z(X)} \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1}) \]

\[ Z(X) = \sum_Y \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1}) \]

practically the same as

- e.g., in speech recognition (what do potentials encode?)

for each \( X=x \), we have a different MRF
Conditional Random Fields: another benefit

\[ P(Y \mid X) = \frac{1}{Z(X)} \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1}) \]

\[ Z(X) = \sum_{Y} \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1}) \]

what if \( \phi_i(X, Y_i) \) instead of \( \phi_i(X_i, Y_i) \) ?

**sparse structure** after conditioning on \( X=x \)

- learning needs inference on this structure *(discussed later)*
- not true for the corresponding MRF
Conditional Random Fields: input structure

\[
P(Y \mid X) = \frac{1}{Z(X)} \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1})
\]

\[
Z(X) = \sum_Y \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1})
\]

How about the structure of the input?

\[
P(Y \mid X) = \frac{1}{Z(X)} \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1}) \gamma_i(X_i, X_{i+1}) = \frac{1}{Z'(X)} \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1})
\]

i.e., input structure can be ignored
(already accounted for in the observations)
Conditional Random Fields: parametrization

\[ P(Y \mid X) = \frac{1}{Z(X)} \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1}) \]

\[ Z(X) = \sum_Y \prod_{i=1}^{5} \phi_i(X_i, Y_i) \prod_{i=1}^{4} \psi_i(Y_i, Y_{i+1}) \]

- In practice we need to learn the potentials
- parameterize them and learn the parameters (e.g., a neural network)
  - traditionally: a log-linear model: \( \phi_i(X_i, Y_i; w_i) \triangleq \exp(\sum_k w_{i,k} f_{i,k}(X_i, Y_i)) \)
    - E.g., for binary input/output:
      \[ \phi_i(X_i, Y_i; w_i) = \exp(w_i \mathbb{1}(X_i = 1, Y_i = 1)) = \exp(w_i X_i Y_i) \]
Local probabilistic models
Local probabilistic models

- conditional probability distributions (CPDs)
  - in prediction $P(Y \mid X_1, \ldots, X_n)$
  - in Bayes-nets $P(X \mid Pa_X)$
    - discrete variables (CPTs)
      - exponential in $|Pa_X|$ 
- how to represent these efficiently? exploit some sort of structure
Deterministic CPDs

\[ P(X \mid Pa_X) \triangleq \mathbb{I}(X_i = f(Pa_{X_i})) \]

determinism produces **additional independencies**:

- **without determinism:** \( (D \perp E \mid A, B) \notin \mathcal{I}(G) \)
- **with determinism:** \( (D \perp E \mid A, B) \in \mathcal{I}(G) \)
Deterministic CPDs

\[ P(X \mid Pa_X) \triangleq \mathbb{I}(X_i = f(Pa_{X_i})) \]

determinism produces **additional independencies**:

- without determinism: \((D \perp E \mid A, B) \notin \mathcal{I}(\mathcal{G})\)
- with determinism: \((D \perp E \mid A, B) \in \mathcal{I}(\mathcal{G})\)

**Deterministic d-separation:** \((X, Y \mid Z)\)?

- add all the variables that deterministically follow \(Z\) to define \(Z^+\)
- run d-separation for \((X, Y \mid Z^+)\)
Noisy-OR model

- for **binary** variables only
- number of parameters is linear in $|Pa_{X_i}|$
- each parent $(x_j = 1)$ is an **independent cause**
- each cause is observed with prob $P(X'_j = 1) = \lambda_j x_j$

noise parameter
Noisy-OR model

- for **binary** variables only
- number of parameters is linear in $|Pa^G_{X_i}|$
- each parent ($X_j = 1$) is an **independent cause**
- each cause is observed with prob $P(X'_j = 1) = \lambda_j X_j$

\[
p(X_i = 0 \mid Pa_{X_i}) = (1 - (\lambda_0)) \prod_{X_j \in Pa_{X_i}} (1 - \lambda_j X_j)
\]

- leak parameter (role of a bias term)
- prob. of **no cause observed**

\[
P(X'_i = 1) = \lambda_j X_j
\]

- noise parameter
Noisy-OR model: visualization

\[ p(X_i = 0 \mid Pa_{X_i}) = (1 - \lambda_0) \prod_{j \in Pa_{X_i}} (1 - \lambda_j X_j) \]

leak parameter (role of a bias term)

prob. of no cause observed
Noisy-OR model: example

Medical diagnosis *(BN2O network)*

various diseases/conditions

symptoms/test results

\[
\begin{align*}
\text{CPDs: } \quad & p(F_i = 0 \mid Pa_{F_i}) = (1 - \lambda_{i,0}) \prod_{D_j \in Pa_{F_i}} (1 - \lambda_{i,j} D_j) \\
& \quad \text{for } i = 1, 2, \ldots, n
\end{align*}
\]
Logistic CPD

for **binary output** variables

\[
P(X_i = 1) = \frac{\exp(\sum_j w_j X_j)}{1 + \exp(\sum_j w_j X_j)}
\]

logistic aggregation function

generally, the input can be **discrete** or **continuous**

- *E.g.*, \(X_j = 2\) or \(X_j, \ldots, X_{j+n} = 0, 1, \ldots, 0\)

---

one-hot coding
Logistic CPD

for **binary output** variables

\[ P(X_i = 1) = \frac{\exp(\sum_j w_j X_j)}{1 + \exp(\sum_j w_j X_j)} \]

logistic aggregation function

generally, the input can be **discrete** or **continuous**

- *E.g.*, \( X_j = 2 \) or \( X_j, \ldots, X_{j+n} = 0, 1, \ldots, 0 \)

one-hot coding

binary input: *each cause has a multiplicative effect on the ratio* \[ \frac{P(X_i=1)}{P(X_i=0)} \]
Softmax CPD

extension for categorical outputs
softmax function for aggregation:
\[ f(z_\ell) = \frac{\exp(z_\ell)}{\sum_{\ell'} \exp(z_{\ell'})} \]

functional form of the CPD:
\[ P(X_i = \ell) = \frac{\exp(\sum_j w_{j,\ell}X_j)}{\sum_{\ell'} \exp(\sum_j w_{j,\ell'}X_j)} \]
**Independence of causal influence**

*Commutative and associative aggregation*

**Logistic CPD**
- **transformation**
  \[ X'_j = w_j X_j \]
- **aggregation**
  logistic function

**Noisy-OR**
\[ P(X'_j = 1) = \lambda_j X_j \quad 0 \leq \lambda \leq 1 \]

OR /Max/...
Linear Gaussian CPD

for continuous input/output variables

\[ P(X_i) = \mathcal{N}(\sum_j w_j X_j; \sigma^2) \]
Linear Gaussian CPD

for **continuous** input/output variables

\[ P(X_i) = \mathcal{N}(\sum_j w_j X_j; \sigma^2) \]

alternatively, a **discrete** input selects among **continuous** coefficients (produces a Gaussian mixture):

\[ P(X_i) = \mathcal{N}(\sum_j w_j X_d X_j; \sigma^2_{X_d}) \]

**conditional** linear Gaussian CPD:

one Gaussian mixture for each discrete assignment
Generalized linear models

\[ \mathbb{E}[X_i] = f(w^T P a_{X_i}) \]

mean function

**Logistic CPD:** \( f \) is the logistic function

**Gaussian CPD:** \( f \) is the identity function

\[ X_j \quad \ldots \quad Pa^G_{X_i} \]

\[ X'_j = w_j X_j \]

\[ X_i \]
Generalized linear models

\[ \mathbb{E}[X_i] = f(w^T P a_{X_i}) \]

**Logistic CPD:** \( f \) is the logistic function

**Gaussian CPD:** \( f \) is the identity function

conditional dist. is a member of the exponential family

\[ p(x_i \mid P a_{X_i}) = h(x_i) \exp(w^T P a_{X_i} - F(w^T P a_{X_i})) \]

(base measure) \hspace{1cm} (integral of \( f \))

*(will come back to this in exp. family lecture)*
Conditional Bayesian networks

use an entire Bayes-net to represent a CPD

\[
P(X_i \mid Pa_{X_i}) = \sum_Z P(X_i, Z \mid Pa_{X_i})
\]
Conditional Bayesian networks: example

can be used for encapsulation in complex models
Neural networks defining CPDs

- this idea is extensively used in *deep generative models*
- alternative strategies:
Neural networks defining CPDs

- this idea is extensively used in *deep generative models*
- alternative strategies:
  - $f$ is a deterministic CPD
    - in Generative Adversarial Networks (GANs)
    - in Normalizing Flows (special family of functions $f$)
Neural networks defining CPDs

- this idea is extensively used in *deep generative models*
- alternative strategies:
  - $f$ is a deterministic CPD
    - in Generative Adversarial Networks (GANs)
    - in Normalizing Flows (*special family of functions f*)
  - $f$ defines parameters in a parametric distribution
    - In Variational Auto Encoders (VAEs) and Auto-Regressive models
Neural networks defining CPDs

- this idea is extensively used in *deep generative models*
- alternative strategies:
  - \( f \) is a deterministic CPD
    - in Generative Adversarial Networks (GANs)
    - in Normalizing Flows (*special family of functions* \( f \))
  - \( f \) defines parameters in a parametric distribution
    - In Variational Auto Encoders (VAEs) and Auto-Regressive models
  - \( f \) is a stochastic function itself
    - In energy-based models or VAEs with stochastic middle-layers
Summary

the conditioned version of directed & undirected models:

- Conditional Random Fields
- Conditional Bayes-nets
Summary

the conditioned version of directed & undirected models:

- Conditional Random Fields
- Conditional Bayes-nets

representing conditional probabilities:

- deterministic CPD
- noisy-OR model
- logistic CPD
- linear Gaussian CPD

part of a bigger family of GLMs
Summary

the conditioned version of directed & undirected models:

- Conditional Random Fields
- Conditional Bayes-nets

representing conditional probabilities:

- deterministic CPD
- noisy-OR model
- logistic CPD
- linear Gaussian CPD

neural networks define expressive CPDs

part of a bigger family of GLMs