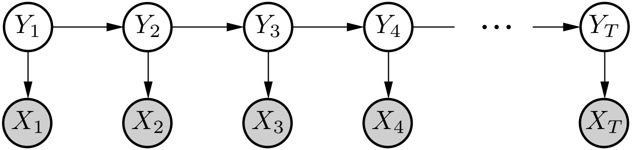
Introduction to Graphical Models
lecture 5 - Conditional Random Fields

Ulf Brefeld
TU Berlin

– Hidden Markov models
– Contidional random fields (CRFs)

Recall: HMMs



- Hidden Markov models
  - generative models for sequential data
  - parameters: prior, transition, and observation probabilities
  - joint probability:

$$P(X_1, \ldots, Y_1 \ldots) = P(Y_1) \prod_{i=1}^{T} P(X_i | Y_i) \prod_{i=2}^{T} P(Y_i | Y_{i-1})$$

Learning HMMs

- given: n labeled sequences $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$

- maximum Likelihood (ML)
  – adapt parameters of HMM to data
  – HMM: ML reduces to counting
  – efficient (one pass over data suffices)
  – easy to implement
  – exact inference (Viterbi algorithm)

- drawbacks
  – $P(\text{unobserved token}|Y_i) = 0$ (remedy: smoothing techniques)
  – generative models optmize the wrong criterion

Today: From HMMs to CRFs

- Use undirected graphical model

  – no assumption on directions of dependencies (i.e., WWW, NLP, images, ...)

  – sequences: factor graph does not change

  – Markov random fields

- Condition joint probability of MRF on observations

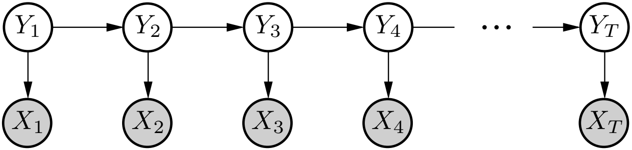
  – criterion: prediction model

  – now: conditional (=discriminative) model

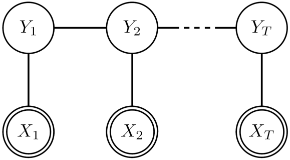
Conditional Random Fields

Markov Random Fields

HMM:

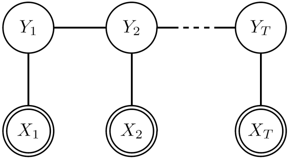


MRF:



- every BN can be translated into equivalent MRF (moralization)
  – but: not always necessary (i.g., web pages)
  – dependencies now bidirectional

MRF: Joint Probability Distribution



- joint probability factorizes across cliques
  - cliques between transitions and label-observation pairs

$$P(X_1, \ldots, Y_1, \ldots) = \frac{1}{Z} \prod_{i=1}^{T} \psi^{obs}(X_i, Y_i) \prod_{i=2}^{T} \psi^{trans}(Y_{i-1}, Y_i)$$

- potential functions $\psi^{trans}(Y_i, Y_{i-1})$, $\psi^{trans}(Y_i, Y_{i-1})$
- Z normalization term (=partition function)

Partition Function

$$P(X_1,\ldots,Y_1,\ldots) = \frac{1}{Z}\prod_{i=1}^{T}\psi^{obs}(X_i,Y_i)\prod_{i=2}^{T}\psi^{trans}(Y_{i-1},Y_i)$$

- the partition function needs to sum over all possible assignments of input and output sequences
  - we have:

$$Z = \sum_{x_1,\ldots,x_T}\sum_{y_1,\ldots,y_T}\prod_{i=1}^{T}\psi^{obs}(X_i,Y_i)\prod_{i=2}^{T}\psi^{trans}(Y_{i-1},Y_i)$$

  - important for $P(X_1,\ldots,Y_1,\ldots)$ being a probability

Potential Functions

$$P(X_1, \ldots, Y_1, \ldots) = \frac{1}{Z} \prod_{i=1}^{T} \psi^{obs}(X_i, Y_i) \prod_{i=2}^{T} \psi^{trans}(Y_{i-1}, Y_i)$$

- potential functions ψ^{trans} (transitions), ψ^{obs} (label-observ.)
  – arbitrary, non-negative, positive functions
  – capture relevant dependencies
  – defined across cliques

- problem:
  – size of largest clique depends on input (i.e., WWW)
  – remedy: represent only cliques of size 2 (=Markov network)

Representation

$$P(X_1, \ldots, Y_1, \ldots) = \frac{1}{Z} \prod_{i=1}^{T} \psi^{obs}(X_i, Y_i) \prod_{i=2}^{T} \psi^{trans}(Y_{i-1}, Y_i)$$

- sequences
  - all cliques are of size 2
  - only their number varies with T

- how to choose ψ^{trans}, ψ^{obs}?
  - (remember they have to capture relevant dependencies)

- common assumption (Hammersley & Clifford theorem):
  - ψ is log-linear combination of basis functions ϕ_j

Members in the Exponential Family

- basis functions:

$$\psi^{trans}(Y_i, Y_{i-1}) = \exp\{\sum_{j=1}^{d_{trans}} w_j^{trans} \phi_j^{trans}(Y_{i-1}, Y_i)\}$$

$$\psi^{obs}(X_i, Y_i) = \exp\{\sum_{j=1}^{d_{obs}} w_j^{obs} \phi_j^{obs}(X_i, Y_i)\}$$

  – math turns out to be nice!

  – write:

$$P(X_1, \ldots, Y_1, \ldots) = \frac{1}{Z} \prod_{i=1}^{T} \exp\{\sum_{j=1}^{d_{obs}} w_j \phi_j^{obs}(X_i, Y_i)\} \prod_{i=2}^{T} \exp\{\sum_{j=1}^{d_{trans}} w_j \phi_j^{trans}(Y_{i-1}, Y_i)\}$$

Basis Functions: label-label

$$\psi^{trans}(Y_i, Y_{i-1}) = \exp\left\{\sum_{j=1}^{d_{trans}} w_j^{trans} \phi_j^{trans}(Y_{i-1}, Y_i)\right\}$$

- simple case: indicator functions

$$\begin{aligned}
\phi_1^{trans}(Y_{i-1}, Y_i) &= [[Y_{i-1} = \text{noun} \wedge Y_i = \text{noun}]] \\
\phi_2^{trans}(Y_{i-1}, Y_i) &= [[Y_{i-1} = \text{noun} \wedge Y_i = \text{verb}]] \\
\vdots \qquad & \qquad \vdots \\
\phi_{d_{trans}}^{trans}(Y_{i-1}, Y_i) &= [[Y_{i-1} = \text{adverb} \wedge Y_i = \text{adverb}]]
\end{aligned}$$

  – similar to HMM

  – later more...

Basis Functions: label-observation

$$\psi^{obs}(X_i, Y_i) = \exp\left\{\sum_{j=1}^{d_{obs}} w_j^{obs} \phi_j^{obs}(X_i, Y_i)\right\}$$

- simple case: indicator functions

$$\begin{aligned}
\phi_1^{obs}(X_i, Y_i) &= [[X_i = \text{Aachen} \wedge Y_i = \text{noun}]] \\
\phi_2^{obs}(X_i, Y_i) &= [[X_i = \text{Aar} \wedge Y_i = \text{noun}]] \\
\vdots \qquad & \qquad\qquad \vdots \\
\phi_{d_{obs}}^{obs}(X_i, Y_i) &= [[X_i = \text{ZZ-top} \wedge Y_i = \text{adverb}]]
\end{aligned}$$

  – similar to HMM

  – later more...

Putting Everything Together...

$$
\begin{aligned}
P(\mathbf{x},\mathbf{y}) &= \frac{1}{Z}\prod_{i=1}^{T}\exp\{\sum_{j=1}^{d_o} w_j^o\phi_j^o(x_i,y_i)\}\prod_{i=2}^{T}\exp\{\sum_{j=1}^{d_t} w_j^t\phi_j^t(y_{i-1},y_i)\} \\
&= \frac{1}{Z}\prod_{i=1}^{T}\exp\{\langle\mathbf{w}^o,\phi^o(x_i,y_i)\rangle\}\prod_{i=2}^{T}\exp\{\langle\mathbf{w}^t,\phi^t(y_{i-1},y_i)\rangle\} \\
&= \frac{1}{Z}\exp\{\sum_{i=1}^{T}\langle\mathbf{w}^o,\phi^o(x_i,y_i)\rangle\}\exp\{\sum_{i=2}^{T}\langle\mathbf{w}^t,\phi^t(y_{i-1},y_i)\rangle\} \\
&= \frac{1}{Z}\exp\{\langle\mathbf{w}^o,\sum_{i=1}^{T}\phi^o(x_i,y_i)\rangle\}\exp\{\langle\mathbf{w}^t,\sum_{i=2}^{T}\phi^t(y_{i-1},y_i)\rangle\} \\
&= \frac{1}{Z}\exp\{\langle\underbrace{\begin{pmatrix}\mathbf{w}^o\\ \mathbf{w}^t\end{pmatrix}}_{=:\mathbf{w}},\underbrace{\begin{pmatrix}\sum_{i=1}^{T}\phi^o(x_i,y_i)\\ \sum_{i=2}^{T}\phi^t(y_{i-1},y_i)\end{pmatrix}}_{=:\Phi(\mathbf{x},\mathbf{y})}\rangle\} \\
&= \frac{1}{Z}\exp\{\langle\mathbf{w},\Phi(\mathbf{x},\mathbf{y})\rangle\}
\end{aligned}
$$

Joint Feature Representation

- joint representation of input and output variables:
  $\Phi(\mathbf{x}, \mathbf{y}) = (\sum_{i=1}^{T} \phi^o(x_i, y_i), \sum_{i=2}^{T} \phi^t(y_{i-1}, y_i))'$

- Example for HMM-alike basis functions:
  – $\Phi(\mathbf{x}, \mathbf{y})$ counts how many times ...
  – ... a *noun* is followed by *verb* (summing over transitions)
  – ... the token *Aachen* is observed as a noun (sum over obs-label)
  – dimensionality of Φ is $dom(x_i) \times dom(y_i) + dom(y_i)^2$

- POS-tagging:
  – dictionary size 20,000 tokens, 36 POS-tags, $dim(\Phi) = 721296$

Example

Features

- Features are engineered to capture important relations/dependencies

- all time favorites for natural language text:
  – n-grams (English: *-ing*, German: *-ung, -heit, -keit*)
  – surface clues (capitalization, all-caps, ...)
  – foreign symbols (α, ω, ...)
  – numbers (42, 1984, ...)

- CRFs allow for rich feature spaces
  – CRFs may contain any number of basis functions
  – basis functions can be defined on the entire input sequence
  – basis functions do need not have a probabilistic interpretation.

More Features / Relation to HMM

- observation-label/transitions can depend on input
  - $\phi^{trans}(y_{t-1}, y_t) \rightarrow \phi^{trans}(y_{t-1}, y_t; x_t)$
  - or even: $\phi^{trans}(y_{t-1}, y_t) \rightarrow \phi^{trans}(y_{t-1}, y_t; \mathbf{x})$
  - similarly: $\phi^{obs}(x_t, y_t) \rightarrow \phi^{obs}(\mathbf{x}, y_t)$
  - (alternative graph structure)

- Implications for HMMs
  - Multi-bernoulli/nomial distribution
  - Generally infeasible

The Exponential Family

$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\{\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle\}$$

- $P(\mathbf{x}, \mathbf{y})$ is a member in the exp. family, rewrite in canonical form

$$P(\mathbf{x}, \mathbf{y}) = \exp\{\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle - \log Z\}$$

- Identify the terms:
  - $\Phi(\mathbf{x}, \mathbf{y})$ is the sufficient statistics
  - $\mathbf{w}$ is the natural parameter
  - $\log Z < \infty$ is the moment generating function

Conditional Markov Random Fields

- joint probability

$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\{\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle\}$$

  – partition function: $Z = \sum_{\mathbf{x}} \sum_{\mathbf{y}} \exp\{\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle\}$

- condition on the observation

  – apply the rule: $P(\mathbf{y}|\mathbf{x}) = P(\mathbf{x}, \mathbf{y})/P(\mathbf{x})$

  – obtain new partition function:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp\{\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle\}$$

- obtain a so-called conditional random field (CRF)

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\{\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle\}$$

Training CRFs with Maximum Likelihood

- given n input output examples $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$

- the log-likelihood is given

$$\log \mathcal{L} = \sum_{i=1}^{n} \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \log Z(\mathbf{w}|\mathbf{x}_i)$$

- differentiating wrt $\mathbf{w}$ gives

$$\frac{\partial}{\partial \mathbf{w}} \log \mathcal{L} = \mathbf{E}_{\hat{p}(X,Y)}[\Phi(X,Y)] - \sum_{i=1}^{n} \mathbf{E}_{p(Y|\mathbf{x}_i;\mathbf{w})}[\Phi(Y, \mathbf{x}_i)]$$

  – empirical distribution of data $\hat{p}$

  – model distribution p

Optimization

- direct optimization is expensive and often infeasible
  – E.g., calculating the partition function is time consuming if at all possible

- Many differerent optimization strategies have been proposed:
  – linear programming (Roth & Li, 2005)
  – iterative scaling (Lafferty et al., 2001)
  – conjugate gradients (Sha & Pereira, 2003)
  – Gauss-Newton subspace optimization (Altun et al., 2004)
  – gradient tree boosting (Dietterich et al., 2004)
  – stochastic meta descent (Vishwanathan et al., 2006)
  – perceptron algorithm (Altun et al., 2003)
  – ...

The Perceptron Algorithm for CRFs

CRF vs. HMM

- characteristics:
  - CRF: undirected graph, conditional models
  - HMM: directed BN,. generative model

- CRFs generalize HMMs
  - CRFs allow for rich feature spaces
  - HMMs restricted to implicit bag-of-words representation

- Optimization
  - CRF: difficult, complex optimization problem
  - HMM: simple, easy to implement

- Similarities:
  - inference algorithms (Viterbi, sum-product)

Posterior vs. MAP

- Once optimal parameters $\mathbf{w}^*$ are found these are used as plug-in estimates $P(\mathbf{y}|\mathbf{x};\mathbf{w}^*)$
  – posterior distribution allows for computing confidence intervals

- However, the full posterior is not always needed
  – often, the maximum a posteriori (MAP) estimate suffices
  – e.g., prediction model $\hat{\mathbf{y}} = \mathrm{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$
  – computing MAP estimates is much cheaper than full posterior!

Computing MAP Estimates

- For MAP estimates compute

$$\begin{aligned}\hat{\mathbf{y}} &= \operatorname*{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \\ &= \operatorname*{argmax}_{\mathbf{y}} \frac{1}{Z(\mathbf{x})} \exp\{\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y})\rangle\} \\ &= \operatorname*{argmax}_{\mathbf{y}} \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y})\rangle\end{aligned}$$

– because $\exp$ is a monotone function and $\frac{1}{Z(\mathbf{x})}$ is constant

- We arrive at:

$$P(\mathbf{x}, \mathbf{y}) \propto \underbrace{\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y})\rangle}_{=: f(\mathbf{x}, \mathbf{y})}$$

Outlook

- adapt $f(\mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$ to data

- perceptron algorithm
  – primal: efficient, nof parameters = $dim(\Phi)$
  – dual: nof parameters = nof possible output sequences

- dual perceptron
  – explicit representation is infeasible
  – solve implicitly by column generation

- examples