Probabilistic Machine Learning:
8. Probabilistic Graphical Models - part 2

Tomasz Kajdanowicz, Piotr Bielak, Maciej Falkiewicz, Kacper Kania, Piotr Zieliński

Department of Computational Intelligence
Wroclaw University of Science and Technology

HR EXCELLENCE IN RESEARCH

Wrocław University of Science and Technology

The presentation has been inspired and in some parts totally based on

1. Machine Learning A Probabilistic Perspective, Kevin Murphy [1] Chapters 19. Undirected graphical models (Markov random fields 22. More variational inference 27. Latent variable models for discrete data
2. Pattern Recognition and Machine Learning, Christopher M. Bishop [2] Chapter 8. Graphical Models

Pre-reading

Build, Compute, Critique, Repeat: Data Analysis with Latent Variable Models. David M. Blei, Columbia University

So far covered in PGM part 1

- Plate notation with examples
- Markov Random Fields
  - Conditional independence
  - Markov blanket
  - Joint distribution factorization (potential functions over maximal clicks)
  - Relation of directed and undirected models (moralization)
- Inference in graphical models
  - Tree
  - Polytree
  - Factor graphs
  - sum-product algorithm

Table of Contents

Markov Random Fields - recap

Conditinal Random Fields

Latent Dirichlet Allocation

Approximate Inference in PGMs
- Introduction
- Loopy Belief Propagation

Table of Contents

Markov Random Fields - recap

Conditinal Random Fields

Latent Dirichlet Allocation

Approximate Inference in PGMs
- Introduction
- Loopy Belief Propagation

Directed graphical models vs undirected graphical model - recap

Advantages of models

UGMs over DGMs

- are symmetric and therefore more "natural" for certain domains (e.g. spatial or relational data)
- discriminative UGMs (aka conditional random fields) which define conditional densities of the form $p(y|x)$, work better than discriminative DGMs, s. 13

DGMs over UGMs

- the parameters are more interpretable
- parameter estimation is computationally less expensive

MRF recap

Joint distribution:

- C denotes maximal cliques
- $\mathbf{y}_C$ is a set of variables in the clique
- $\psi_C(\mathbf{y}_C)$ are *potential functions*

$$P(\mathbf{y}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{y}_C), \tag{1}$$

where

$$Z = \sum_{\mathbf{y}} \prod_C \psi_C(\mathbf{y}_C) \tag{2}$$

acts as a normalizing factor (*partition function*)

If we parametrize the edges of the GM rather than the maximal cliques, we have a *pairwise MRF* with

$$P(\mathbf{y}) \propto \prod_{s \sim t} \psi_{st}(\mathbf{y}_s, \mathbf{y}_t) \prod_t \psi_t(\mathbf{y}_t), \tag{3}$$

Wrocław University of Science and Technology

Examples of MRFs

1. Ising model
   - for modeling the behavior of magnets
   - represent the spin of an atom (binary model)
   - defines the pairwise clique potential (for pair of random variables)
2. Hopfield networks
   - fully connected Ising model
   - possible to interpret this model as a recurrent neural network
   - *Boltzmann machine* generalizes the Hopfield / Ising model by including some hidden nodes
3. Potts model
   - Ising model with multiple discrete states
   - used as a prior for image segmentation, since it says that neighboring pixels are likely to have the same discrete label and hence belong to the same segment
4. Gaussian MRFs
   - pairwise MRF
   - potentials are miltivariate Gaussian like

Table of Contents

Markov Random Fields - recap

Conditinal Random Fields

Latent Dirichlet Allocation

Approximate Inference in PGMs
- Introduction
- Loopy Belief Propagation

Conditinal Random Fields

CRF [3] is just a version of an MRF where all the clique potentials are conditioned on input features

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{Z(\boldsymbol{x}, \boldsymbol{w})} \prod_c \psi_c(\boldsymbol{y}_c|\boldsymbol{x}, \boldsymbol{w}) \tag{4}$$

where $\boldsymbol{w}$ are the parameters.
Ussual way is to assume a log-linear representation of potentials:

$$\psi_c(\boldsymbol{y}_c|\boldsymbol{x}, \boldsymbol{w}) = \exp(\boldsymbol{w}_c^T \phi(\boldsymbol{x}, \boldsymbol{y}_c)) \tag{5}$$

where $\phi(\boldsymbol{x}, \boldsymbol{y}_c)$ is a feature vector derived from the global inputs $\boldsymbol{x}$ and the local set of labels $\boldsymbol{y}_c$

Wrocław University of Science and Technology

CRF vs MRF

Advantages of models

CRF over an MRF

- the same advantage of a discriminative classifier over a generative classifier
- we don't need to "waste resources" modeling things that we always observe
- potentials (or factors) can be data-dependent, e.g.
  - in image processing - "turn off" the label smoothing between two neighboring nodes *s* and *t* if there is an observed discontinuity in the image intensity between pixels *s* and *t*

MRF over an CRF

- do not require labeled training data
- faster to train

Wrocław University of Science and Technology

Chain-structured CRFs

- most widely used kind of CRF
- uses a chain-structured graph to model correlation amongst neighboring labels
- useful for a variety of sequence labeling tasks

Various models for sequential data

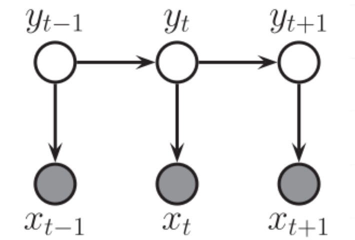


Figure: A generative directed HMM

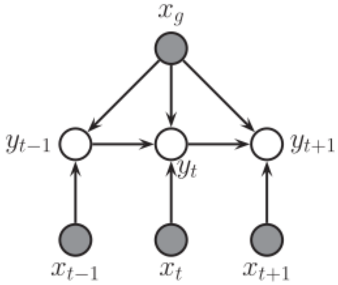


Figure: A discriminative directed MEMM

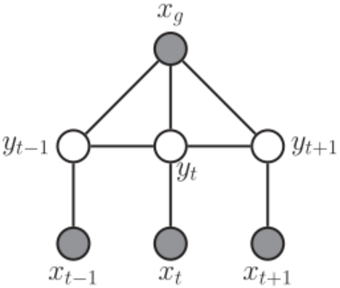


Figure: A discriminative undirected CRF

HMM

- if we observe both $\boldsymbol{x}_t$ and y_t for all t, it is very easy to train such models, e.g. EM algorithm
- HMM requires specifying a generative observation model $\mathrm{P}(\boldsymbol{x}_t|y_t, \boldsymbol{w})$

$$\mathrm{P}(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{w}) = \prod_{t=1}^{T} \mathrm{P}(y_t|y_{t-1}, \boldsymbol{w})\mathrm{P}(\boldsymbol{x}_t|y_t, \boldsymbol{w})$$

Maximum Entropy Markov model

- discriminative version of an HMM
- state transition probabilities are conditioned on the input features
- label bias problem: local features at time t do not influence states prior to time t

$$\mathrm{P}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) = \prod_{t} \mathrm{P}(y_t|y_{t-1}, \boldsymbol{x}, \boldsymbol{w})$$

where $\boldsymbol{x} = (\boldsymbol{x}_{1:T}, \boldsymbol{x}_g)$, $\boldsymbol{x}_g$ are global features, $\boldsymbol{x}_t$ are features specific to node t.

CRF

- label bias problem no longer exists, since y_t does not block the information from x_t from reaching other y_t' nodes

$$P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}) = \frac{1}{Z(\boldsymbol{x},\boldsymbol{w})} \prod_{t=1}^{T} \psi(y_t|\boldsymbol{x},\boldsymbol{w}) \prod_{t=1}^{T-1} \psi(y_t, y_{t+1}|\boldsymbol{x},\boldsymbol{w})$$

- label bias problem in MEMMs: because directed models are locally normalized (CPD sums to 1)
- MRFs and CRFs are globally normalized - local factors do not need to sum to 1 (there is the partition function Z, which sums over all joint configurations)
- then we need to compute all factors to obtain Z, will
- CRFs are not useful for online or real-time inference
- CRFs is much slower to train than DGMs

Applications of CRFs

Figure: Handwriting recognition

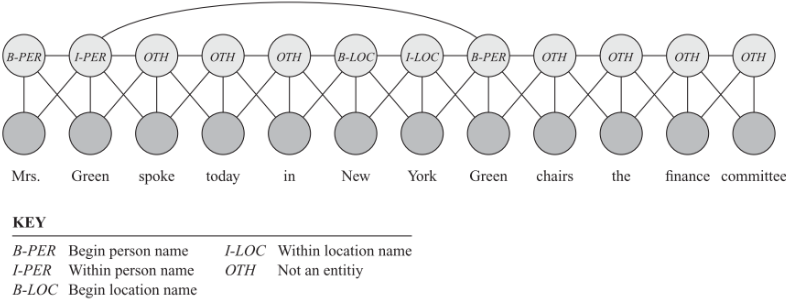


Figure: A skip-chain CRF for named entity recognition

Wrocław University of Science and Technology

Gradient method for CRF training

Consider an CRF in log-linear form:

$$\mathrm{P}(\boldsymbol{y}|\boldsymbol{w}) = \frac{1}{Z(\boldsymbol{w})} \exp\left(\sum_c \boldsymbol{w}_c^T \phi_c(\boldsymbol{y})\right) \tag{6}$$

then the scaled log-likelihood becomes:

$$\ell(\boldsymbol{w}) = \frac{1}{N}\sum_i \log \mathrm{P}(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}) = \frac{1}{N}\sum_i \left[\sum_c \boldsymbol{w}_c^T \phi_c(\boldsymbol{y}_i, \boldsymbol{x}_i) - \log Z(\boldsymbol{w}, \boldsymbol{x}_i)\right] \tag{7}$$

Since CRFs are in the exponential family, we know that this function is convex in $\boldsymbol{w}$, so it has a unique global maximum which we can find using gradient-based optimizers.
The gradient:

$$\frac{\partial \ell}{\partial \boldsymbol{w}_c} = \frac{1}{N}\sum_i \left[\phi_c(\boldsymbol{y}_i, \boldsymbol{x}_i) - \frac{\partial}{\partial \boldsymbol{w}_c} \log Z(\boldsymbol{w}, \boldsymbol{x}_i)\right] = \frac{1}{N}\sum_i \left[\boldsymbol{\phi}_c(\boldsymbol{y}_i, \boldsymbol{x}_i) - \mathbb{E}[\phi_c(\boldsymbol{y}_i, \boldsymbol{x}_i)]\right] \tag{8}$$

Inference must be done for every single training case inside each gradient because the partition function depends on the inputs $\boldsymbol{x}_i$.

Wrocław University of Science and Technology

Table of Contents

Markov Random Fields - recap

Conditinal Random Fields

Latent Dirichlet Allocation

Approximate Inference in PGMs
- Introduction
- Loopy Belief Propagation

Latent Dirichlet Allocation

- generative probabilistic model
- topic modelling
- the composites: documents, the parts: words
- Possible application:
  - DNA and nucleotides,
  - pizzas and toppings,
  - molecules and atoms,
  - employees and skills

The probabilistic topic model estimated by LDA consists of:

1. a table that describes the probability or chance of selecting a particular **word** when sampling a particular **topic**
2. a table that describes the chance of selecting a particular **topic** when sampling a particular **document**

Example model configutration

	Topic 0	Topic 1	Topic 2
☀	0.000	1.000	0.000
🎩	0.000	0.000	0.559
🐱	1.000	0.000	0.441
	Topic 0	Topic 1	Topic 2

Figure: Example word-topic distribution. Credits - working demo: https://lettier.com/projects/lda-topic-modeling/

	Topic 0	Topic 1	Topic 2
Document 0	0.486	0.116	0.399
Document 1	0.094	0.638	0.268
Document 2	0.377	0.616	0.007
Document 3	0.007	0.899	0.094
	Topic 0	Topic 1	Topic 2

Figure: Example document-topic distribution. Credits - working demo: https://lettier.com/projects/lda-topic-modeling/

LDA Model

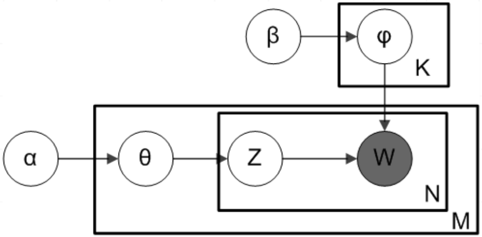


Figure: LDA with Dirichlet-distributed topic-word distributions. Credits: wikipedia

- M denotes the number of documents
- N is number of words in a given document (document i has N_i words
- α is the parameter of the Dirichlet prior on the per-document topic distributions
- β is the parameter of the Dirichlet prior on the per-topic word distribution
- θ_i is the topic distribution for document i
- φ_k is the word distribution for topic k
- z_{ij} is the topic for the j-th word in document i
- w_{ij} is the specific word

LDA model remarks

- W is grayed out - words w_{ij} are the only observable variables
- all other variables are *latent variables*
- following the intuition that the probability distribution over words in a topic is skewed, a sparse Dirichlet prior can be used to model the topic-word distribution
- only a small set of words have high probability
- documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over all the words

LDA generative procedure

For a corpus D consisting of M documents each of length N_i:

1. Choose $\theta_i \sim \mathrm{Dir}(\alpha)$ where $i \in \{1, \ldots, M\}$ and $\mathrm{Dir}(\alpha)$ is a Dirichlet distribution with a symmetric parameter α which typically is sparse (α <1)
2. Choose $\varphi_k \sim \mathrm{Dir}(\beta)$, where $k \in \{1, \ldots, K\}$ and β typically is sparse
3. For each of the word positions i,j, where $i \in \{1, \ldots, M\}$ and $j \in \{1, \ldots, N_i\}$
   a. Choose a topic $z_{i,j} \sim \mathrm{Categorical}(\theta_i)$
   b. Choose a word $w_{i,j} \sim \mathrm{Categorical}(\varphi_{z_{i,j}})$

Variable	Type	Meaning
K	integer	number of topics (e.g. 50)
V	integer	number of words in the vocabulary (e.g. 50,000 or 1,000,000)
M	integer	number of documents
$N_{d=1\dots M}$	integer	number of words in document *d*
N	integer	total number of words in all documents; sum of all N_d values, i.e. $N = \sum_{d=1}^{M} N_d$
$\alpha_{k=1\dots K}$	positive real	prior weight of topic *k* in a document; usually the same for all topics; normally a number less than 1, e.g. 0.1, to prefer sparse topic distributions, i.e. few topics per document
$\boldsymbol{\alpha}$	*K*-dimensional vector of positive reals	collection of all α_k values, viewed as a single vector
$\beta_{w=1\dots V}$	positive real	prior weight of word *w* in a topic; usually the same for all words; normally a number much less than 1, e.g. 0.001, to strongly prefer sparse word distributions, i.e. few words per topic
$\boldsymbol{\beta}$	*V*-dimensional vector of positive reals	collection of all β_w values, viewed as a single vector
$\varphi_{k=1\dots K,w=1\dots V}$	probability (real number between 0 and 1)	probability of word *w* occurring in topic *k*
$\boldsymbol{\varphi}_{k=1\dots K}$	*V*-dimensional vector of probabilities, which must sum to 1	distribution of words in topic *k*
$\theta_{d=1\dots M,k=1\dots K}$	probability (real number between 0 and 1)	probability of topic *k* occurring in document *d*
$\boldsymbol{\theta}_{d=1\dots M}$	*K*-dimensional vector of probabilities, which must sum to 1	distribution of topics in document *d*
$z_{d=1\dots M,w=1\dots N_d}$	integer between 1 and *K*	identity of topic of word *w* in document *d*
$\mathbf{Z}$	*N*-dimensional vector of integers between 1 and *K*	identity of topic of all words in all documents
$w_{d=1\dots M,w=1\dots N_d}$	integer between 1 and *V*	identity of word *w* in document *d*
$\mathbf{W}$	*N*-dimensional vector of integers between 1 and *V*	identity of all words in all documents

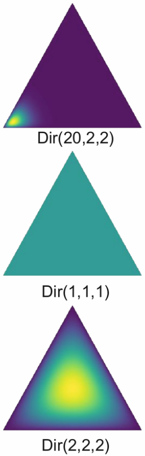
Figure: Definition of variables in the model. Credits: wikipedia

Remark on Dirichlet distribution

- K - way categorical events
- α - number of observed outcomes
- multivariate generalization of the Beta distribution

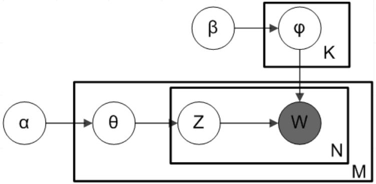
$$\mathrm{Dir}(x|\alpha) = \frac{1}{\mathrm{B}(\alpha)} \prod_{k=1}^{K} x_k^{\alpha_k - 1} \quad (9)$$

where $\mathrm{B}(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)}$ and $\alpha = (\alpha_1, \ldots, \alpha_K)$



Figure

Probability factorization



- M documents
- N words in document
- K topics
- α is the parameter on the per-document topic distributions
- β is the parameter on the per-topic word distribution
- θ_i topic distribution for document i
- φ_k word distribution for topic k
- z_{ij} topic for the j-th word in document i
- w_{ij} specific word

$$\mathrm{P}(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}, \alpha, \beta) = \prod_{k=1}^{K} \mathrm{P}(\varphi_k, \beta) \prod_{i=1}^{M} \mathrm{P}(\theta_i, \alpha) \prod_{j=1}^{N} \mathrm{P}(z_{ij}|\theta_i)\mathrm{P}(w_{ij}|\varphi_{z_{ij}}) \tag{10}$$

Wrocław University of Science and Technology

Probability of document

- Integrating over θ and summing over Z, we obtain the marginal distribution of a document

$$P(w|\alpha, \beta) = \int P(\theta|\alpha) \left(\prod_{j=1}^{N} \sum_{z_j} P(z_j|\theta) P(w_j|z_j, \beta) \right) d\theta \tag{11}$$

- taking the product of the marginal probabilities of single documents, we obtain the probability of a corpus D (set of documents)

$$P(\mathrm{D}|\alpha, \beta) = \prod_{i=1}^{M} \int P(\theta_i|\alpha) \left(\prod_{j=1}^{N_i} \sum_{z_{ij}} P(z_{ij}|\theta) P(w_{ij}|z_{ij}, \beta) \right) d\theta_i \tag{12}$$

Classical estimation in LDA

Classically: Gibbs sampling - estimates the topic assignments for each of words

Initialize $x^{(0)} \sim q(x)$

for iteration $i = 1, 2, \ldots$ **do**

$$x_1^{(i)} \sim p(X_1 = x_1 | X_2 = x_2^{(i-1)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$$

$$x_2^{(i)} \sim p(X_2 = x_2 | X_1 = x_1^{(i)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$$

$$\vdots$$

$$x_D^{(i)} \sim p(X_D = x_D | X_1 = x_1^{(i)}, X_2 = x_2^{(i)}, \ldots, X_D = x_{D-1}^{(i)})$$

end for

Figure: Gibbs sampler. Credits [1]

Wrocław University of Science and Technology

Table of Contents

Markov Random Fields - recap

Conditinal Random Fields

Latent Dirichlet Allocation

Approximate Inference in PGMs
- Introduction
- Loopy Belief Propagation

Table of Contents

Markov Random Fields - recap

Conditinal Random Fields

Latent Dirichlet Allocation

Approximate Inference in PGMs

- Introduction
- Loopy Belief Propagation

Approximate Inference in PGMs

- we already know *belief propagation*, also known as *sum-product message passing*
- it was originally designed for acyclic graphical models
- however the algorithm can be used in general graphs

Loopy belief propagation

initialization and scheduling of message updates are **slightly adjusted**, because graphs might not contain any leaves in comparison to *belief propagation*

Initiallization and passing:

- all variable messages initialize to 1
- updates all messages at every iteration
- messages coming from known leaves or tree-structured subgraphs may no longer need updating after sufficient iterations
- not well understood the precise conditions under which loopy belief propagation will converge

There are other approximate methods for marginalization including variational methods and Monte Carlo methods.

Table of Contents

Markov Random Fields - recap

Conditinal Random Fields

Latent Dirichlet Allocation

Approximate Inference in PGMs

- Introduction
- Loopy Belief Propagation

Loopy belief propagation

The basic idea is extremely simple: we apply the *belief propagation* algorithm to the graph, even if it has loops (i.e., even if it is not a tree).

Algorithm 22.1: Loopy belief propagation for a pairwise MRF

Input: node potentials $\psi_s(x_s)$, edge potentials $\psi_{st}(x_s, x_t)$;
Initialize messages $m_{s \to t}(x_t) = 1$ for all edges $s - t$;
Initialize beliefs $\mathrm{bel}_s(x_s) = 1$ for all nodes s;
repeat
  Send message on each edge
  $m_{s \to t}(x_t) = \sum_{x_s} \left(\psi_s(x_s) \psi_{st}(x_s, x_t) \prod_{u \in \mathrm{nbr}_s \setminus t} m_{u \to s}(x_s) \right)$;
  Update belief of each node $\mathrm{bel}_s(x_s) \propto \psi_s(x_s) \prod_{t \in \mathrm{nbr}_s} m_{t \to s}(x_s)$;
until *beliefs don't change significantly*;
Return marginal beliefs $\mathrm{bel}_s(x_s)$;

Figure: credits: [1]

Bibliography I

[1] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] John Lafferty, Andrew McCallum, and Fernando CN Pereira. “Conditional random fields: Probabilistic models for segmenting and labeling sequence data”. In: (2001).