Data Science
Lecture 6 : Unsupervised Learning

Thierry Artières and QARMA people

Option DIGITALE - Ecole Centrale Marseille
Master IAAA

3 novembre 2019

1 Basics

2 KMeans

3 EM

4 Latent Variable models

Outline

1 Basics
2 KMeans
3 EM
4 Latent Variable models

Unsupervised learning

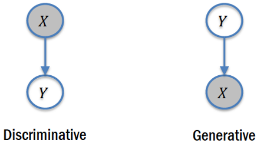
Setting

- You are given a training set of N samples $T = \{x^1, ..., x^N, \forall i, x^i \in \mathcal{X}\} \Rightarrow$ without any supevision
- The question is : what can you say about the data ?
- Useful setting : most often gathering labeled data is long / expensive / impossible

What to do

- Clustering : identify few typical samples
- Density estimation : learn $p(x)$ (e.g. fit a Gaussian distribution... or a more complex one)
- Identify factors of variations that explain the data (at a finer level than clustering)

Unsupervised learning



Clustering : An ill posed problem ?

- Somehow the methods are designed to learn some specific something, so will learn it...
- Hard to evaluate the goodness of a solution !

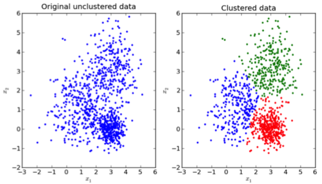
Outline

1 Basics
2 KMeans
3 EM
4 Latent Variable models

Clustering

Objectif

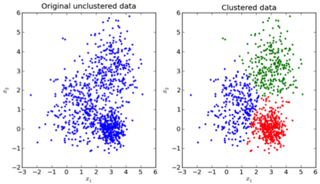
- Find similarities between data i.e. cluster, group
- What for ?
  - Identify typical user profiles ⇒ allow personalization
  - From continuous to discrete (quantization) simplifies model learning, relax hypothesis etc
- But : it is a combinatorial problem
  - 14 samples in 4 categories ⇒ 10 millions different clusterings.
  - Clustering algorithmes rely on hypothesis on the data distribution



Clustering

Hypothesis based algorithms

- For instance : A cluster has an isotropic distribution
- $\Rightarrow$ Implementation : Cluster samples by their distance wrt. a limited set of centers of clusters partitions (a codebook)
- The codebook defines the clustering : A sample belongs to a cluster according to its nearest neighbor in the codebook
- Quality criterion : $\sum_{i=1}^{N} \|x^i - p(x^i)\|^2$
  - where $p(x^i)$ is element in the codebook which is the closest to x^i

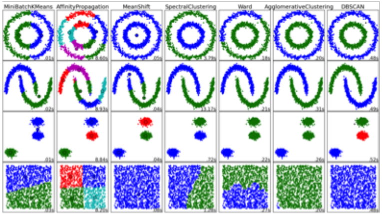


This may work well !

Clustering

Hypothesis based algorithms

- For instance : A cluster has an isotropic distribution
- $\Rightarrow$ Implementation : Cluster samples by their distance wrt. a limited set of centers of clusters partitions (a codebook)
- The codebook defines the clustering : A sample belongs to a cluster according to its nearest neighbor in the codebook
- Quality criterion : $\sum_{i=1}^{N} \|x^i - p(x^i)\|^2$
  - where $p(x^i)$ is element in the codebook which is the closest to x^i



But not always

Clustering

Hypothesis based algorithms

- For instance : A cluster has an isotropic distribution
- $\Rightarrow$ Implementation : Cluster samples by their distance wrt. a limited set of centers of clusters partitions (a codebook)
- The codebook defines the clustering : A sample belongs to a cluster according to its nearest neighbor in the codebook
- Quality criterion : $\sum_{i=1}^{N} \|x^i - p(x^i)\|^2$
  - where $p(x^i)$ is element in the codebook which is the closest to x^i

Other hypothesis lead to different algorithms and quality criterion

- In unsupervised learning you have to give something $\Rightarrow$ you need an hypothesis from which you may derive an algorithm
- Alternative hypothesis : Two samples that are close are likely to belong to the same cluster
  - $\Rightarrow$ Propagation methods
- Alternative hypothesis : Frontiers between clusters lie in low density areas
- ...

KMeans : a best seller

Algorithm 1 KMeans

Input: Codebook size K
Input: Dataset T of N samples in $\mathcal{X}$

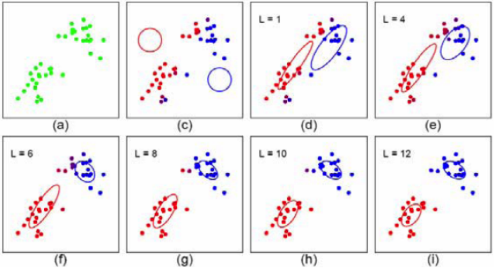
Initialize a codebook of K elements in $\mathcal{X}$
repeat
    Determine clusters for all x^i in T
    Define codebook as centers of subsets of samples from T in each cluster
until Stop Criteria satisfied
return Codebook

Kmeans Algorithm

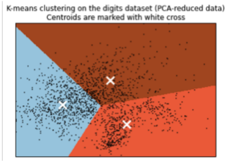
- Isotropic cluster distribution assumption
- Size K is set by hand
  - Hierarchical (ascendent or descent) algorithms
- Requires a good initialization (convergence to a local minima)
- Dependent on the distance used
- Many variants

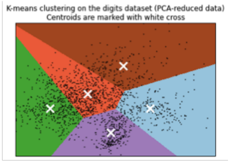
KMeans in action

Along iterations



Choosing K





Outline

1. Basics
2. KMeans
3. EM
4. Latent Variable models

KMeans as an instance of the EM algorithm

Algorithm 2 EM like algorithm

Input: Observed uncompletly observed dataset T of N samples in $\mathcal{X}$, $T = \{x^1, ...x^N\}$.
Input: There exists hidden (unobserved) information for every sample h^i, with $H = \{h^1, ...h^N\}$, such that learning the model with joint knowledge of T and H would be trivial.

1: Initialize model parameters W
2: **repeat**
3: Infer $\hat{H}$, a guess on H given T and W
4: Set W through learning from complete (approximated) data $(T, \hat{H})$
5: **until** Stop Criteria satisfied

Kmeans instance

- If the hidden variable (partition indicator) $h^i \in \{1, .., K\}$ was known for every $x^i \in \mathcal{X}$, learning the codebook would be trivial
- Every iteration one uses the learned parameters (the codebook) to infer a guess on H

KMeans as an instance of the EM algorithm

Standard (Soft) EM algorithm

- Actually there is uncertainty on the hidden variable
- Rather than hard assignment to e.g. the most likely $\hat{H}$, one infers the distribution on H on the E-step, and uses these distribution in the M-step

---

Algorithm 3 EM like algorithm

Input: Observed uncompletely observed dataset T of N samples in $\mathcal{X}$, $T = \{x^1, ...x^N\}$.
Input: There exists hidden (unobserved) information for every sample h^i, with $H = \{h^1, ...h^N\}$, such that learning the model with joint knowledge of T and H would be trivial.

1: Initialize model parameters W
2: **repeat**
3:     Infer a distributionq_H on H, given T and W
4:     Set W to the model through learning from data T and considering the distribution on Q_H
5: **until** Stop Criteria satisfied

Hard and soft KMeans algorithm

Algorithm 4 Hard KMeans

Input: Codebook size K
Input: Dataset T of N samples in $\mathcal{X}$
1: Initialize a codebook of K elements in $\mathcal{X}$
2: **repeat**
3:     (hard) Assign clusters for all x^i in T given codebook
4:     Define the new codebook as centers of subsets of samples from T in each cluster
5: **until** Stop Criteria satisfied

Soft version takes into account uncertainty on clusters assignment

Algorithm 5 Soft KMeans

Input: Codebook size K
Input: Dataset T of N samples in $\mathcal{X}$
1: Initialize a codebook of K elements in $\mathcal{X}$
2: **repeat**
3:     Assign scores $\alpha_{i,k}$ (e.g. likelihoods) for all x^i in T to belong to all clusters k
4:     Define codebook as weighted centers of subsets of samples from T in each cluster where weights are above scores (new center of cluster k : $c_k \propto \sum_i \alpha_{i,k} x^i$)
5: **until** Stop Criteria satisfied

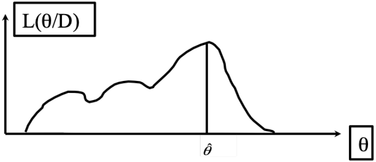
Maximum Likelihood Estimation (MLE)

Optimization criterion

- Based on a dataset $T = \{x^1, .., x^N\}$ infer the model that most likely produced the data
- Express the likelihood of the data, given the standard i.i.d. assumption

$$p(T|\theta) = p(x^1, .., x^N|\theta) = \prod_{i=1}^{N} p(x^i|\theta) = L(\theta|T)$$

- MLE principle : choose $\hat{\theta} = \arg\max_\theta L(\theta|T)$
- Solving requires setting the gradient to 0
  - Analytical solution
  - Iterative algorithms (gradient, EM, ...)



Coupling variables in optimization

The probabilistic view of KMeans : Gaussian mixture model

- Assume data have been generated by a Gaussian mixture (with K components)

  $p(x|\theta) = \sum_{k=1}^{K} p_i \times p(x|\theta_k)$ with : $p(x|\theta_k) = \frac{1}{\sqrt{2\Pi}\sigma_k} e^{-\frac{(x-\mu_k)^2}{\sigma_k^2}}$
- How to estimate all parameters $\{(p_k, \mu_k, \sigma_k), k = 1...K\}$?
- $\Rightarrow$ Expressing the log-likelihood (noted ll)
  $ll(\theta) = \sum_{i=1}^{N} \log p(x|\theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} p_i \times p(x|\theta_k)$
  - Not so easy to implement (entangled parameters) ! ! !
- While assuming knowledge of hidden variables h^i would yield (noting complete ll as cll)
  - $\Rightarrow$ Parameters are decoupled in separated terms and may be optimized independently

$$cll(\theta) = \log p(T, H|\theta) = \log p(T|H, \theta) + \log p(H|\theta)$$
$$= \sum_{i=1}^{N} \log p(x^i, h^i|\theta) = \sum_{i=1}^{N} \log p(x^i|h^i, \theta) + \sum_{i=1}^{N} \log p(h^i|\theta)$$

- EM algorithm brings a solution when hidden variables are missing

EM mechanics

Likelihood

- What we want to optimize : $ll(\theta) = \sum_i \log p(x^i|\theta) = \sum_i \log \sum_{h^i} p(x^i, h^i|\theta)$
- We may rewrite :

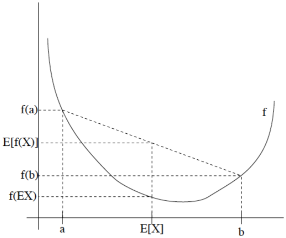
$$\log p(x|\theta) = \log \sum_h q(h|x) \frac{p(x,h|\theta)}{q(h|x)} \geq \sum_h q(h|x) \log \frac{p(x,h|\theta)}{q(h|x)}$$

$$\Rightarrow ll(\theta) \geq \sum_i \sum_{h^i} q(h^i|x^i) \log \frac{p(x^i, h^i|\theta)}{q(h^i|x^i)} \stackrel{\text{def}}{=} J(q,\theta)$$

- Because of Jensen inequality with f convex (reciproqual result for concave case)

$$\forall \lambda_j \geq 0 \text{ s.t. } \sum_j \lambda_j = 1, f(\sum_j \lambda_j x_j) \quad \leq \sum_j \lambda_j f(x_j)$$

$$f\,[E_x[x]] \qquad \leq E_x[f(x)]$$



EM mechanics

Likelihood

- Rewriting :

$$J(q,\theta) = \sum_i \sum_{h^i} q(h^i|x^i) \log \frac{p(x^i, h^i|\theta)}{q(h^i|x^i)} \leq ll(\theta)$$

- Then $J(q,\theta)$ is a lower bound of $ll(\theta)$
- We are looking for the tightest lower bound of $ll(\theta)$
  - Actually we may find q such that $J(q,\theta) = ll(\theta)$
  - Choosing $q(h^i|x^i) = p(h^i|x^i,\theta) = \frac{p(h^i|x^i,\theta)}{p(x^i|\theta)}$

$$\Rightarrow \sum_{h^i} q(h^i|x^i) \log \frac{p(x^i, h^i|\theta)}{q(h^i|x^i)} = \sum_{h^i} p(h^i|x^i,\theta) \log \frac{p(x^i, h^i|\theta)}{p(h^i|x^i,\theta)}$$

$$= \log p(x^i|\theta) \sum_{h^i} q(h^i|x^i,\theta) = \log p(x^i|\theta)$$

  - Hence with $q_{posterior} = p(h^i|x^i,\theta)$: $J(q_{posterior},\theta) = ll(\theta)$

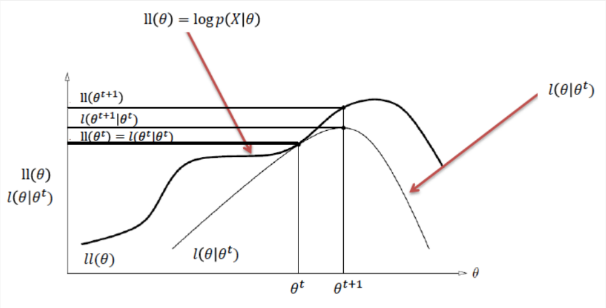
EM mechanics

EM (iterative) algorithm

- At iteration t, we note current parameters θ_t :
- We note $l(\theta, \theta_t) = J(q_{posterior_t}, \theta)$
  - with $q_{posterior_t}$: $q(h^i|x^i) = p(h^i|x^i, \theta_t)$

Wrapping it up

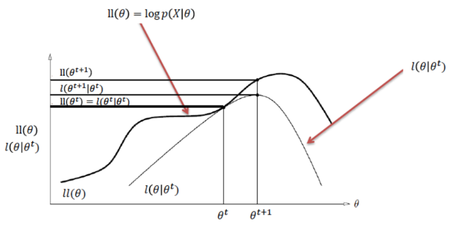
- $l(\theta|\theta_t) \leq ll(\theta)$
- $l(\theta_t|\theta_t) = ll(\theta_t)$
- $\Rightarrow ll(\theta_t) \leq \max_\theta l(\theta|\theta_t) \leq ll(\theta)$



EM Algorithm

Algorithm 6 EM

1: Initialize parameters θ^0
2: **repeat**
3: E-Step : Compute $q^{t+1} = q_{posterior}$
4: M-Step : $\theta^{t+1} = \arg\max_\theta l(\theta, \theta_t)$
5: **until** Convergence



EM mechanics

Auxiliary function

- At iteration t, we actually look to maximize $Q(\theta, \theta_t)$, with

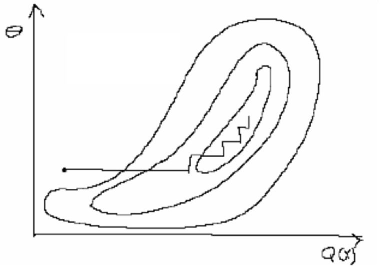
$$Q(\theta, \theta_t) = \sum_i \left[\sum_{h^i} p(h^i | x^i, \theta_t) \log p(x^i, h^i | \theta) \right]$$

- Same maximization problem as $l(\theta, \theta_t)$ since $l(\theta, \theta_t) = Q(\theta, \theta_t) + H[q_{Posterior_t}]$

EM Algorithm : alternative view

Algorithm 7 EM

Initialize parameters θ^0
repeat
    E-Step : $q^{t+1} = \arg\max_q J(q, \theta^t)$
    M-Step : $\theta^{t+1} = \arg\max_\theta J(q^{t+1}, \theta)$
until Convergence



EM variants

Main variants to account for specific settings

- Classifying EM (CEM)
  - Hard decision in the E-Step : e.g. KMeans
  - Does converge but not to the same solution
- Generalized EM
  - Maximizing $I(\theta|\theta_t)$ might no be so easy
  - Increasing $I(\theta|\theta_t)$ over $I(\theta_t)$ is enough for convergence proofs
- Variational EM
  - Approximate the $I(\theta|\theta_t)$ because it might too hard to optimize

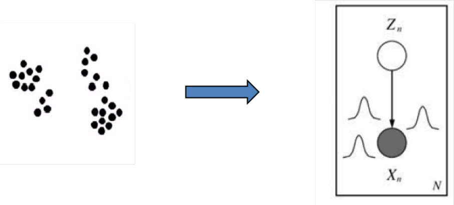
Outline

1. Basics
2. KMeans
3. EM
4. Latent Variable models

Latent variable models

Kmeans is a latent variable model :

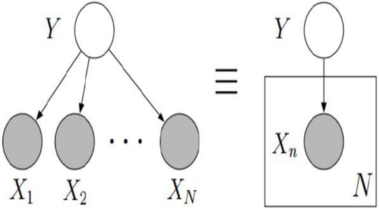
- It assumes a latent factor explaining the data and aims at learning it



Latent variable models

Plate Notation for Graphical models

- Nodes are random variables
- Edges denote possible dependence
- Observed variables are shaded
- Plates denote replicated structure



Latent variable models

A series of models for explaining text data (and more)

- Going further one may try to discover many (more or less independent) latent variables fro the data. Many applications
  - Author expertise and Topics discovery from collaborative scientific papers
  - Exploring the Enron case
  - ...
- Many models
  - Probabilistic Latent Semantic Analysis (PLSA)
  - Latent Dirichlet Allocation (LDA)
  - Author Topic Models
  - ...

At the begining : the Unigram model

Basic model

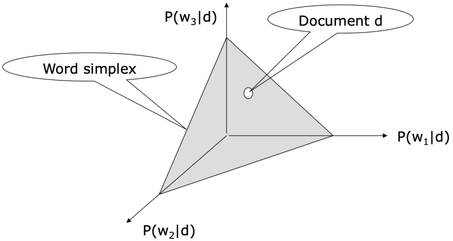
- To learn densities over text documents (combinatorial problem), assumptions are required
- Strong assumption : All words in a document are independent ! A document is a distribution over the dictionnary
- Leads to simple models that are easily learnable but not much useful

$$p(W|d\theta) = \prod_{i=1\dots N_d} p(w_i|\theta)$$

- where W stands for the words in d, and d is seen as a distribution over documents !

Generation process

- Choose (sample) a length N_d
- Repeat N_d times
  - Generate a word with $p(.|\theta)$ defined on the vocabulary



PLSA

Introducing latent variables for topics

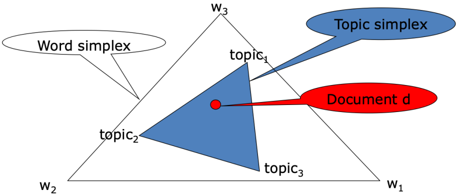
- A document is a distribution on topics
- A topic is a distribution on words of the vocabulary
- Leads to more useful models

$$p(d|\theta) = \prod_{i=1\dots N_d} \sum_z p(z|d) \times p(w_i|\theta_z)$$

- Application : Given a set of texts learn (in an unsupervised way) simultaneously the various topics the set of documents deal with, and the topics that are discussed in each of the document.

Generation process

- Choose (sample) a length N_d
- Repeat N_d times
  - Generate a topic with distribution $p(z|d)$
  - Generate a word with topic z, according to distribution $p(.|\theta_z)$



PLSA

Introducing latent variables for topics

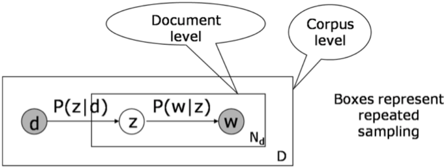
- A document is a distribution on topics
- A topic is a distribution on words of the vocabulary
- Leads to more useful models

$$p(d|\theta) = \prod_{i=1\ldots N_d} \sum_z p(z|d) \times p(w_i|\theta_z)$$

- Application : Given a set of texts learn (in an unsupervised way) simultaneously the various topics the set of documents deal with, and the topics that are discussed in each of the document.

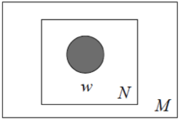
Generation process

- Choose (sample) a length N_d
- Repeat N_d times
  - Generate a topic with distribution $p(z|d)$
  - Generate a word with topic z, according to distribution $p(.|\theta_z)$

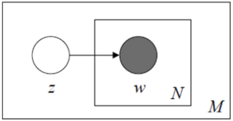


Comparing V.L. models

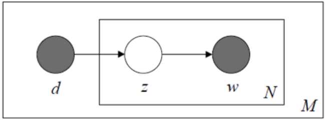
In terms of assumptions on the generation process



(a) unigram



(b) mixture of unigrams

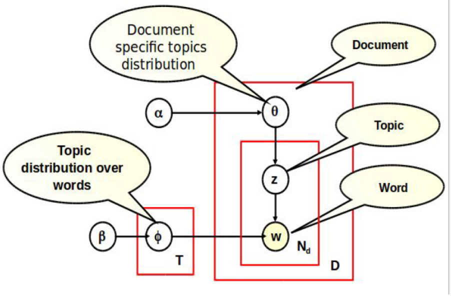


(c) pLSI/aspect model

LDA

Extending PLSA

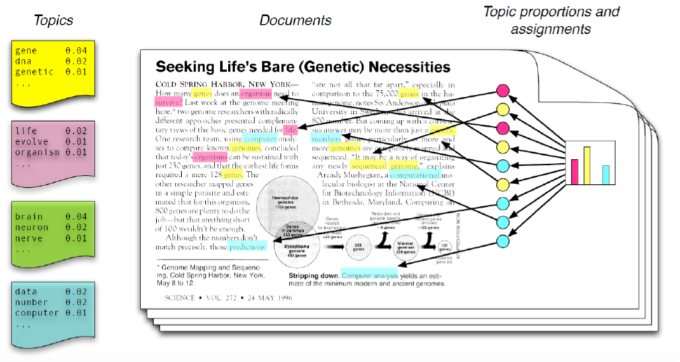
- PLSA is not actually a generative model of documents
- LDA extends PLSA by adding priors on distributions
- For convenience, we use Dirichlet priors (with parameter α and β), because posteriors may be computed analytically.
- Learning (for LDA and hereafter models) : EM or, preferred, Gibbs sampling



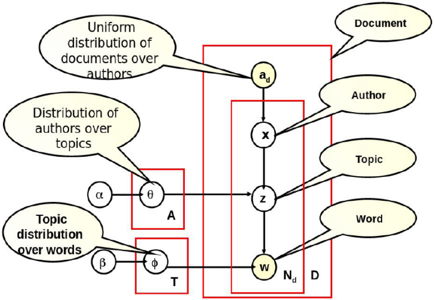
Author Topic Models

Main idea

- Encode in the model what we want to discover
- Main assumptions
  - Authors have dedicated topics where they are specialists
  - Documents may deal with various topics
  - In multiauthors documents each word is written by one of the author
  - A topic is a distribution on the words in the vocabulary

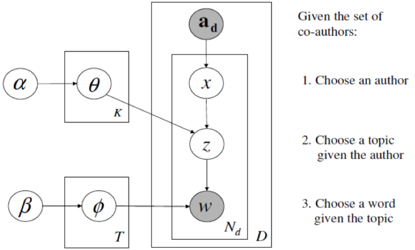


Model and generation process



Author Topic Models

Model and generation process



Author Topic Models on Citeseer

[AUTH1=Scholkopf_B (69%, 31%)]
[AUTH2=Darwiche_A (72%, 28%)]

A method[1] is described which like the kernel[1] trick[1] in support[1] vector[1] machines[1] SVMs[1] lets us generalize distance[1] based[2] algorithms to operate in feature[1] spaces usually nonlinearly related to the input[1] space This is done by identifying a class of kernels[1] which can be represented as norm[1] based[2] distances[1] in Hilbert spaces It turns[1] out that common kernel[1] algorithms such as SVMs[1] and kernel[1] PCA[1] are actually really distance[1] based[2] algorithms and can be run[2] with that class of kernels[1] too As well as providing[1] a useful new insight[1] into how these algorithms work the present[2] work can form the basis[1] for conceiving new algorithms

This paper presents[2] a comprehensive approach for model[2] based[2] diagnosis[2] which includes proposals for characterizing and computing[2] preferred[2] diagnoses[2] assuming that the system[2] description[2] is augmented with a system[2] structure[2] a directed[2] graph[2] explicating the interconnections between system[2] components[2] Specifically we first introduce the notion of a consequence[2] which is a syntactically[2] unconstrained propositional[2] sentence[2] that characterizes all consistency[2] based[2] diagnoses[2] and show[2] that standard[2] characterizations of diagnoses[2] such as minimal conflicts[1] correspond to syntactic[2] variations[1] on a consequence[2] Second we propose a new syntactic[2] variation on the consequence[2] known as negation[2] normal form NNF and discuss its merits compared to standard variations Third we introduce a basic algorithm[2] for computing consequences in NNF given a structured system[2] description We show that if the system[2] structure[2] does not contain cycles[2] then there is always a linear size[2] consequence[2] in NNF which can be computed in linear time[2] For arbitrary[1] system[2] structures[2] we show a precise connection between the complexity[2] of computing[2] consequences and the topology of the underlying system[2] structure[2] Finally we present[2] an algorithm[2] that enumerates[2] the preferred[2] diagnoses[2] characterized by a consequence[2] The algorithm[2] is shown[1] to take linear time[2] in the size[2] of the consequence[2] if the preference criterion[1] satisfies some general conditions

Author Topic Models on Citeseer

TOPIC 95	
WORD	**PROB.**
PATTERNS	0.1965
PATTERN	0.1821
MATCHING	0.1375
MATCH	0.0337
TEXT	0.0242
PRESENT	0.0207
MATCHES	0.0167
PAPER	0.0126
SHOW	0.0124
APPROACH	0.0099
AUTHOR	**PROB.**
Navarro_G	0.0133
Amir_A	0.0099
Gasieniec_L	0.0062
Baeza-Yates_R	0.0048
Baker_B	0.0042
Arikawa_S	0.0041
Crochemore_M	0.0037
Rytter_W	0.0034
Raffinot_M	0.0032
Ukkonen_E	0.0032

TOPIC 293	
WORD	**PROB.**
USER	0.3290
INTERFACE	0.1378
USERS	0.1060
INTERFACES	0.0498
SYSTEM	0.0434
INTERACTION	0.0296
INTERACTIVE	0.0214
USABILITY	0.0132
GRAPHICAL	0.0092
PROTOTYPE	0.0086
AUTHOR	**PROB.**
Shneiderman_B	0.0051
Rauterberg_M	0.0046
Harrison_M	0.0025
Winiwarter_W	0.0024
Ardissono_L	0.0021
Billsus_D	0.0019
Catarci_T	0.0017
St_R	0.0017
Picard_R	0.0016
Zukerman_I	0.0016

TOPIC 29	
WORD	**PROB.**
MAGNETIC	0.0155
STARS	0.0145
SOLAR	0.0135
EMISSION	0.0127
MASS	0.0125
OBSERVATIONS	0.0120
STAR	0.0118
RAY	0.0112
GALAXIES	0.0105
OBSERVED	0.0098
AUTHOR	**PROB.**
Falcke_H	0.0140
Linsky_J	0.0082
Butler_R	0.0077
Knapp_G	0.0067
Bjorkman_K	0.0065
Kundu_M	0.0060
Christensen-D_J	0.0057
Mursula_K	0.0054
Cranmer_S	0.0051
Nagar_N	0.0050

TOPIC 58	
WORD	**PROB.**
METHODS	0.5319
METHOD	0.1403
TECHNIQUES	0.0442
DEVELOPED	0.0216
APPLIED	0.0162
BASED	0.0153
APPROACHES	0.0133
COMPARE	0.0113
PRACTICAL	0.0112
STANDARD	0.0102
AUTHOR	**PROB.**
Srinivasan_A	0.0018
Mooney_R	0.0018
Owren_B	0.0018
Warnow_T	0.0016
Fensel_D	0.0016
Godsill_S	0.0014
Saad_Y	0.0014
Hansen_J	0.0013
Zhang_Y	0.0013
Dietterich_T	0.0013

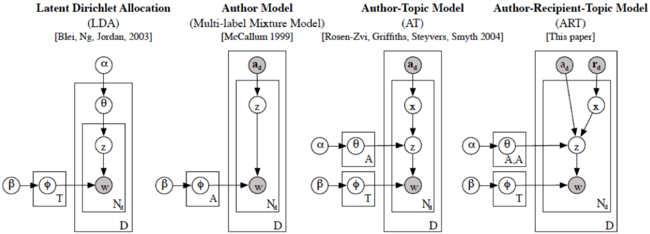
TOPIC 52	
WORD	**PROB.**
DATA	0.1622
MINING	0.0657
DISCOVERY	0.0408
ATTRIBUTES	0.0343

TOPIC 68	
WORD	**PROB.**
PROBABILISTIC	0.0869
BAYESIAN	0.0791
PROBABILITY	0.0740
MODEL	0.0533

TOPIC 298	
WORD	**PROB.**
RETRIEVAL	0.1208
INFORMATION	0.0613
TEXT	0.0461
DOCUMENTS	0.0385

TOPIC 139	
WORD	**PROB.**
QUERY	0.1406
QUERIES	0.0947
DATABASE	0.0932
DATABASES	0.0468

Beyond author Topic Models



Results on the Enron dataset

Topic 5 "Legal Contracts"		Topic 17 "Document Review"		Topic 27 "Time Scheduling"		Topic 45 "Sports Pool"	
section	0.0299	attached	0.0742	day	0.0419	game	0.0170
party	0.0265	agreement	0.0493	friday	0.0418	draft	0.0156
language	0.0226	review	0.0340	morning	0.0369	week	0.0135
contract	0.0203	questions	0.0257	monday	0.0282	team	0.0135
date	0.0155	draft	0.0245	office	0.0282	eric	0.0130
enron	0.0151	letter	0.0239	wednesday	0.0267	make	0.0125
parties	0.0149	comments	0.0207	tuesday	0.0261	free	0.0107
notice	0.0126	copy	0.0165	time	0.0218	year	0.0106
days	0.0112	revised	0.0161	good	0.0214	pick	0.0097
include	0.0111	document	0.0156	thursday	0.0191	phillip	0.0095
M.Hain	0.0549	G.Nemec	0.0737	J.Dasovich	0.0340	E.Bass	0.3050
J.Steffes		B.Tycholiz		R.Shapiro		M.Lenhart	
J.Dasovich	0.0377	G.Nemec	0.0551	J.Dasovich	0.0289	E.Bass	0.0780
R.Shapiro		M.Whitt		J.Steffes		P.Love	
D.Hyvl	0.0362	B.Tycholiz	0.0325	C.Clair	0.0175	M.Motley	0.0522
K.Ward		G.Nemec		M.Taylor		M.Grigsby	

Topic 34 "Operations"		Topic 37 "Power Market"		Topic 41 "Government Relations"		Topic 42 "Wireless"	
operations	0.0321	market	0.0567	state	0.0404	blackberry	0.0726
team	0.0234	power	0.0563	california	0.0367	net	0.0557
office	0.0173	price	0.0280	power	0.0337	www	0.0409
list	0.0144	system	0.0206	energy	0.0239	website	0.0375
bob	0.0129	prices	0.0182	electricity	0.0203	report	0.0373
open	0.0126	high	0.0124	davis	0.0183	wireless	0.0364
meeting	0.0107	based	0.0120	utilities	0.0158	handheld	0.0362
gas	0.0107	buy	0.0117	commission	0.0136	stan	0.0282
business	0.0106	customers	0.0110	governor	0.0132	fyi	0.0271
houston	0.0099	costs	0.0106	prices	0.0089	named	0.0260
S.Beck	0.2158	J.Dasovich	0.1231	J.Dasovich	0.3338	R.Haylett	0.1432
L.Kitchen		J.Steffes		R.Shapiro		T.Geaccone	
S.Beck	0.0826	J.Dasovich	0.1133	J.Dasovich	0.2440	T.Geaccone	0.0737
J.Lavorato		R.Shapiro		J.Steffes		R.Haylett	
S.Beck	0.0530	M.Taylor	0.0218	J.Dasovich	0.1394	R.Haylett	0.0420
S.White		E.Sager		R.Sanders		D.Fossum	