A Generic Approach to Topic Models

Gregor Heinrich

Fraunhofer IGD + University of Leipzig
Darmstadt, Germany
heinrich@igd.fraunhofer.de

Abstract. This article contributes a generic model of topic models. To define the problem space, general characteristics for this class of models are derived, which give rise to a representation of topic models as "mixture networks", a domain-specific compact alternative to Bayesian networks. Besides illustrating the interconnection of mixtures in topic models, the benefit of this representation is its straight-forward mapping to inference equations and algorithms, which is shown with the derivation and implementation of a generic Gibbs sampling algorithm.

1 Introduction

Mixture models [1] are a powerful tool to model complex probabilistic distributions by convex sums of component densities, $p(x) = \sum_k p(z{=}k)p(x|\vartheta_k)$, where z is an index variable that indicates which component k the observation x originates from. Among the large class of such models, mixture models with discrete component densities $p(x|\vartheta_k)$ are of particular interest because in this case the component densities can serve as weighting functions for other mixtures, which themselves can have again discrete or non-discrete component densities. This fact makes it possible to construct models that consist of cascades or even networks of coupled discrete mixtures as generative structure underlying one or more observable mixtures with arbitrary (e.g., discrete or Gaussian) component densities.

Such a coupling of mixtures can be considered a defining characteristic of topic models, a class of probabilistic models that has become a central subject of research in text mining, computer vision, bioinformatics and other fields. Following the idea proposed by the seminal work on latent Dirichlet allocation (LDA [2]), topic models exploit the conjugacy of Dirichlet and multinomial/discrete distributions to learn discrete latent variables from discrete co-occurrence data (e.g., [3,4,5]) or from the co-occurrence of discrete and continuous features (e.g., [6]). Via the interrelation of the latent variables across different mixture levels, structures assumed in the data can be accounted for in specialised topic models, which renders the topic model approach a powerful and flexible framework.

However, the published work on topic models only defines this framework implicitly; authors tend to analyse and derive probabilistic properties and inference algorithms on a model-specific basis, typically using results from particular prior work. Although on the other hand frameworks for automatic inference in (more general) Bayesian networks exist that are in principle capable of handling topic models as special cases (e.g., WinBUGS [7], HBC [8], AutoBayes [9], or VIBES [10]), the generality of this software

makes it difficult (1) to gain insights from the result of the automatic inference derivation process, and (2) to make performance improvements that may be possible for more restricted model structures, which is desirable especially because topic models have serious scalability issues. Such improvements may be based on the recent advances in massively parallel hardware, along with general-purpose programming platforms like OpenCL [11], or heterogeneous computing architectures including specialised FPGA processor designs, along with programming interfaces like the hArtes toolchain [12]. For such high-performance computing architectures, a generic approach to topic model inference may permit to reuse highly optimised kernels across models and therefore allow to focus optimisation effort.

Apart from theoretical interest, these practical considerations motivate a closer look on topic models with the intent to characterise their properties in a generic manner. Specifically, we generalise the probabilistic properties of topic models in Sec. 2. Motivated by this general characterisation, we propose a specialised representation of topic models in Sec. 3: mixture networks. Subsequently, as a basis for actual implementations we present a generic approach to inference in mixture networks in Sec. 4 for the case of Gibbs sampling, which has been implemented as a generic Gibbs sampling tool described in Sec. 5. We finish with conclusions and future work directions in Sec. 6.

2 Generalising topic models

In this section, we present a generic characterisation to topic models. As a basis for the following derivations, consider an arbitrary Bayesian network (BN [13]) with variables $U_n \in U$. Its likelihood can be generally formulated as:

$$p(U) = \prod_n p(U_n \,|\, \mathrm{pa}(U_n)) \tag{1}$$

where the operator $\mathrm{pa}(U_n)$ refers to the set of parents of some BN node that belongs to variable U_n.

Characteristics. As has been outlined in the Introduction, the first notable characteristic of topic models is their use of the conjugate Dirichlet and multinomial/discrete distributions. Focussing on discrete observations, such models can be structured entirely into "mixture levels", each of which consists of a set of multinomial components $\vec{\vartheta}_k \in \Theta \triangleq \{\vec{\vartheta}_k\}_{k=1}^K$ that are themselves drawn from Dirichlet priors with some set of hyperparameters $\vec{\alpha}_j \in A \triangleq \{\vec{\alpha}_j\}_{j=1}^J$. Based on one or more discrete values from parent nodes in the BN, a component k among the multinomial mixture is chosen and a discrete value x_i sampled from it, which is part of the observation sequence $X \triangleq \{x_i\}_{i \in I}$ with index sequence I. The corresponding generative process for one mixture level can be summarised as:

$$\begin{aligned} x_i \,|\, \vec{\vartheta}_k, k{=}g(\uparrow x_i, i) &\sim \mathrm{Mult}(x_i|\Theta, \uparrow x_i) \\ \vec{\vartheta}_k \,|\, \vec{\alpha}_j, j{=}f(\uparrow X) &\sim \mathrm{Dir}(\vec{\vartheta}_k|A, \uparrow X) \end{aligned} \tag{2}$$

where the component index k is some function of the incoming discrete values or their indices that maps to components of the local mixture level. For this, the parent variable

operator $\uparrow x_i$ is introduced that collects all parent variables of x_i (excluding parameters: $\uparrow X = \mathrm{pa}(X) \backslash \Theta$), and the component selection function can consequently be expressed as $k = g(\uparrow x_i, i)$. Hyperparameter indices j can be chosen either to be global for all k, i.e., $j \equiv 1$, or similarly to the component indices, assigned to a group of components with some grouping function $j = f(\uparrow X)$. This grouping can be used to model clustering among components (see, e.g., [14,4]).

The generative process in Eq. 2 reveals the second characteristic of topic models: Mixture levels are solely connected via discrete parent variables ($\uparrow X$), which ensures a simple form of the joint likelihood of the model.[1] Based on Eq. 1, the complete topic model can be constructed from the mixture levels $\ell \in L$, yielding the likelihood:

$$p(X, \Theta | A) = \prod_{\ell \in L} p(X^\ell, \Theta^\ell \,|\, A^\ell; \uparrow X^\ell) \tag{3}$$

$$= \prod_{\ell \in L} \left[\prod_{i \in I} \mathrm{Mult}(x_i \,|\, \Theta, \uparrow x_i) \prod_{k=1}^{K} \mathrm{Dir}(\vec{\vartheta}_k \,|\, A, \uparrow X) \right]^{[\ell]} \tag{4}$$

where for simplicity we mark up variables specific to a level with a superscript ℓ, in brackets $[\cdot]^{[\ell]}$ for all their contents or for entire equations in the text. Without this mark-up, symbols are assumed model-wide sets of variables X, parameters Θ, hyperparameters A, etc. Eq. 4 shows the structure of the joint likelihood common for topic models: Multinomial observations factorise over the sequence of tokens generated by the model, and the Dirichlet priors factorise between the components.

Mixture level likelihood. Due to the conjugacy of the Dirichlet and multinomial/discrete distributions, the inner terms of Eq. 4 can be simplified further after a transformation from tokens with index $i \in I$ (part of a sequence) to counts over component dimensions with index over $t \in [1, T]$ (part of a "vocabulary"), each specific to a mixture level ℓ. For every ℓ, the following holds for the total count of co-occurrences between outcomes $x_i{=}t$ and mixture components $k{=}g(\uparrow x_i, i)$ responsible for them:

$$n_{k,t} = \sum_{i \in I} \delta(k - g(\uparrow x_i, i))\, \delta(t - x_i) \tag{5}$$

where $\delta(x)$ is the delta function, $\delta(x) = \{1 \text{ if } x{=}0, 0 \text{ otherwise}\}$. Using these counts, the likelihood of one mixture level becomes:

$$p(X^\ell, \Theta^\ell \,|\, A^\ell; \uparrow X^\ell) = \left[\prod_{k=1}^{K} \frac{1}{\Delta(\vec{\alpha}_j)} \prod_{t=1}^{T} \vartheta_{k,t}^{n_{k,t} + \alpha_{j,t} - 1} \right]^{[\ell]} \tag{6}$$

where the product over t in Eq. 6 is the integrand of a Dirichlet integral and $\Delta(\vec{\alpha})$ is the partition function of the Dirichlet distribution, a T-dimensional generalisation of the beta function:

$$\Delta(\vec{a}) \triangleq \frac{\prod_{t=1}^{T} \Gamma(a_t)}{\Gamma(\sum_{t=1}^{T} a_t)} \,. \tag{7}$$

[1] With the hyperparameters dependent on $j = f(\cdot)$, formally there is an additional dependency between mixture levels, but this is dropped by assuming the set A known; common EM-type inference methods estimate hyperparameters independently inside their M-step; see Sec. 4.

Mixture level variants. The topic model framework is not restricted to the Dirichlet–multinomial type of mixture level that forms its core. Several possibilities exist to extend the framework and plug as levels into Eq. 3:

- *Symmetric hyperparameters* are a common variant to the standard vectors, see, e.g., the original LDA model [2]. The Dirichlet partition function simplifies to $\Delta_T(a) \triangleq \Gamma(a)^T / \Gamma(Ta)$.
- *Observed parameters* introduce known mixture proportions or fixed observations like labels (see, e.g., the author–topic model [3]), which leads to $p(X|\Theta; \uparrow X) = \prod_i \mathrm{Mult}(x_i|\Theta, \uparrow x_i) = \prod_{k,t} \vartheta_{k,t}^{n_{k,t}}$ for a level, i.e., the Dirichlet vanishes in Eq. 3.
- *Infinite mixtures* allow model adaptation to data dimensionalities and typically use Dirichlet process (DP) mixtures or generalisations [15]. Mixture interrelations in topic models are handled typically using hierarichcal DPs [16]. Especially the stick-breaking representation of the DP and its finite approximations [17,18] promise to preserve a high similarity to finite-dimensional models.
- *Non-Dirichlet priors* like logistic–normal allow a more flexible combination of topics in particular mixture levels, as in the correlated topic model [19].
- *Non-discrete observation components* use, e.g., Gaussian distributions in the final mixture level (e.g., in Corr-LDA [6]). Parameters are preferrably drawn from conjugate priors to simplify inference. The likelihood of a non-discrete mixture level is $p(\tilde{X}, \tilde{\Theta}|\tilde{A}; \uparrow\tilde{X}) = \prod_i h(\tilde{x}_i, \tilde{\Theta}| \uparrow\tilde{x}_i) \prod_k g(\tilde{\vartheta}_k|\tilde{\alpha}_j)$ with component distribution h and prior g.

To keep this paper focussed, we restrict ourselves to the first two variants mentioned, which already cover a vast body of models in the literature.

3 Mixture networks

The dependency structure characteristic for topic models shown in Eq. 3 gives rise to the idea of a specialised graphical representation. In addition to the fact that in BN diagrams of more complex topic models, interrelations between mixtures are easily hidden in complex network structures, the introduction of a domain-specific graphical representation of topic models may help simplify derivation of their likelihood structure and inference equations.

To obtain such a representation, we use the two characteristics discussed in Sec. 2: Dirichlet–multinomial mixture levels and the connections between levels via discrete variables, which can be seen as nodes and edges in a new network structure. This leads to a representation of topic models as "mixture networks".

3.1 Definition

A mixture network (MN) is defined as a digraph $G(\mathcal{N}, \mathcal{E})$ that consists of (1) a set of nodes, $\mathcal{N}$, where (a) an inner node represents a mixture level as described in Sec. 2, i.e., a sampling operation from a mixture component and (b) a terminal node represents an observable (discrete) value, as well as (2) a set of directed edges, $\mathcal{E} : \mathcal{N} \times \mathcal{N}$, where an

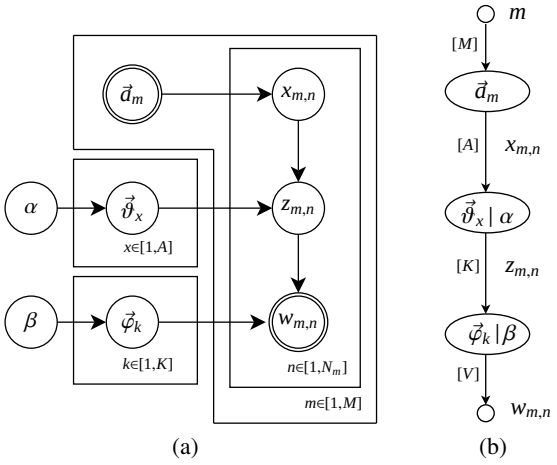

Fig. 1. The author–topic model, (a) Bayesian network and (b) mixture network.

edge propagates a discrete value from its parent node to its child node. The child node then uses the value to choose one of its components.

Graphical notation. A graphical notation for mixture networks is proposed in Fig. 1(b) via the example of the author–topic model (ATM [3]), which models the topic association with authors with three mixture levels and whose BN is shown in Fig. 1(a). Opposed to BNs that visualise dependencies between random variables and express the repetitions of data points (plate notation), MNs focus on the interrelations between discrete mixtures (in the example: document–author, $\vec{a}_m$, author–topic, $\vec{\vartheta}_x|\alpha$, and topic–word, $\vec{\varphi}_k|\beta$, distributions), along with component numbers and dimensionalities ($[M]$, $[K]$ and $[V]$). The frequency of sampling a particular variable is encoded in subscripts (in the example: $\vec{\vartheta}_x$, $\vec{\varphi}_k$ and $w_{m,n}$ referring to author-, topic- and word-wise sampling schedules, respectively). Note that the top (inner) node does not indicate a hyperparameter because of its observed parameter (see mixture level variants in Sec. 2).

3.2 Example models

In illustrate the applicability of the MN representation, the mixture network diagrams of some topic models from the literature are drawn in Fig. 2. Fig. 2(a) shows the MN of the model of latent Dirichlet allocation (LDA [2]), which served as a design paragon to all other models. It has two mixture levels: a document–topic mixture $\vec{\vartheta}_m|\alpha$ and a topic–term mixture $\vec{\varphi}_k|\beta$. The extension of this model gives illustrative insight into the organisation of mixtures to account for logical and semantic structure assumed in the data. A simple extension to the plain LDA model is the author–topic model shown in Fig. 2(b) as explained above.

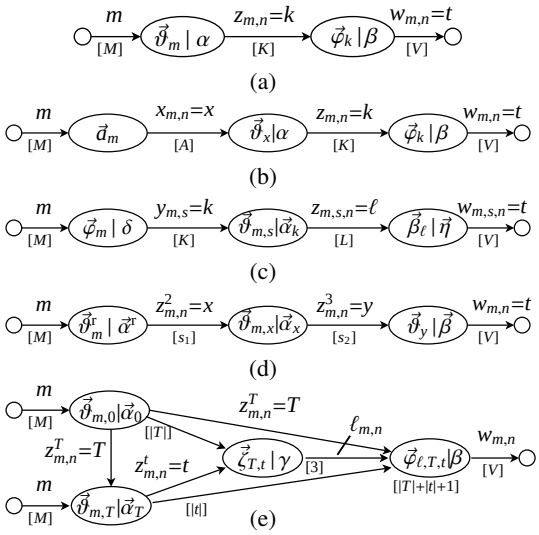

Fig. 2. Mixture networks of example models from the literature: (a) latent Dirichlet allocation, (b) author–topic model, (c) latent Dirichlet co-clustering model, (d) 4-level pachinko allocation, (e) hierarchical pachinko allocation (hPAM1).

Co-clustering. The model of latent Dirichlet co-clustering (LDCC [14]) in Fig. 2(c) uses aggregation to infer an additional logical layer of topics from the data: For each section s, a topic distribution $\vec{\vartheta}_{m,s}$ can be inferred, and document topic distributions $\vec{\varphi}_m$ index word-topics $z_{m,s,n}$ indirectly via section topics $y_{m,s}$, allowing a finer-grained handling of topic structure across documents with component selection function $g(\uparrow z_{m,s,n}, (m, s, n)) = (m, s)$. Further, segment topics $\vec{\vartheta}_{m,s}$ are coupled across documents via the topic-hyperparameters $\vec{\alpha}_y$ with component group function $f(\uparrow z) = k$.

Pachinko allocation. Another multi-level MN is the class of pachinko allocation models (PAM), of which the four-level variant as described in [4] is depicted in Fig. 2(d). For each word, a path through a topic hierarchy is sampled consisting of the indicators (z^1, z^2, z^3) where z^1=1 provides the root of the tree, associated with LDA-type document topics $\vec{\vartheta}^r_m$, and based on its sample, a document- and topic-dependent level $\vec{\vartheta}_{m,x}$ is sampled ($g(\uparrow z^3_{m,n}, (m, n)) = (m, x)$), finally indexing word-topics $\vec{\vartheta}_y$. Similar to the LDCC model, component grouping is used: $f(\uparrow z^3) = x$.

Hierarchical pachinko allocation [5] (hPAM) as shown in Fig. 2(e) is an example for a more complex model that allows a hierarchy of topic–term distributions: As in PAM, the topic hierarchy consists of document-specific root- and super- as well as global subtopics, but each node k in the hierarchy is associated with a topic–term distribution $\vec{\varphi}_k$, and for each word $w_{m,n}$, a complete topic path (root–super–sub) is sampled along with a level $\ell_{m,n}$ from $\vec{\zeta}_{T,t}$ specific to super- and sub-topics (hPAM1 in [5]). The topic sample on level $\ell_{m,n}$ selects from the set $k = \{1, 1 + T, 1 + |T| + t\}$ the component $\vec{\varphi}_k$ that finally generates the word.

Although with a different goal in mind, the concept of pachinko allocation models is closely related to the approach pursued with mixture networks because it allows to connect different levels of mixtures with great flexibility. In fact, MNs can be considered a generalisation of PAMs that allows free interconnection of nodes in general DAG structures with different types of mixture levels (observed, unobserved parameters) and with observable variables (edges or parameters) at arbitrary points in the network. By appropriate choice of index transformations $g^\ell(\uparrow x_i^\ell, i^\ell)$, even the component-dependent subtrees mentioned as the most flexible version of the PAM concept [4] may be realised with mixture networks.

4 Inference in mixture networks

Inference in the context of mixture networks refers to finding the parameters Θ and hyperparameters A given the observations. With the model variables X divided into sets of visible (observed) and latent (hidden) variables, $X = \{V, H\}$, this is typically a two-part process of (1) Bayesian inference for the posterior distribution,

$$p(H, \Theta|V, A) = \frac{p(V, H, \Theta|A)}{p(V|A)} \,, \tag{8}$$

and (2) estimation of the hyperparameters, for which ML or MAP estimators are commonly sufficient because of the simpler search space.

As in many latent-variable models, determining the posterior Eq. 8 is generally intractable in mixture networks because of excessive dependencies between the latent variables H and parameters Θ in the marginal likelihood for the observations V in the denominator, $p(V|A) = \sum_H \int p(V, H, \Theta|A)\, \mathrm{d}\Theta$. To circumvent this intractability, approximate inference methods have been proposed, for topic models including mean-field variational Bayes [2], collapsed variational Bayes [20], expectation propagation [21] and collapsed Gibbs sampling [22].

For our purposes, a method is needed that has feasible complexity with reasonable accuracy even when it comes to modelling dependencies between variables. The full factorisation of variational mean-field distributions may be adverse for model fitting [23], and structured approaches become complicated quickly [10]. Expectation propagation on the other side has not been commonly used with more complex topic models. Thus, Gibbs sampling appears to be the most straight-forward method for a formulation of approximate inference for mixture networks.

4.1 Gibbs sampling

Gibbs sampling [24] is an approximative inference method particularly suited for models where the marginals of the posterior can be expressed in closed form, in particular for high-dimensional discrete models. As a Markov-chain Monte Carlo (MCMC) method, Gibbs sampling uses a Markov chain that upon convergence approximately generates samples according to the posterior distribution. By sampling one dimension of the posterior at a time, Gibbs sampling avoids computationally complex Metropolis-Hastings acceptance calculations. One step in the Gibbs sampling inference approach

thus corresponds to sampling dependent hidden variables h_i for each data token v_i from the full conditional distribution, $h_i \sim p(h_i|H_{\neg i}, V, \Theta, A)$, where $\cdot_{\neg i}$ refers to the complete set of tokens except i. Analogously, $\vec{\vartheta}_k$ must be sampled in such an approach [25].

With topic models, it has been shown, however, that collapsed approaches to Gibbs sampling, i.e., those that integrate out parameters Θ [26], lead to particularly good convergence behaviour [22] (which is attributed to the high independence of the remaining hidden variables). Therefore, the posterior considered for Gibbs sampling is $p(H|V, A) = \int p(H, \Theta|V, A)\,\mathrm{d}\Theta$. The Markov state of the Gibbs sampler then reduces to H, and the resulting mixture network inference approach can be considered a form of stochastic EM algorithm [27] that trains the latent variables H in its E-step and hyperparameters A in its M-step.

To sample from posteriors of collapsed MNs, for each independent latent variable H^ℓ (generic variables, complete sequence: upper case) with tokens $h_i^\ell \in H^\ell \triangleq \{h_{i'}^\ell\}_{i' \in I^\ell}$ (tokens: lower case; convention: $h_i^\ell \equiv h_{i^\ell}^\ell$ unless otherwise noted), a separate full conditional distribution $p(h_i^\ell|H_{\neg i}^\ell, H^{\neg \ell}, V, A)$ must be formulated for each token $h_i^\ell \in H^\ell$ with $\cdot^{\neg \ell}$ used analogous to $\cdot_{\neg i}$. Typically, however, several hidden variables are dependent and need to be drawn as a block. Therefore, with dependency groups denoted by H^d with $H^\ell \subseteq H^d \subseteq H$ as sequences of groups of dependent tokens h_i^d, the full conditionals sought are: $p(h_i^d|H_{\neg i}^d, H^{\neg d}, V, A)$ for each group d and each token $i = i^d$. Remember that subscripts refer to sequence indices and superscripts to levels. Further note that h_i^d is a configuration of hidden variables that corresponds to a unique combination of components k^ℓ and outputs t^ℓ of the mixture levels involved.

Derivation. To find the full conditional distributions, we start from the joint likelihood, Eq. 3, and for a collapsed approach integrate out its parameters via Dirichlet integrals:

$$\begin{aligned} p(V, H|A) &= \prod_{\ell \in L} \left[\int \prod_{k=1}^{K} \frac{1}{\Delta(\vec{\alpha}_j)} \prod_{t=1}^{T} \vartheta_{k,t}^{n_{k,t}+\alpha_{j,t}-1} \,\mathrm{d}\Theta \right]^{[\ell]} \\ &= \prod_{\ell \in L} \left[\prod_{k=1}^{K} \frac{\Delta(\vec{n}_k + \vec{\alpha}_j)}{\Delta(\vec{\alpha}_j)} \right]^{[\ell]} \end{aligned} \tag{9}$$

where the level-specific $\vec{n}_k$ are vectors of co-occurrence counts $n_{k,t}$.

This equation shows that the joint likelihood of the model variables is a product of ratios of Dirichlet partition functions for each component on each individual mixture level in the model. Interestingly, using the identity $\Gamma(a+n) = \Gamma(a) \prod_{c=0}^{n-1}(a+c)$ with real $a > 0$ and integer $n \geq 0$, we obtain a ratio of finite product sequences:

$$\frac{\Delta(\vec{a} + \vec{n})}{\Delta(\vec{a})} = \frac{\prod_{t=1}^{T} \prod_{c=0}^{n_t-1}(a_t + c)}{\prod_{c=0}^{[\sum_t n_t]-1}([\sum_{t=1}^{T} a_t] + c)}\,, \tag{10}$$

which for a unit difference in a single element u, $\Delta(\vec{a}+\delta(t\text{–}u))/\Delta(\vec{a})$, reduces to $a_u / \sum_t a_t$. Note that with Eq. 10, we can alternatively expand Eq. 9 into products without any special functions, which comes at the cost of obtaining denominator terms in Eq. 9 specific to components k.

The next step to obtain full conditionals is to determine dependent edges $H^d \subseteq H$: Analogous to the "Bayes ball" algorithm in Bayesian networks, in MNs we can identify

dependent hidden edges by finding subgraphs that extend through nodes whose component selection function $g(\uparrow x_i, i)$ contains the respective hidden edges. In the examples given in Sec. 2, ATM, PAM and hPAM models have dependent edges; LDCC does not because the hidden variables are connected via hyperparameters (assumed given in the full conditional). Further, edges of nodes adjacent to subgraph H^d but independent of H^d are collected in a set $S^d \subset \{V, H\}$ with token sets s_i^d. We use the notation $\cdot^{\neg s}$ to denote the exclusion of S^d. With these definitions, full conditional distributions can be derived generically by applying the chain rule:

$$\begin{aligned} p(h_i^d | H_{\neg i}^d, H^{\neg d}, V, A) &= \frac{p(h_i^d, s_i^d | H_{\neg i}^d, S_{\neg i}^d, H^{\neg d, \neg s}, V^{\neg s}, A)}{p(s_i^d | H_{\neg i}^d, S_{\neg i}^d, H^{\neg d, \neg s}, V^{\neg s}, A)} \\ &\propto p(h_i^d, s_i^d | H_{\neg i}^d, S_{\neg i}^d, H^{\neg d, \neg s}, V^{\neg s}, A) \\ &= \frac{p(H, V|A)}{p(H_{\neg i}^d, S_{\neg i}^d, H^{\neg d, \neg s}, V^{\neg s}|A)} \\ &= \prod_{\ell \in \{H^d, S^d\}} \left[\prod_{k=1}^{K} \frac{\Delta(\vec{n}_k + \vec{\alpha}_j)}{\Delta(\vec{n}_{k,\neg i^d} + \vec{\alpha}_j)} \right]^{[\ell]} . \end{aligned} \tag{11}$$

Generic full conditionals. In Eq. 11, all terms except those with a count difference between numerator and denominator cancel out. The remainder of terms can be simplified by applying Eq. 10 with $\vec{a} = \vec{n}^{\ell}_{k^\ell, \neg i^d} + \vec{\alpha}^{\ell}_j$, and the resulting full conditional becomes a product of the following form if all mixture levels $\in \{H^d, S^d\}$ exclude only a single token with $\neg i^d$:

$$p(h_i^d | H_{\neg i}^d, H^{\neg d}, V, A) \propto \prod_{\ell \in \{H^d, S^d\}} \left[\frac{n_{k,t,\neg i^d} + \alpha_{j,t}}{\sum_{t=1}^{T} n_{k,t,\neg i^d} + \alpha_{j,t}} \right]^{[\ell]} . \tag{12}$$

The factors in Eq. 12 can be interpreted as posterior means of Dirichlet distributions with hyperparameters $\vec{\alpha}_j$ and observation counts $\vec{n}_{k,\neg i^d}$, $\langle \mathrm{Dir}(\cdot | \vec{n}_{k,\neg i^d} + \vec{\alpha}_j) \rangle$ on level ℓ. Although this form of full conditional factors is prevalent in a majority of topic models, with the scope of models considered in this paper alternative forms are possible:

- If $[g(\uparrow x_i, i)]^{[\ell]}$ contains no hidden edges, the denominator can be omitted (e.g., nodes with m as only component index).
- If one index i^d at a mixture level input aggregates a whole sequence of i^ℓ at its output, $\neg i^d$ corresponds to more than one token in the factor denominator in Eq. 11 (e.g., in LDCC, section topics $y_{m,s}$ aggregate word topic sequences $\{z_{m,s,n}\}_n$), which yields a factor analogous to Eq. 10:

$$\left[\frac{\prod_{t=1}^{T} \prod_{c=0}^{n_{k,t}-1} (c + \alpha_{j,t})}{\prod_{c=0}^{[\sum_{t=1}^{T} n_{k,t}]-1} (c + \sum_{t=1}^{T} \alpha_{j,t})} \right]^{[\ell]} . \tag{13}$$

- Finally, mixture levels with observed parameters have components $\vec{\vartheta}_k$ as factors. In this case, few non-zero elements in $\vec{\vartheta}_k$ support sparse representations, while symmetric non-zero values cancel out.

4.2 Parameter estimation

Generally, estimation of parameters and hyperparameters is part an M-step dual to the Gibbs E-step in a stochastic EM procedure. It can be performed on a per-node basis in mixture networks.

Hyperparameters. In many topic models, hyperparameters are of decisive importance, e.g., to couple component groups or to model data dispersion. As there is no closed-form solution for estimation of Dirichlet parameters from count data, iterative or sampling-based approaches are commonly employed. Extending results from [28] yields the following fixed-point iterations for node-specific standard and symmetric Dirichlet distributions that result in maximum likelihood estimates:

$$\alpha_{j,t} \leftarrow \alpha_{j,t} \frac{\left(\sum_{\{k:f(k)=j\}} \Psi(n_{k,t} + \alpha_{j,t})\right) - K_j \Psi(\alpha_{j,t})}{\left[\sum_{\{k:f(k)=j\}} \Psi(\sum_{t=1}^{T} n_{k,t} + \alpha_{j,t})\right] - K_j \Psi(\sum_{t=1}^{T} \alpha_{j,t})} \,, \tag{14}$$

$$\alpha \leftarrow \alpha \frac{\left(\sum_{k=1}^{K} \sum_{t=1}^{T} \Psi(n_{k,t} + \alpha)\right) - KT\Psi(\alpha)}{T\left[\left(\sum_{k=1}^{K} \Psi([\sum_{t=1}^{T} n_{k,t}] + T\alpha)\right) - K\Psi(T\alpha)\right]} \,. \tag{15}$$

where $\Psi(x) = \mathrm{d}/\mathrm{d}x \log \Gamma(x)$ is the digamma function and level indicators ℓ are omitted. For the case $j \not\equiv 1$ we use $f(\uparrow X) = f(k)$ for notational simplicity. Each $\alpha_{j,t}$ then is estimated from K_j components for each of the J component groups. Estimators are initialised with a coarse-grained heuristic or a previous estimate and converge within few iterations.

Component parameters. Estimation of component parameters Θ is possible directly from the statistics of the collapsed state H and estimated hyperparameters A. Using the posterior mean of Dirichlet distributions given obervation counts $\vec{n}_k$ for each level ℓ, $\mathrm{Dir}(\vec{\vartheta}_k|\vec{\alpha}_j + \vec{n}_k) = \prod_i \mathrm{Mult}(x_i|\vec{\vartheta}_k) \cdot \mathrm{Dir}(\vec{\vartheta}_k|\vec{\alpha}_j)$, leads to the point estimate:

$$\vartheta_{k,t} = \frac{n_{k,t} + \alpha_{j,t}}{\sum_{t=1}^{T} n_{k,t} + \alpha_{j,t}} \tag{16}$$

where $\alpha_{j,t} \equiv \alpha$ for the symmetric case. Usually several samples $H^{(r)}$, $r \in [1, R]$ are taken from the stationary Markov chain with a sampling lag in between to ensure decorrelation. Finally parameters are averaged: $\vec{\vartheta}_k \approx R^{-1} \sum_r \vec{\vartheta}_k^{(r)}$.

4.3 Predictive inference

In many applications, it is necessary to predict the topics of some query data set V' given the model $\mathcal{M}$ trained on the observations V. Regarding the information required to represent the model $\mathcal{M}$, two different types of node can be distinguished:

- *Topic nodes*, $\ell \in L^*$, represent mixtures whose components are not specific to documents, i.e., $g(\uparrow x_i, i) \equiv g(\uparrow x_i)$, and $\mathcal{M}$ contains their parameters $\Theta^* = \{\Theta^\ell\}_{\ell \in L^*}$,
- *Sequence nodes*, $\ell \in L'$, represent mixtures specific to documents, and $\mathcal{M}$ contains their hyperparameters $A' = \{A^\ell\}_{\ell \in L'}$ that allow to find parameters $\Theta' = \{\Theta^\ell\}_{\ell \in L'}$.

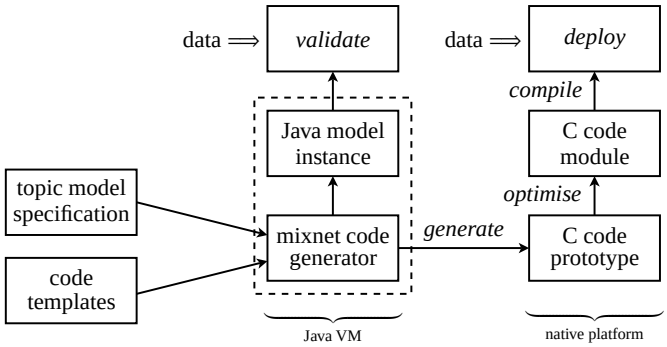

Fig. 3. Mixture network Gibbs sampler development workflow.

Thus we can define $\mathcal{M} \triangleq \{\Theta^*, A'\}$, and finding the association of unseen data V' with a state H' can be achieved using Gibbs sampling with a predictive full conditional analogous to Eq. 11, only that now it is possible (1) to treat parameters of topic nodes Θ^* as observed and (2) to restrict sampling to the query state H' without M-step updates, which both accelerates convergence of H' compared to H:

$$p(h_i'^d | H'^d_{\neg i}, H'^{\neg d}, V', \mathcal{M}) \propto \prod_{\ell \in \{S^{*d}, H^{*d}\}} [\vartheta_{k,t}]^{[\ell]} \cdot \prod_{\ell \in \{S'^d, H'^d\}} \left[\prod_{k=1}^{K} \frac{\Delta(\vec{n}_k + \vec{\alpha}_j)}{\Delta(\vec{n}_{k,\neg i^d} + \vec{\alpha}_j)} \right]^{[\ell]} . \quad (17)$$

With this equation, all findings on generic full conditionals that were derived from the analogous Eq. 11 can be reused, including Eqs. 12 and 13. Parameters can be estimated again using Eq. 16.

5 Implementation

The coherence of Eqs. 12–17 across models leads to the conclusion that Gibbs sampler implementations can be achieved based on a small number of computation kernels. Few reusable kernels are desirable when targeting architectures that require high optimisation effort. In this section, a proof-of-concept implementation of the MN approach is outlined that, although it targets a CPU-based architecture, may be a basis for topic model implementation on massively parallel and FPGA-based architectures.

5.1 Generic Gibbs samplers

The implementation of MN Gibbs samplers is based on a multi-stage workflow that allows to construct software modules with increasing levels of optimisation. This is intended to keep the interface for the researcher simple while retaining flexibility with respect to target architectures. An overview of the workflow is given in Fig. 3.

The central block in this process is the Java-based mixture network code generator, which is fed with a simple text script of a given MN (for example that shown in Fig. 4)

```
data:                          # input data
    w[m,n] : M * N[m] -> V     # word tokens (vocabulary size V)
state:                         # latent variables (E-step)
    x[m,n] : M * N[m] -> X     # supertopics (defines dimension X)
    y[m,n] : M * N[m] -> Y     # subtopic (defines dimension Y)
est:                           # estimated parameters (M-step)
    thetar : M * X             # document-supertopic level 1
    theta  : M * X * Y         # supertopic-subtopic level 2
    phi    : Y * V             # subtopic-term level 3
    alphar : 1                 # level 1 hparam (scalar)
    alpha  : X * Y             # level 2 hparam (with grouping)
    beta   : 1                 # level 3 hparam (scalar)
network:                       # format: parent_values >>
                               #     param[g(pav,i)] | hparam[f(pav)]
                               #     >> child_edge[sequence] = value
    m   >>   thetar[m]  | alphar    >>   x[m,n] = x
    x   >>   theta[m,x] | alpha[x]  >>   y[m,n] = k
    k   >>   phi[k]     | beta      >>   w[m,n]
```

Fig. 4. Commented mixture network script for 4-level PAM.

and allows two modes of operation: (a) The generator can create an instance of a Java-based Gibbs sampling class directly from the model script, e.g., for model validation purposes, and: (b) Based on a set of code templates, it generates C source code of the Gibbs sampler kernels that can then be further optimised and integrated with other code before it is compiled for the native computing platform.

In both cases, the generator applies the results of Sec. 4 to the information parsed from the script, creating Gibbs sampling algorithms as outlined in Fig. 5. Across different MN models, the design follows a stochastic EM approach that after initialisation loops over alternating sampling (E) and hyperparameter estimation (M) steps until convergence, after which samples can be drawn from the posterior. Important data structures in the generated code include the Markov state H, its count statistics as well as the arrays for multinomial sampling from the full conditional. The main computation kernels are those for full conditionals, Eq. 12 (including filling of the multinomial masses of $p(h_i^d|\cdot)$), for parameter estimation, Eqs. 14–16, as well as for convergence monitoring, which is described below. Currently, in addition to standard Dirichlet–multinomial nodes models can include observed nodes and parameters but are restricted to a single token sequence, which excludes aggregation as in the LDCC example.

Convergence monitoring and model quality. Gibbs sampling and other MCMC methods pose the general problem to determine when their Markov chain reaches a stationary state that allows to sample from the posterior distribution. With standard convergence diagnostics [29] difficult to apply to the high-dimensional discrete problem at hand, an alternative approach is to use some measure of model quality that reaches an optimum at convergence. Because of its generalisability and frequent use of similar approaches in topic model evaluation, the likelihood of test data given the trained model $\mathcal{M}$ (as defined in Sec. 4.3) has been chosen as quality measure, whose generalisation can be outlined as follows:

Algorithm `mixnetGibbs(`V, V'`)`
Input: training and test observations V, V'
Global data: level-specific dimensions $K^H = \{K^\ell\}_{\ell \in H}$, $T^H = \{T^\ell\}_{\ell \in H}$, selection functions f and g, count statistics $N^\ell = [\{\vec{n}_k\}_{k=1}^K]^\ell$, $N^\ell \in N$ and their sums $\Sigma^\ell = [\{\sum_x n_{k,t}\}_{k=1}^K]^\ell$, $\Sigma^\ell \in \Sigma$ for each node with hidden parameters, memory for full conditional array $p(h_i^d|\cdot)$, likelihood $\mathcal{L}$
Output: topic associations H, parameters Θ and hyperparameters A

```
// initialise
```
for all nodes ℓ in topological order **do**
    random initialise hidden sequences $h_i^\ell \sim \text{Mult}(1/T^\ell)$, update counts N^ℓ and Σ^ℓ

```
// Gibbs EM over burn-in period and sampling period
```
while not (converged and R samples taken) **do**
    `// stochastic E step to sample collapsed state`
    for all dependency groups $H^d \subseteq H$ **do**
        for all joint tokens $h_i^d \in H^d$ **do**
            decrement counts N^d and sums Σ^d according to current state h_i^d
            assemble array for $p(h_i^d|H_{\neg i}^d, H^{\neg d}, V)$ acc. to Eq. 12
            sample new state $h_i^d \sim p(h_i^d|H_{\neg i}^d, H^{\neg d}, V)$
            increment counts N^d and sums Σ^d according to changed state h_i^d
    `// M step to estimate parameters`
    for all nodes ℓ **do**
        update hyperparameters A^ℓ acc. to Eqs. 14 and 15
    for all nodes ℓ **do**
        find parameters Θ^ℓ according to Eq. 16
    `// monitor convergence using test data likelihood`
    $\mathcal{L} \leftarrow$ **call** `testLik(`Θ, A, V'`)` using Eqs. 16–18
    if $\mathcal{L}$ converged and L sampling iterations since last read out **then**
        `// different parameter read outs are averaged`
        $\bar{\Theta} \leftarrow \bar{\Theta} + \Theta$

```
// Complete parameter average
```
$\Theta = \bar{\Theta}/R$

Fig. 5. Generic Gibbs sampling algorithm.

- For each sequence node of the network, the hidden state H' is trained on test data $V' = \{v_i'\}_i$ according to Eq. 17, resulting in predictive parameters $\Theta' = \{\Theta'^\ell\}_\ell$.
- For each test-data token v_i', the likelihood given parameters $\{\Theta', \Theta^*\}$ is calculated:

$$p(v_i' \,|\, \Theta', \Theta^*) \;=\; \sum_{h_i'} \prod_{\ell \in L} \left[\vartheta_{k,t}\right]^{[\ell]} \tag{18}$$

  where the sum over h_i' refers to marginalisation of all hidden variables. To calculate Eq. 18 efficiently, the mixture network is traversed level by level according to its generative process, multiplying the respective level parameters (elements of Θ'^ℓ or $\Theta^{*\ell}$) and summing over values of latent variables $h_i'^\ell$ not indexing components k^ℓ of child levels. Further, duplicate v_i' have identical likelihood.
- The log likelihood of held-out test documents is accumulated from the token likelihoods: $\mathcal{L}(V') = \sum_i \log p(v_i' \,|\, \Theta', \Theta^*)$.

As a variant, the test-set likelihood $\mathcal{L}(V')$ can be exponentiated and normalised with the number of tokens in the test data W' to obtain the perplexity: $\mathcal{P}(V') = \exp(-\mathcal{L}(V')/W')$, i.e., the inverse geometric mean of the token likelihoods. Both $\mathcal{L}(V')$ and $\mathcal{P}(V')$ are measures of how well a model is able to explain unseen data. Specifically, perplexity

can be intuitively interpreted as the expected size of a vocabulary with uniform word distribution that the model would need to generate a token of the test data. A model that better captures co-occurrences in the data requires fewer possibilities to choose tokens given their context (document etc.). Due to the stochastic nature of the states H and H', values of $\mathcal{L}(V')$ and $\mathcal{P}(V')$ are not strictly monotonic over iterations. Thus, convergence of their moving-average process is used as indicator of Markov chain stationarity.

5.2 Validation

At this point, the focus of validation was on algorithms generated for a single-processor PC architecture, providing a basis for future investigation of specific high-performance architectures. In order to validate the implementation taken, generated and manually developed mixture network Gibbs samplers have been compared, including the examples LDA, ATM and PAM from Fig. 2 as well as several other models with two and three dependent hidden variables that handle labelled texts. Beside verification of the generated kernels, the code has been tested on the NIPS1-12[2] and Reuters-21578[3] data sets that in addition to text contain label information (authors, categories). Temporal performance achieved with the generated C-based algorithms came close to the respective manual implementations ($\Delta t < 2.5\%$). With equal seeds for random number generators, numerical behaviour turned out to be identical, considering Θ, A and $\mathcal{P}(V')$. Validation results are presented in further detail in a technical report [30].

6 Conclusions

We have presented a generic approach to topic models that covers a broad range of models in the literature. From their general characteristics, we have developed a representation of topic models as "mixture networks" along with a domain-specific graphical representation that complements Bayesian networks. Based on the mixture network representation, Gibbs sampling full conditionals were derived, which resulted in a generic Gibbs sampling algorithm and a "meta-Gibbs sampler" implementation based on code generation for specific models.

Future work can depart from these results in various directions. Extensions like the ones listed in Sec. 2 are desirable to widen the scope of the the mixture network approach, e.g., towards non-discrete observations as in the Corr-LDA model [6] and infinite mixtures with Dirichlet process priors [16]. Furthermore, the generic approach for Gibbs sampling may be applied analogously to collapsed variational Bayes [20].

The foremost research direction is, however, related to the actual motivation of this article discussed in the Introduction: to extend the code generation to high-performance computing architectures to help tackle the scalability issues common with topic models. The vision of this is a high-level language as a user front-end for implementations with optimised computing kernels. In addition to targeting computing platforms, improvements may be gained from heuristics like the statistically motivated acceleration of multinomial samplers proposed in [31], especially for the large sampling spaces of dependent latent variables.

[2] http://www.cs.toronto.edu/~roweis/data.html.

[3] http://www.daviddlewis.com/resources/testcollections/reuters21578/.

References

1. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley (2000)
2. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. Journal of Machine Learning Research **3** (2003) 993–1022
3. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.: Probabilistic author-topic models for information discovery. In: The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2004)
4. Li, W., McCallum, A.: Pachinko allocation: DAG-structured mixture models of topic correlations. In: ICML '06: Proceedings of the 23rd international conference on Machine learning, New York, NY, USA, ACM (2006) 577–584
5. Mimno, D., Li, W., McCallum, A.: Mixtures of hierarchical topics with pachinko allocation. In: ICML '07: Proceedings of the 24th International Conference on Machine Learning, New York, NY, USA, ACM (2007) 633–640
6. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D., Jordan, M.: Matching words and pictures. JMLR – Special Issue on Machine Learning Methods for Text and Images **3** (2003) 1107–1136
7. Lunn, D., Thomas, A., Best, N., , Spiegelhalter, D.: WinBUGS – a Bayesian modelling framework: concepts, structure, and extensibility. Statistics and Computing **10** (2000) 325–337
8. Daumé, H.I.: HBC: Hierarchical Bayes Compiler. (2007)
9. Gray, A.G., Fischer, B., Schumann, J., Buntine, W.L.: Automatic derivation of statistical algorithms: The EM family and beyond. In: NIPS. (2002) 673–680
10. Winn, J.M.: Variational Message Passing and its Applications. PhD thesis, University of Cambridge (2004)
11. Khronos OpenCL Working Group: The OpenCL Specification, version 1.0.29. (2008)
12. Rashid, M., Ferrandi, F., Bertels, K.: hArtes design flow for heterogeneous platforms. In: Proc. 10th International Symposium on Quality of Electronic Design (ISQED). (2009) 330–338
13. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. Technical report, EECS Dept., University of California, Berkeley (2003)
14. Shafiei, M.M., Milios, E.E.: Latent Dirichlet co-clustering. In: ICDM '06: Proceedings of the Sixth International Conference on Data Mining, Washington, DC, USA, IEEE Computer Society (2006) 542–551
15. Teh, Y.W., Jordan, M.I.: Hierarchical Bayesian nonparametric models with applications. In Hjort, N., Holmes, C., Müller, P., Walker, S., eds.: To appear in Bayesian Nonparametrics: Principles and Practice. Cambridge University Press (2009)
16. Teh, Y., Jordan, M., Beal, M., Blei, D.: Hierarchical Dirichlet processes. Journal of the American Statistical Association **101** (2006) 1566–1581
17. Blei, D.M., Jordan, M.I.: Variational methods for the Dirichlet process. In: Proc. ICML. (2006)
18. Ishwaran, H., James, L.F.: Gibbs sampling methods for stick breaking priors. Journal of the American Statistical Association **96** (2001) 161–??
19. Blei, D., Lafferty, J.: A correlated topic model of science. Annals of Applied Statistics **1** (2007) 17–35.
20. Teh, Y.W., Newman, D., Welling, M.: A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In: Advances in Neural Information Processing Systems. Volume 19. (2007)
21. Minka, T., Lafferty, J.: Expectation-propagation for the generative aspect model. In: Proc. UAI. (2002)

22. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences **101** (2004) 5228–5235
23. Dueck, D., Frey, B.J.: Probabilistic sparse matrix factorization. Technical report PSI TR 2004-023., U. Toronto, September 28 (2004)
24. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE 6 **6** (1984) 721–741
25. Pritchard, J.K., Stephens, M., Donnelly, P.: Inference of population structure using multilocus genotype data. Genetics **155** (2000) 945–959
26. Liu, J.: The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problems. Journal of the American Statistical Association **89(427)** (1994) 958–966
27. Jank, W.: Stochastic variants of EM: Monte Carlo, quasi-Monte Carlo and more. In: Proc. American Statistical Association, Minneapolis, Minnesota. (2005)
28. Minka, T.: Estimating a Dirichlet distribution. Web (2000)
29. Robert, C., Casella, G.: Monte Carlo Statistical Methods. 2nd edn. New York: Springer-Verlag (2004)
30. Heinrich, G.: Generic topic models. Technical report 09RP008-FIGD, Fraunhofer IGD, Darmstadt (2009)
31. Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., Welling, M.: Fast collapsed Gibbs sampling for latent Dirichlet allocation. In: KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2008) 569–577