Higher-Order Markov Tag-Topic Models for Tagged Documents and Images

Jia Zeng, Wei Feng, William K. Cheung and Chun-Hung Li

Abstract—This paper studies the topic modeling problem of tagged documents and images. Higher-order relations among tagged documents and images are major and ubiquitous characteristics, and play positive roles in extracting reliable and interpretable topics. In this paper, we propose the tag-topic models (TTM) to depict such higher-order topic structural dependencies within the Markov random field (MRF) framework. First, we use the novel factor graph representation of latent Dirichlet allocation (LDA)-based topic models from the MRF perspective, and present an efficient loopy belief propagation (BP) algorithm for approximate inference and parameter estimation. Second, we propose the factor hypergraph representation of TTM, and focus on both pairwise and higher-order relation modeling among tagged documents and images. Efficient loopy BP algorithm is developed to learn TTM, which encourages the topic labeling smoothness among tagged documents and images. Extensive experimental results confirm the incorporation of higher-order relations to be effective in enhancing the overall topic modeling performance, when compared with current state-of-the-art topic models, in many text and image mining tasks of broad interests such as word and link prediction, document classification, and tag recommendation.

Index Terms—Topic models, Latent Dirichlet allocation, Markov random fields, Bayesian networks, factor graph, hypergraph, higher-order relation, tagged documents and images, belief propagation, message passing, hierarchical Bayesian models.

1 INTRODUCTION

The goal of this work is to model and infer semantically meaningful word clusters, referred to as topics, from large-scale tagged documents and images. In a broad sense, we define a tag as a label that characterizes certain properties of documents and images. For example, the "author" tag identifies the authorship of document, and the "time stamp" tag marks when the document is published. On the other hand, we can treat images as documents composed of visual words. The users often manually annotate images by semantic tags such as "building" or "tree" to label local contents or objects of interests. Generally, one document may be associated with multiple tags, and one tag may be attached to multiple documents. Fig. 1A illustrates an example of tagged documents with the tags being authors, where the link denotes that the author writes the document. Fig. 1B shows another example of tagged images, where four images are annotated with three tags "sky", "building", and "people". We can conveniently represent tagged documents and images by a bipartite graph in Fig. 1C, which is composed of the tag nodes \( \gamma \) and the document or image nodes \( \delta \) connected by links. In the bipartite graph, tags often connect multiple documents or images so that build higher-order relations among documents or images.

Besides the pairwise relations, the higher-order relations among tagged documents and images formed by multiple tags are major and ubiquitous characteristics. For example, the authors \( \{\gamma_1, \gamma_2\} \) collaborate to write the document \( \delta_1 \) on the topic "Machine Learning" denoted by the intersection subset of two circles \( \{\gamma_1, \gamma_2\} \) in Fig. 1D. Similarly, \( \{\gamma_1, \gamma_3\} \) collaborate to write \( \delta_2 \) on the topic "Computer Vision", and \( \{\gamma_2, \gamma_3\} \) collaborate to write \( \delta_4 \) on the topic "Data Mining". These three authors \( \{\gamma_1, \gamma_2, \gamma_3\} \) also jointly collaborate to write the document \( \delta_4 \) denoted by the intersection subset of three circles \( \{\gamma_1, \gamma_2, \gamma_3\} \) in Fig. 1D. If we simply decompose the higher-order relation \( \{\gamma_1, \gamma_2, \gamma_3\} \) into three pairwise relations \( \{\gamma_1, \gamma_2\}, \{\gamma_1, \gamma_3\}, \{\gamma_2, \gamma_3\} \), we may just come to the conclusion that \( \delta_4 \) focuses on the combined topics of "Machine Learning", "Computer Vision", and "Data Mining". Nevertheless, the possibility that \( \delta_2 \) is in fact modeling a totally new topic like "Computational Biology" in the intersection subset of three circles will be excluded as shown in Fig. 1D. Obviously, \( \delta_2 \) lies in the specific subset of \( \{\gamma_1, \gamma_2, \gamma_3\} \), which is quite different from the union of subsets \( \{\gamma_1, \gamma_2\}, \{\gamma_1, \gamma_3\}, \{\gamma_2, \gamma_3\} \). So, the explicit modeling the higher-order relation among documents constituted by multiple tags \( \{\gamma_1, \gamma_2, \gamma_3\} \) is needed to distinguish the specific topic of \( \delta_2 \) from the combined topics of \( \delta_1, \delta_3, \delta_4 \), and \( \delta_4 \). Furthermore, modeling such higher-order relations also reflects the truth that the tags \( \{\gamma_1, \gamma_2, \gamma_3\} \) are often attached to the document \( \delta_2 \) jointly rather than separately to explain the document content. Similar higher-order relations among images induced by
multiple tags can be also found in Fig. 1B. However, prior efforts at pairwise relation modeling in topic models [11, 11] rarely consider higher-order relations that may encode specific topic structural dependencies among tagged documents and images. Therefore, in this paper, we propose the tag-topic models (TTM) to describe such higher-order topic structural dependencies within the Markov random field (MRF) framework. This approach extends our previous work in modeling higher-order relations of coauthors [12] to the more generic tagged documents and images, allowing us to develop more efficient inference and parameter estimation algorithms within the theoretically well-founded MRF framework.

First, we reformulate the topic modeling task as a labeling problem from the novel MRF perspective. We represent the latent Dirichlet allocation [11] (LDA)-based topic models as factor graphs [13, 14], and develop the classic loopy belief propagation (BP) algorithm to make approximate inference and parameter estimation. Second, we represent TTM using the factor hypergraph [15] according to the bipartite graph in Fig. 1C, and focus on both pairwise and higher-order relation modeling within the higher-order MRF framework. Indeed, such higher-order MRF has recently found important applications in modeling high-level image structural priors in many computer vision problems, including image restoration, disparity estimation and object segmentation [16, 17].

Generally, inferring the higher-order MRF is intrinsically a computationally expensive problem since even encoding $M$-order topic structural dependencies of $J$ topics requires $J^M$ labeling configurations. However, similar to image structural priors, higher-order relations used in topic modeling also have certain properties such as smoothness or sparsity [17–19], which makes them easy to handle. Intuitively, the co-tagged documents and images tend to have a higher likelihood to share the similar topic labeling configuration. Based on the smoothness or sparsity prior, many higher-order topic labeling configurations are equally unlikely and thus need not to be encouraged. Therefore, we encourage only a total of $J^M$ smooth topic labeling configurations in TTM, which avoids encoding $J^M$ arbitrary topic structural dependencies. To this end, we design the higher-order functions to encode the major and representative smoothness relations, and develop the loopy BP algorithm [14, 16] to make efficient inference and parameter estimation of TTM.

The rest of this paper is organized as follows. Section 2 introduces related work. Sections 3 presents MRF for topic modeling and develops loopy BP algorithms for approximate inference and parameter estimation. Section 4 proposes TTM and focuses on pairwise and higher-order relation modeling among tagged documents and images. Section 5 shows extensive experimental results on several challenging text and image mining tasks of broad interests. Finally, Section 6 draws conclusions and envisions future work.

2 RELATED WORK

The use of probabilistic topic models for text mining is the state-of-the-art approach such as learning terminological ontologies [22]. LDA [11] is the basic topic model in Fig. 2A. It allocates a topic label $z$ to each word $w$ in the document $d$ based on the document-specific topic proportion $\theta_d$ and the topic-specific multinomial distribution $\phi$ over vocabulary words, where $\theta_d$ and $\phi$ are smoothed by two conjugate Dirichlet hyperparameters $\alpha$ and $\beta$, respectively. The plates indicate replication. For example, the document repeats $D$ times in the corpus, the word repeats $I_d$ times in the document $d$, and there are a total of $J$ topics. LDA builds implicit links between two documents $\{d, d'\}$ by sharing the same topic distribution $\phi$, and it encourages similar topic labeling configurations if two documents contain similar words. However, LDA uses only $\phi$ to exchange topic information among documents, but ignores the rich link information like citations or hyperlinks between documents. This motivates the recent variants of LDA that regularize topic distribution $\phi$ through pairwise relations between documents.

Pairwise topic models focus on the link generation process, which in turn influences the topic allocation for words. Link LDA [2] uses the document-specific topic proportion and the topic-specific distribution over documents to generate a cited document by the citing doc-

---

**Fig. 1.** Examples of tagged documents and images: (A) the tagged documents with the tags being authors, (B) the tagged images with the tags being annotations, (C) the bipartite graph representation, and (D) the higher-order relation distinguishes the specific topic of $\delta_2$ from the combined topics of $\delta_1$, $\delta_3$, and $\delta_4$. 

---

**Table 1.**

| Topic Tag | Frequency |
|-----------|-----------|
| $\phi_1$  | 0.1       |
| $\phi_2$  | 0.2       |
| $\phi_3$  | 0.3       |
| $\phi_4$  | 0.4       |

**Fig. 2.** Diagram of TTM with multiple tags.
...pairwise topic models have limited expressive power, because they are insufficient to depict higher-order relations among tagged documents and images.

Author-topic models (ATM) [20] (Fig. 2C) and labeled LDA (L-LDA) [21] (Fig. 2D) are able to associate observed tags with words directly. ATM uses a documentspecific uniform distribution \( u_d \) to generate a tag, and further uses the tag-specific topic proportions \( \theta_t \) to generate a topic label for the word. The plate on \( \theta \) indicates that there are \( T \) unique tags. All documents with the tag \( t \) will share the same tag-specific topic proportions \( \theta_t \), which implicitly encodes the pairwise relation of documents associated with the tag \( t \). L-LDA constrains latent topics to be observed tags generated from the document-specific topic proportions \( \theta_d \) over tags, where each tag is further associated with a multinomial distribution \( \phi_t \) to generate words. In this sense, L-LDA is a supervised topic model because it replaces latent topic labels as observed tags. All documents with the tag \( t \) will share the same multinomial distribution \( \phi_t \), which also encodes statistical information of documents associated with the tag \( t \). However, the higher-order relations among documents and images due to multiple connected tags have been largely neglected in both ATM and L-LDA, which motivates us to explore a more specific higher-order TTM in this study.

### 3 MRF FOR TOPIC MODELING

#### 3.1 The Labeling Problem

Table 1 summarizes some important notations in this paper. From a new perspective, this subsection formulates the topic modeling as the labeling problem within the MRF framework. The objective of topic modeling is to assign a set of semantic topic labels \( z = \{ z_{w,d} \} \) to explain the observed bag of words \( w = \{ w, d \} \), where \( 1 \leq w \leq W \) is the word index in the vocabulary and \( 1 \leq d \leq D \) is the document index in the corpus. Generally, the topic label \( z_{w,d} \) takes one of the topic index \( 1 \leq j \leq J \) and partitions all words into \( J \) topic groups, so that the topic modeling technique is often viewed as

![Fig. 2. (A) LDA [1], (B) RTM [4], (C) ATM [20], and (D) L-LDA [21]. The shaded circles are observed variables, while others are latent variables and parameters. The plates indicate replication. The hyperparameter \( \beta \) in (B)-(C) and some subscripts are omitted for simplicity.](image-url)
As far as topic modeling is concerned, we only encourage smoothness of neighboring topic labels, i.e., the neighboring topic labels tend to be the same. To this end, we define the neighborhood system of the topic label \( z_{w,d} \) as \( z_{w,d} \) and \( z_{w,-d} \), where \( z_{w,d} \) denotes a set of topic labels associated with all word indices in the document \( d \) excluding the word index \( w \), and \( z_{w,-d} \) denotes a set of topic labels associated with the word index \( w \) in all documents excluding \( d \). Furthermore, we use the factor graph [13], [14] Chapter 8.4.3 to represent LDA, and treat parameters \( \theta \) and \( \phi \) as factors with parameterized functions [14]. By designing the proper factor functions, which are equivalent to clique potentials, we can encourage or penalize different local labeling configurations in the neighborhood system. More specifically, we encourage the topic labeling smoothness among \( \{ z_{w,d}, z_{w,d}, z_{w,-d} \} \).

In this paper, we consider a type of LDA with fixed symmetric Dirichlet hyperparameters \( \alpha \) and \( \beta \) [23] in order to avoid the complex full Bayesian inference of \( \alpha \) and \( \beta \), respectively. We transform the generative graphical representation of LDA in Fig. 2A to the factor graph [13], [14] in Fig. 3A from the MRF perspective. We illustrate the factors \( \theta_d \) and \( \phi_w \) as squares, and denote their connected variables \( z_{w,d} \) as circles. Obviously, the factors \( \theta_d \) and \( \phi_w \) connects the neighboring topic labeling configurations \( \{ z_{w,d}, z_{w,d}, z_{w,-d} \} \). In this way, the hierarchically directed graphical model of LDA in Fig. 2A becomes a more generic undirected graphical model in Fig. 3A. We absorb the observed word index \( w \) as the index of the factor \( \phi_w \), which is similar to absorbing the observed document index \( d \) as the index of the factor \( \theta_d \) in Fig. 2A. Because the factors can be parameterized functions [14], both \( \theta_d \) and \( \phi_w \) can be the same multinomial functions smoothed by the Dirichlet priors defined in LDA [23]. Also, both hyperparameters can be viewed as pseudo-counts in estimating the corresponding multinomial distributions. This resembles the collapsed GS [23] that integrates out parameter variables \( \theta \) and \( \phi \) and treats hyperparameters \( \alpha \) and \( \beta \) as pseudo topic counts in order to perform inference on the collapsed hidden variable space \( z \).

Recently, LDA has been reformulated as a Bayesian network [24], which is one of the constrained undirected graphical models (MRF) with causal dependencies between hidden variables. Indeed, Fig. 2A and Fig. 3A reflect two facets of LDA, in which the former focuses more on the generative process of the observed words hierarchically, while the latter emphasizes more on the topic labeling smoothness within the MRF framework.

The original factor graph representation for MRF [13] can be naturally extended to describe the generative process of a probabilistic model. For example, one extension is the directed factor graph [25] that enhances the visual language to represent LDA. Because the topic modeling task can be formulated as the labeling problem from the MRF perspective, the original undirected factor graph has enough expressive power to represent LDA directly. Although the undirected graph does not explicitly emphasize the generative process as the directed counterpart [25], it still captures the underlying structural dependencies of hidden variables without loss of information. In this sense, the factor graph may be a more generic visual representation for both directed and undirected graphical models in various real-world applications.

Although the factor graph in Fig. 3A is slightly different from the directed graphical model in Fig. 2A, it can fulfill the same topic modeling task using the specific neighborhood systems and factor functions. First, both Figs. 3A and 2A have the same neighborhood system because the connection of hidden variables remains the same. Second, in the next subsection, we shall design specific factor functions to realize the same topic modeling goal as Fig. 3A without loss of information.

### 3.2 Inference and Parameter Estimation

The loopy BP [14] Chapter 8) algorithms such as the sum-product and max-sum algorithms provide efficient and approximate solutions to inference problems for graphs with loops in Fig. 3A. Rather than directly calculating the posterior probability \( p(z|w) \), we turn to calculating the posterior marginal probability \( p(z_{w,d}) \), referred to as the message \( \mu(z_{w,d}) \), which can be normalized efficiently using a local computation. Message passing proceeds...
from variables to factors, and in turn from factors to variables until convergence after several iterations. In this subsection, we adopt the sum-product algorithm to infer the marginal posterior probability $\mu(z_{w,d})$.

The message passing scheme is an instantiation of the E-step of expectation-maximization (EM) algorithm [14], which has been widely used to infer the marginal probabilities of hidden variables in various graphical models according to the maximum-likelihood estimation. For example, the E-step inference for Gaussian mixture models (GMM) [26], the forward-backward algorithm for hidden Markov models (HMM) [27], and the probabilistic relaxation labeling algorithm for MRF [28] can be all formulated within the message passing framework [14], [29], [30]. After the E-step, we can estimate parameters $\theta$ and $\phi$ based on the inferred marginal probabilities $\mu(z_{w,d})$ at the M-step of EM algorithm, which is almost the same as those EM algorithms for learning other finite mixture models like GMM and GMM-based HMM. More details on learning finite mixture models using the EM algorithm can be found in the book [14].

Fig. 3 shows the message passing from two factors to the variable. The message $\mu(z_{w,d})$ is the product of both input messages,

$$\mu(z_{w,d}) \propto \mu_{\theta_d \rightarrow z_{w,d}}(z_{w,d}) \times \mu_{\phi_w \rightarrow z_{w,d}}(z_{w,d}),$$

where we use the arrows to denote the message passing directions. The normalized message $\mu(z_{w,d})$ is in turn passed back to the factors. In Fig. 3C and 3D, the messages from factors to variables can be further calculated based on all input messages from neighboring variables as follows,

$$\mu_{\theta_d \rightarrow z_{w,d}}(z_{w,d}) = \sum_{z_{w,d}} f_{\theta_d} \prod_{z_{w,d}} \mu(z_{w,d}) \alpha, \tag{2}$$

$$\mu_{\phi_w \rightarrow z_{w,d}}(z_{w,d}) = \sum_{z_{w,d}} f_{\phi_w} \prod_{z_{w,d}} \mu(z_{w,d}) \beta, \tag{3}$$

where $z_{w,d}$ and $z_{w,d}$ represent all possible neighboring labeling configurations of $z_{w,d}$, and $f(\cdot)$ is the factor function that evaluates the topic structural dependencies of input topic messages. The topic labeling smoothness prior implies that only $J$ topic configurations are encouraged in (2) and (3). Thus, we can rewrite (2) and (3) as

$$\mu_{\theta_d \rightarrow z_{w,d}}(z_{w,d} = j) = f_{\theta_d} \prod_{z_{w,d}} \mu(z_{w,d} = j) \alpha, \tag{4}$$

$$\mu_{\phi_w \rightarrow z_{w,d}}(z_{w,d} = j) = f_{\phi_w} \prod_{z_{w,d}} \mu(z_{w,d} = j) \beta. \tag{5}$$

In practice, Eqs. (2) and (3) often cause the product of multiple input messages close to zero [31]. To avoid arithmetic underflow, we approximate the product of messages by the sum of messages because the product value increases when the sum value increases,

$$\prod_{z_{w,d}} \mu(z_{w,d}) \alpha \propto \sum_{z_{w,d}} \mu(z_{w,d}) + \alpha, \tag{6}$$

$$\prod_{z_{w,d}} \mu(z_{w,d}) \beta \propto \sum_{z_{w,d}} \mu(z_{w,d}) + \beta. \tag{7}$$

Such approximations as (6) and (7) transform the standard sum-product to the sum-sum algorithm, which is still good at passing messages in MRF with acceptable performance [30], [31].

For convenience we use the shorthand notations $\mu(z_{w}) = \sum_{w} \mu(z_{w,d})$, $\mu(z_{w}) = \sum_{w} \mu(z_{w,d})$, $\mu(z_{w,d}) = \sum_{w} \mu(z_{w,d})$, and $\mu(z_{w,d}) = \sum_{d} \mu(z_{w,d})$ in the subsequent formulas. In MRF [14], the factor functions $f(\cdot)$ correspond to the clique potentials, which can be designed arbitrarily to encode our prior knowledge on encouraging or penalizing topic labeling configurations. Indeed, the higher value of $f(\cdot)$ encourages passing more neighboring messages. Here, we design $f_{\theta_d}$ and $f_{\phi_w}$ as

$$f_{\theta_d} = \frac{1}{\sum_{j} \mu(z_{w,d} = j) + \alpha}, \tag{8}$$

$$f_{\phi_w} = \frac{1}{\sum_{j} \mu(z_{w,d} = j) + \beta}. \tag{9}$$

Eq. (8) normalizes the input messages by the total number of topics associated with the document $d$ in order to make output messages comparable across different documents. Eq. (9) normalizes the input messages by the total number of messages of all word indices in the vocabulary in order to make output messages comparable across different vocabulary words.

Combining (1) to (7) yields the complete message...
update equation,

\[ \mu(z_{w,d} = j) \propto \frac{\mu(z_{w,d} = j) + \alpha}{\sum_j [\mu(z_{w,d} = j) + \alpha]} \times \frac{\mu(z_{w,d} = j) + \beta}{\sum_w [\mu(z_{w,d} = j) + \beta]}, \]

where the notations \(-w\) and \(-d\) denote all word indices except \(w\) and all document indices except \(d\), and the notations \(\mu(z_{w,d})\) and \(\mu(z_{w,d})\) represent the sum of all possible neighboring messages excluding the current message \(\mu(z_{w,d})\). We normalize the updated message locally in terms of \(j\) so that \(\sum_j \mu(z_{w,d} = j) = 1\).

In practice, after finite \(N\) iterations, the message will converge in the factor graph as shown in Fig. 3A. BP usually converges fast with \(N \leq 500\). Note that we need to multiply the number of word counts or the relative word frequencies to the corresponding word topic message during message passing and parameter estimation.

Given the inferred marginal posterior probability \(\mu(z_{w,d})\), the parameter estimation of \(\theta\) and \(\phi\) can be performed simply using (4) and (5) (Figs. 4C and 4D) by adding all input messages including \(\mu(z_{w,d})\) evaluated by the corresponding factor functions,

\[ \theta_d(j) = \frac{\mu(z_{d,d} = j) + \alpha}{\sum_j [\mu(z_{d,d} = j) + \alpha]}, \]

\[ \phi_w(j) = \frac{\mu(z_{w,d} = j) + \beta}{\sum_w [\mu(z_{w,d} = j) + \beta]}, \]

Alternatively, we can also derive the parameter estimation equations using the EM algorithm [14]. In the E-step, we calculate the marginal posterior probability \(\mu(z_{w,d})\). Employing the multinomial-Dirichlet conjugacy and Bayes’ rule, we get the following marginal Dirichlet distributions [23],

\[ p(\theta_d) = \text{Dir}(\theta_d | \mu(z_{d,d}) + \alpha), \]

\[ p(\phi_w) = \text{Dir}(\phi_w | \mu(z_{w,d}) + \beta), \]

In the M-step, maximizing (13) and (14) with respect to \(\theta_d\) and \(\phi_w\) also results in the same parameter estimation equations (11) and (12).

### 3.3 Discussion

LDA is the hierarchical Baysian model that maximizes the objective \(p(w,z)\) to generate topic labels for words, while MRF is the undirected model that maximizes the objective \(p(z|w)\) to assign the best topic labels to words. Their objectives are identical according to the Bayes’ rule \(p(z|w) \propto p(w,z)\) since \(p(w)\) is a constant in terms of \(z\). The collapsed Gibbs sampling (GS) [23] and variational Bayes (VB) [11] have been two commonly-used approximate inference algorithms for LDA-based topic models. In this paper, we provide an alternative inference method for LDA-based topic models using the BP algorithm from the novel MRF perspective.

GS resembles the proposed BP except that it randomly samples a topic label from the marginal posterior probability \(\mu(z_{w,d})\) for each word token, and immediately updates parameters based on the currently sampled topic label. Therefore, GS needs to sample a topic label for each word token in the document, while BP only calculates the message for each word index in the vocabulary within the document. Due to the word sparsity in the document, BP significantly lowers the computational cost than GS. In addition, the randomly sampled topic label in GS always loses some information compared with the marginal posterior probability in BP. As a result, BP is more accurate than GS in parameter estimation because it keeps and uses the complete messages at each learning iteration without loss of information.

VB uses the Jensen’s inequality to get an adjustable lower bound on the objective function, and maximizes the objective through maximizing the lower bound by tuning a set of variational parameters. VB also resembles the proposed BP except that it calculates the topic messages by minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior distribution. Thus, the variational message update equations in VB differ significantly from those in BP by involving the more complicated digamma functions.

For each learning iteration, both BP and VB have the same computational cost \(O(JDW_d)\), but GS requires \(O(JD_{d1})\), where \(W_d\) is the average vocabulary size and \(I_d\) is the average number of word tokens per document. Because in a document the number of word indices is usually much smaller than the total number of word tokens due to the word sparsity, i.e., \(W_d \ll I_d\), BP and VB generally scale much better than GS to the large-scale corpus. More detailed comparisons among VB, GS and BP can be found in [32].

### 4 Tag-Topic Models

#### 4.1 Factor Hypergraph Representation

Fig. 4A shows the factor hypergraph representation of TTM, which directly combines the factor graph in Fig. 3A with the bipartite graph in Fig. 4C. Note that the undirected hypergraph is equivalent to the bipartite graph in Fig. 4C [15], where the factor hyperedge \(\delta_d\) (denoted by the yellow block) connects the tag factors \(\gamma_t\) attached to the document \(d\) in Fig. 4C. In this factor hypergraph representation, we absorb the observed tag \(t\) as the index of the factor \(\gamma_{t'}\) which connects the variable \(z_{w,d}\) with its neighbors \(z_{d,d}\) using the solid black line as shown in Fig. 4B. We assume that the document pair \(\{d, d'\}\) share the same tag \(t\), the document pair \(\{d, d''\}\) share the same tag \(t'\), the the document \(d\) are associated with three tags \(\{t, t', t''\}\). Although Fig. 4A does not follow the standard definition of factor graphs due to the factor hyperedge \(\delta_d\), this variant of factor graph can represent both pairwise and higher-order relations among tagged documents and images as shown.
in Fig. 4B and Fig. 4C. For example, the topic labeling configurations $z_{..d}$ or $z_{..d'}$ can influence its neighboring label $z_{w,d}$ separately through the factor $\gamma_t$ or $\gamma_{t'}$ based on the pairwise relation in Fig. 4B. In the meanwhile, $\{z_{..d'}, z_{..d''}\}$ can also influence their neighbor $z_{w,d}$ jointly through the factor hyperedge $\delta_d$ based on the higher-order relation resulted from the connected tag factors $\{\gamma_t, \gamma_{t'}\}$ in Fig. 4C. As a result, the factor function $f_{\gamma_t}$ encodes the pairwise relation between $\{z_{..d'}, z_{..d''}, z_{w,d}\}$, while the factor function $f_\delta$ depicts the higher-order relation among $\{z_{..d'}, z_{..d''}, z_{w,d}\}$.

4.2 Credit Attribution

Each attached semantic tag usually accounts for parts of words in documents or local contents in images. The **credit attribution** task is to associate individual words in a document with their most appropriate tags [21]. In the probabilistic framework, we assume that all tags in the document associate the same word with different likelihoods, which can be calculated based on the pairwise relation formed by each tag $\gamma_t$ as shown in Fig. 4B. Specifically, if $x_{w,d}$ is the tag label associated with the word $w$ in the document $d$, we calculate the likelihood $p(x_{w,d} = t)$ based on the following similarity in terms of topic messages,

$$p(x_{w,d} = t) = \sum_{j=1}^{J} \mu(z_{w,d} = j) \mu_{\gamma_t \rightarrow z_{w,d}}(z_{w,d} = j), \quad (15)$$

where the message $\mu_{\gamma_t \rightarrow z_{w,d}}(z_{w,d} = j)$ from the factor $\gamma_t$ will be introduced in the next subsections. Intuitively, Eq. (15) measures the similarity between the tag and the word content in the latent topic space. In practice, we randomly initialize $p(x_{w,d} = t)$ for all tags per word, and iteratively update and normalize it using (15).

4.3 Pairwise and Higher-order Relation Modeling

In Fig. 4B, the factor function $f_{\gamma_t}$ describes the pairwise topic dependencies between all pairs of documents connected with the tag $t$,

$$f_{\gamma_t} = \frac{\sum_{d,d' \in \text{ne}(t)} \mu_{\gamma_t \rightarrow d} \circ \mu_{\gamma_{t'} \rightarrow d'}}{|d, d' \in \text{ne}(t)|}, \quad (16)$$

where the $\circ$ notation denotes the Hadamard (element-wise) product of two vectors [4], the $\text{ne}(t)$ notation denotes the set of all connected documents with the tag $t$, the $|d, d' \in \text{ne}(t)|$ notation indicates the total number of document pairs connected with the tag $t$, and

$$\mu_{\gamma_t \rightarrow d} = \sum_w \mu(z_{w,d}) p(x_{w,d} = t)$$

where $p(x_{w,d} = t)$ is defined in (15). The Hadamard product captures the similarity between two connected documents with the tag $t$ in the latent topic representations. As a result, Eq. (16) is the average Hadamard product of all pairs of documents connected with the tag $\gamma_t$ that encodes the dominant pairwise topic structural dependencies. Eq. (17) is the weighted sum of all word messages in the document $d$ with respect to the tag $t$, which can be viewed as the normalized message passed from all words in the document $d$ to the tag $t$.

In Fig. 4C, the factor function $f_{\delta_d}$ depicts the higher-order topic dependencies among documents and images through connected tags $\{t, t', t''\} \in \text{ne}(d)$. Generally, modeling the 3-order and 4-order relations is sufficient in practice because most documents and images often contain less than four tags in Table 2. Without loss of generality, here we present the 3-order relation modeling, and the higher than 3-order relation can be modeled...
similarly. We design $f_{\delta_d}$ for 3-order relation based on the Hadmard product as follows,
\[
 f_{\delta_d} = \frac{\sum_{d, d', d'' \in \text{ne}(t, t', t'')} \prod_{d, d', d''} \circ \prod_{d, d', d''}}{|(d, d', d'') \in \text{ne}(t, t', t'')}|, \tag{18}
\]
where the $|d, d', d'' \in \text{ne}(t, t', t'')|$ notation denotes the total number of 3-order document or image triples constituted by the connected tags $t$, $t'$ and $t''$, respectively. Obviously, Eq. (18) is the average Hadmard product of all triples of documents or images, capturing the major and representative 3-order topic structural dependencies among tagged documents and images.

### 4.4 Inference and Parameter Estimation

TTM in Fig. 4A contains loops so that we develop the loopy BP algorithm [14, 16] for approximate inference and parameter estimation. In subsection 3.2, we have calculated the messages $\mu_{\theta_d \rightarrow z_{w,d}}$ and $\mu_{\phi_w \rightarrow z_{w,d}}$ in (4) and (5). In this subsection, we focus on computing the message $\mu_{\gamma_t \rightarrow z_{w,d}}$ and $\mu_{\delta_d \rightarrow z_{w,d}}$ based on the sum-product algorithm, which involves not only pairwise relations but also higher-order relations among documents in Figs. 4B and 4C, respectively. Under the topic smoothness constraint, the message from the factor $\gamma_t$ to the variable $z_{w,d}$ is
\[
 \mu_{\gamma_t \rightarrow z_{w,d}} = f_{\gamma_t} \sum_{m \in \text{ne}(t) \backslash d} \prod_{m,t}, \tag{19}
\]
where $m \in \text{ne}(t) \backslash d$ denotes all connected document with the tag $t$ except the current document $d$ in Fig. 4B. Similarly, the message from the factor hyperedge $\delta_d$ to the variable $z_{w,d}$ is
\[
 \mu_{\delta_d \rightarrow z_{w,d}} = f_{\delta_d} \sum_{m, m' \in \text{ne}(t, t') \backslash d} \prod_{m,t} \prod_{m',t'}, \tag{20}
\]
where $m, m' \in \text{ne}(t, t') \backslash d$ contains all document pairs except the current document $d$ connected with the tags $\{t, t'\}$, respectively. Eq. (19) passes messages from the neighboring documents by the individual factor $\gamma_t$, while (20) passes messages by the factor hyperedge $\delta_d$, which connects multiple tag factors $\{\gamma_t, \gamma_t'\}$. Therefore, Eq. (19) influences the word message $\mu(z_{w,d})$ through the pairwise relation across the individual tag $t$, and (20) plays the similar role through the higher-order relation across multiple tags $\{t, t'\}$. Similar to (5) and (7), we replace the product operation by the sum operation in (19) and (20) for all neighboring input messages in order to avoid arithmetic underflow.

In the standard sum-product algorithm, we calculate the marginal posterior probability $\mu_{z_{w,d}}$ by the product of all input messages according to Fig. 4A. However, the direct product is not flexible to balance the messages from factors $\theta_d$, $\gamma_t$ and $\delta_d$ in Fig. 4A. Conceivably, the message $\mu_{\theta_d \rightarrow z_{w,d}}$ measures the topic labeling influence within the document $d$, the message $\mu_{\gamma_t \rightarrow z_{w,d}}$ captures the influence from neighboring documents by pairwise relations, and the message $\mu_{\delta_d \rightarrow z_{w,d}}$ plays the similar role but through higher-order relations. Because these three messages are in the document level, we balance them by a simple convex combination, and rewrite (1) as
\[
 \mu(z_{w,d}) \propto \left[ (1 - \omega_1 - \omega_2) \mu_{\theta_d \rightarrow z_{w,d}} + \omega_1 \sum_{t \in \text{ne}(d)} \mu_{\gamma_t \rightarrow z_{w,d}} + \omega_2 \mu_{\delta_d \rightarrow z_{w,d}} \right] \times \mu_{\phi_w \rightarrow z_{w,d}}, \tag{21}
\]
where $\omega_1 = 0, \omega_2 = 0, \omega_1 \mu_2 \leq 1$ are the weights to balance three messages from factors $\theta_d$, $\gamma_t$ and $\delta_d$. In (21), we sum the messages $\mu_{\gamma_t \rightarrow z_{w,d}}$ in terms of all individual tag $t$ attached to the document $d$, which accumulates the influence from all attached tags. Obviously, Eq. (21) shows that the current word message is regularized by both pairwise and higher-order relations of tagged documents. When $\omega_1 = \omega_2 = 0$, Eq. (21) reduces to (1), so that TTM becomes LDA without tag information. When $\omega_3 = 0$, we depict only pairwise relations between tagged documents. Automatic estimating the best weights in TTM requires further studies in future work. In this paper, we manually tune these weights based on the training data sets.

The inference and parameter estimation equations for TTM are almost the same as those for LDA except that the update equation of message $\mu(z_{w,d})$ is replaced by (21). Fig. 5 summarizes the loopy BP algorithm for learning TTM. At each learning iteration, we need to estimate both the pairwise and higher-order topic structural dependencies using (15) and (18), so that the computational cost of learning TTM is $O(JDLW)$, where $L$ is the total number of pairwise and higher-order relations among tagged documents and images.

### 5 Experimental Results

#### 5.1 Data Sets

We use four data sets of tagged documents and images:
- **CORA** [33] and MEDLINE (MED) [34]: The former contains abstracts from the Cora research paper search engine in machine learning area, and the latter contains abstracts from the MEDLINE biomedical paper search engine. We use the author names as the tags for each paper. CORA documents can be classified into 7 major categories, and MED documents fall broadly into 4 categories.
- **COREL5K (CSK)** and COREL30K (C30K) [35]: They originate from the Corel stock photograph collection. They contain all kinds of images, ranging from natural scenes to people portraits or sports.

| Data sets | $D$ | $T$ | $W$ | $I_d$ | $W_d$ | $T_d$ |
|-----------|-----|-----|-----|-------|-------|-------|
| CORA      | 2410| 2480| 2961| 57    | 43    | 3.0   |
| MED       | 2317| 8906| 8918| 104   | 66    | 5.8   |
| CSK       | 5000| 371 | 128 | 11970 | 90    | 3.5   |
| C30K      | 31695| 1035| 128 | 11970 | 105   | 3.6   |
5.2 Performance of Tag-Topic Models

In the following experiments, we randomly divide the entire CORA and MED documents into training (80%) and test (20%) sets. For C5K, we use the same training and test set partition of [35], in which 4500 images constitute the training set and the remaining 500 images constitute the test set. For C30K, we randomly partition the entire images into training (90%) and test (10%) sets. We manually tune the weights $\omega_1$ and $\omega_2$ in (21) based on the perplexity [1] for training data. When $\omega_1 = 0.2, \omega_2 = 0$, we refer to this TTM as TTM-P for only pairwise relation modeling. When $\omega_1 = 0.1, \omega_2 = 0.05$, we refer to this TTM as TTM-H for both pairwise and higher-order relation modeling. Through the comparative study between TTM-P and TTM-H, we can explore the effectiveness of modeling higher-order topic interactions among tagged documents and images.

We compare TTM with three current state-of-the-art topic models such as RTM (Fig. 2B) with the exponential link probability function [4] ATM (Fig. 2C) [20] and L-LDA (Fig. 2D) [21] using the same experimental settings. As discussed in Section 2 the above benchmark topic models are able to handle pairwise relations between tagged documents. In contrast, TTM additionally considers higher-order relations induced by connected tags among documents. Because L-LDA is a supervised topic model, we only compare TTM with L-LDA in the tag recommendation task. In all experiments, we assume that the tags are unobserved in test data, and use the estimated topic distributions from training data to predict words, links, tags as well as class labels for documents in the test set.

5.2.1 Word Prediction

The word prediction task is to evaluate the likelihood that the learned topic distributions generate the unseen test data. Fig. 6 compares the test set perplexity of ATM, RTM, TTM-P and TTM-H. The lower perplexity corresponds to the higher likelihood that the learned topics can generate the unseen test set. For all data sets, TTM-H consistently achieves the lowest perplexity in different topics, which implies the best generalization ability to predict words in unseen test sets. Unlike RTM, ATM does not explicitly model the pairwise topic representations between tagged documents and images, so that it insufficiently benefit from the rich relational information for regularizing topic distributions. On the other hand, RTM estimates the link probability function for all document pairs connected by different tags, while TTM-P estimates the tag-specific pairwise relations using (16). As a result, TTM-P has the potential to capture the subtle topic structural dependencies between documents or images through specific tags. Fig. 6 shows that TTM-P achieves almost 15% reduction on average in perplexity compared with ATM and RTM. Furthermore, TTM-H gains on average 5% reduction of perplexity as compared with TTM-P, which indicates that joint influence through higher order relations may paly positive roles in topic distribution regularization. Although TTM-H has a higher computational complexity than TTM-P, it is worth gaining a better word prediction performance in real-world applications. Generally, the predictive perplexity decreases as the number of topics increases, which
implies that the more latent topics provide the higher likelihood to predict words.

To measure the interpretability of a topic model, the word intrusion and topic intrusion are proposed to involve subjective judgements [37]. The basic idea is to ask volunteer subjects to identify the number of word intruders in the topic as well as the topic intruders in the document, where intruders are defined as inconsistent words or topics based on prior knowledge of subjects. Due to lack of volunteer subjects, Fig. 7 shows only three consistent topics with top ten words on the CORA training set for qualitative evaluation. We see that most topics share similar words with different ranking orders. Nevertheless, both ATM and RTM extract first two topics that contain the word intruder “paper”, and RTM even extracts three word intruders “design”, “research” and “university” in the third topic. Obviously, both TTM-P and TTM-H show much better interpretability at least for the top ten words, which do not contain irrelevant common words such as “paper”. Moreover, TTM-H is slightly better than TTM-P in that it has a more natural word ranking order in each topic.

5.2.2 Link Prediction
The link prediction task is to predict if two documents share the same tag. The natural real-world application of link prediction is to suggest tags of a document to a linking document. If the tags are author names, we may use the link prediction to find reviewers or collaborators for the linking document. Also, the link prediction can help retrieve related documents with similar tags. The effectiveness of these applications depend highly on the link prediction accuracy. We define the link prediction as a binary classification problem. We use the Hadmard product of a pair of document topic proportions as the link feature, and train an SVM [38] to decide if there is a link between them. We evaluate link prediction performance using the same number of linking/non-linking training and test samples.

Fig. 8 compares the F-measure of link prediction. Because ATM does not encode the Hadmard link features of pairs of documents, its prediction results are almost random guess with F-measure close to 0.5 for all data sets. In contrast, RTM shows a significantly better link prediction performance using the generalized linear models estimated from the link features, which efficiently differentiate links from non-links. For all data sets, TTM-P deviates slightly from RTM because both TTM-P and RTM encode only pairwise relations of tagged documents. However, TTM-H outperforms RTM around 8% F-measure for link prediction. One possible
reason is that TTM-H incorporates much richer higher-order topic structural dependencies so that it makes the topic proportions of documents sharing tags more differentiable from those documents without sharing tags. Interestingly, F-measure does not always increase as the number of topics increases. Although more latent topics can predict unseen words better as shown in Fig. 8, they cannot consistently enhance the link prediction performance on the test set as shown in Fig. 8. This phenomenon suggests that the content similarity between documents alone cannot completely account for the link information. Additional information such as partially observed links of some documents may help for a better link prediction performance.

5.2.3 Document Classification

Document classification partitions a set of documents into several mutually exclusive categories. Topic models can be used as a dimensionality reduction method to reduce the high-dimensional word vector space for classification [1]. We may use the document topic proportions as the reduced feature vectors and study their discriminative ability in document classification. To this end, we train SVMs on the document topic proportions given class labels, and compare the document classification accuracy on the test set. In CORA, we randomly select 100 training samples for each of the seven categories. In MED, we randomly select 200 training samples for each of the five categories. In C5K and C30K, we choose four tags as class labels: sky, water, trees, and people. We use those images associated with only one of four tags for training purposes. In C5K, we randomly select 300 training samples for each class. In C30K, we randomly select 1500 training samples for each class. The remaining documents and images are test samples.

Fig. 9 shows the classification accuracy based on low-dimensional document topic proportions. We see that ATM generally outperforms RTM, which is inconsistent with their word prediction performance in Fig. 6. The reason may lie in that RTM treats sharing tags as equal links, but in reality different tags may encode different topic structural dependencies between documents. Thus, RTM may erroneously encourage the topic smoothness of documents through different tags, which often has the close correspondence with the class labels of documents, especially when tags are used as class labels for C5K and C30K. In contrast, TTM-P relaxes the limitation in RTM by encouraging the smoothness of document topic proportions using the tag-specific pairwise relation modeling. Furthermore, TTM-H still outperforms TTM-P with 6% higher classification accuracy on average by forcing tag-specific smoothness constraint through both pairwise and higher-order relations. Image classification performance on C5K and C30K is generally worse than that on CORA and MED, partly because the tags tend to describe individual image components, which are not exactly equivalent to class labels that describe the global image contents. Similar to the link prediction task, the more latent topics does not enhance the overall document classification performance.

5.2.4 Tag Recommendation

Tag recommendation is a multi-label classification problem that suggests a set of tags to query documents or images, which has found many real-world applications such as credit attribution [21], expert finding [39], image annotation [35], [40]. Due to lack of benchmark data to evaluate the expert finding performance, we focus on tag recommendation for image annotation in this section. We propose an TTM-based tag recommendation system including two SVMs:

1) Each tag is a class label. We train a multiclass SVM called \textit{svm} \textsubscript{1} to classify the image topic proportions into \( T \) tags, where the training samples are images associated with each tag. Some images may have more than one tag and will be used as training samples for multiple tags. For each training sample \( d \), \textit{svm} \textsubscript{1} predicts a vector of likelihoods \( p^d = [p_1, \ldots, p_t, \ldots, p_T] \) for all tags.
2) We also train a total of $T$ binary SVMs called $svm_2$ for all tags. For the tag $t$, the positive sample is the tagged image $d$ with the feature vector $p^d_t = [p_{t,i} | i \in \text{ne}(t)]$ predicted by $svm_1$, where $\text{ne}(t)$ is a set of connected tags of the tag $t$. This feature encodes information of connected tags for robust prediction. To balance the training data for each tag, we choose the same number of positive/negative samples. For each training sample $d$, $svm_2$ predicts a vector of likelihoods $\mu^d_t = [\mu^d_{t,1}, \sum_i \mu^d_{t,i}] = 1, i = \{1, 2\}$, where $\mu^d_{t,1}$ is the likelihood that the tag $t$ is recommended to $d$.

3) For the test image $d$, we use $svm_1$ to predict its likelihoods $p^d_t$ to all $T$ tags. Then, we use $svm_2$ to predict $\mu^d_t, 1 \leq t \leq T$ for all tags. To balance the prediction results of $svm_1$ and $svm_2$, we linearly combine the two likelihoods $y = \omega p^d_t + (1 - \omega)\mu^d_t$, where $\omega = 0.25$ estimated from the training set. We follow the standard image annotation evaluation protocol [35], [40], and suggest top five tags to the query image with highest $y$.

In this system, $svm_1$ uses only image content information to suggest tags, while $svm_2$ uses connected tags to refine the tag recommendation result. The basic idea is that if the tag $t$ is suggested to the image $d$, its connected tags also have a high likelihood to be suggested.

The performance measures for image tag recommendation include recall and precision rates per tag [35], [40]. More specifically, for a given tag, let $N_h$ be the number of images in the test set that are labeled with this tag by human, $N_s$ be the number of images in the test set that are labeled with tag by the tag recommendation system, and $N_c$ be the number of images that the system gives correct tag recommendation. The recall and precision rates are defined as recall = $N_c$/$N_h$ and precision = $N_c$/$N_s$. We also evaluate the coverage rate $R_{c+}$ of recommended tags, which is calculated as the number of tags with positive recall divided by the total number of tags in the test set. The higher $R_{c+}$ implies a better generalization ability that can achieve relative high recall and precision rates on a large set of tags.

Table 3 compares TTM-H and TTM-P with two state-of-the-art tag recommendation methods L-LDA [21] and SML [40]. With the similar coverage rate, TTM-H provides the competitive image annotation performance with SML. Although L-LDA shows the comparable or better tag recommendation performance than SVM for tagged web pages, it does not show clear advantages in image annotation problem especially on the C30K data set. Indeed, L-LDA does not use the connected tag information from training data, which play major roles to rule out many false positives to enhance the average precision. We see that TTM-H still outperforms TTM-P, which is consistent with its superior document classification performance as shown in Fig. 9. Furthermore, the more latent topics does not improve the tag recommendation performance, so that we show only the best results of TTM-H and TTM-P when the number of latent topics $J = 20$.

### 6 Conclusions

This paper has presented TTM and discussed its effectiveness in encoding smoothness pairwise and higher-order topic interactions among tagged documents and images. Within the MRF framework, TTM allows the efficient loopy BP algorithm for inference and parameter estimation. On four large-scale data sets, TTM consistently outperforms current state-of-the-art topic models,
such as ATM, RTM and L-LDA, in several real-world text and image mining applications.

Furthermore, we observe that the higher-order relations also exist in many important computer vision and text mining applications. For example, the unsupervised activity perception in crowded and complicated scenes involves lots of higher-order interactions of multiple agents, which can be encoded in topic models for discovering more specific motion patterns. Another example is tracking historical topics from time-stamped documents. We speculate that the higher-order temporal topic interactions may characterize some specific long-range topic evolution patterns, which can be also studied in our future work.

ACKNOWLEDGEMENTS

This work is supported by NSFC (Grant No. 61003154) and the Shanghai Key Laboratory of Intelligent Information Processing, China (Grant No. I IPL-2010-009).

REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” J. Mach. Learn. Res., vol. 3, pp. 993–1022, 2003.
[2] E. Eroshova, S. Fienberg, and J. Lafferty, “Mixed-membership models of scientific publications,” Proc. Natl. Acad. Sci., vol. 101, pp. 5220–5227, 2004.
[3] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen, “Joint latent topic models for text and citations,” in KDD, 2008, pp. 542–550.
[4] J. Chang and D. M. Blei, “Hierarchical relational models for document networks,” Annals of Applied Statistics, vol. 4, no. 1, pp. 124–150, 2010.
[5] A. Gruber, M. Rosen-Zvi, and Y. Weiss, “Latent topic models for hypertext,” in IJAI, 2008, pp. 230–239.
[6] J. Zeng, W. K.-W. Cheung, C.-H. Li, and J. Liu, “Topic-Link LDA: Joint models of topic and community authorship,” in ICML, 2009, pp. 665–672.
[7] J. Zeng, W. K.-W. Cheung, C.-H. Li, and J. Liu, “Multirelational topic models,” in ICDM, 2009, pp. 1070–1075.
[8] H. Daume III, “Markov random topic fields,” in ACL-IJCNLP, 2009, pp. 293–296.
[9] L. Dietz, S. Bickel, and T. Scheffer, “Unsupervised prediction of citation influences,” in ICML, 2007, pp. 233–240.
[10] Q. Mei, D. Cai, D. Zhang, and C. X. Zhai, “Topic modeling with network regularization,” in WWW, 2008, pp. 101–110.
[11] C. Wang, B. Thiesson, C. Meek, and D. Blei, “Markov topic models,” in AISTATS, 2009, pp. 583–590.
[12] J. Zeng, W. K. Cheung, C.-H. Li, and J. Liu, “Coauthor network topic models with application to expert finding,” in IEEE/WIC/ACM WI-IAT, 2010, pp. 366–373.
[13] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” IEEE Transactions on Inform. Theory, vol. 47, no. 2, pp. 498–519, 2001.
[14] C. M. Bishop, Pattern recognition and machine learning. Springer, 2006.
[15] S. Klamt, U. U. Haus, and F. Theis, “Hypergraphs and cellular networks,” PLoS Comput Biol, vol. 5, no. 5, p. e1000385, 2009.
[16] X. Lan, S. Roth, D. Huttonlocher, and M. Black, “Efficient belief propagation with learned higher-order Markov random fields,” in ECCV, 2006, pp. 269–282.
[17] C. Rother, P. Kohli, W. Feng, and J. Jia, “Minimizing sparse higher order energy functions of discrete variables,” in CVPR, 2009, pp. 1382–1389.
[18] S. Z. Li, Markov Random Field Modeling in Image Analysis. New York: Springer-Verlag, 2001.
[19] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, “A comparative study of energy minimization methods for Markov random fields with smoothness-based priors,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 6, pp. 1068–1080, 2008.
[20] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, “The author-topic model for authors and documents,” in UAI, 2004, pp. 487–494.
[21] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, “Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora,” in Empirical Methods in Natural Language Processing, 2009, pp. 248–256.
[22] W. Wei, P. Barnaghi, and A. Bargiela, “Probabilistic topic models for learning terminological ontologies,” IEEE Trans. Knowledge Data Eng., vol. 22, no. 7, pp. 1028–1040, 2010.
[23] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” Proc. Natl. Acad. Sci., vol. 101, pp. 5228–5235, 2004.
[24] G. Heinrich, “Parameter estimation for text analysis,” University of Leipzig, Tech. Rep., 2008.
[25] L. Dietz, “Directed factor graph notation for generative models,” Max Planck Institute for Informatics, Tech. Rep., 2010.
[26] J. Zeng, L. Xie, and Z.-Q. Liu, “Type-2 fuzzy Gaussian mixture models,” Pattern Recognition, vol. 41, no. 12, pp. 3636–3643, 2008.
[27] J. Zeng and Z.-Q. Liu, “Type-2 fuzzy hidden Markov models and their application to speech recognition,” IEEE Trans. Fuzzy Syst., vol. 14, no. 3, pp. 454–467, June 2006.
[28] J. Zeng and Z.-Q. Liu, “Type-2 fuzzy Markov random fields and their application to handwritten Chinese character recognition,” IEEE Trans. Fuzzy Syst., vol. 16, no. 3, pp. 747–760, 2008.
[29] B. J. Frey and N. Jojic, “A comparison of algorithms for inference and learning in probabilistic graphical models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 9, pp. 1392–1416, 2005.
[30] T. Werner, “A linear programming approach to max-sum problem: A review,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 7, pp. 1165–1179, 2007.
[31] J. Zeng and Z.-Q. Liu, “Markov random field-based statistical character structure modeling for handwritten Chinese character recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 5, pp. 767–780, 2008.
[32] J. Zeng, W. K. Cheung, and J. Liu, “Learning topic models by belief propagation,” arXiv:1109.3437v1 [cs.LG], 2011.
[33] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, “Automating the construction of internet portals with machine learning,” Information Retrieval, vol. 3, no. 2, pp. 127–163, 2000.
[34] S. Zhu, J. Zeng, and H. Mamitsuka, “Enhancing MEDLINE document clustering by incorporating MeSH semantic similarity,” Bioinformatics, vol. 25, no. 15, pp. 1944–1951, 2009.
[35] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth, “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary,” in ECCV, 2002, pp. 97–112.
[36] G. Qiu, “Indexing chromatic and achromatic patterns for content-based colour image retrieval,” Pattern Recognition, pp. 1675–1686, 2002.
[37] J. Chang, J. Boyd-Graber, S. Gerris, C. Wang, and D. Blei, “Reading tea leaves: How humans interpret topic models,” in NIPS, 2009, pp. 288–296.
[38] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 27:1–27:27, 2011.
[39] D. Mimno and A. McCallum, “Expertise modeling for matching papers with reviewers,” in KDD, 2007, pp. 500–509.
[40] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, “Supervised learning of semantic classes for image annotation and retrieval,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 3, pp. 394–410, 2007.