Transfer Learning using a Nonparametric Sparse Topic Model

Ali Faisal[1,*], Jussi Gillberg, Gayle Leen[2], Jaakko Peltonen[1]

Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, Aalto University, P.O. Box 15400, FI-00076 Aalto, Finland

Abstract

In many domains data items are represented by vectors of counts; count data arises for example in bioinformatics or analysis of text documents represented as word count vectors. However, often the amount of data available from an interesting data source is too small to model the data source well. When several data sets are available from related sources, exploiting their similarities by *transfer learning* can improve the resulting models compared to modeling sources independently. We introduce a Bayesian generative transfer learning model which represents similarity across document collections by *sparse sharing of latent topics* controlled by an Indian Buffet Process. Unlike a prominent previous model, Hierarchical Dirichlet Process (HDP) based multi-task learning, our model decouples topic sharing probability from topic strength, making sharing of low-strength topics easier. In experiments, our model outperforms the HDP approach both on synthetic data and in first of the two case studies on text collections, and achieves similar performance as the HDP approach in the second case study.

Keywords:
Transfer learning, latent Dirichlet allocation, nonparametric Bayesian inference, sparsity, small sample size, topic models

1. Introduction

Traditionally machine learning methods learn models for data from a single data source, for example learning a model of news articles posted to a newsgroup or scientific papers submitted to a conference track. Learning the model can be called a *task*. In particular, we consider learning models for *count data*, a prominent type of data that arises in bag-of-words representations of text documents, in bioinformatics for example as counts of active genes over pathways, and in other domains. Latent structure in count data has often been modeled with *topic models* [1], in domains from document collections [2] to bioinformatics [3, 4].

When few training samples are available for the learning task, methods may overfit or have too little information to infer complicated models. To gain more information for the learning task, *transfer learning* [5] methods transfer knowledge from earlier tasks to a new one, and *multi-task learning* [6] methods learn several tasks together from their respective data sets, exploiting their underlying relationships. For example, the data of these related tasks may be articles from other newsgroups or papers from other tracks in the conference.

A particular interesting setting is the case when one task is more interesting than others: in the text data case this could correspond to focusing on creating a model for a particular newsgroup which could be of strong interest to advertisers analyzing the newsgroup that correspond to their business field. Similarly a model for articles from a particular conference section would interest researchers whose research topic matches well with the conference section. In some cases the task of interest may be a new task (a recent newsgroup or conference track) for which less data is available, and multi-task learning is then crucial to learn a good model for it.

When set in the probabilistic modeling framework, transfer learning or multi-task learning approaches typically build a hierarchical model describing how model parameters vary among tasks; models for all tasks are then learned simultaneously. The success of transfer learning and multi-task learning models depends on whether the assumed kinds of relationships between data sources match the real relationships.

In this paper we introduce a multi-task learning (transfer learning) method for an unsupervised multi-task learning problem, *generative modeling of count data in multiple tasks*, such as bag-of-words text documents from several collections. We will model each data source with the topic model family. We propose a nonparametric extension where both the number of topics and their strengths are learned from data. To model sharing of information among tasks, we allow topics to be shared among tasks. We use an Indian Buffet Process (IBP; [7]) to model how many topics are active overall and which topics each task uses to model its respective documents; we allow a further sparsity-inducing step to turn off some topics from each task. Finally we generate the strengths of active topics in each task from a Gamma prior. We use Bayesian inference (MCMC sampling) to infer the posterior over topics and make predictions about new documents as in any Bayesian model.

The most relevant earlier work is the Hierarchical Dirichlet

*Corresponding author
Email addresses: ali.faisal@aalto.fi (Ali Faisal), jussi.gillberg@aalto.fi (Jussi Gillberg), gayle.leen@decode.is (Gayle Leen), jaakko.peltonen@aalto.fi (Jaakko Peltonen)
[1]Authors with equal contribution
[2]now at deCODE Genetics, Reykjavik, Iceland

Process model (HDP; [8]) which extends the single-task Latent Dirichlet Allocation model (LDA; [1]) and learns the number of topics from data by a Dirichlet Process (DP) prior; it is also extended to multi-task problems by modeling topic strengths in each task as draws from an upper-level Dirichlet Process prior; we denote the multi-task version by MT-HDPLDA.

Due to the way topic strengths are hierarchically drawn from Dirichlet Processes, MT-HDPLDA implicitly assumes that the topics most likely to be shared are also the strongest topics, (contributing most of the words in documents). This neglects the possibility of sharing weak topics, and can make it hard to learn such weak shared topics from data. Here "weak shared topics" denotes shared topics that are either weak overall so that they in total contribute only few words in documents, or topics whose overall strength is moderate but whose strength is relatively small in some subset of tasks. The term "weak shared topic" is used only as an informal description of why HDP may poorly represent sharing of some topics; the above described implicit assumption in HDP affects strength and sharing of all topics, and the weaker a shared topic is in some tasks, the harder it may be to represent it properly in an HDP model.

In contrast to MT-HDPLDA, our IBP-based sharing separates the choice of which topics to share from generation of topic strengths, allowing more flexible sharing between multiple tasks. In experiments our model outperforms MT-HDPLDA on several data domains. Another related model is the single-task model in [9], which uses an IBP prior to control which topics are active in each document and draws strengths of active topics from Gamma priors. The model in [9] is for single-task learning only. Our model can be seen as a multi-task counterpart, where the "IBP+Gamma" type generation of topic strengths is used across multiple tasks rather than across documents in one task.

This paper extends our conference paper [10]; the main changes in this journal version are a comparative analysis of our proposed model with the multi-task HDP based LDA approach under varying number of total tasks in a simulation study, a new comparison between the two models on newsgroup data, a discussion of the topics learned by our model for a multi-task collection of scientific articles, and an extended description of the method including detailed equations and derivations for the model inference.

The rest of the paper is organized as follows: Section 2 describes related earlier models, Section 3 describes our model, Section 4 details the inference scheme and equations, Section 5 explains the experimental results while Section 7 concludes the paper.

2. Background

In this section we discuss selected prominent earlier models for count data. We first describe the basic single-task topic model, then describe a nonparametric model where the number of available topics is not restricted, and lastly describe a multi-task extension of the nonparametric model which we will use as a comparison method.

2.1. Single-task topic model

The basic single-task topic model Latent Dirichlet Allocation (LDA; [1]) generates a document through activity of latent *topics*; to generate a document d, a topic distribution $\boldsymbol{\pi}_d$ is drawn from a prior so that $\boldsymbol{\pi}_d \sim \text{Dirichlet}(\alpha)$, and then the words are generated one by one. To generate the nth word in the document, a topic index $z_{d,n}$ is drawn from the topic distribution so that $z_{d,n} \sim \text{Multinomial}(\boldsymbol{\pi}_d)$, and the word is then drawn from a topic-wise word distribution: $w_{d,n} \sim \text{Multinomial}(\boldsymbol{\beta}_{z_{d,n}})$ where $\boldsymbol{\beta}_k = \{\beta_{w|k}\}_w$ are probabilities of each word w in the kth topic. The available topics are the same for all documents. Typically the topic-wise word distributions are drawn from a prior $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\eta)$, where η is the topic hyperparameter. A plate diagram for this generative process is presented in Figure 1. Note that in LDA each word is generated independently given the topic and the order of the word occurrences does not matter; LDA is thus suitable for count data such as bag-of-words representations of text, where only the overall occurrence count of each different word is observed.

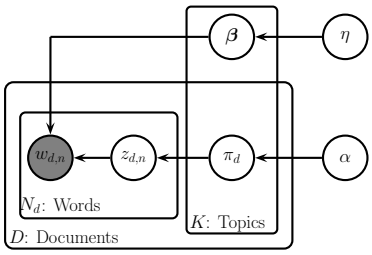

Figure 1: Plate diagram for the basic single task topic model (Latent Dirichlet Allocation). Topic-to-word distributions $\boldsymbol{\beta}$ are first sampled from Dirichlet priors governed by η, then for each document d, topic proportions (topic probabilities) $\boldsymbol{\pi}_d$ are sampled from another Dirichlet prior governed by α, and finally the words in the document are generated by sampling a topic $z_{d,n}$ and sampling the word $w_{d,n}$ itself from the corresponding topic-to-word distribution. Dark shade denotes that the observed variables are counts of how many times each word in the vocabulary appears in a document.

Given a data set of documents, the LDA model can be fitted to the data by maximum a posteriori methods. Note that when the LDA topic model is learned from a data set, the Dirichlet priors for the word distribution somewhat mitigate overfitting when large vocabularies are used, so that words that do not appear in the training set are still assigned some probability to appear in future documents.

The use of Dirichlet priors in LDA stems from convenient properties of the Dirichlet distribution, in particular it has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution. These properties allow some of the parameters to be integrated out analytically when fitting an LDA model; similarly, we will use these properties of Dirichlet distribution in development of inference and parameter estimation algorithms for our model in Section 3.

The LDA model assumes the number K of available topics to be specified in advance. This restriction can be problematic especially for complicated count data sets, where the number of actual underlying topics can be large, and expert knowledge for choosing the correct number of topics may not be available. If

the number of topics is chosen to be too small, fitting the model effectively forces the model to merge some of the real topics in the data. On the other hand, if the number of topics is chosen to be at least as large as the true number of topics, then the model can in principle represent the data correctly; however, fitting the model by maximum likelihood methods will in practice overfit to the limited number of documents and will effectively split some of the real topics according to artifacts in the observed data.

2.2. Nonparametric model for count data: Hierarchical Dirichlet Process

The Hierarchical Dirichlet Process (HDP; [8]) is a Bayesian hierarchical nonparametric model that can be used to generalize LDA to learn the number of topics from data, and can also be used to model multiple document collections (data sets). We first discuss the mathematical form of the hierarchical Dirichlet process and discuss how it is used to create a single-task topic model. We then discuss the multi-task version in the next subsection.

Preliminary: the (hierarchical) Dirichlet process. The Hierarchical Dirichlet Process is a nonparametric prior based on Dirichlet processes (DP; [11]). Dirichlet processes are prior distributions over probability measures; intuitively it is an infinite dimensional generalization of Dirichlet distribution. Measures drawn from a Dirichlet process are discrete with probability one, meaning that the measure gives nonzero probability to a finite number of discrete choices, but the number of available choices can differ between different draws from the Dirichlet process.

The Dirichlet process is defined based on a 'base measure', and a draw from a Dirichlet process effectively redistributes weight among the choices in the base measure, possibly shutting off some of those choices. The choices in a draw from the Dirichlet process are a subset of the choices in the base measure. The draw itself can be used as a base measure for another Dirichlet process.

Formally, a Dirichlet process has two parameters: a base probability measure H, which defines the mean of draws from the process, and a strength parameter $\gamma > 0$ that controls the variability around H. A draw G_0 from a DP is represented as $G_0 \sim \mathrm{DP}(\gamma, H)$ and with probability one G_0 can be represented as $G_0 = \sum_{k=1}^{\infty} \pi_k \delta_{\beta_k}$, where the β_k are random variables distributed according to H and δ_{β_k} is an atom at β_k. The sequence of probabilities $\boldsymbol{\pi} = (\pi_k)_{k=1}^{\infty}$ is defined by the stick-breaking construction [8, 12] of a DP as follows:

$$G_0 \sim \mathrm{DP}(\gamma, H), \quad G_0 = \sum_{k=1}^{\infty} \pi_k \delta_{\beta_k}$$

$$\pi_k = \pi'_k \prod_{l=1}^{k-1} (1 - \pi'_l), \quad \pi'_k \sim \mathrm{Beta}(1, \gamma) \tag{1}$$

where $(\pi'_k)_{k=1}^{\infty}$ are independent sequences of i.i.d. random variables.

Using the Dirichlet process in a topic model. The HDP based single task topic model (HDPLDA; [8]) uses the Dirichlet process to allow a potentially infinite number of topics. The Dirichlet process (or hierarchical Dirichlet process) merely generates a sequence of probabilities; to have a full generative model, the probabilities must be connected to a generative model of the finally observed variables. In HDPLDA, observed variables are counts of words in documents as usual, but now the topics are no longer chosen from a pre-fixed finite number of choices, instead the topics used in a document are drawn from a Dirichlet process.

The topics are drawn as the atoms in a Dirichlet process (DP). Each document has its own DP; to allow sharing of the topics (atoms) among different documents, a shared global DP G_0 is placed as a prior over document level DPs G_d, so that the base measure of each document-level DP is a draw from the global DP. Since the global DP has support (nonzero probability) at the points (topics) $\boldsymbol{\beta} = (\beta_k)_{k=1}^{\infty}$, each G_d necessarily has support at a subset of these points. Then G_d can be written as:

$$G_d \sim \mathrm{DP}(\alpha_0, G_0), \quad G_d = \sum_{k=1}^{\infty} \pi_{d,k} \delta_{\beta_k}$$

$$\pi_{d,k} = \pi'_{d,k} \prod_{l=1}^{k-1} (1 - \pi'_{d,l}), \quad \pi'_{d,k} \sim \mathrm{Beta}\left(\alpha_0 \pi_k, \alpha_0 \left(1 - \sum_{l=1}^{k} \pi_l\right)\right)$$

To sample a topic for a word in document d, the probabilities $\pi_{d,k}$ are used as the topic probabilities. The rest of the model is essentially the same as the basic LDA: the observed word are generated from the topic-to-word distribution of the chosen topic. The topic-to-word distribution of each topic is sampled from a Dirichlet prior; the distribution only needs to be sampled only for those topics that are actually used over the document collection.

2.3. Multi-task extension of the HDPLDA model

The multi-task extension of HDPLDA models several document collections (data sets, also denoted as tasks), by taking the hierarchy of Dirichlet processes one level higher: in single-task HDPLDA the topics over the document collection were controlled by an overall DP, but in the multi-task extension each document collection has its own overall DP, which are in turn drawn from a top-level DP which controls topics over all the document collections.

Technically, a data set level DP $G_c \sim \mathrm{DP}(\alpha_0, G_0)$ is introduced in the HDP prior: Inside each document collection (data set) c, a document level DP $G_d \sim \mathrm{DP}(\alpha_c, G_c)$ is drawn for each document from a data set level DP. The data set level DP, G_c, can in turn be drawn from an overall DP across data sets, with base measure H. The rest of the model is again similar to the basic LDA: topic-to-word distributions are drawn for the topics in use, and after drawing a topic the observed word is drawn from the corresponding topic-to-word distribution. See Figure 2 for the plate diagram of the resulting multitask HDPLDA, based on the stick-breaking representation.

In this hierarchical generative process, the topmost DP in the hierarchy determines which topics are active overall and their

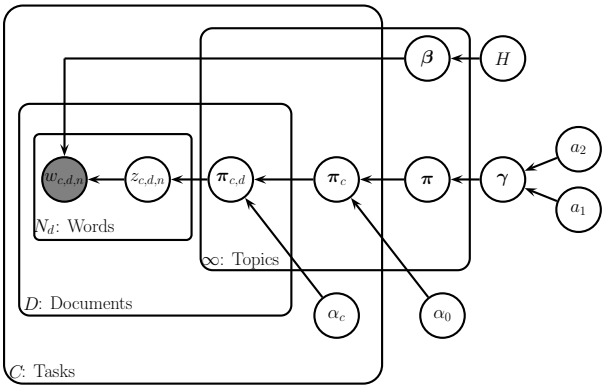

Figure 2: Plate diagram for multitask HDPLDA, a nonparametric topic model for multiple collections. In each document collection (task) the overall topic distribution is controlled by a task-specific Dirichlet process, which are in turn drawn from an overall Dirichlet process controlled by a base measure H. Otherwise the generative process is the same as for the single-task HDPLDA.

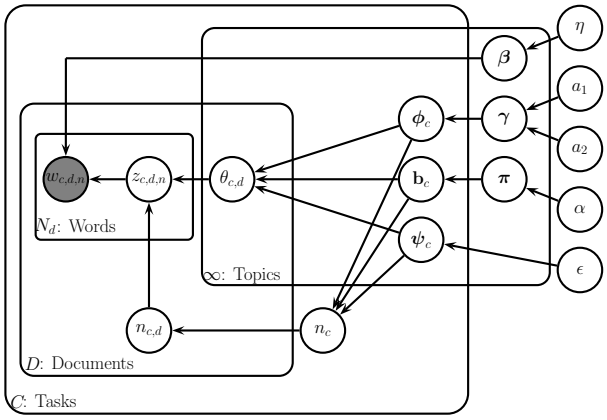

Figure 3: Plate diagram for our sparse transfer learning topic model. Notice the parameters γ, π and hyperparameters a_1 and a_2 have a different meaning that the MT-HDPLDA model. See Table 1 for notation and Section 3 for explanation of the generative process.

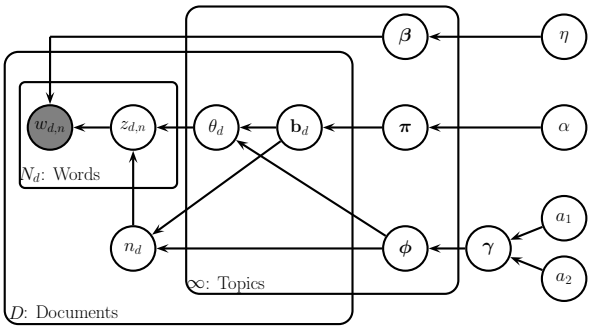

Figure 4: Plate diagram for the single task model of [9]. Topic-to-word distributions $\boldsymbol{\beta}$ are first sampled from Dirichlet priors governed by η, then for each document d, topic proportions (topic probabilities) $\boldsymbol{\theta}_d$ are sampled from another Dirichlet prior governed by document level topic presence; ($\mathbf{b}_d$) and global topic strength parameters; $\phi_k \sim$ Gamma(γ, 1). Otherwise the generative process is the same as for the single-task LDA. Notice the parameters γ, π and hyperparameters a_1 and a_2 have a different meaning than in the MT-HDPLDA and our model. For details about the model the reader should refer to [9].

strengths; lower-level DPs choose among their parent-level active topics, varying their strengths by the previously detailed stick-breaking construction, to yield differing topic distributions at each branch of the hierarchy. When inferring topics from data, the topmost DP can activate new topics as well as change their strength, and the activated new topics can then be assigned nonzero probability on the next hierarchy level for each document collection; the HDP can thus infer the number of topics from data. See [8] for further details.

The HDPLDA model has a potential problem due to an implicit assumption about the topic sharing: since the sharing is done by the topic strength hierarchy (topic probability hierarchy), with the stick-breaking construction the strongest topics (which generate many words overall) are the most likely to survive in several branches of the hierarchy and thus be shared across data sets. This property can make the HDPLDA model a bad fit for multi-task problems with low-strength shared topics (topics discussed in many document collections but not at great length).

2.4. Single-task topic model with flexible sharing

Recently a single-task model with more flexible topic sharing was proposed [9] using an Indian Buffet Process Compound Dirichlet Process prior which can be seen as a *spike-and-slab prior* [13] over topic strengths. An Indian Buffet Process prior is placed on binary flags of whether topics are present in documents, rather than on the strengths of the topics; rows of the IBP correspond to documents and columns correspond to topics. The topic strengths are generated separately from Gamma variables. In this way, the model avoids the coupling of topic strength and topic sharing implicit in the HDP model. The plate diagram for the model is presented in Figure 4.

Technically, in the HDP based prior the strength of topics was generated at the same time as their sharing, through the stick-breaking construction: topics that occurred later in the order of the stick breaking process were likely to get lower strength.[1] In contrast, in the IBP Compound DP prior the strength of topics is assigned independently of their order in the IBP process that generates the binary sharing matrix.

Note that like the HDP based prior, the IBP Compound DP prior allows a potentially infinite number of active topics, yet sampling only requires finite computational effort. Because the sampling of the IBP matrix is based on a stick-breaking construction, the sampled binary IBP vector for each document almost surely contains a finite number of active topics, hence the whole document collection will contain a finite number of active topics. The overall prior for strengths of the topics can then be sampled by sampling a Gamma-distributed strength variable for each active topic in the collection. The topic probability vector for each individual document is then sampled by turning off topics that are inactive in the document according to the IBP, and sampling the probabilities of the remaining topics according to their strengths in the prior.

[1] It is easy to show that topic weights in the top-level Dirichlet process are

The model of [9] is for single-task learning only, and the IBP is defined to model the sharing of topics among documents from a single data source (document collection). It cannot model relationships between several data sources. When only few data are available from each data source, a multi-task solution is needed. The model that we propose in the next section is for a multi-task scenario, and we will use an IBP based construction to model sharing of topics among several data sources. The essential difference between our new model and [9] is then that we handle the multi-task learning case, and focus our modeling effort on modeling the sharing between tasks.

3. New sparse nonparametric topic model for transfer learning

We present a new hierarchical Bayesian multi-task (transfer learning) model which allows flexible sharing of low-strength and high-strength topics across multiple data sets, with a spike-and-slab prior. Learning the model for each data set is called a task; our model performs transfer learning by learning the tasks together.

Preliminary: the Indian Buffet Process. In our model, we will draw a binary matrix that indicates which topics are present in each task. The matrix will be drawn from an Indian Buffet Process (IBP; [7]), which is a nonparametric prior over binary matrices. The use of the IBP prior ensures that the number of topics does not need to be fixed and can instead be learned from the data. The IBP prior allows a potentially infinite number of active topics, but each draw from the prior yields some finite number of active topics.

The IBP prior can be derived by as a limit of finite-sized binary matrices: if K is the number of columns in a binary matrix, then the IBP is the limit, when K approaches infinity, of a finite $C \times K$ binary matrix $\mathbf{B}$ whose elements $b_c^{(k)}$ are distributed according to: $\pi^{(k)} \sim \text{Beta}(\alpha/K, 1)$ and $b_c^{(k)} \sim \text{Bernoulli}(\pi^{(k)})$, where the cth row of $\mathbf{B}$ is $\mathbf{b}_c$. The $\pi^{(k)}$ is probability of turning on an entry in the kth column of the matrix.

In the limit when $K \to \infty$ the $\pi^{(k)}$ has been shown ([14]) to obey the following stick-breaking construction:

$$v^{(k)} \overset{iid}{\sim} \text{Beta}(\alpha, 1)$$
$$\pi^{(k)} = v^{(k)}\pi^{(k-1)} = \prod_{j=1}^{k} v^{(j)} \tag{2}$$

The construction can be understood as follows; consider a stick of length 1, at each iteration $k = 1, 2, ...$, we break off a piece at a point $v^{(k)}$ relative to the current length of the stick $\pi^{(k-1)}$. We record the length $\pi^{(k)}$ of the stick we just broke off and recurse on this piece. The sequence produces a decreasing ordering of latent probabilites $\pi^{(k)}$ which can be used as a prior over unbounded binary matrices;

$$b_c^{(k)} \sim \text{Bernoulli}(\pi^{(k)}) \qquad \text{for each } c. \tag{3}$$

In our model, the columns of the IBP correspond to topics and the rows represent different tasks. Thus an entry in the matrix indicates which topic contributes to which task.

Our model: nonparametric transfer learning topic model based on the IBP. In our model the rows of the matrix $\mathbf{B}$ represent different tasks (the number of document collections), the columns represent topics, and the individual binary entries $b_c^{(k)}$ indicate whether topic k is present in task c. To draw a topic for a new task, the IBP chooses one of the existing topics according to how many tasks they are already present in, or activates a new topic. Therefore the IBP can choose to increase the number of topics with no upper limit; when fitting a topic model with an IBP prior, the number of active topics is then inferred from data.

Note that we use the IBP prior differently from the single-task model [9] discussed in Section 2.4; that model used the IBP to draw the presence of topics across different documents of the same collection, we use the IBP in a multi-task context, to draw the presence of topics across different document collections (tasks), consequently our IBP matrix has only one row per each task (not one row per document as in [9]).

We empirically found that in our setting IBP by itself does not provide enough sparsity. This is because the IBP matrix has just one row per task, so the IBP parameters are learned from few observations (the matrix rows), which leaves the IBP uncertain about the number of active topics and hence causes it to activate more topics than really needed. This formulation of IBP is necessary to decouple topic sharing from topic strength. To combat the unwanted effect of activating too many topics, we incorporated an additional new *sparsity-inducing masking step*: for each topic in each task, the sparsity inducing masking step simply turns off the topic with probability ϵ.

After the two topic selection operations (IBP and the additional masking, together denoted *IBP-masking*) have been done, the strength of remaining active topics is drawn from Gamma distribution within each task; these strengths define the prior distribution of topic activities within the task. The combination of the Gamma-distributed topic strengths and the IBP-masking can be seen as an infinite *spike and slab* prior, where the IBP-masking generates the spikes (possibility for a topic to be turned completely off) and the Gamma distribution acts as a slab (which generates the strengths of topics that are not turned off). The use of the independent topic strength variables avoids the restrictions imposed by the DP construction of Section 2.3; it makes inference easier and is able to model weak topics by decoupling the strength and presence of a topic.

When the task-specific topic priors have been generated, the rest of the generative process proceeds within each task as in LDA: for each topic that is active in any task, a topic-to-word distribution is drawn from a Dirichlet prior, and documents

upper bounded by a monotonously decreasing sequence; we can simply rewrite Equation (1) for the kth stick weight as $\pi_k = \pi'_k\tilde{\pi}_k$ where $\tilde{\pi}_k = \prod_{l=1}^{k-1}(1-\pi'_l)$ and π'_l are random variables between 0 and 1. We thus have $\pi_k \leq \tilde{\pi}_k$ where the $\tilde{\pi}_k$ are a monotonously decreasing sequence, therefore topics far in the stick breaking process (having large k) are likely to get small weights. On the lower levels of the HDP, the topic weights are sampled using the upper-level DP as a base distribution, and therefore topics with very small weight at the top level are unlikely to get large weight on the lower levels anymore.

Algorithm 1 Pseudo-code for our multi-task topic model Gibbs sampler. Text after symbol '▷' are comments.

1: **for** *iter* = 1 to ITER **do**
2: **for** $c = 1$ to TOTAL_TASKS **do**
3: **for** $d = 1$ to TOTAL_DOCUMENTS_IN_TASK **do**
4: **for** Each word $w_{c,d,n}$ in document d **do**
5: $k \leftarrow z(n)$ ▷ Get topic assignment
6: Decrease $n^{(k)}_{w_{c,d,n}}$ and $n^{(k)}_{(.),(.)}$ by 1
7: Decrease $n^{(k)}_{c,d}$ and $n^{(k)}_{c}$ by 1
8: **for** $k = 1$ to TOTAL_TOPICS +1 **do** ▷ The +1 is for the inactive topic
9: $p(k) \leftarrow \frac{n^{(k)}_{w_{c,d,n}}+\eta}{n^{(k)}_{(.),(.)}+\eta \text{VocabSize}} \text{E}[\theta^{(k)}_{c,d}]$ ▷ For expectation use; Eq. (A.11), (A.12) and (A.13)
10: **end for** ▷ 'VocabSize' is the number of different words in the vocabulary.
11: $k \leftarrow \text{sample}(p)$
12: $z(n) \leftarrow k$
13: Increase $n^{(k)}_{w_{c,d,n}}$ and $n^{(k)}$ by 1
14: Increase $n^{(k)}_{c,d}$ and $n^{(k)}_{c}$ by 1
15: **if** $k >$ ACTIVE_TOPICS **then** [3]
16: $p(b^{(k)}_c = 1) \leftarrow$ Eq. (8)
17: $p(\psi^{(k)}_c = 1) \leftarrow$ Eq. (8) by replacing $b^{(k)}_c$ by $\psi^{(k)}_c$ and $\pi^{(k)}$ by ϵ
18: Sample $\pi^{(k)^\bullet}$ and $\pi^{(k+1)^\circ}$ using Eq. (6) and Eq. (7) with details in [14]
19: Sample $\phi^{(k)}_{c,d}$ using Eq. (9)
20: Sample $\gamma^{(k)}$ using Eq. (10)
21: **end if**
22: **end for**
23: **end for**
24: **for all** $k = 1$ to TOTAL_TOPICS **do**
25: Reinitialize $b^{(k)}_c$ and $\psi^{(k)}_c$ as before
26: Sample $\pi^{(k)}$ and $\phi^{(k)}_c$ as before
27: **end for**
28: **end for**
29: **end for**

within a task are generated as usual by drawing a topic distribution from the task-specific topic prior and then drawing the words for each document.

The notation we use for our model is summarized in Table 1. The full generative scheme for our model (corresponding to the plate model in Figure 3) is as follows:

1. For each topic $k = 1, 2, ...,$ draw,
   (a) topic strength prior $\gamma^{(k)} \sim \text{Gamma}(a_1, a_2)$
   (b) IBP probablility of topic activation $\pi^{(k)}$ from Eq. 2
   (c) topic-to-word distributions $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\eta)$

2. For each topic k in task $c = 1, 2, .., C$ draw,
   (a) topic strength $\phi^{(k)}_c \sim \text{Gamma}(\text{shape} = \gamma, \text{scale} = 1)$
   (b) IBP topic activation $b^{(k)}_c$ from Eq. 3
   (c) additional sparsity masking $\psi^{(k)}_c \sim \text{Bernoulli}(\epsilon_c)$

3. Draw the size of the task; total number of word occurrences, $n^{(.)}_c \sim \text{NB}(\sum_k b^{(k)}_c \phi^{(k)}_c \psi^{(k)}_c, \frac{1}{2})$[2]

4. For every document $d = 1, 2, ...D$ in task c,
   (a) draw distribution over topics $\boldsymbol{\theta}_{c,d} \sim \text{Dirichlet}(\mathbf{b}_c.\boldsymbol{\phi}_c.\boldsymbol{\psi}_c)$
   For each word $n = 1, 2, ..., N_d$ in the document
   (b) Draw the topic index $z_{c,d,n} \sim \text{Multinomial}(\boldsymbol{\theta}_{c,d})$
   (c) Draw the word term $w_{c,d,n} \sim \text{Multinomial}(\boldsymbol{\beta}_{z_{c,d,n}})$

Note that the Dirichlet distribution is defined based on pseudocounts, which are here an elementwise multiplication of the binary IBP flags $\mathbf{b}_c$, the additional sparsity-inducing masking $\boldsymbol{\psi}_c$, and the topic strengths $\boldsymbol{\phi}_c$; any topic which has been turned off by the IBP or the additional masking gets a zero pseudocount, hence draws from the Dirichlet distribution always yield zero probability for such topics, as desired.

[2] As a simplification, our model generates the total numbers of words per task but not how this total is divided among the individual documents. Essentially this means that fitting the model does not draw information from the size variation between documents, only from the total size variation between tasks. In the plate diagram of Figure 3 we mark the sizes $n_{c,d}$ of individual documents for clarity since they affect the generation of the document content.

[3] In our implementation we have the *if* clause (line no. 15 – 21) outside the *for* loop over words (line no. 4); this helps us speeds up our implementation. To cater for the new inactive topics that might emerge for subsequent words; we sample a series of inactive topic stick parameters before entering the *for* loop.

In the above-described generative process, the set of hyper-parameters are: $\{\alpha, \epsilon, \eta, a_1, a_2\}$ and the unknown model parameters are: $\{\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{B}, \boldsymbol{\psi}, \boldsymbol{\phi}\}$. To fit the model to data in a Bayesian fashion, we infer the posterior of the model parameters given the observed word counts in all documents of all tasks. Inference by sampling, discussed next, is efficient and only processes a finite number of topics at each step as is usual in non-parametric models.

4. Bayesian inference for our model

To infer our model from the multi-task data sets (document collections), we use a combination of collapsed Gibbs sampling and the Metropolis-Hastings algorithm to sample from the posterior distribution of the model parameters. It turns out it is possible to directly integrate out some of the 'nuisance' model parameters; then the posterior of the rest of the variables can be sampled more efficiently. We integrate out the topic specific distribution over words $\boldsymbol{\beta}$, the topic mixture distribution $\boldsymbol{\theta}$ and the binary IBP matrix $\mathbf{B}$; sampling is needed only over the remaining variables. In the Gibbs sampling we cyclically sample the topic assignment z, the topic strength ϕ and the IBP prior π (stick-breaking parameters) for topic activation. Algorithm 1 presents the complete pseudocode for the algorithm and includes references to the sampling distributions discussed in the following section.

4.1. Sampling z_k and the stick parameter π_k

To sample topic assignments within a document d in task c, we integrate out the topic distribution $\boldsymbol{\theta}_{c,d}$ of the document. The posterior probability that the n^{th} word in document d of task c comes from topic k is

$$\begin{aligned} &p(z_{c,d,n} = k|\mathbf{z}_{\setminus c,d,n}, w_{c,d,n}, \Delta) \\ &= \frac{p(w_{c,d,n}|z_{c,d,n} = k)p(z_{c,d,n} = k|\mathbf{z}_{\setminus c,d,n})p(\mathbf{z}_{\setminus c,d,n}, \Delta)}{p(w_{c,d,n}, \mathbf{z}_{\setminus c,d,n}, \Delta)} \\ &\propto p(w_{c,d,n}|z_{c,d,n} = k)p(z_{c,d,n} = k|\mathbf{z}_{\setminus c,d,n}, \Delta) \\ &\propto (n^{(k)}_{w_{c,d,n},\setminus c,d,n} + \eta) \int d\boldsymbol{\theta}_{c,d}\, p(z_{c,d,n} = k|\boldsymbol{\theta}_{c,d})p(\boldsymbol{\theta}_{c,d}|\mathbf{z}_{\setminus c,d,n}, \Delta) \end{aligned} \tag{4}$$

where $\mathbf{z}_{\setminus c,d,n}$ denotes the current values of all other topic assignments except the one whose probability we are computing, $\Delta = \{\boldsymbol{\phi}^{\bullet}_c, \boldsymbol{\pi}^{\bullet}, \boldsymbol{\gamma}, \alpha, \epsilon\}$, and the superscript $^{\bullet}$ denotes active topics. The first equality follows from the Bayes rule since the word $w_{c,d,n}$ only depends on the topic $z_{c,d,n}$. The second proportionality follows by explicitly writing out how the (posterior) word probability in topic k depends on word counts and prior pseudocounts, and by explicitly writing the probability of choosing topic k as an integral over the posterior of the latent topic probability variable $\boldsymbol{\theta}_{c,d}$.

On the right-hand side of (4), we simply have $p(z_{c,d,n} = k|\boldsymbol{\theta}_{c,d}) = \boldsymbol{\theta}^{(k)}_{c,d}$ which is the kth value of the topic-probability vector $\boldsymbol{\theta}_{c,d}$; therefore the integral on the right-hand side of (4) is an expectation of the topic probability. However, that expectation is taken over a complicated posterior distribution of topic probabilities, where $p(\boldsymbol{\theta}_{c,d}|\mathbf{z}_{\setminus c,d,n}, \Delta) \propto \int d\boldsymbol{\phi}^{\circ}_c \sum_{\mathbf{b}_c} \sum_{\boldsymbol{\psi}_c} p(\boldsymbol{\theta}_{c,d}|\boldsymbol{\psi}_c, \mathbf{b}_c, \boldsymbol{\phi}_c, \mathbf{z}_{\setminus c,d,n}) p(\mathbf{b}_c, \boldsymbol{\psi}_c, \boldsymbol{\phi}^{\circ}|\Delta)$.This likelihood involves a combinatorial integration over values of the sparse IBP matrix, but since we only need the posterior for taking topic k, it can be shown (refer to equation (A.2) in Appendix A for derivation) that the integral on the right-hand side of (4) ultimately simplifies to

$$E[\boldsymbol{\theta}^{(k)}_{c,d}|\mathbf{z}_{\setminus c,d,n}, \Delta] \propto E\left[\frac{(n^{(k)}_{c,d,\setminus c,d,n} + \phi^{(k)}_c) b^{(k)}_c \psi^{(k)}_c}{n^{(.)}_{c,d,\setminus c,d,n} + \sum_j b^{(j)}_c \psi^{(j)}_c \phi^{(j)}_c}\right]. \tag{5}$$

In the above equation, $n^{(k)}_{c,d,\setminus c,d,n}$ is the number of words in the document assigned to topic k not counting the nth word, and $n^{(.)}_{c,d,\setminus c,d,n}$ is the total number of words in the document not counting the nth word. While not combinatorial, the expectation in (5) is inefficient to evaluate in closed form as we would need to do so for every word during the Gibbs sampling. We use an approximation similar to [9], using 1st order Taylor expansion for the three possible cases: topic k is active in the current task (data set); topic k does not appear in the current task but is active in the corpus (all data sets); or topic k is inactive in the whole corpus. Details of the approximation are provided in Appendix A.

During the Gibbs sampling (including the above-mentioned approximation) we must process *inactive topics* in case the sampling activates one; the ability to activate new topics is essential so we can learn the number of topics from data instead of pre-specifying it. In particular, we must be able to sample the IBP prior (stick-breaking parameters) for both inactive and active topics. A topic is inactive (denoted by a superscript $^{\circ}$) if it is never used in the whole corpus, i.e. the total number of word occurrences assigned to topic k is $n^{k}_{(.)(.)} = 0$, and active otherwise. Note that a topic without any word occurrences assigned to it is considered inactive even if the IBP had enabled the topic for some tasks so that $\sum_c b^{(k)}_c > 0$; for sampling active and inactive topics we follow [14]. For active topics stick lengths $\pi^{\bullet}_k$ have the conditional distribution;

$$p(\pi^{\bullet}_k|\mathbf{B}) \sim \text{Beta}\left(\sum_{c=1}^{C} b^{(k)}_c, 1 + C - \sum_{c=1}^{C} b^{(k)}_c\right). \tag{6}$$

where B is the current value of the IBP (the binary matrix).The posterior can be sampled directly using Gibbs sampling. To sample the stick parameters for the inactive topics we follow the *semi-ordered stick breaking construction* [14]; consider $K^{\dagger}$ be an index such that all active topics have index $k < K^{\dagger}$; thus all topics beyond index $K^{\dagger}$ have no word occurrences assigned to them, denote this by $\mathbf{z}_{k:k>K^{\dagger}} = 0$. The inactive topics have an ordering of decreasing stick lengths: the stick length distribution of the inactive topic k, given the stick length of the previous

Table 1: Notation used for our model

Parameter	Meaning
n	index for the n^{th} word token in a document.
$w_{c,d,n}$	contains the vocabulary index of the n^{th} word token in document d of task c.
$z_{c,d,n}$	topic assignment of the n^{th} word token in document d of task c.
$n_{c,d}$	total no. of words in document d of task c.
n_c	total no. of words in all document of task c.
$n^{(k)}_{(.)(.)}$	total no. of words assigned to topic k in the whole corpus.
$n^{(k)}_{c,d}$	total no. of words assigned to topic k in the document d of task c.
$n^{(k)}_{w_{c,d,n}}$	total no. of times the term $w_{c,d,n}$ has been assigned to topic k in the document d of task c.
$n^{(.)}_{c,d,\backslash c,d,n}$	total no. of words in the document not counting the n^{th} word.
$n^{(k)}_{c,d,\backslash c,d,n}$	total no. of words in the document assigned to topic k not counting the n^{th} word.
$\boldsymbol{\theta}_{c,d}$	topic mixture distribution for the document d of task c.
$\boldsymbol{\beta}_k$	topic specific distribution over words for topic k.
η	prior for the topic specific distribution over $\boldsymbol{\beta}_k$.
$\mathbf{B}$	a $C \times K$ binary IBP matrix where C is the number of tasks and K is the current number of topics in the IBP matrix, and the rows and columns index tasks and topics respectively.
$\mathbf{b}_c$	a binary vector which is the c^{th} row of the IBP matrix and indicates which topics should be turned off in task c.
$\pi^{(k)}$	probability of turning on the k^{th} topic in the IBP matrix.
$^\bullet$	The superscript $^\bullet$ denotes active topics; the ones that are currently represented in the corpus.
$^\circ$	The superscript $^\circ$ denotes inactive (unused) topics; their corresponding parameter values are unknown.
$\phi^{(k)}_c$	strength parameter for topic k in task c. Strenghts of all topics in task c are together denoted as $\boldsymbol{\phi}_c$.
$\gamma^{(k)}$	parameter for topic k in the prior for topic strengths; the parameters together are denoted by $\boldsymbol{\gamma}$ and they define the prior for all $\boldsymbol{\phi}_c$.
a_1, a_2	hyperparameters for the gamma prior over the topic strength prior: a_1 is the shape and a_2 is the scale parameter. Each topic strength prior parameter $\gamma^{(k)}$ is drawn from the gamma distribution defined by a_1 and a_2.
$\boldsymbol{\psi}_c$	sparsity inducing binary masking vector which tells which topics should be turned off in task c.
ϵ	probability of turning on a topic in the sparsity inducing binary vector; $\boldsymbol{\psi}_c$.

topic, is

$$P(\pi^\circ_k|\pi^\circ_{k-1}, \mathbf{z}_{k:k>K^\dagger} = 0)$$
$$\propto \exp\Big(\sum_{i=1}^{N}\frac{1}{i}(1-\pi^\circ_k)^i\Big)(\pi^\circ_k)^{\alpha-1}(1-\pi^\circ_k)^N \cdot \mathbb{I}(0 \le \pi^\circ_k \le \pi^\circ_{k-1}) \quad (7)$$

where $\mathbb{I}(0 \le \pi^\circ_k \le \pi^\circ_{k-1})$ is 1 when the statement inside the parenthesis is true and 0 otherwise. Using (7), we sample the stick parameters for the inactive topics by adaptive rejection sampling (ARS) [4] [15]. ARS samples from a distribution $p(x)$ by first constructing an envelope function for $\log(p(x))$. The envelope function is then used for rejection sampling. Whenever a sample is rejected, the envelope function is updated to correspond better to the underlying density. The R package *ars* [15] is used to generate samples using ARS.

4.2. Reinstantiating the IBP and Bernoulli masking matrices

Even though topic assignments can be sampled while integrating over the binary IBP matrix, the IBP matrix is still temporarily required here for sampling the stick parameters for the active topics in (6); more precisely, the values $b^{(k)}_c$ in all rows c of the kth matrix column are needed to sample the stick parameter of active topic k. For this purpose, the current value of the IBP matrix is reinstantiated based on the known values of the other parameters, according to

$$p(b^{(k)}_c = 1|\pi^{(k)}, \phi^{(k)}_c, \psi^{(k)}_c, n^{(k)}_c) =$$
$$\begin{cases} 1: & \text{if } n^{(k)}_{c,(.)} > 0 \\ \pi^{(k)}: & \text{if } n^{(k)}_{c,(.)} = 0, \psi^{(k)}_c = 0 \\ \frac{\pi^{(k)}}{\pi^{(k)}+2^{\phi^{(k)}_c}(1-\pi^{(k)})}: & \text{if } n^{(k)}_{c,(.)} = 0, \psi^{(k)}_c = 1 \end{cases} \quad (8)$$

where on the right-hand side, the topmost choice simply means that the topic must be activated for the task if some word is already assigned to it; the middle choice means that if the topic is unused and moreover the additional masking has turned it off, then the activation probability comes from the prior and the bottom choice means that if the topic is unused but the additional masking has not turned it off, then the activation probability is derived through the IBP and the total number of words assigned to the k-th topic in task c. The additional masking vector $\boldsymbol{\psi}_c$ is initialized by a similar equation as the IBP matrix by interchanging $b^{(k)}_c$ with $\psi^{(k)}_c$ and π_k with ϵ.

4.3. Sampling topic strength parameters

Lastly, to sample the topic strength parameters, we first compute the joint probability of the strength $\phi^{(k)}_c$ of topic k in task

[4] Multiple samples were generated and an average was used to get a better approximation.

c and the total number of counts assigned to topic k in the task; the joint probability depends on the corresponding topic strength prior parameter $\gamma^{(k)}$, the IBP matrix value $b_c^{(k)}$, and the additional masking value $\psi_c^{(k)}$, as follows:

$$p(\phi_c^{(k)}, n_{(.)}^{(k)} | \gamma^{(k)}, b_c^{(k)}, \psi_c^{(k)}) = \frac{(\phi_c^{(k)})^{\gamma^{(k)}-1} e^{-\phi_c^{(k)}}}{\Gamma(\gamma^{(k)})} \prod_{c: b_c^{(k)}.\psi_c^{(k)}=1}^{C} \frac{\Gamma(n_c^{(k)} + \phi_c^{(k)})}{\Gamma(\phi_c^{(k)})\, n_c^{(k)}!\, 2^{(\phi_c^{(k)} + n_c^{(k)})}} \tag{9}$$

where the right-hand side follows because the topic strength has a Gamma prior with parameter $\gamma^{(k)}$ and the total number of words assigned to the k-th topic in the c-th task is distributed according to $n_c^{(k)} \sim \text{NB}(b_c^{(k)}\phi_c^{(k)}\psi_c^{(k)}, 1/2)$.

We use Metropolis-Hastings to compute the posterior for $\phi_c^{(k)}$. We sample the prior topic strength parameter $\gamma^{(k)}$ in a similar manner from the joint posterior for $\gamma^{(k)}$ and the topic strengths $\phi_{(.)}^{(k)}$: the result is

$$p(\gamma^{(k)}, \phi_{(.)}^{(k)} | n_c^{(k)}, b_c^{(k)}, \psi_c^{(k)}, a_1, a_2) = p(\gamma^{(k)} | a_1, a_2) \prod_{c: b_c^{(k)}.\psi_c^{(k)}=1}^{C} p(\phi_{(c)}^{(k)} | n_c^{(k)}, \gamma^{(k)}, b_c^{(k)}, \psi_c^{(k)}) \tag{10}$$

5. Empirical results

We compare our model to the nearest method Hierarchical Dirichlet Process based multi-task learning (MT-HDPLDA).

Model Selection. The hyperparameters ϵ, η, a_1, a_2 and α can have a clear effect on the results. The precise values are listed in the *Experiment* sections that follow, here we briefly discuss their roles; Smaller values of ϵ lead to less active topics. In the experiments we set ϵ by a very simple manner according to the average number of documents per task: since the artificial data experiments have few documents per task we use the same moderately large ϵ value 0.01 in all artificial data runs; since the real data experiments have more documents per task we use a small ϵ value 0.0001 in all real data runs. The topic distribution prior controlled by η is also found in MT-HDPLDA and has the same meaning; small η would yield more specific topics; for a discussion of the parameter see [2] (in that paper η is called β). Our real data experiments (Sections 5.3 and 5.4) are similar to the ones used by the authors of MT-HDPLDA in [8], so we follow them and use the same value of η. In our simulated data experiments (Sections 5.1 and 5.2) the data size is small and we aim to extract fine grained topics; therefore we use a smaller value of η. The hyperparameters of our model a_1, a_2 and α have the same meanings as in the IBP compound Dirichlet prior of [9]; a_1 and a_2 are the 'shape' and 'scale' hyperparameters for the Gamma distribution of topic strengths (large a_1 linearly increases mean and variance of topic strenghts; large a_2 linearly increases the mean and quadratically increases variance of topic strenghts), and α sets the prior for the stick-breaking in the IBP (large α decreases the number of active topics). We set a_1, a_2 and α as in [9].

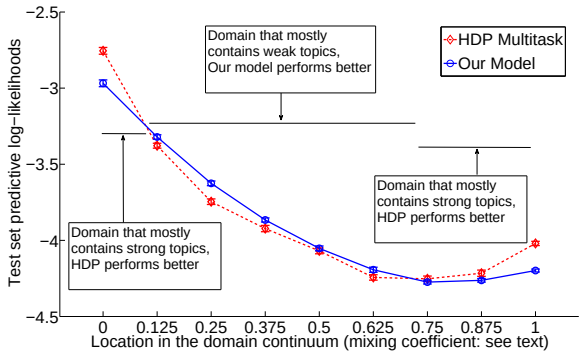

Figure 5: Experiment results: test set predictive likelihoods for simulated data continuum, error bars show ±1 standard deviation over 10 random datasets.

5.1. Experiment 1: Continuum of problem domains

We expect our model to perform well in the case of multi-task problems where some shared topics are strong in all tasks where they appear whereas other shared topics are only weakly present in several tasks; we build a continuum of multi-task problem domains where this situation occurs. At either end of the continuum, data is generated from a model where shared topics are strong (they generate many words in all tasks where they appear); the left end is a simpler case where both models can work well, and the right end is a complicated case especially suitable for MT-HDPLDA. Interesting domains lie between the two ends: in these intermediate domains, the topic generation mechanisms from either end are mixed together linearly, yielding small shared topics from both generators in each individual task. We create nine domains across the continuum, identified by the mixing coefficient (0 to 1) between the generators. See Appendix B for a detailed description of the construction of the synthetic data continuum.

Each problem domain is a multi-task scenario where each learning problem has 10 tasks (data sets). We use the setting where one task is more interesting than others; the interesting task has 24 documents with 8 words each, other tasks have 8 documents with 8 words each, all generated from 10 topics with a vocabulary of 150 words. We generate 10 such learning problems in each domain and run our method and MT-HDPLDA on each problem. We initialize the Gibbs sampler randomly, take 1500 burnin iterations and draw 100 samples 15 iterations apart.

For setting the hyperparameters we follow [8] for MT-HDPLDA and for ours we use $\alpha = 5$ and $\gamma \sim Gamma(5, 0.1)$ following [9]. We fixed $\epsilon = 0.01$ as discussed in the Model selection paragraph earlier. In both our model and the MT-HDPLDA we use a relatively small value of $\eta = 0.00005$.

The results are evaluated by predictive likelihood on held-out documents from the interesting task using the empirical likelihood based approach [16]. Figure 5 shows that in the intermediate domains where weak topics are shared in the interesting task, we outperform MT-HDPLDA.

It should be noted that the horizontal axis in Figure 5 is over a continuum of very different prediction problems, and the scale of results is not intended to be comparable between different parts of the continuum: rather, the take-home message is that

our method is better at five locations in the middle of the continuum where weak shared topics are likely to appear in the interesting task.

5.2. *Experiment 2: Model performance under varying number of total tasks*

In this experiment we evaluate the performance of the two models when the total number of tasks are varied. We fix the location in the intermediate domain continuum at point 0.25 of the mixing coefficient (from Figure 5) such that there are weak topics in the data generation and expand it further such that the total number of tasks is varied from 5 to 30. The rest of the experimental setting is the same as before. We use the same evaluation criterion as before, predictive likelihood on held-out documents from the interesting task. Figure 6 shows the results: under the interesting case when the total number of tasks is relatively small we outperform MT-HDPLDA. When the number of tasks grows, performance of both methods increases and the methods become comparable at the end of many tasks.

We further investigate the effect of total number of tasks on the performance of two models at two other points in the domain continuum, corresponding to mixture coefficient 0.5 and to mixture coefficient 1; for the latter coefficient the data generation assumptions match those of MT-HDPLDA. The resulting predictive likelihoods are plotted in Figure 7 and Figure 8 respectively. In these domains MT-HDPLDA performs better for a large number of tasks; however, if the number of tasks is small (near the left end of the horizontal axes in the figures) our model performs better than MT-HDPLDA, even in the case of the domain with mixture coefficient 1 (Figure 8) which was expected to favor MT-HDPLDA. The good performance of our model on small numbers of tasks is therefore consistent in all our simulated experiments (mixture coeff. 0.25, 0.5 and 1).

Another interesting factor affecting performance is the number of documents in the task of interest; in many scenarios the task of interest may be a newer task with fewer documents available, for example a recently started newsgroup or a recently introduced track in a conference. We study the model performance with different numbers of documents in the task of interest in the following real data experiments (20newsgroups and NIPS conference articles).

5.3. *Experiment 3: 20 newsgroups data*

We next compare our method to MT-HDPLDA on a real-life collection of count data.

We take the computational group of the 20newsgroups data[5]. This group (often abbreviated as *comp*) is divided into five subgroups; some of the subgroups such as *comp.sys.ibm.pc.hardware* and *comp.sys.mac.hardware* have closely related topics and therefore the *comp* group may be well suited for a multitask problem. The data contains 11293 documents. We remove common words like *and* and *you* from the whole collection. We choose the *comp.sys.ibm.pc.hardware*

[5]We use the *stemmed* version of the data downloaded from http://web.ist.utl.pt/~acardoso/datasets/

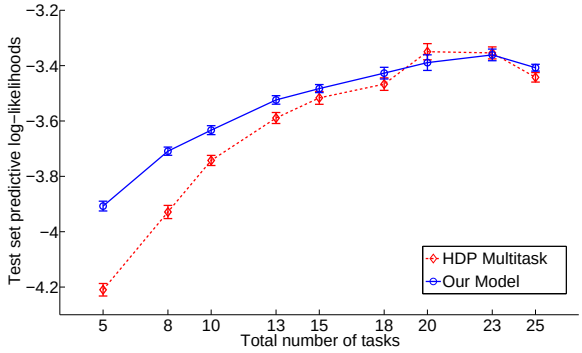

Figure 6: Experiment results: test set predictive likelihoods for datasets with different number of total tasks. The multi-task domain in this experiment is one of the domains in the domain continuum of Experiment 1, corresponding to mixture coefficient 0.25 in Figure 5. The error bars show ±1 standard deviation over 10 random datasets. Our model outperforms MT-HDPLDA ("HDP Multitask") when the number of tasks is small.

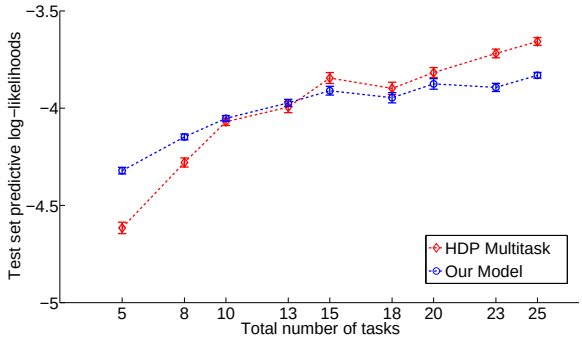

Figure 7: Experiment results: test set predictive likelihoods for datasets with different number of total tasks. The multi-task domain in this experiment is one of the domains in the domain continuum of Experiment 1, corresponding to mixture coefficient 0.5 in Figure 5. The error bars show ±1 standard deviation over 10 random datasets. Our model again outperforms MT-HDPLDA ("HDP Multitask") when the number of tasks is small.

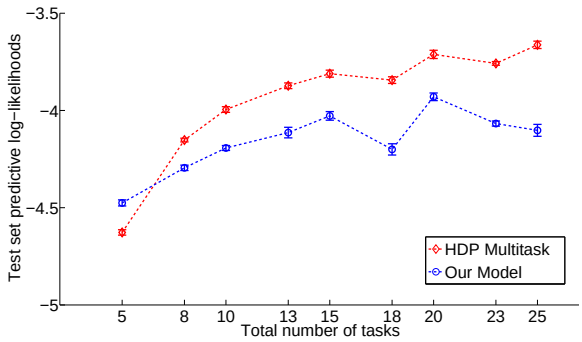

Figure 8: Experiment results: test set predictive likelihoods for datasets with different number of total tasks. The multi-task domain in this experiment is one of the domains in the domain continuum of Experiment 1, corresponding to mixture coefficient 1 in Figure 5. The error bars show ±1 standard deviation over 10 random datasets. This multi-task domain was designed to favor MT-HDPLDA ("HDP Multitask"), but our model still outperforms MT-HDPLDA when the number of tasks is very small.

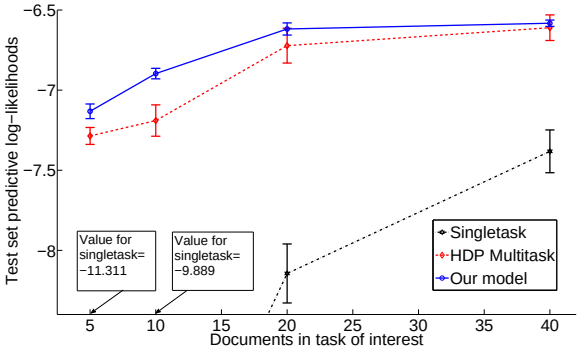

Figure 9: Test set Predictive likelihoods for 20newsgroups, error bars show ±1 standard deviation over 5 folds.

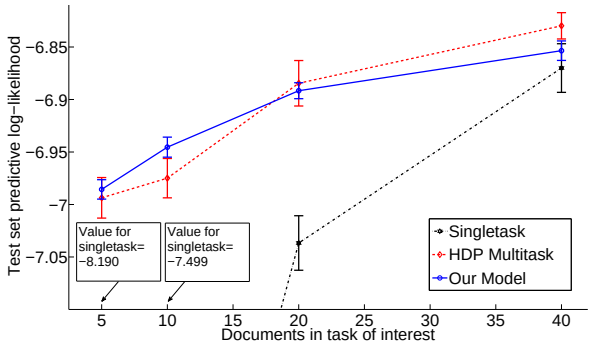

Figure 10: Test set Predictive likelihoods for the NIPS dataset, error bars show ±1 standard deviation over 5 folds.

subgroup as the interesting task. We run our model, MT-HDPLDA and single-task HDP as a baseline; we follow [8] for MT-HDPLDA and set $\eta = 0.5$ for both models. For our model we set $\alpha = 5$, $\gamma \sim Gamma(5, 0.1)$ and set $\epsilon = 0.0001$ as discussed in the Model Selection paragraph earlier. For sampling we initialize the Gibbs samplers randomly, take 1000 burn-in iterations, and then draw a total of 10 samples 50 iterations apart. We learn models for different sizes of training data in the interesting task (5-40 documents) with 50 documents in each other task, and use 5-fold cross-validation in each case. Results are again evaluated by average predictive log-likelihood of held-out documents from the interesting task. Figure 9 shows the results. Single-task learning naturally works poorly, and our model outperforms MT-HDPLDA in scenarios where training data is small and hence multi-task learning is most needed.

5.4. Experiment 4: NIPS data

We compare our model to MT-HDPLDA on another real-life collection of count data, a collection of scientific articles represented as bags-of-words.

We take the five most frequent sections of NIPS articles from 1987 to 1999 (http://www.gatsby.ucl.ac.uk/~ywteh); in total they contain 1147 documents with vocabulary size 1321 and average document length ~ 950 words. The most frequent group is "Algorithms and Architecture", which we choose as the interesting task. Like the 20newsgroups experiment we run our model, MT-HDPLDA and single-task HDP models and evaluate performance over the held-out dataset in a 5-fold cross validation setting. The number of documents per task, and the hyperparameters and the other experimental settings are the same as the ones used in 20newsgroups experiment. Figure 10 shows the results: single-task learning works poorly as before. There is not a large performance improvement for our model against MT-HDPLDA; however we observe essentially similar shapes of the peformance curves in both the NIPS and the 20 newsgroups data, and additionally observe consistent difference in several domains in the artificial continuum, which demonstrates an overall better predictive performance for our model especially under limited tasks and limited numbers of documents in the task of interest.

To illustrate the topic model our method has learned for the NIPS collection, we show learned topics for the problem setting where the number of documents in task of interest is 10 and each supplementary task has 50 documents (second location on the horizontal axis in Figure 10). We extract the top ten words from the strongest two topics for each task and from the weakest shared topic. Table 2 lists the top words. The first topic (first column of the table) is the strongest in all of the tasks; it lists words about general machine learning concepts. The next-to-strongest topics are listed in from column two to five: The next-to-strongest topic is different in each task (NIPS section), except that tasks LT and AA (task of interest) have the same next-to-strongest topic. Note that these "next-to-strongest" topic are all relatively weak even in their respective NIPS sections, compared to the strongest topic listed in the first column. Note also that these next-to-strongest topics are also used in other tasks (NIPS sections) but to a weaker extent; they can be interpreted as concepts encountered in many NIPS papers and most commonly in the particular section where their inferred probability was greatest. The strongest topic can be interpreted as general concepts of learning from data including neural learning (appropriately for the NIPS conference), the topic most active in CNP can be interpreted as general concepts of reinforcement learning especially in robotics; the topic most active in NS can be interpreted as biological concepts of neural learning; in LT and AA the most active topic is somewhat varied but can be interpreted as general concepts of probabilistic and kernel learning; in AP the most active topic can be interpreted as concepts of rules and schedules (for example for learning agents).

The last topic (last column of the table) is another weak topic which is uniformly present in all NIPS tasks. It can be interpreted as a mixture of concepts related to the structure of a paper (words like "discussion" and "conclusion") and to general experimental settings which might appear across several NIPS sections like neuroscience and control, navigation and planning (words such as "positions", "recorded" and "threshold").

Strongest Topic	CNP	NS	LT and AA	AP	Weakest Shared Topic
learning	control	neurons	variables	rules	recorded
network	reinforcement	neuron	kernel	coarse	side
model	robot	cortical	markov	instruction	positions
time	learned	arbors	conditional	fine	discussions
input	policy	dendritic	group	schedule	technical
neural	tangent	competition	likelihood	instructions	conclusions
algorithm	interpolation	cells	face	dec	scales
data	steps	axonal	database	blocks	fire
set	initial	cell	generalized	rl	exploration
system	grid	modules	matrix	resolution	threshold

Table 2: Top ten words in the strongest topic, four next-to-strongest but relatively weak topics, and the weakest shared topic, for the NIPS article collection. The NIPS collection is divided into five sections (tasks): CNP - Control Navigation and Planning, NS - Neuroscience, LT - Learning Theory, AA - Algorithms and Architecture, and AP - Applications. The strongest topic turned out to be the same in every tasks; the top words in that topic are listed in the first column. The next strongest topic is different in every task (except the task of interest AA), and its top words are listed under each task's name. The weakest shared topic is weakly present in all tasks.

6. Discussion

Overall we observe that under limited tasks and documents our model has better predictive performance than MT-HDPLDA. Our experiments suggest that this happens particularly in data domains where there are weak shared topics in some tasks. Since the MT-HDPLDA model makes too strong assumptions (it couples topic sharing with topic strength), our decoupled IBP based method performs better in such scenarios; a probable reason is the lack of strong evidence for the presence of topics in individual tasks in such settings.

Our model outperformed MT-HDPLDA in Figures 9 and 10 when the number of documents in the task of interest was small. One potential reason is that MT-HDPLDA couples topic strength with sharing, thus it assumes the topics shared in the task of interest are likely to be the ones that are strong in the other tasks: then weaker shared features of the task of interest (topics that are present in that task and in other tasks, but which are not always strongly present in the tasks where they appear) might not be learned well by MT-HDPLDA when few data are available. In Figures 9 and 10, performance increases for both our model and MT-HDPLDA as the number of data in the task of interest grows; this suggests that if a sufficient number of data points is available, MT-HDPLDA may be able to learn also weaker topics in the task of interest since the data provides sufficient evidence to make them visible in the posterior despite the coupling assumption.

In theory we expect that in both MT-HDPLDA and in our method, learning both strong and weak topics will benefit from having more tasks: as more tasks become available, the topics that are shared across most of the tasks can be learned from more data. In MT-HDPLDA the evidence for topics accumulates through the HDP hierarchy, and the benefit of many tasks will be greatest for strong shared topics due to the coupling assumption of sharing and strength. In our method the evidence of sharing accumulates in the learning of the IBP matrix and the extra sparsity vector, without a tight coupling to learning the topic strengths. Both our model and MT-HDPLDA increase their performance as more tasks become available; for example, in Figure 8 all topics in the domain are relatively strongly present in their respective tasks, thus here the performance increase is due to learning strong topics well. The fact that our method outperforms MT-HDPLDA in the intermediate domains where shared topics are weakly present in some tasks (see Figure 5) suggests our model is useful in such domains.

The continuum of multi-task domains studied in Experiment 1 (Figure 5) is not an exhaustive list of all multi-task domains; although the continuum already showed an advantage to our method in domains where some shared topics are weakly present in tasks, even larger differences between our method and MT-HDPLDA might be available in other multi-task scenarios.

In our case studies we evaluated the predictive performance for the task-of-interest, however the benefit of our model is not an artifact of the particular newsgroup/NIPS section choice that we used: the artificial experiment shows that our method has an advantage even when average over a large number of multi-task scenarios. More symmetric scenarios (e.g. predictions in all tasks with within-task and across-task accuracies) are also very important and will be considered in further work. Moreover in addition to the predictive likelihood we believe it is crucial to measure the comprehensibility of the extracted topics, for example in terms of their semantic coherence. Recently [17] have proposed the topic coherence score (a pointwise mutual information score) this can be used in future work as additional evaluation criterion for our model and other topic models.

The IBP has a "rich-get-richer" property where new matrix rows are likely to use frequently activated old topics rather than activating new topics; this keeps the IBP matrix sparse. However, in our setting there is only one IBP matrix row per task (data collection), thus none of the topics can become very rich: with C tasks, each topic is activated at most C times. Then the rich-get-richer property has only a weak effect, and new rows of the IBP matrix are likely to activate more topics than needed. To keep the topics sparsely used and prevent activating too many topics, our model uses an additional sparsity inducing step.

In this paper we set the prior for the additional sparsity by a

very simple manner according to the average number of documents per task, as described in Section 5 in the *Model Selection* paragraph. A cross-validation approach or a Bayesian prior, could also have been used; however, this simple first choice already worked well. Additionally, it is possible that a variant of IBP (e.g. through some variant of Hierarchical Beta Processes [18]) could achieve the same effect as our additional sparsity step; this would be an interesting direction of future work.

Another recent line of research is on statistic domain adaptation involving HDP and related models specially in the context of sequential data. In several works time dependence is incorporated to model time evolving topics among documents appearing in a sequence. For instance, the dynamic HDP model in [19] and [20] models the time evolution of topics and encourages topic sharing among temporally proximal data. These models are for a single task setting; in contrast, our model and MT-HDPLDA consider settings with multiple document collections. The recent single-task topic model of [21] studies sequential evolution not over time but rather within documents; it uses a two parameter generalization of Dirichlet process prior; a Poisson Dirichlet prior (Pitman-Yor process). It simultaneously models the hierarchical and the sequential topic structures within subparts (groups of sentences of paragraphs) of documents. The model is again for a single task setting whereas our model and MT-HDPLDA consider settings with multiple document collections. Another interesting approach is the HDP based evolutionary model in [22], which models the time evolution of topics both within and across multiple corpora. The paper focuses on mixture models rather than topic models; each document is generated by a mixture component, and strengths of mixture components over time and corpora are modeled through a HDP construction; evolution is modeled following a Markovian assumption. In contrast, we focus on topic models and unlike HDP we decouple component (here topic) strength from its sharing.

7. Conclusions

We have introduced a sparse multi-task topic model that is a robust and flexible method to model strong and weak sharing of topics in multiple heterogeneous collections of documents in an unsupervised manner. The generative model decouples the sharing of topics from the generation of the topic strengths by using a spike-and-slab prior. The proposed non-parametric model outperforms a state of the art Hierarchical Dirichlet Process based topic model on a simulated data continuum and in case studies on real data with small training sets. In our real-data experiments (20 newsgroups and NIPS data sets) our model and the state of the art MT-HDPLDA method are both much better than the single-task topic model, and our model still achieves further improvement: the error bars show that we get a consistent improvement over MT-HDPLDA. In particular, our experiments suggest that our model extracts *weak topics* better than the previous method, when the number of available tasks and documents per task is low. This shows that our new multi-task approach is a promising alternative to the standard approach in methods like MT-HDPLDA. Thus we recommend our method in cases where weak shared topics are likely to exist, and there are not very many documents or tasks to learn the models from.

8. Acknowledgments

Authors belong to the Finnish Centre of Excellence in Computational Inference Research (COIN). The work was supported by Academy of Finland decisions 123983 and 252845; Finnish Doctoral Programme in Computational Sciences and in part by PASCAL2 NoE, ICT 216886. We thank Samuel Kaski for fruitful discussions and support throughout the project. The calculations presented above were performed in part using computer resources within the Aalto University School of Science "Science-IT" project.

APPENDIX A

As described in Section 4.1, in order to sample the topic assignment z we wish to approximate the expectation over $\boldsymbol{\theta}^{(k)}$. In this section we describe the approximation; it is an extension of the technique used to approximate the expectation of topic mixture $\boldsymbol{\theta}$ for a single task model in [9].

We first rewrite the expectation as

$$\begin{aligned}
E[\boldsymbol{\theta}^{(k)}_{c,d}|\mathbf{z}_{\setminus c,d,n},\Delta] &\propto \int \tilde{\boldsymbol{\theta}}^{(k)}_{c,d}\, p(\tilde{\boldsymbol{\theta}}_{c,d},\mathbf{z}_{\setminus c,d,n}|\Delta)d\tilde{\boldsymbol{\theta}}_{c,d}\\
&\propto \int \tilde{\boldsymbol{\theta}}^{(k)}_{c,d}\, p(\mathbf{z}_{\setminus c,d,n}|\tilde{\boldsymbol{\theta}}_{c,d},\Delta)p(\tilde{\boldsymbol{\theta}}_{c,d}|\Delta)d\tilde{\boldsymbol{\theta}}_{c,d}\\
&\propto \int \tilde{\boldsymbol{\theta}}^{(k)}_{c,d}\, p(\mathbf{z}_{c,d\setminus n}|\tilde{\boldsymbol{\theta}}_{c,d})p(\mathbf{z}_{\setminus c,d}|\tilde{\boldsymbol{\theta}}_{c,d},\Delta)p(\tilde{\boldsymbol{\theta}}_{c,d}|\Delta)d\tilde{\boldsymbol{\theta}}_{c,d}
\end{aligned}$$

and approximating $p(\mathbf{z}_{\setminus c,d}|\tilde{\boldsymbol{\theta}}_{c,d},\Delta) \approx p(\mathbf{z}_{\setminus c,d}|\Delta)$ which is constant with respect to $\boldsymbol{\theta}^{(k)}_{c,d}$, we further write

$$\begin{aligned}
&E[\boldsymbol{\theta}^{(k)}_{c,d}|\mathbf{z}_{\setminus c,d,n},\Delta]\\
&\propto \int \tilde{\boldsymbol{\theta}}^{(k)}_{c,d} \int_{\boldsymbol{\phi}^\circ_c} \sum_{\mathbf{b}^\circ_c:b^{(k)}_c=1} \sum_{\boldsymbol{\psi}^\circ_c:\psi^{(k)}_c=1} p(\mathbf{z}_{c,d\setminus n}|\tilde{\boldsymbol{\theta}}_{c,d})\\
&\quad p(\tilde{\boldsymbol{\theta}}_{c,d}|\boldsymbol{\psi}_c,\mathbf{b}_c,\boldsymbol{\phi}_c)d\tilde{\boldsymbol{\theta}}_{c,d}\, p(\boldsymbol{\phi}^\circ_c|\boldsymbol{\gamma})p(\mathbf{b}^\circ_c|\boldsymbol{\pi}^\bullet,\alpha)p(\boldsymbol{\psi}^\circ_c|\epsilon)d\boldsymbol{\phi}^\circ_c\ .
\end{aligned}$$

Since $p(\mathbf{z}_{c,d\setminus n}|\tilde{\boldsymbol{\theta}}_{c,d})$ is the value of a Multinomial distribution and $p(\tilde{\boldsymbol{\theta}}_{c,d}|\boldsymbol{\psi}_c,\mathbf{b}_c,\boldsymbol{\phi}_c)$ is the value of a Dirichlet, their product is proportional to the value of another Dirichlet; we can then further rewrite the equation as

$$\begin{aligned}
&E[\boldsymbol{\theta}^{(k)}_{c,d}|\mathbf{z}_{\setminus c,d,n},\Delta]\\
&\propto \int_{\boldsymbol{\phi}^\circ_c} d\boldsymbol{\phi}^\circ_c \sum_{\mathbf{b}^\circ_c:b^{(k)}_c=1} \sum_{\boldsymbol{\psi}^\circ_c:\psi^{(k)}_c=1} \int \tilde{\boldsymbol{\theta}}^{(k)}_{c,d}\mathrm{Dir}(\tilde{\boldsymbol{\theta}}_{c,d}|\mathbf{n}_{c,d,\setminus c,d,n}+\boldsymbol{\phi}_c)d\tilde{\boldsymbol{\theta}}_{c,d}\\
&\quad p(\boldsymbol{\phi}^\circ_c|\boldsymbol{\gamma})p(\mathbf{b}^\circ_c|\boldsymbol{\pi}^\bullet,\alpha)p(\boldsymbol{\psi}^\circ_c|\epsilon)
\end{aligned}$$

and since the integral over $\tilde{\boldsymbol{\theta}}_{c,d}$ simply takes the k:th element from the mean of the Dirichlet distribution, we finally arrive at

$$E[\boldsymbol{\theta}_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\Delta] \propto \int d\boldsymbol{\phi}_c^{\circ} \sum_{\mathbf{b}_c^{\circ}:b_c^{(k)}=1} \sum_{\boldsymbol{\psi}_c^{\circ}:\psi_c^{(k)}=1} \frac{(n_{c,d,\backslash c,d,n}^{(k)} + \phi_c^{(k)})}{n_{c,d,\backslash c,d,n}^{(.)} + \sum_j b_c^{(j)}\psi_c^{(j)}\phi_c^{(j)}} p(\boldsymbol{\phi}_c^{\circ}|\boldsymbol{\gamma}) p(\mathbf{b}_c^{\circ}|\boldsymbol{\pi}^{\bullet},\alpha) p(\boldsymbol{\psi}_c^{\circ}|\epsilon) \; . \quad \text{(A.1)}$$

On the right-hand side, the sums over the binary vectors $\mathbf{b}_c^{\circ}$ and $\boldsymbol{\psi}_c^{\circ}$ are only over values whose kth entry is 1. This is equivalent to taking the sum over all possible vectors but multiplying the summed function by the binary flags $b_c^{(k)}$ and $\psi_c^{(k)}$, and the above equation can therefore be rewritten as

$$E[\boldsymbol{\theta}_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\Delta] \propto E\left[\frac{(n_{c,d,\backslash c,d,n}^{(k)} + \phi_c^{(k)})b_c^{(k)}\psi_c^{(k)}}{n_{c,d,\backslash c,d,n}^{(.)} + \sum_j b_c^{(j)}\psi_c^{(j)}\phi_c^{(j)}}\right] . \quad \text{(A.2)}$$

Let us divide $\sum_j b_c^{(j)}, \psi_c^{(j)}, \phi_c^{(j)}$ into active topics corresponding to entries of b in $b^{\bullet}$ (these are the topics represented in the corpus) and inactive topics corresponding to elements in b°

$$\sum_j b_c^{(j)}\psi_c^{(j)}\phi_c^{(j)} = \sum_{j:n_{c,(\cdot),\backslash n}^{(k)}>0} \phi_c^{(j)} + \sum_{j:n_{c,(\cdot),\backslash n}^{(k)}=0} b_c^{(j)}\psi_c^{(j)}\phi_c^{(j)} \quad \text{(A.3)}$$

$$= X + Y \quad \text{(A.4)}$$

We further split the inactive term into two components Y_1 and Y_2 thus:

$$\sum_j b_c^{(j)}\psi_c^{(j)}\phi_c^{(j)} = \sum_{j:n_{c,(\cdot),\backslash n}^{(k)}>0} \phi_c^{(j)} + \sum_{j\in J_1} b_c^{(j)}\psi_c^{(j)}\phi_c^{(j)} + \sum_{j\in J_2} b_c^{(j)}\psi_c^{(j)}\phi_c^{(j)}$$

$$= X + Y_1 + Y_2 \quad \text{(A.5)}$$

where:

$$J_1: \quad n_{c,(\cdot),\backslash n}^{(j)} = 0 \text{ and } n_{(\cdot),(\cdot),\backslash n}^{(j)} > 0 \quad \text{(A.6)}$$

$$J_2: \quad n_{(\cdot),(\cdot),\backslash n}^{(j)} = 0 \quad \text{(A.7)}$$

Thus (A.2) becomes:

$$E[\boldsymbol{\theta}_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\Delta] \propto E\left[\frac{(n_{c,d,\backslash c,d,n}^{(k)} + \phi_c^{(k)})b_c^{(k)}\psi_c^{(k)}}{n_{c,d,\backslash c,d,n}^{(.)} + X + Y_1 + Y_2}\right] \quad \text{(A.8)}$$

The expectation of Y is:

$$E[Y|\pi^{\bullet},\alpha,\boldsymbol{\gamma},\epsilon] = E[Y_1|\pi^{\bullet},\boldsymbol{\gamma},\epsilon] + E[Y_2|\alpha,\boldsymbol{\gamma},\epsilon] \quad \text{(A.9)}$$

$$E[Y_1|\pi^{\bullet},\boldsymbol{\gamma},\epsilon] = \sum_{j\in J_1} \pi^{(j)}\gamma^{(j)}\epsilon$$

$$E[Y_2|\alpha,\boldsymbol{\gamma},\epsilon] = \alpha a_1 a_2 \epsilon \quad \text{(A.10)}$$

Since it is not feasible to evaluate the above expectation in closed form as we would need to evaluate it for every word in each Gibbs sampling so we perform an approximation. The $\mathrm{E}[f(X)|Y]$ can be approximated by the first order Taylor expansion $\mathrm{E}[f(X)|Y] \approx f(\mathrm{E}[X|Y])$. We approximate the expectation under the following cases:

Case 1: $n_{c,d,\backslash c,d,n}^{(k)} = 0$ and $n_{c,(.),\backslash n,d,c}^{(k)} > 0$, i.e. the *k-th* topic is active in the task c which means $\psi_c^{(k)} = b_c^{(k)} = 1$, Eq (A.8) becomes:

$$\mathrm{E}[\boldsymbol{\theta}_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\Delta] \propto (n_{c,d,\backslash c,d,n}^{(k)} + \phi_c^{(k)})E\left[\frac{1}{n_{c,d,\backslash c,d,n}^{(.)} + X + Y_1 + Y_2}\right]$$

$$\propto \frac{(n_{c,d,\backslash c,d,n}^{(k)} + \phi_c^{(k)})}{n_{c,d,\backslash c,d,n}^{(.)} + \left[\sum_{j:n_{c,(.),\backslash n}^{(j)}>0} \phi_c^{(j)}\right] + \left[\sum_{j:n_{(.),(.),\backslash n}^{(j)}>0} \pi^{(j)}\gamma^{(j)}\epsilon\right] + \alpha\epsilon a_1 a_2} \quad \text{(A.11)}$$

Case 2: $n_{c,(.),\backslash n,d,c}^{(k)} = 0$ and $n_{(.),(.),\backslash n,d,c}^{(k)} > 0$ i.e. the *k-th* topic does not appear in the current task but is active in the corpus, so the expectation in Eq (A.8) is

$$\mathrm{E}[\boldsymbol{\theta}_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\Delta] \propto \gamma^{(k)}\pi^{(k)}\epsilon \; \mathrm{E}\left[\frac{1}{n_{c,d,\backslash c,d,n}^{(.)} + X + Y_{\backslash k} + \gamma^{(k)}}\right]$$

$$\propto (\gamma^{(k)}\pi^{(k)}\epsilon)\Big/\Bigg(n_{c,d,\backslash c,d,n}^{(.)} + \left[\sum_{j:n_{c,(.),\backslash n}^{(j)}>0} \phi_c^{(j)}\right] + \left[\sum_{j:n_{(.),(.),\backslash n}^{(j)}>0 \,\&\, j\neq k} \pi^{(j)}\gamma^{(j)}\epsilon\right] + \gamma^{(k)} + \alpha\epsilon a_1 a_2\Bigg) \quad \text{(A.12)}$$

Case 3: $n_{(.),(.),\backslash n,d,c}^{(k)} = 0$ implying the topic is inactive in the whole corpus. In this case we evaluate the probability of assigning any of the infinite number of components:

$$\mathrm{E}[\boldsymbol{\theta}_{c,d}^{(k)}|\mathbf{z}_{\backslash c,d,n},\Delta] \propto \mathrm{E}\left[\frac{Y_2}{n_{c,d,\backslash c,d,n}^{(.)} + X + Y_1 + Y_2}\right] \quad \text{(A.13)}$$

$$\propto \frac{(\alpha\epsilon a_1 a_2)}{n_{c,d,\backslash c,d,n}^{(.)} + \left[\sum_{j:n_{c,(.),\backslash n}^{(j)}>0} \phi_c^{(j)}\right] + \left[\sum_{j:n_{(.),(.),\backslash n}^{(j)}>0} \pi^{(j)}\gamma^{(j)}\epsilon\right] + \alpha\epsilon a_1 a_2} \quad \text{(A.14)}$$

These cases together suffice to compute the approximated expectation.

APPENDIX B

In our simulated experiment of Figure 5 we construct a continuum of synthetic domains. From each domain we generate several multi-task learning problems: each multi-task learning problem consists of several data sets (tasks).

In detail, each learning problem is generated from the model structure of our model, that is, from a multi-task topic model. There are 10 active topics across 10 tasks. The overall sum of pseudocounts across topics was set to 300 for each task. The division of pseudocounts across topics in each task was fixed

according to the continuum as described below. For each task c, elements of the vector $\mathbf{b}_c$ were set to 1 if the corresponding topic had been allocated nonzero pseudocounts, and to 0 otherwise. In order to generate small training data, the topic-to-word distributions were generated according to the sparse topic model in [23].

At each intermediate point (domain) in the continuum, the prior topic strength vector is generated by linearly mixing two extreme values of $\boldsymbol{\phi}_c$, with a mixing coefficient u between zero and one: $\boldsymbol{\phi}_c = (1-u)\boldsymbol{\phi}_c^{\text{Left}} + u\boldsymbol{\phi}_c^{\text{Right}}$ where "Left" and "Right" denote the two extreme choices. The mixing coefficient corresponds to the position in the domain continuum (horizontal axis of Figure 5) so that $u = 0$ at the left end of the continuum and $u = 1$ at the right end, and u takes intermediate values between the two ends.

The *first extreme choice* for the topic pseudocount vectors $\boldsymbol{\phi}_c^{\text{Left}}$ in each task c is as follows. In the task of interest $c = 1$, only the first topic is active (pseudocount 300), all others have pseudocount 0. In a supplementary task $c > 1$, the first topic is active with pseudocount 150, and additionally a task-specific topic (topic index c, same as the task index) is active with pseudocount 150; all other topics have pseudocount 0. In this extreme choice, the first topic is very strong (it is active in all tasks with high pseudocount) and other topics are also strong (active in one task with pseudocount 150). The *second extreme choice* for pseudocount vectors $\boldsymbol{\phi}_c^{\text{Right}}$ is as follows. For each task c, three randomly picked topics were activated (their pseudocount was set to 100 each) and other topics were inactive (pseudocount 0). The overall strength of each topic then depends on how many tasks picked them; all topics active in at least one task have total pseudocount at least 100.

In both extreme choices $\boldsymbol{\phi}_c^{\text{Left}}$ and $\boldsymbol{\phi}_c^{\text{Right}}$ the active topics are strongly active in their respective tasks. However, the set of which topics are active in which tasks differs between the extremes. In particular, in "Left" the task of interest (TOI) uses only topic 1 so that $\boldsymbol{\phi}_{\text{TOI}}^{\text{Left}} = [300\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$ whereas in "Right" the task of interest uses a random three topics so that for example $\boldsymbol{\phi}_{\text{TOI}}^{\text{Right}} = [0\ 0\ 100\ 0\ 100\ 0\ 0\ 100\ 0\ 0]$. Then the interpolated weight vector for the task of interest contains weak values, for example $u = 0.1$ yields $\boldsymbol{\phi}_{\text{TOI}} = (1-u)\boldsymbol{\phi}_{\text{TOI}}^{\text{Left}} + u\boldsymbol{\phi}_{\text{TOI}}^{\text{Right}} = [270\ 0\ 10\ 0\ 10\ 0\ 0\ 10\ 0\ 0]$. The topic strength vectors of other tasks also get weak values through the interpolation.

We sample the two extreme choices several times; each time we generate the learning problems for the whole continuum (for both extreme positions in the continuum, and for each intermediate position by interpolating the pseudocount vectors as described above).

References

[1] D. M. Blei, A. Y. Ng, M. I. Jordan, J. Lafferty, Latent Dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[2] T. L. Griffiths, M. Steyvers, Finding scientific topics, Proceedings of the National Academy of Sciences 101 (2004) 5228–5235.

[3] A. Perina, P. Lovato, V. Murino, M. Bicego, Biologically aware Latent Dirichlet allocation for the classification of expression microarray, in: International Conference on Pattern Recognition in Bioinformatics, 2010.

[4] J. Caldas, N. Gehlenborg, E. Kettunen, A. Faisal, M. Ronty, A. Nicholson, S. Knuutila, A. Brazma, S. Kaski, Data-driven information retrieval in heterogeneous collections of transcriptomics data links *SIM2s* to malignant pleural mesothelioma., Bioinformatics 28 (2) (2012) i246–i253.

[5] S. Thrun, Is learning the n-th thing any easier than learning the first, in: Advances in Neural Information Processing Systems, The MIT Press, 1996, pp. 640–646.

[6] R. Caruana, Multitask learning, Machine Learning 28 (1997) 41–75.

[7] T. Griffiths, Z. Ghahramani, Infinite latent feature models and the Indian buffet process, in: Advances in Neural Information Processing Systems 18, MIT Press, 2006, pp. 475–482.

[8] Y. W. Teh, M. I. Jordan, M. J. Beal, D. M. Blei, Hierarchical Dirichlet processes, Journal of the American Statistical Association 101 (476) (2006) 1566–1581.

[9] S. Williamson, C. Wang, K. A. Heller, D. M. Blei, The IBP compound Dirichlet process and its application to focused topic modeling, in: Proceedings of the 27th International Conference on Machine Learning, Omnipress, 2010, pp. 1151–1158.

[10] A. Faisal, J. Gillberg, J. Peltonen, G. Leen, S. Kaski, Sparse nonparametric topic model for transfer learning, in: Proceedings of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2012.

[11] T. S. Ferguson, A Bayesian analysis of some nonparametric problems, The Annals of Statistics 1 (2) (1973) 209–230.

[12] J. Sethuraman, A constructive definition of Dirichlet priors, Statistica Sinica 4 (1994) 639650.

[13] H. Ishwaran, J. Rao, Spike and slab variable selection: Frequentist and Bayesian strategies, Annals of Statistics 33 (2) (2005) 730–773.

[14] Y. W. Teh, Stick-breaking construction for the Indian buffet process, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2007, pp. 1–10.

[15] W. R. Gilks, P. Wild, Adaptive rejection sampling for gibbs sampling, Applied Statistics 41 (1992) 337–348.

[16] W. Li, A. Mccallum, Pachinko Allocation: DAG-structured mixture models of topic correlations, in: International Conference on Machine Learning, 2006.

[17] D. Newman, E. Bonilla, W. Buntine, Improving topic coherence with regularized topic models, in: Advances in Neural Information Processing Systems, The MIT Press, 2011.

[18] R. Thibaux, M. Jordon, Hierarchical Beta processes and the Indian buffet process, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2007, pp. 564–571.

[19] L. Ren, D. Dunson, L. Carin, The dynamic hierarchical Dirichlet process, in: Proceedings of the 25th International conference on Machine learning, 2008, pp. 824–831.

[20] L. Ren, D. Dunson, S. Lindroth, L. Carin, Dynamic nonparametric Bayesian models for analysis of music, Journal American Statistical Association 105 (490) (2010) 458–472.

[21] L. Du, W. Buntine, H. Jin, Modelling sequential text with an adaptive topic model, in: Empirical Methods in Natural Language Processing, 2012.

[22] J. Zhang, Y. Song, C. Zhang, S. Liu, Evolutionary hierarchical Dirichlet processes for multiple correlated time-varying corpora, in: Proceedings of the ACM SIGKDD International Conference on Knowledge discovery and data mining, 2010, pp. 1079–1088.

[23] C. Wang, D. M. Blei, Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process.