On the Integration of Topic Modeling and Dictionary Learning

Lingbo Li LL83@DUKE.EDU
Mingyuan Zhou MZ1@EE.DUKE.EDU
Guillermo Sapiro[†] GUILLE@UMN.EDU
Lawrence Carin LCARIN@EE.DUKE.EDU
Department of Electrical and Computer Engineering, Duke University, Durham, NC
[†]Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN

Abstract

A new nonparametric Bayesian model is developed to integrate dictionary learning and topic model into a unified framework. The model is employed to analyze partially annotated images, with the dictionary learning performed directly on image patches. Efficient inference is performed with a Gibbs-slice sampler, and encouraging results are reported on widely used datasets.

1. Introduction

Statistical topic models, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003), originally developed for text analysis, have been applied successfully for image-analysis tasks. In this setting researchers typically represent an image as a bag of visual words (Fei-Fei & Perona, 2005; Li & Fei-Fei, 2007). Using such methods, there has been interest in developing models for automatic clustering, classification and annotation of images, based on image features as well as available meta-data such as image annotations (Barnard et al., 2003; Blei & Jordan, 2003; Blei & MaAuliffe, 2007; Wang et al., 2009; Li et al., 2009; Du et al., 2009).

In such research one typically treats image feature extraction as a pre-processing step, decoupled from the subsequent statistical analysis. Local image descriptors, *e.g.*, scale-invariant feature transform (SIFT) (Lowe, 1999), are commonly used to extract features from local patches (Fei-Fei & Perona, 2005; Li & Fei-Fei, 2007; Wang et al., 2009), segments (Li et al., 2009), or super-pixels (Du et al., 2009). In such research the extracted local features are typically used to design a discrete codebook (*i.e.*, vocabulary), with vector quantization (VQ). When analyzing images, each local descriptor is subsequently assigned to one of the codewords, with these codes playing the role of discrete words in traditional documents (Fei-Fei & Perona, 2005).

Although the above research has realized significant success, there is no principled way to define the codebook size; this parameter must be tuned and is in general a function of the dataset considered. Further, since feature extraction is performed separately from the subsequent statistical analysis, it is unclear which features should be used and why one class of features should be preferred.

In this paper we integrate feature learning and topic modeling within a unified setting. The feature extraction is performed using dictionary learning, with this integrated within topic modeling. Recent research on dictionary learning and sparse coding has demonstrated superior performance in a number of challenging image processing applications, including image denoising, inpainting and sparse image modeling (Mairal et al., 2008; Zhou et al., 2009). Recent advances in image classification show that substantially improved performance may be achieved by extracting features from local descriptors with dictionary learning and sparse coding, this replacing VQ (Yang et al., 2009). In the work reported here we also replace VQ, with the number of features (dictionary atoms) and their characteristics inferred via a new application of the hierarchical beta process (Thibaux & Jordan, 2007).

We develop a novel hierarchical Bayesian model that integrates dictionary learning, sparse coding and topic modeling, for joint analysis of multiple images and (when present) associated annotations. The model defines topics in terms of the probabilities with which dictionary atoms are used, with the dictionary learned jointly while performing topic modeling. The learned

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

model clusters all images into groups, based upon dictionary usage; a statistical distribution is also provided for words that may be associated with previously non-annotated images (only a subset of the images are assumed annotated when learning the model). The encouraging performance of the framework is demonstrated on several commonly analyzed datasets, with comparisons to previous related research. We also quantitatively examine the utility of jointly performing image feature learning and topic modeling, *vis-a-vis* treating these as two disjoint processes. Additionally, we compare the performance of learned features applied directly to the image, as opposed to first doing feature extraction using such methods as SIFT. To the authors' knowledge, this paper is the first to unify dictionary learning and statistical topic modeling.

2. Model Construction

We wish to analyze M images, and a subset of the images have accompanying words or an annotation; the vocabulary of such annotations is assumed to be of dimension L. The vector $\boldsymbol{x}_m$ represents the pixels associated with image m, and $\boldsymbol{y}_m = (y_{m1}, \ldots, y_{mL})^T$ represents a vector of word counts for that image, when available (y_{ml} represents the number of times word $l \in \{1, \ldots, L\}$ is present in the annotation). The objective is to organize/sort/cluster the images, utilizing annotations when available.

The M images are assumed characterized by the following hierarchy. Each image is assumed to have an associated category/class. For example, some images may be characterized as city scenes, while others may be forest or beach scenes. The number of such categories is not set or defined *a priori*, and is to be inferred by the data under analysis. At the next level of the hierarchy, each image category is characterized in terms of a distribution of objects/entities that may appear in the image (these image objects are analogous to topics in topic models). Again, the number of such objects is to be inferred by the data, and the partial presence of annotations plays an important role in defining an appropriate number of objects.

Finally, each object (or topic) is characterized at the patch level in terms of a distribution over dictionary atoms. The number of dictionary atoms and their composition are also inferred based on the data under test. The dictionary atoms play the role of words in topic models. In classical topic models (Blei et al., 2003) each topic is characterized by a distribution over words. In the analysis that follows, each topic is characterized by a set of probabilities, defining the probabilities with which particular dictionary atoms ("words") are selected to represent a particular object.

2.1. Hierarchical BP & Dictionary learning

When presenting the model we start at the level of the observed pixels, and then work our way up to the top (image-class) level. As is customary in dictionary learning applied to image analysis, we divide each image into partially overlapping patches, where each patch consists of a contiguous subset of pixels. Specifically, the mth image is divided into N_m patches, where the ith patch is denoted $\boldsymbol{x}_{mi} \in \mathbb{R}^P$ with $i = 1, \ldots, N_m$.

Each patch $\boldsymbol{x}_{mi}$ is represented as a sparse linear combination of learned dictionary atoms. Further, each patch is assumed associated with an object/entity ("topic"); the probability of which dictionary atoms are employed for a given patch is dictated by the object associated with it. The connection between the different topics, the dictionary usage, and the dictionary form is constituted via a *hierarchical* beta process (HBP) (Thibaux & Jordan, 2007), in the following manner.

Each patch is represented as $\boldsymbol{x}_{mi} = \mathbf{D}(\boldsymbol{z}_{mi} \odot \boldsymbol{s}_{mi}) + \epsilon_{mi}$, where $\odot$ represents the element-wise/Hadamard product, $\mathbf{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_K] \in \mathbb{R}^{P \times K}$, K is the truncation level on the possible number of dictionary atoms, $\boldsymbol{z}_{mi} = [z_{mi1}, \cdots, z_{miK}]^T$, $\boldsymbol{s}_{mi} = [s_{mi1}, \cdots, s_{miK}]^T$, $z_{mik} \in \{0, 1\}$ indicates whether the kth atom is *active* within patch i in image m, $s_{mik} \in \mathbb{R}^+$, and $\boldsymbol{\epsilon}_{mi}$ is the residual error. Note that $\boldsymbol{z}_{mi}$ represents the specific sparseness pattern of dictionary usage for $\boldsymbol{x}_{mi}$. The hierarchical form of the model is

$$\begin{aligned} \boldsymbol{x}_{mi} &\sim \mathcal{N}(\mathbf{D}(\boldsymbol{z}_{mi} \circ \boldsymbol{s}_{mi}), \gamma_\epsilon^{-1} \mathbf{I}_P) \\ \boldsymbol{d}_k &\sim \mathcal{N}(0, \frac{1}{P} \mathbf{I}_P) \\ \boldsymbol{s}_{mi} &\sim \mathcal{N}_+(0, \gamma_s^{-1} \mathbf{I}_K) \\ \boldsymbol{z}_{mi} &\sim \prod_{k=1}^{K} \text{Bernoulli}(\pi_{h_{mi}k}) \end{aligned} \tag{1}$$

where gamma priors are placed on both γ_ϵ and γ_s. Unlike conventional dictionary learning (Zhou et al., 2009), positive weights $\boldsymbol{s}_{mi}$ (truncated normal, $\mathcal{N}_+(\cdot)$) are imposed, which we have found to yield improved results.

In (1) the indicator variable h_{mi} defines the topic associated with $\boldsymbol{x}_{mi}$, and this will be controlled via higher layers of the model; we discuss this below. We now focus on how the probabilities π_{hk} are constituted, in terms of an HBP. Specifically, the K-dimensional vector $\boldsymbol{\pi}_h$ defines the probability that each of the K columns of $\mathbf{D}$ is employed to represent object type $h \in \{1, \ldots, J\}$, where the kth component of $\boldsymbol{\pi}_h$ is

π_{hk}. Using an HBP construction as in (Thibaux & Jordan, 2007), these probability vectors are defined as $\boldsymbol{\pi}_h \sim \prod_{k=1}^{K} \text{Beta}(c_1\eta_k, c_1(1-\eta_k))$, $\eta_k \sim \text{Beta}(c_0\eta_0, c_0(1-\eta_0))$ where η_k represents the "global" probability of using dictionary atom $\boldsymbol{d}_k$ across all topics (object types), and π_{hk} represents the probability of using $\boldsymbol{d}_k$ for object type h. Although the model is truncated to J topics, in practice J is set to a large value, and the model infers which subset of $\{\boldsymbol{\pi}_h\}$ are actually needed to represent the observed data. Similarly, K is set to a large value, and the model infers the subset of dictionary atoms ("words") needed to represent the data.

In the Indian buffet metaphor (Griffiths & Ghahramani, 2005; Thibaux & Jordan, 2007), each of the topics is a customer at a buffet of dictionary atoms ("words" in the context of a topic model). The vector $\boldsymbol{\pi}_h$ defines the probability of dictionary atom selection for topic/customer h. While each topic shares the same buffet of dictionary atoms, the probability with which such are selected is topic-dependent.

2.2. Topic-modeling component

The generative model has now constituted a set of topic-dependent dictionary-usage probabilities $\{\boldsymbol{\pi}_h\}$, and a given image patch $\boldsymbol{x}_{mi}$ is linked to an indicator variable $h_{mi} \in \{1, \ldots, J\}$ defining the topic associated with patch i in image m. What remains is to define probabilities with which objects/topics may be found in an image, and to link this probability vector to the specific image class under test.

Let $r_m \in \{1, \ldots, T\}$ represent the image class associated with image m, which we seek to cluster. Then the remainder of the generative process may be expressed as

$$h_{mi} \sim \sum_{j=1}^{J} \nu_{r_m j}\delta_j \ , \ \boldsymbol{\nu}_t \sim \text{Dir}(\alpha_\nu/J, \cdots, \alpha_\nu/J)$$
$$r_m \sim \sum_{t=1}^{T} \mu_t\delta_t \ , \ \boldsymbol{\mu} \sim \text{Dir}(\alpha_\mu/T, ..., \alpha_\mu/T) \qquad (2)$$

where δ_α is a unit measure at the point α. The J-dimensional probability vector $\boldsymbol{\nu}_t$ defines the probability with which each of the J objects are manifested in image class t, while $\boldsymbol{\mu}$ defines the probability with which the T image classes are manifested across the M images.

Summarizing the generative process thus far, for image m we draw a latent $r_m \in \{1, \ldots, T\}$, this defining the image class. For each of the image patches $\{\boldsymbol{x}_{mi}\}$ in this image we draw an associated object type or topic, with probability of topics defined by $\boldsymbol{\nu}_{r_m}$. Latent $h_{mi} \in \{1, \ldots, J\}$ defines which object/topic is associated with patch i in image m, defined by $\boldsymbol{x}_{mi}$. Finally, the vector of probabilities $\boldsymbol{\pi}_{h_{mi}}$ defines the associated probabilities with which columns of $\mathbf{D}$ ("image words") are used.

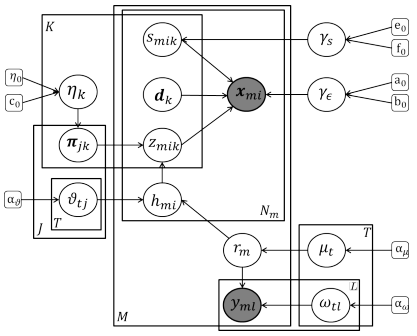


Figure 1. The graphical representation of the model.

2.3. Handling words/annotations

If annotations are available for at least a subset of the M images, it is desirable to leverage the information they provide. For each image class $t \in \{1, \ldots, T\}$ there is a unique distribution over the L words, and therefore the observed count of words for image m (when words are available) is drawn

$$\begin{aligned} \boldsymbol{y}'_m &\sim \text{Mult}(\boldsymbol{\omega}_{r_m}, N_m) \\ \boldsymbol{\omega}_t &\sim \text{Dir}(\alpha_\omega/L, \cdots, \alpha_\omega/L) \end{aligned} \qquad (3)$$

where $\boldsymbol{y}'_m = \boldsymbol{y}_m N_m/|\boldsymbol{y}_m|$ and $|\boldsymbol{y}_m|$ represents the total number of words associated with image m. Recall that r_m is the topic/class associated with image m.

Note that we have scaled the observed count of words $\boldsymbol{y}_m$ to produce $\boldsymbol{y}'_m$, and the total number of words used in $\boldsymbol{y}'_m$ equals N_m, the number of image patches used in the analysis of image m. This has been found important in our numerical studies, as it places the image features and words on equal footing, when words are present. Typically $|\boldsymbol{y}_m| \ll N_m$, and therefore if this rescaling is not performed the contribution to the likelihood from the image features far overwhelms the likelihood contribution from the words. This rescaling of the word count is equivalent to raising the multimonial contribution to the likelihood function from $\boldsymbol{y}_m$ by power $N_m/|\boldsymbol{y}_m|$.

A graphical representation of the model is summarized in Fig. 1, in which shaded and unshaded nodes indicate observed and latent variables, respectively. An arrow indicates dependence between variables. The boxes denote repetition, with the number of repeti-

tions indicated by the variables in the corner of boxes.

2.4. Discussion

While the hierarchical form of the model may appear relatively complicated, we have found it to be robust and relatively insensitive to parameter settings. There has been no tuning performed for any hyperparameters to achieve the results presented below, with parameters set in a "standard" way for such models. Specifically, the hyperparameters for the gamma distributions on the precisions were set as $(10^{-6}, 10^{-6})$. For the hierarchical beta process we set $c_0 = 10$, $\eta_0 = 0.5$ and $c_1 = 1$. The parameters on the Dirichlet distributions were set as $\alpha_\nu = 1$, $\alpha_\mu = 1$ and $\alpha_\omega = 1$.

The manner in which annotations are handled in the proposed model is more flexible than how such were considered in (Du et al., 2009). Specifically, in the latter paper a single word was associated with each object class in the scene, and therefore the number of objects J was required to be equal to the number of words L. In our model J and L are in general different, and the number of inferred objects need not be equal to the number of words; this implies that multiple words may be used to represent the same object type.

In the course of developing the proposed model, we considered different details on the model construction. For example, we considered a stick-breaking representation for the beta process, with (Teh et al., 2007) $\eta_k = \prod_{i=1}^{k} u_l$ with $u_l \sim \text{Beta}(\beta, 1)$. The advantage of this construction is that it associates the important (large) η_k with small indices k. This is of interest particularly when truncating the beta process to K atoms, as done here. We found that the model above worked the same as when the stick-breaking form of the beta process was employed, and therefore the former was adopted for its simplicity.

Note that each image was above assumed associated with a particular class, with an image class defined by a distribution over topics, $\boldsymbol{\nu}_t$. This was done to address the specific applications discussed below, of image clustering. In this setting all images in class r_m share the same distribution over topics, $\boldsymbol{\nu}_{r_m}$. In typical topic models (Blei et al., 2003) each image has a unique distribution over topics, and this may also be considered here if desired. In this case rather than clustering images via the indicator r_m, each image may have a unique distribution over topics, drawn for example from a hierarchical Dirichlet process (Teh et al., 2004). We also considered drawing the probability vector $\boldsymbol{\mu}$ over image categories via a stick-breaking representation (Sethuraman, 1994) rather than from a Dirichlet distribution, with results similar to those reported below.

3. Model Inference

Because all consecutive layers except for η_k in the hierarchical model are in the conjugate-exponential family, we employ Gibbs sampling for each parameter except η_k, for which slice sampling is utilized in (Zhou et al., 2011). The inference equations for the dictionary $\mathbf{D}$, the binary sparse codes $\boldsymbol{z}$ and the real non-negative sparse codes $\boldsymbol{s}$ are similar to that in (Zhou et al., 2009), and are omitted for brevity.

Sampling $\boldsymbol{\pi}_j$: $p(\boldsymbol{\pi}_j|-) = \text{Beta}(\boldsymbol{\pi}_j; \psi_{1j}, \psi_{2j})$, where $\psi_{1j} = c_1\boldsymbol{\eta} + \sum_{m=1}^{M}\sum_{i=1}^{N_m}\delta(h_{mi} = j)\boldsymbol{z}_{mi}$ and $\psi_{2j} = c_1(1-\boldsymbol{\eta}) + \sum_{m=1}^{M}\sum_{i=1}^{N_m}\delta(h_{mi} = j)(1-\boldsymbol{z}_{mi})$.

Sampling r_m and h_{mi}:

$$p(r_m = t|-) \propto \mu_t \prod_{j=1}^{J} \nu_{tj}^{\sum_{i=1}^{N_m}\delta(h_{mi}=j)} \prod_{l=1}^{L} \omega_{tl}^{y'_{ml}} \quad (4)$$

$$p(h_{mi} = j|-) \propto \nu_{r_m j} \prod_{k=1}^{K} \pi_{jk}^{z_{mik}}(1-\pi_{jk})^{1-z_{mik}}. \quad (5)$$

Sampling ν_{tj}, ω_{tl}, and μ_t : The prior has $p(\nu_{tj}|-) = \text{Dir}(\nu^*_{t1}, ..., \nu^*_{tJ})$, $p(\omega_{tl}|-) = \text{Dir}(\omega^*_{t1}, ..., \omega^*_{tL})$ and $p(\mu_t|-) = \text{Dir}(\mu^*_1, ..., \mu^*_T)$, where $\nu^*_{tj} = \frac{\alpha_\nu}{L} + \sum_{m=1}^{M}\big[\sum_{i=1}^{N_m}\delta(h_{mi} = j)\big]\delta(r_m = t)$, $\omega^*_{tl} = \frac{\alpha_\omega}{L} + \sum_{m=1}^{M}\delta(r_m = t)y'_{ml}$, $\mu^*_t = \frac{\alpha_\mu}{T} + \sum_{m=1}^{M}\delta(r_m = t)$.

4. Experimental Results

We test our model with one relatively simple but illustrative dataset (MNIST handwritten digits) and three real-world image data sets (MSRC, LabelMe and UIUC-Sport); the latter three contain annotations. For all experiments, we process patches from each image. For the MNIST data we randomly select 50 partially overlapping patches in each image, with 15×15 patch size, and for the other three datasets we collect all $32 \times 32 \times 3$ non-overlapping patches from the color image (we could also consider overlapping patches in this case, but it was found unnecessary). These patches are used to constitute the data matrix $\mathbf{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]$, where $\boldsymbol{x}_i \in \mathbb{R}^P$, with P the number of pixels in each patch ($P = 225$ for MNIST, and $P = 3072$ for the other three data); N is the total number of patches in the dataset. The matrix $\mathbf{X}$ is pre-whitened with principal component analysis (PCA) and the first 200 principle components are employed (200 keeps about 95% of the energy of the original data, achieving a good balance between accuracy and complexity). To initialize the dictionary, we can use random initialization or some fixed redundant

bases, such as over-completed DCT. In this paper, we use the covariate-dependent HBP (with the covariates linked to the relative locations between data samples) to learn an initial set of dictionary atoms, which are found to match the local latent features (Zhou et al., 2011).

In all experiments we set the truncation levels as $K = 400$, $J = 100$ and $T = 30$. Similar results were found for larger truncations. Note that these truncation levels are upper bounds on the associated parameter, while the model infers the number of components needed. For each experiment, we run 1000 MCMC iterations, and collect the last 500 samples.

4.1. MNIST Handwritten Digits

For the MNIST handwritten digit database, we randomly choose 100 samples per digit (digits 0 through 9), and therefore 1000 samples are considered in total; the original digit images are of size 28×28. In this experiment annotations are not considered.

Each collection sample manifests a number of unique image classes, and often more than 10 classes are inferred, since some digits tend to occupy more than one image class (as a consequence of different styles of writing the digits). Fig. 2 displays five random examples associated with each image class inferred, at a typical collection sample. From Fig. 2 we see that there is more than one way some digits may be expressed, and the different writing styles constitute unique image classes inferred by the model.

As seen from Fig. 2, the inferred clusters are readily labeled in terms of truth, based upon the large frequency with which a particular cluster is associated with one digit. In Fig. 3(a) we present a confusion matrix, which quantifies the probability that a given digit is clustered "properly", in the sense that it is in a cluster dominated by the same digit type (this quantifies the "purity" of the clusters, in the context of being associated with the same image type). The average clustering accuracy is 81.4%, and we note that this performance is achieved with an *unsupervised* model, with dictionary learning and clustering performed simultaneously.

4.2. Microsoft Data

The experiments with the MNIST data demonstrate the ability of the model to cluster images accurately; henceforth we do such in the presence of annotations, considering natural images. We use the same settings

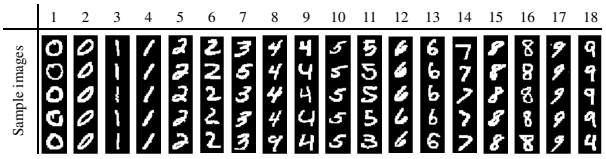
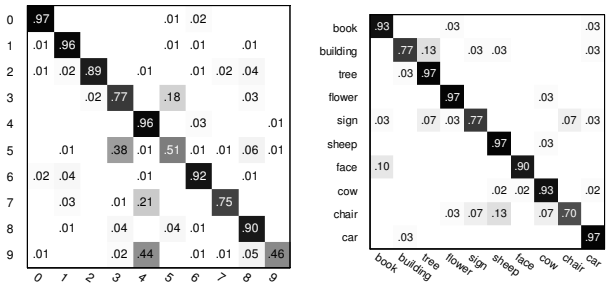


Figure 2. Example images associated with 18 inferred classes, with each column representing one unique class.



(a) MNIST (b) MSRC

Figure 3. (a) Confusion matrix of MNIST data, with average accuracy of 81.04%. (b) Confusion matrix of MSRC data, with average accuracy of 89.06%.

of images and annotations from the MSRC data[1] as considered in (Du et al., 2009), to allow a direct comparison. We choose 320 images from 10 categories of images with manual annotations available. The categories are "tree", "building", "cow", "face", "car", "sheep", "flower", "sign", "book" and "chair". The numbers of images are 45 and 35 in the "cow" and "sheep" classes, respectively and 30 in all the other classes. Each image has size 213×320 or 320×213. For annotations, we remove all annotation-words that occur less than 8 times (approximately 1% of them), and obtain 15 unique annotation-words, thus $L = 15$. For each category, we randomly choose 10 images, and remove their annotations, treating them as non-annotated images within the analysis.

We inferred 11 clusters, and found that the "chair" image class is divided into two types. Using such labeling of clusters based on truth, we may constitute a confusion matrix, defining the probability that an image from a given class is associated with the appropriate mixture component, as was done with the MNIST data. The confusion matrix as computed form the collection samples is depicted in Fig. 3(b). The average accuracy is 89.06%, outperforming the results in (Du et al., 2009) by 6.16% under the same test set-

[1] *http://research.microsoft.com/en-us/projects/objectclassrecognition/*

Figure 4. Example images inferred for each class. Each row is for one category. The first three columns on the left show 3 examples of correctly inferred images, the last column on the right shows an example of incorrectly recognized image.

tings (note that in (Du et al., 2009) predefined features were extracted from super-pixels, and VQ was employed). By contrast, the proposed model does image clustering (topic modeling) and feature design simultaneously, without VQ. Fig. 4 shows three example images correctly assigned to each of the clusters. In Fig. 4 we observe that many of the "inaccurate" classifications that cause errors in Fig. 3(b) actually make a lot of sense. For example the "face" image at the top-right in Fig. 4 is "incorrectly" assigned to the "book" class, as a consequence of the books in the background of the face picture. As another example, sheep are misclassified as cows.

Each image class is characterized by a distribution over objects, and these objects may be linked to words via the annotation, when available. A good connection is inferred between words and image classes (clusters), with no further details here, for brevity. Below we show detailed word associations for the UIUC-Sport data.

For the above results, the dictionary is performed simultaneously with topic modeling, with the dictionary learning performed directly on image patches (below we refer to this as "online"). As comparisons, we consider the following alternatives. In one test, the dictionary atoms, initialized with the method discussed in (Zhou et al., 2011), are fixed, and we use the dictionary in the topic model as before (below we refer to this as "offline"). This permits us to examine the benefit of simultaneously doing dictionary learning and topic modeling, which allows the dictionary atoms to be matched to the topic-modeling objective. As another example, topic modeling and dictionary learning are performed simultaneously, but the dictionary learning is performed using SIFT features extracted from the same local region patches used in the previous dictionary learning. Finally, we remove dictionary learning altogether, and learn a codebook of dimension $K = 400$ (consistent with the dictionary-learning truncation level), with VQ codebook design performed directly on the image patches. The quantitative comparisons between these tests are summarized in Table 1. It is observed that dictionary learning performed directly on the patches yields best results, with an improvement manifested by the full online analysis (joint topic modeling and dictionary learning). There is a marked improvement in doing dictionary learning directly on the image patches, compared to doing such on the SIFT features.

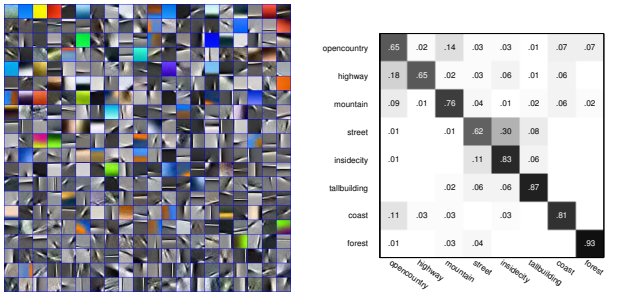
Table 1. Performance comparisons with different settings of features and dictionary, for the MSRC data.

Feature	Dictionary setting	Accuracy
Image patches	Online learning	89.06%
Image patches	Offline learning	87.50%
Image patches	K-means	67.81%
SIFT	Online learning	80.94%

4.3. LabelMe Data

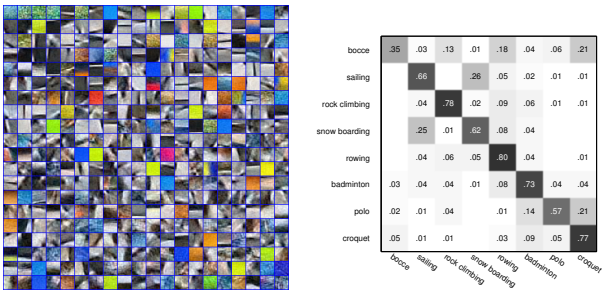
We next consider the LabelMe dataset together with annotations[2]. The LabelMe data contain 8 image classes: "coast", "forest", "highway", "inside city", "mountain", "open country", "street" and "tall building". We use the same settings of images and annotations as (Wang et al., 2009): we randomly select 200 images for each class, thus the total number of images is 1600. Each image is resized to be 256×256 pixels. For the annotations, we remove terms that occur less than 3 times, and obtain a vocabulary of 186 unique words, thus $L = 186$. There are 6 terms per annotation in the LabelMe data on average. We then randomly

[2] *http://www.cs.princeton.edu/ chongw/*



(a) (b)

Figure 5. Results for the LabelMe data. (a) The inferred dictionary with elements sorted in a decreasing order, (b) confusion Matrix over the 800 non-annotated images, with the average performance of 76.25%.



(a) (b)

Figure 6. For the UIUC-Sport data, (a): The inferred dictionary with elements sorted in a decreasing order of importance. (b): Confusion Matrix over the 688 non-annotated images.

select 800 images, and remove their annotations treating them as non-annotated images, so that the total set of images analyzed are partially annotated, as for the MSRC example.

Fig. 5(a) shows the inferred dictionary atoms, demonstrating both color and texture features. The model inferred 13 image classes, and 77 unique objects/topics. Although the model is learned using both annotated and non-annotated images, we focus on the confusion matrix for the 800 non-annotated images in Fig. 5(b), computed as above (each of the inferred image classes may be unambiguously associated with one of the true classes). In Table 2 we summarize average clustering accuracy on the annotated and non-annotated images, with results also summarized there for the UIUC-Sport data we consider next. In Table 2 we also provide a comparison to results from (Wang et al., 2009).

Table 2. Performance comparisons of confusion matrix. 'annotated' and 'non-annotated' separately denote the accuracy of confusion matrix computed over the annotated images and the non-annotated images. 'Wang' represents the result reported in (Wang et al., 2009).

	annotated	non-annotated	Wang
LabelMe	92.25%	76.25%	76%
UIUC-Sport	91.03%	69.11%	66%

4.4. UIUC-Sport Data

Finally we test our model on the UIUC-Sport dataset. The UIUC-Sport dataset contains 8 types of sports: "badminton"(200 images), "bocce"(137 images), "croquet"(236 images), "polo"(182 images), "rock climbing"(194 images), "rowing"(250 images), "sailing"(190 images), and "snow boarding"(190 images). The total number of images is 1579. With the purpose of comparison, we use the same settings of images as (Wang et al., 2009)[3]. Since the tags contain too many arbitrarily noisy words, we first obtain candidate tags belonging to 'physical entity' (Li et al., 2009) by using WordNet synsets[4], and then select the 30 most frequent words from these candidate tags; thus $L = 30$. We evenly split each class and remove annotations of half, treating them as non-annotated images. The inferred dictionary and confusion matrix is shown in Fig. 6, with the average accuracy of 69.11%, summarized in Table 2. Based on the learned posterior word distribution $\boldsymbol{\omega}_t$ for the tth image class, we can further infer which words are most probable for each image class (category). Fig. 7 shows the $\boldsymbol{\omega}_t$ for 8 classes, with the five largest-probability words displayed. A good connection is manifested between the words and image classes. The model clearly learns a good statistical distribution over words, matched to the latent image class/category. Further, the confusion matrices demonstrate that the model can infer the image class well. Therefore, the model performs well in *statistically* annotating non-annotated images (not further detailed, for brevity). The presence of the annotations assists with the clustering of the images into categories. Linkages are inferred between objects in the images and associated words (when present), and this assists clustering of images, even for those images without annotations.

The experiments above have been performed in 64-bit Matlab on a machine with 2.27 GHz CPU and 4Gbyte RAM. One MCMC run of the proposed model takes around 5, 2, 11 and 10 minutes respectively

[3]The total number reported in the paper is 1792. According to the resources that also provided in the paper (*http://vision.stanford.edu/lijiali/*), there are actually 1579 images available.

[4]*http://wordnet.princeton.edu/*

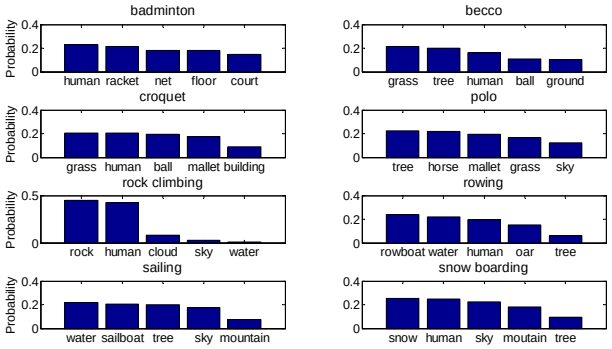


Figure 7. Inferred distributions over words for UIUC-Sport data, as a function of inferred image category. Names on the horizontal represents the annotation terms, the order of which varies across the categories. The vertical axis represents the distribution.

for the MNIST, MSRC, LabelMe and UIUC experiments (in which we simultaneously analyzed respectively 1000, 320, 1600, and 1579 total images). The proposed model could also be implemented via variational Bayesian (VB) analysis, that may yield to efficiency.

5. Conclusion

A new model has been developed to integrate topic modeling and dictionary learning into a unified Bayesian setting. In comparison with previous models, based on image features which were carefully defined (e.g., superpixels, SIFT, shape, texture, etc.), the proposed model achieves performance as good or better as existing published results. This is realized by executing the dictionary-learning component of the model directly on patches from the original image. The model is therefore not specialized to imagery, and may be applied to other problems, for example annotated audio signals. The research reported here was supported by AFOSR, ARO, DOE, ONR and NGA.

References

Barnard, K., Duygulu, P., Forsyth, D., Freitas, N., Blei, D., and Jordan, M. Matching words and pictures. *JMLR*, 2003.

Blei, D. and Jordan, M. Modeling annotated data. In *SIGIR*, 2003.

Blei, D. and MaAuliffe, J. Supervised topic models. In *NIPS*, 2007.

Blei, D., Ng, A., and Jordan, M. Latent dirichlet allocation. *JMLR*, 2003.

Du, L., Ren, L., Dunson, D., and Carin, L. Bayesian model for simultaneous image clustering, annotation and object segmentation. In *NIPS*, 2009.

Fei-Fei, L. and Perona, P. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

Griffiths, T. L. and Ghahramani, Z. Infinite latent feature models and the indian buffet process. In *NIPS*, 2005.

Li, L.-J. and Fei-Fei, L. What, where and who? classifying events by scene and object recognition. In *ICCV*, 2007.

Li, L.-J., Socher, R., and Fei-Fei, L. Towards total scene understanding: classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009.

Lowe, D. Object recognition from local scale-invariant features. In *ICCV*, 1999.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. Discriminative learned dictionaries for local image analysis. In *CVPR*, 2008.

Sethuraman, J. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4, 1994.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. Hierarchical dirichlet processes. *J. Am. Stat. Ass.*, 101, 2004.

Teh, Y. W., Görür, D., and Ghahramani, Z. Stick-breaking construction for the Indian buffet process. In *AISTATS*, volume 11, 2007.

Thibaux, R. and Jordan, M. Hierarchical beta processes and the indian buffet process. In *AISTATS*, 2007.

Wang, C., Blei, D., and Fei-Fei, L. Simultaneous image classification and annotation. In *CVPR*, 2009.

Yang, J., Yu, K., Gong, Y., and Huang, T. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

Zhou, M., Chen, H., Paisley, J., Ren, L., Sapiro, G., and Carin, L. Non-parametric bayesian dictionary learning for sparse image representations. In *NIPS*, 2009.

Zhou, M., Yang, H., Sapiro, G., Dunson, D., and Carin, L. Dependent hierarchical beta process for image interpolation and denoising. In *AISTATS*, 2011.