Modeling images using transformed IBPs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Latent feature models are an appropriate choice for image modeling, since images generally contain multiple objects or features. However, many latent feature models either do not account for the fact that objects can appear at different locations in different images or they require pre-segmentation of images and cluster the resulting segments. The recently-proposed transformed Indian buffet process (tIBP, [1]) provides a method for modeling transformation-invariant features without the need for pre-segmentation of images. In this paper we demonstrate how the tIBP can be combined with an appropriate likelihood to create a model applicable to real images. Using the the cross-correlation between images and features, we develop a novel Metropolis-Hastings inference algorithm that is both theoretically and empirically faster than previous inference techniques.

1 Introduction

Latent feature models offer an unsupervised approach to discover the hidden structure of data. Such models assume each data point is generated as a combination of latent features, and that these latent features are shared across the data set. Moreover, such models assume *all* properties of a latent feature are common to all data points – i.e. that each feature appears exactly the same in each data point. For many data sets this is a reasonable assumption. For example, microarray data are designed so that each cell consistently corresponds to a specific experiment or condition across all observations. However, for images, this assumption is less well formed.

Consider a series of images depicting a ball rolling across a floor. If a model is forced to create a new feature to explain every position of the ball as it moves across the floor, it will devote less attention to other aspects of the image and it will be unable to generalize across the ball's path, treating every position as a distinct feature. Instead, we would like to treat *some* properties of a feature – for example shape – as common across data points, but other properties – for example location – as observation-specific.

There are many applications for models that can generalize across images to discover translation-invariant features. Image tracking, for instance, allows for the discovery of mislaid bags or illegally stopped cars. Image reconstruction is used to restore partially corrupted images. Image / movie compression depends on the ability to recognize recurring image patches that can be cached across frames.

Consider a set of desirable properties for latent feature models of images. Such a model should: 1) Generalize across translations, so that a feature can appear in different locations in different images. 2) Learn both the number of features in an image and the total number of features in a dataset. 3) Be appropriate for real-valued images, without heavy preprocessing or segmentation. In Sections 2 and 6 we discuss a number of models that meet some, but not all, of these criteria.

The model that comes closest to our stated goals is the noisy-OR transformed Indian buffet process (NO-tIBP, [1]). The transformed Indian buffet process is a Bayesian nonparametric prior appropriate

for latent feature image models where the number of features is not known *a priori*, and where features can appear in different locations in different images. However, the model described in [1] uses a likelihood that is only appropriate for binary data. In addition, the inference method proposed in [1] do not scale well to large images.

In this paper, we combine the transformed IBP with a linear Gaussian likelihood model, which is appropriate for real-valued data. This model, which we call the LG-tIBP, satisfies the properties described above. In addition, we describe a new Metropolis-Hastings based inference scheme that significantly reduces the inference costs of the transformed IBP, making our model applicable to much larger datasets.

We begin by describing the Indian buffet process and related models in Section 2, with specific focus on applications in image modeling. We then present our model in Section 3, and describe the inference in Section 4. In Section 5, we show that LG-tIBP is able to recover ground truth features in simulated data, and out-performs other IBP-based image models on real-world and simulated data. Finally, in Section 6, we discuss relationships with other nonparametric models and possible future extensions.

2 Background

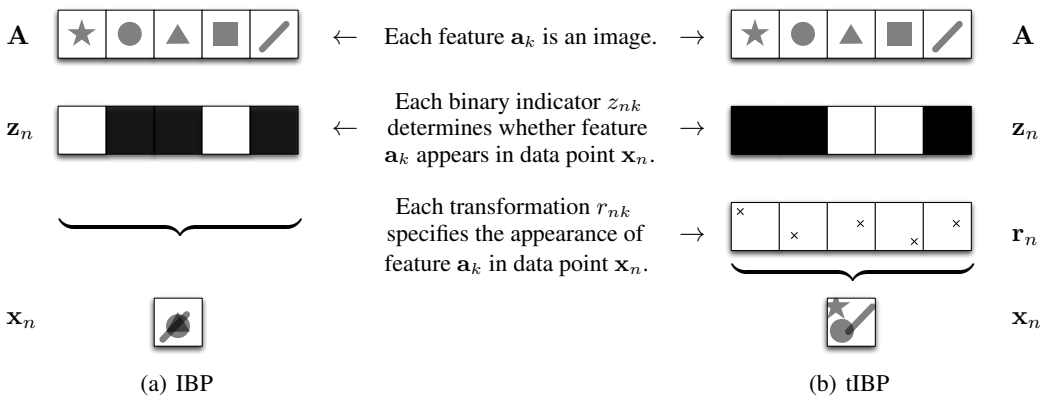
The Indian buffet process (IBP, [2]) is a distribution over binary matrices with exchangeable rows and countably infinite columns. The number of columns with non-zero entries is unbounded and grows in expectation with the number of rows. If we associate each row with a data point and each column with a latent feature, this allows us to define a nonparametric latent feature model. A row expresses a feature if the corresponding column has a non-zero entry. Thus, the number of features is not known *a priori* and can increase as we observe more data. This often matches our intuition about the real world - we do not in general know how many latent features we expect to find in our data, and we do not necessarily expect to see all possible latent features in a given data set.

Samples from the IBP can be described using the following restaurant analogy. A customer enters a restaurant serving a buffet consisting of an infinite number of dishes. She chooses a Poisson(α) number of dishes. Subsequent customers enter the restaurant, with the ith customer selecting the kth previously chosen dish with probability m_k/i, where m_k is the number of times that dish has been chosen. The ith customer then samples Poisson(α/i) new dishes. Each customer is represented as a row $\mathbf{z}_i$ of a matrix $\mathbf{Z}$, and each column represents a dish. We place a one in the element z_{ik} to indicate that the ith customer has chosen dish k.

In order to use the IBP to model data, we must select a likelihood model that determines the form of the features corresponding to the columns of the binary matrix, and the manner in which the subset of features selected by a row of the matrix are combined to generate a data point. A number of likelihoods have been proposed for the IBP, several of which are appropriate for modeling images. A linear Gaussian model [2] assumes each feature is distributed according to a multivariate Gaussian. Each row of the IBP selects a subset of these features, and generates the corresponding data point by additively superimposing these features and adding Gaussian noise. Figure 1(a) demonstrates how the nth row of the IBP-distributed binary matrix selects a subset of features to be combined in this manner. Infinite sparse factor analysis (isFA) and infinite independent component analysis (iICA) [3] extend this model by associating each non-zero entry of the IBP-distributed matrix with a Gaussian or Laplacian weight. This has the effect of allowing features to appear in different images with different intensities, and allows us to invert features. A noisy-OR model [4] associates each feature with a binary image. Each pixel in a given data point is obtained by flipping a weighted coin for each of the features (and also for the noise parameter) and setting the pixel to one if any of the coin flips were successful.

These models assume that if a feature appears in an image, it will always appear in the same location. This obviously limits their application in image modeling, where (for example) a car may appear at different locations in different images. The basic linear Gaussian, isFA, iICA, and noisy-OR models would learn a different feature for each location that a car appears in; a more appropriate model would learn that each of these features are in fact translations of a common feature.

The transformed Indian buffet process (tIBP, [1]) extends the IBP to such a setting. In the tIBP, each column of the IBP is again associated with a feature. In addition, each non-zero element of



(a) IBP (b) tIBP

Figure 1: Comparing the traditional IBP with a transformed IBP. In the IBP (left), features $\mathbf{a}_k$ are shared across the data set; data point-specific binary indicators z_{nk} determine which features contribute to data point $\mathbf{x}_{nk}$. In the transformed IBP (right), features $\mathbf{a}_k$ are shared across the data set; data point-specific binary indicators z_{nk} and transformations r_{nk} determine whether they contribute to the data point, and if so the manner in which they are transformed (respectively).

the IBP is associated with a transformation, sampled from some discrete distribution over possible transformations. Data points are obtained by transforming the subset of features selected by the IBP according to these transformations, and then combining them using some likelihood model. For the NO-tIBP [1], the features are binary, and are combined via a noisy-OR model, as before.

A binary model, such as the NO-tIBP, is inappropriate for real-valued data. In addition, the inference method proposed in [1] does not scale well to large images, as each Gibbs sweep involves evaluating the likelihood for each possible transformation of each possible feature of each data point. In the following sections, we will describe how the tIBP can be used with a linear Gaussian likelihood to give a model appropriate for real-valued images, and describe a new Metropolis-Hastings-based inference scheme that is applicable to both the noisy-OR and linear-Gaussian models.

3 Modeling real-valued images using transformed IBPs

In Section 2, we saw how the IBP can be used as the basis of an probabilistic model of images by associating each column of the IBP with an image feature. The tIBP extends such a model by associating a transformation with each element of the IBP. Returning to the restaurant analogy, we can consider each customer selecting dishes with probability proportional to their popularity. For each selected dish, the customer chooses a location on her plate independently of the locations chosen by other customers.

In the original tIBP paper, the features are binary, and are combined using a noisy-OR model. However, to better represent real data, we use Gaussian-distributed features combined them via additive superposition. We will refer to this model as the linear Gaussian tIBP (LG-tIBP), to distinguish between it and the noisy-OR tIBP (NO-tIBP) described in the previous section. More formally, we assume data are generated as follows:

1. Sample a binary matrix $\mathbf{Z} \sim \text{IBP}(\alpha)$. This determines which features (columns) are present in which data points (rows).
2. For each feature $k = 1, 2, \ldots,$ sample a Gaussian feature $\mathbf{a}_k \sim \mathcal{N}(\mathbf{0}, \sigma_a^2 \mathbf{I})$.
3. For each element z_{nk} of the binary matrix $\mathbf{Z}$, sample a transformation $r_{nk}(\cdot) \sim f(r)$.
4. For each data point $n = 1, \ldots, N$, sample an image $\mathbf{x}_n \sim \mathcal{N}(\sum_{k=1}^{\infty} z_{nk} r_{nk}(\mathbf{a}_k), \sigma_x^2 \mathbf{I})$.

Figure 1(b) shows how the IBP-distributed matrix $\mathbf{Z}$ selects *which* features contribute to an image, and the matrix of transformations $\mathbf{R}$ determines *where* they appear in that image. Note that if r_{nk} is always the identity transformation, the tIBP becomes the IBP.

In this paper, we focus on linear translations only, assuming a uniform prior over all position translations. These translations are parametrized by a vector (r_x, r_y), and the transformed feature $r(\mathbf{a}_k)$ is obtained by shifting each pixel in $\mathbf{a}_k$ by (r_x, r_y). Note that only those pixels of the shifted feature that lie within the boundaries of the final image contribute to the data.

4 Inference

We perform inference in the LG-tIBP using MCMC. At each iteration of our sampler, we sample the Gaussian-distributed features $\mathbf{A}$, the IBP-distributed binary matrix $\mathbf{Z}$, the transformations $\{r_{nk}\}$ corresponding to non-zero elements of $\mathbf{Z}$, and the hyperparameters α, σ_x and σ_a.

4.1 Sampling the binary matrix Z and the transformation matrix R

In the tIBP, the binary indicator matrix $\mathbf{Z}$ and the matrix of transformations $\mathbf{R}$ are closely coupled. In the original tIBP paper [1], each element z_{nk} of $\mathbf{Z}$ was sampled by summing over all possible values of r_{nk} to obtain

$$p(z_{nk} = 1|\mathbf{X}, \mathbf{Z}_{-nk}, \mathbf{R}_{-nk}, \mathbf{A}) = \textstyle\sum_r p(z_{nk} = 1|r_{nk}, \mathbf{X}, \mathbf{Z}_{-nk}, \mathbf{R}_{-nk}, \mathbf{A}) p(r_{nk}) \,. \tag{1}$$

The transformation r_{nk} is then sampled separately. While this scheme is applicable to real-valued features, explicitly computing a conditional distribution for each possible transformation for each feature simply cannot scale to even moderate-sized data sets.

Instead, we sample z_{nk} and r_{nk} jointly via a Metropolis-Hastings proposal. Let K_+ be the highest feature index represented in the data, excluding the current data point. Our proposal distribution for $z_{nk}, k \leq K_+$ is

$$q(z_{nk} \to z^*_{nk}) = \begin{cases} 1 & \text{if } z^*_{nk} \neq z_{nk} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Our proposal distribution for the previously unseen features follows that of Griffiths and Ghahramani [2]: We sample a number K^* of new features according to Poisson(α/N), where N is the total number of data points.

To obtain a proposal distribution for r_{nk} that closely matches our intuitions about the true posterior, we look at the *cross-correlation* between the kth feature $\mathbf{a}_k$ and the residual $\tilde{\mathbf{x}}_{n,k} = \mathbf{x}_n - \sum_{j \neq k} z_{nj} r_{nj}(\mathbf{a}_k)$. Cross-correlation is a standard tool in classical image analysis and template-matching [5]. Recall that the cross-correlation $\mathbf{u} \star \mathbf{v}$ between two real-valued images $\mathbf{u}$ and $\mathbf{v}$ is a measure of the similarity between $\mathbf{u}$ and a translated version of $\mathbf{v}$, i.e., $(\mathbf{u} \star \mathbf{v})(t) := \sum_{\tau=1}^{T} \mathbf{u}(\tau)\mathbf{v}(t+\tau)$.

Since our proposal distribution for r^*_{nk} must be strictly positive, we use

$$q(r|\mathbf{a}_k, \tilde{\mathbf{x}}_{n,k}) = (\tilde{\mathbf{x}}_{n,k} \star \mathbf{a}_k)(r) - \min\left((\tilde{\mathbf{x}}_{n,k} \star \mathbf{a}_k)(r)\right) + \epsilon \tag{3}$$

as our proposal distribution, where ϵ is some small constant to ensure ergodicity of the sampler.

The approach allows inference to be scaled to larger images than the direct method proposed by [1]. For images with D total pixels, evaluating the likelihood $p(\mathbf{x}_n|\mathbf{r}_n, z_{nk} = 1, \mathbf{Z}_{-nk}, \mathbf{A})$ requires $O(DK)$ computations, where K is the number of non-zero elements in $\mathbf{z}_n$. Therefore, evaluating Equation 1, up to a normalizing constant, is $O(D^2K)$ because the likelihood must be evaluated at every possible translation. Calculating the proposal distribution given by Equation 3 can be done via the fast Fourier transform in $O(D \log D)$. The likelihood need only be evaluated twice in the Metropolis-Hastings step, making our sampling method $O(D \max(K, \log D))$ time for each feature and observation combination.

In Figure 2, we show the proposal distribution for r^*_{nk} for a single feature and three data points. Notice that the proposal distribution peaks in the locations that best match the pattern of pixels in the feature. If there are locations that are a good match for the feature, the proposal distribution is relatively entropic. Thus, the cross-correlation proposal distribution will cause us to consider good candidates for r_{nk}.

In addition to sampling z_{nk} and r_{nk} jointly, we also resample those r_{nk} for which the corresponding $z_{nk} = 1$, using Metropolis-Hastings with proposal density given by Equation 3.

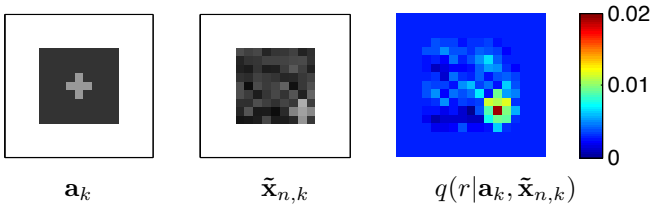


Figure 2: We use cross-correlation as our proposal distribution for the per-image, per-feature transformation r. Here, we need to propose a transformation $r_{n,k}$ of a feature $\mathbf{a}_k$ that best explains the residual $\tilde{\mathbf{x}}_{n,k}$. Note that r_{nk} need not lie within the boundaries of the image, so the borders for $\mathbf{a}_k$ and $\tilde{\mathbf{x}}_{n,k}$ indicate the range of possible values for r_{nk}.

4.2 Sampling the feature matrix A and hyper-parameters

Conditioned on $\mathbf{Z}$, $\mathbf{R}$, the data $\mathbf{X}$, and the other features $\mathbf{A}_{-k}$, the k-th feature $\mathbf{a}_k$ is Gaussian-distributed with mean $\mu_{\mathbf{a}_k}$ and covariance $\Sigma_{\mathbf{a}_k}$:

$$\mu_{\mathbf{a}_k} = \mathbf{M}\mathbf{z}_k^\top \mathbf{X}', \qquad \Sigma_{\mathbf{a}_k} = \sigma_x^2 \mathbf{M}, \qquad \mathbf{x}'_n = r_{nk}^{-1}\left(\mathbf{x}_n - \sum_{j \neq k} z_{nj} r_{nj}(\mathbf{a}_k)\right), \quad (4)$$

where we have defined $\mathbf{M} = \left(\mathbf{z}_k^\top \mathbf{z}_k + \frac{\sigma_x^2}{\sigma_a^2}\mathbf{I}\right)^{-1}$ for notational convienience. The hyperparameters α, σ_x and σ_a can be Gibbs sampled via closed form equations [6].

5 Experimental evaluation

We evaluate our model on both simulated and real-world data over three models – the linear Gaussian transformed IBP (LG-tIBP); the linear Gaussian IBP (IBP); and the noisy-OR IBP (NO-tIBP) [1]. The experiments on simulated data show that LG-tIBP is able to recover true underlying features and locations more effectively than IBP and NO-tIBP. In the experiments on real-world data, we show that our model out-performs them both on four data sets. All data sets were scaled to have zero mean and unit variance.

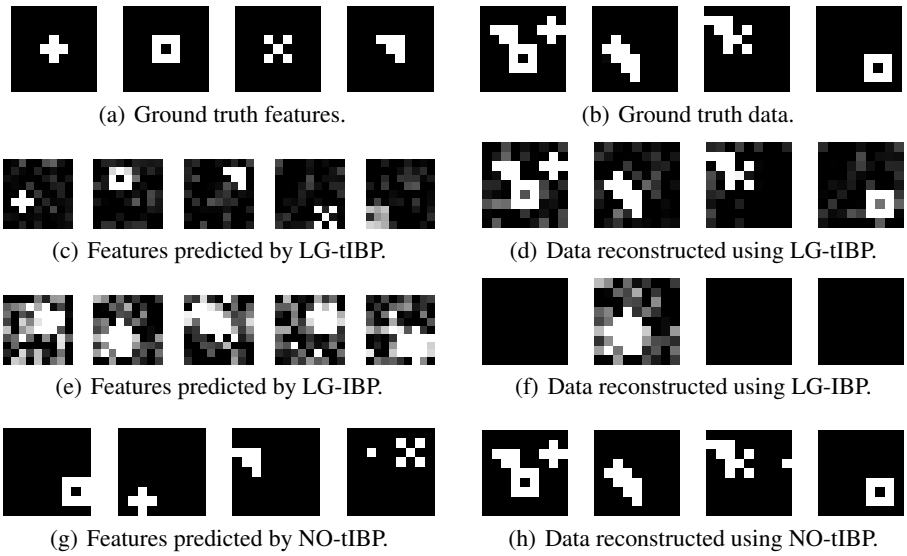
5.1 Simulated data

To qualitatively assess the ability of the LG-tIBP to find translated features, we generated data using the four features – square, triangle, cross ($+$), and St. Andrew's cross ($\times$) – shown in Figure 3(a). We generate 1000 images by selecting each feature independently with probability 0.5, and sampling a translation for each feature uniformly from the set of possible translations. The translated features were superimposed, and Gaussian noise was added (Figure 3(b)).

We trained all three models on data generated from this process. For the noisy-OR model, the data was thresholded to be binary. For the two linear Gaussian models, we ran our code for 1000 iterations. Due to speed limitations of the NO-tIBP code, we were only able to run the NO-tIBP on 100 images for 500 iterations (which took 14 hours). By comparison, 1000 iterations of the LG-tIBP on 1000 images took around 3 hours. The IBP was implemented using the same code as the LG-tIBP, with the transformations r_{nk} fixed to the identity transformation.

Figure 3 shows the features found using three methods. The NO-tIBP is designed to be run on binary data, and so, unsurprisingly, achieves good results. We can see that, while the LG-tIBP is able to generalize across locations and achieves similar results to the NO-tIBP, the IBP struggles to find common structure and performs poorly.

[1] Source code and data sets will be available online after blind review.



(a) Ground truth features. (b) Ground truth data.

(c) Features predicted by LG-tIBP. (d) Data reconstructed using LG-tIBP.

(e) Features predicted by LG-IBP. (f) Data reconstructed using LG-IBP.

(g) Features predicted by NO-tIBP. (h) Data reconstructed using NO-tIBP.

Figure 3: Comparing LG-tIBP with LG-IBP on synthetic data. The ground truth data has Gaussian noise ($\mu = 0, \sigma = 0.05$).

5.2 Real-world data

To show that the performance achieved on simulated data in Section 5.1 carry over to real images, we evaluated the LG-tIBP on four image datasets (Figure 4).

(a) 171 screen shots from 1981 video game "Donkey Kong,"[2] cropped to a frame size of 39×39 pixels (Figure 4(a)).

(b) 181 screen shots from 1989 video game "Super Mario Land,"[3] cropped to a frame size of 39×41 pixels (Figure 4(b)).

(c) 226 frames taken from a video from the CAVIAR dataset of people walking in a Lisbon shopping center[4] cropped to a frame size of 39×39 pixels (Figure 4(c)).

(d) 186 frames taken from a video of an road intersection[5] cropped to a frame size of 41×39 pixels (Figure 4(d)).

(a) Donkey Kong (b) Super Mario (c) People (d) Cars

Figure 4: Sample images from four data sets (four images per data set). The datasets reflect varying levels of complexity from simple videogames to real-world scenes.

For each data set, we trained both LG-tIBP and IBP on random 80% of the images, with the remaining 20% (randomly selected) for testing. We did not evaluate the NO-tIBP on this data due to its slowness

[2] `http://www.youtube.com/watch?v=EhFV5-qbbIw`

[3] `http://www.youtube.com/watch?v=EV0qJd-phvQ`

[4] Raw data available at `http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/`

[5] Raw AVSS PV Easy data available at `http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html`

	Predictive log likelihood				RMSE			
	Donkey Kong	Super Mario	People	Cars	Donkey Kong	Super Mario	People	Cars
LG-IBP	-1.64	-1.40	-1.32	-1.43	1.20	0.961	0.822	1.01
LG-tIBP	-1.16	-1.15	-1.03	-1.20	0.508	0.578	0.441	0.616

Table 1: Test set predictive log likelihood and RMSE on four data sets.

and because our data is not binary. To evaluate, we randomly choose half of the pixels in every test image, using the rest to estimate matrix $\mathbf{Z}$ and $\mathbf{R}$. We then computed the held-out likelihood of the masked pixels. Table 1 shows the average per-pixel predictive log likelihood and RMSE for the datasets. As we can see, the LG-tIBP achieves better predictive performance across all four data sets.

Figure 5 shows the reconstructions obtained and features used by both the LG-tIBP and the IBP on some test set images. The first column shows the true image, and the second column shows the subset of pixels that are used to learn $\mathbf{z}_n$ and $\mathbf{r}_n$. The third and fifth columns show the images recovered by the IBP and LG-tIBP, respectively. In the fourth and sixth columns, we have colored each feature a different color, and superimposed the features for which $z_{nk} = 1$ onto the original image. Colors are consistent across each task for each model – i.e. the red feature in the first Donkey Kong IBP reconstruction is the same as the red feature in the second Donkey Kong reconstruction. We can see that the IBP, in general, only matches the background of an image. In images without a strong background, the IBP often does not predict any features – since the training set may not have included features in the same location as the image being predicted. Conversely, the LG-tIBP is able to identify shapes that appear in different locations in different images. For example, in the first two rows of Figure 5, the LG-tIBP identifies Donkey Kong (magenta), a fireball (yellow), and a pie (blue), in addition to the background (green); conversely, the IBP only identifies the background.

6 Discussion and further work

In this paper, we have shown how the transformed Indian buffet process can be combined with a linear Gaussian likelihood (LG-tIBP) to model images where the number of features is unknown and where the properties of features can vary between images. While we have focused on translation of features, this can in principle be extended to other transformations such as scaling, rotation or changes in brightness. Since the inference scheme for the original NO-tIBP does not scale well to larger images, we propose a more efficient inference scheme based on the cross-correlation between images and features.

In this paper, we have focused on Bayesian nonparametric image models based on the Indian buffet process. In addition, a number of Bayesian nonparametric image models have been developed based on the Dirichlet process. The Dirichlet process [7] is a distribution over discrete probability distributions with a countably infinite number of atoms. Of particular relevance to this paper, the transformed Dirichlet process [8] applies an image-specific transformation to each feature distribution, so that a feature may appear different (e.g. different location, different scale) in different images.

Models that use the Dirichlet process can generally be seen as a nonparametric version of finite mixture models – each observation is associated with a single cluster, and its properties are sampled i.i.d. from that cluster's distribution. In order to model images using Dirichlet processes, we must represent the images as a collection of "objects" – either pixels, super-pixels, or segments of the image. Each object is assumed to be distributed according to one of an unbounded number of feature distributions.

In contrast, the IBP can be seen as a nonparametric version of a latent variable model. Under IBP-based models such as the transformed IBP, each data point is treated not as a collection of exchangeable observations (e.g. pixels or segments) but as a whole. As a result, we do not need to segment our image and obtain descriptors for each segment – our method can be applied to the raw image itself. However, additional information or pre-processing could result in better features and faster convergence.

The transformed Indian buffet process assumes that each feature appears only once in each image. One avenue for future work is to extend this to allow multiple instances of a feature in a given image.

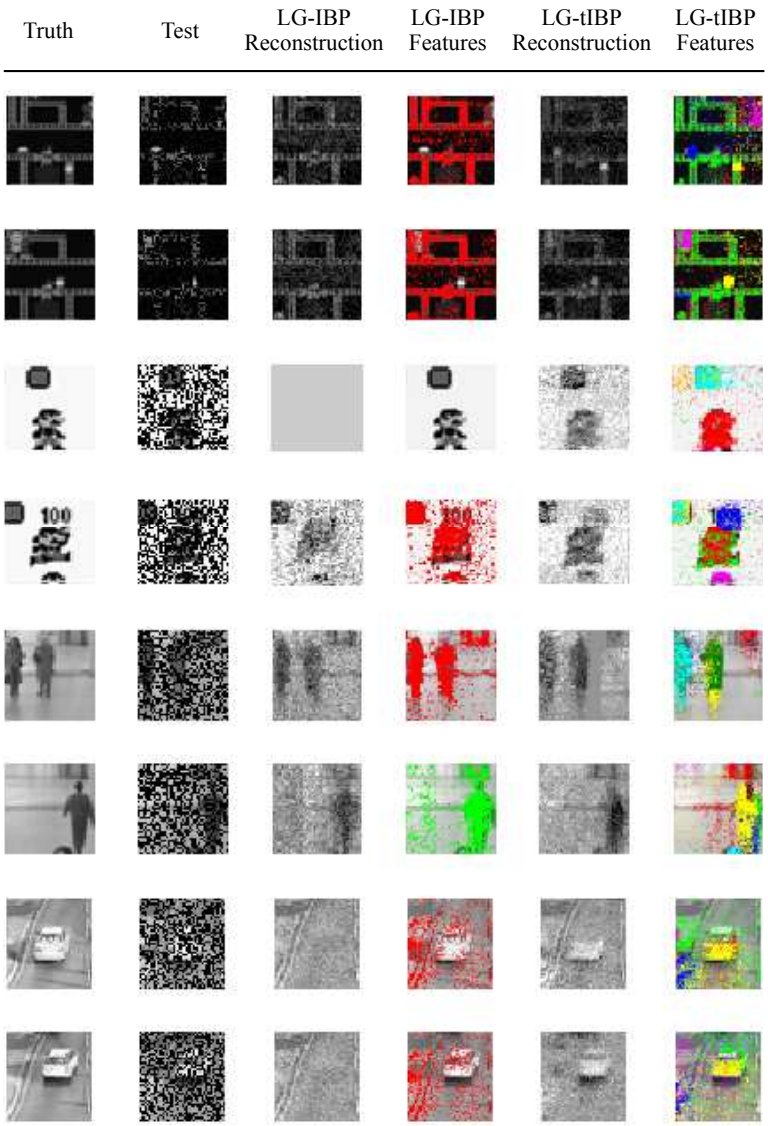


Figure 5: Reconstructions of test data by the LG-IBP and LG-tIBP. First column: True image. Second column: 50% of true image, used to learn $\mathbf{z}_n$ and $\mathbf{r}_n$. Third column: Image reconstructed by LG-IBP. Fourth column: Features used by LG-IBP, superimposed on true image. Each color is a feature; colors are consistent between columns. Fifth column: Image reconstructed by LG-tIBP. Sixth column: Features used by LG-tIBP, superimposed on true image. Each color is a feature; colors are consistent between columns.

The infinite gamma-Poisson process [9] is a distribution over infinite non-negative integer valued matrices, that has been used for image modeling. The likelihood model used in [9] is similar to that used in [8] and requires pre-segmentation of images. The ideas behind the transformed Indian buffet process could be extended to a transformed gamma-Poisson process, which would allow repetition of features and would be appropriate for non-segmented images.

While we have substantially improved inference both theoretically and practically over NO-tIBP, inference still scales poorly to large datasets. Adapting this method to variational inference or techniques that can distribute computation across processing units would help enable real world applications like image tracking, enabling better unsupervised discovery of translation-invariant features in image data.

References

[1] Austerweil, J. L., T. L. Griffiths. Learning invariant features using the transformed Indian buffet process. In *Advances in Neural Information Processing Systems*. 2010.

[2] Griffiths, T. L., Z. Ghahramani. Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems*. 2005.

[3] Knowles, D., Z. Ghahramani. Infinite sparse factor analysis and infinite independent component analysis. In *7th International conference on Independent Component Analysis and Signal Separation*. 2007.

[4] Wood, F., T. L. Griffiths, Z. Ghahramani. A non-parametric Bayesian method for inferring hidden causes. In *Uncertainty in Artificial Intelligence*. 2006.

[5] Duda, R. O., P. E. Hart. *Pattern classification and scene analysis*. Wiley, 1973.

[6] Doshi-Velez, F. *The Indian buffet process: Scalable inference and extensions*. Master's thesis, University of Cambridge, 2009.

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

[8] Sudderth, E. B., A. Torralba, W. T. Freeman, et al. Describing visual scenes using the transformed Dirichlet process. In *Advances in Neural Information Processing Systems*. 2005.

[9] Titsias, M. The infinite gamma-Poisson feature model. In *Advances in Neural Information Processing Systems*. 2007.