

# Epitomized Priors for Multi-labeling Problems

Jonathan Warrell, Simon J.D. Prince and Alastair P. Moore  
University College London  
Malet Place, London, WC1E 6BT  
{j.warrell, s.prince, a.moore}@cs.ucl.ac.uk

## Abstract

Image parsing remains difficult due to the need to combine local and contextual information when labeling a scene. We approach this problem by using the epitome as a prior over label configurations. Several properties make it suited to this task. First, it allows a condensed patch-based representation. Second, efficient E-M based learning and inference algorithms can be used. Third, non-stationarity is easily incorporated. We consider three existing priors, and show how each can be extended using the epitome. The simplest prior assumes patches of labels are drawn independently from either a mixture model or an epitome. Next we investigate a ‘conditional epitome’ model, which substitutes an epitome for a conditional mixture model. Finally, we develop an ‘epitome tree’ model, which combines the epitome with a tree structured belief network prior. Each model is combined with a per-pixel classifier to perform segmentation. In each case, the epitomized form of the prior provides superior segmentation performance, with the epitome tree performing best overall. We also apply the same models to denoising binary images, with similar results.

## 1. Introduction

We investigate epitome priors for *image parsing*, the task of estimating a label for each pixel in a scene corresponding to its object category (see Figure 1). In general, the prior embodies knowledge about which configurations of labels are likely, and which not. For instance, we may see a chair next to a table, above the floor and surrounded by wall, but are unlikely to see a chair above a table on top of a window.

Recent approaches (e.g. [6, 15]) use a local unary classifier to provide an initial labeling, and a prior to disambiguate these estimates using larger context. This prior must be able to describe complex high-dimensional distributions over label configurations. Further desirable characteristics include (i) efficient parameterization, (ii) effective learning and inference algorithms, and (iii) the ability to incorporate aspects such as symmetry constraints and non-stationarity (i.e. variation of the prior with location in an image).

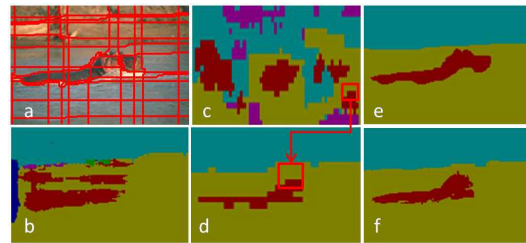


Figure 1. Using an epitome prior for multi-class segmentation. We first divide the image into a regular lattice of superpixels (a). A unary classifier provides an initial per-pixel class estimate (b). We refine this estimate by combining it with patches from an epitome prior learned at the superpixel level (c) to produce a coarse segmentation (d). This segmentation is refined (iteratively) using per-image class color models to produce a pixel level segmentation (f); the ground truth is shown in (e).

The simplest priors use undirected models such as Markov random fields (MRFs) or conditional random fields (CRFs) to describe relationships between neighboring labels [5, 10, 15]. Efficient inference algorithms are known for these models. Unfortunately, since only local relationships are considered, these priors only include information about pairwise object adjacency, and cannot represent interactions of more than two objects, or object shape.

These undirected models have been extended to incorporate potentials over larger clique sizes. He and Zemel [6] learn a prior based on overlapping global and local ‘label features’ in their Multiscale Conditional Random Field model. This allows for a rich representation, but sampling is required during inference. In contrast, Kohli et al. presented efficient algorithms to deal with large clique sizes [9], but with restrictions on the types of potential function modeled and hence limited representational ability.

Another approach is to use directed models [1, 3, 4, 17]. Contrasting possibilities include the Markov structured prior of Domke et al. [3] and the tree structured prior of Feng et al. [4]. These directed models have algorithmic advantages and only mild impositions on representation. Like most priors though, they have difficulties incorporating non-

stationarity without a drastic increase in parameters.

The epitome [8] forms a condensed representation of an image from which patches can be drawn to generate the original. The epitome was not intended as a prior and originally described distributions over continuous values rather than multi-valued labels. However, it has recently been used with discrete features [13] and several characteristics encourage its use as a prior. First, it can represent complex image data with a small number of parameters. Second, its generative form means that efficient E-M algorithms can be used. Third, non-stationarity is easily incorporated.

In this paper we apply the epitome to image labeling. We compare three priors and their ‘epitomized’ versions. The first involves independent patches, that are either drawn from a mixture model, or an epitome. In the second model, we extend the directed prior of Domke et al. [3] to form a ‘conditional epitome’. In the third we adapt the tree structured belief network [4] to form an ‘epitome tree’.

The structure of the paper is as follows. Section 2 outlines our three models. In section 3 our priors are used to denoise handwritten digits. In section 4 we investigate multi-class segmentation on the Corel and Sowerby databases. Section 5 then offers a summary and future directions.

## 2. Methods

For each model we assume that we observe a feature vector  $\mathbf{x}_s$  at each pixel  $s$ , where  $s$  indexes the pixel sites  $S = \{s_1 \dots s_S\}$ . For each site, we wish to infer the values of an unobserved label  $l_s$  that represents the class of object present at this pixel. The label takes values  $1 \dots K$  where  $K$  is the total number of object classes being considered. We assume that we also have the output of a generative or discriminative unary classifier  $\psi_{sk}$  providing a measure of evidence for the class  $k$  being present at site  $s$ .

This paper concerns prior distributions over configurations of the label field  $\mathbf{l}$ . Rather than explicitly model the joint configuration of the entire image, our models operate on square patches of the image/label fields  $p_n$ ,  $n = 1 \dots N$ ,  $p_n \subset S$ . These may or may not overlap depending on the particular model. We sometimes use relative notation so that  $\mathbf{l}_{p_n(i)}$  denotes the label  $l$  that is associated with the  $i$ ’th pixel of the  $n$ ’th patch.

### 2.1. Epitomized Mixture of Multinomials

We first consider a mixture of multinomials (MoM) prior model (illustrated for the 1-d case in Figure 2). In this model, non-overlapping patches are modeled independently. There is a hidden variable  $h$  associated with each patch, representing which of  $C$  possible components from the multinomial mixture is relevant for this patch. When component  $c$  is active (i.e.  $h = c$ ), the labels  $l_{p(i)}$  are modeled as independent draws from the multinomial distribu-

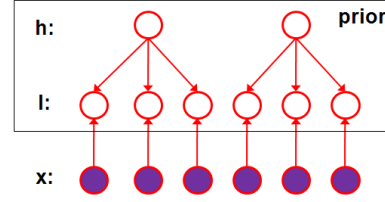


Figure 2. (Epitomized) Mixture of Multinomials Model. The labels,  $\mathbf{l}$ , are dependent on the observations at each pixel,  $\mathbf{x}$ , and hidden variables,  $\mathbf{h}$ , which select components from a multinomial mixture or epitome for non-overlapping patches.

tions for  $c$  at each site  $i$  across the patch  $p$ . The parameters  $\theta_{cik}$  describe the probability of observing label  $k$  at pixel  $i$  within the patch when component  $c$  is active. We can write this prior as:

$$\begin{aligned} Pr(\mathbf{l}) &= \prod_n \sum_c Pr(h_n = c) Pr(\mathbf{l}_{p_n} | h_n = c) \\ &= \prod_n \sum_c \alpha_c \prod_i \theta_{c,i,l_{p_n(i)}} \end{aligned} \quad (1)$$

where  $\alpha_c$  is the weight of the  $c$ ’th mixture component.

One disadvantage of this model is that it is rather wasteful of parameters: since the grid of image patches is regular and non-overlapping, the mixture components must explicitly encode all small translations of label configurations. This problem can be resolved by introducing the epitomized mixtures of multinomials (EMoM) model.

As before we model the image as consisting of regular, non-overlapping image patches. However, we now represent the parameters of the prior as a single 2-d array termed an epitome. An epitome with sites  $T = \{t_1 \dots t_T\}$  is organized into  $T$  overlapping epitome patches  $q_{1 \dots T}$  of the same size as the image patches. Here, we have assumed a toroidal structure so patches that start at the very bottom of the epitome are completed at the top. As in the MoM model, there is a hidden variable  $h$  associated with each image patch. However, now the hidden variable takes values from  $1 \dots T$  and indexes the position within the epitome from which the patch of multinomial parameters are taken. The EMoM model can be written as:

$$\begin{aligned} Pr(\mathbf{l}) &= \prod_n \sum_t Pr(h_n = t) Pr(\mathbf{l}_{p_n} | h_n = t) \\ &= \prod_n \sum_t \alpha_t \prod_i \theta_{q_t(i),l_{p_n(i)}} \end{aligned} \quad (2)$$

where  $\alpha_t$  is the probability of selecting patch  $q_t$  of parameters from the epitome and  $\theta_{tk}$  are the multinomial parameters at epitome position  $t$  for class  $k$ . Hence the term  $\theta_{q_t(i),l_{p_n(i)}}$  evaluates the probability of observing label  $l_{p_n(i)}$  at the  $i$ ’th pixel of the  $n$ ’th image patch  $p_n$  by taking the probability value stored at the  $i$ ’th pixel of the  $t$ ’th epitome patch  $q_t$ . Figure 3 summarizes the main notation.

	Image Domain	Epitome Domain
Sites:	$s = 1 \dots S$	$t = 1 \dots T$
Patches:	$p_1 \dots p_N$	$q_1 \dots q_T$
Inter-patch site index:	$i = 1 \dots I$	$i = 1 \dots I$
Class index	$k = 1 \dots K$	$k = 1 \dots K$
Associated variables:	$x_s$ (observed data) $l_s$ (labels) $\psi_{sk}$ (unary classifier outputs)	$\alpha_t$ (epitome prior) $\theta_k$ (class likelihoods)

Figure 3. Summary of notation.

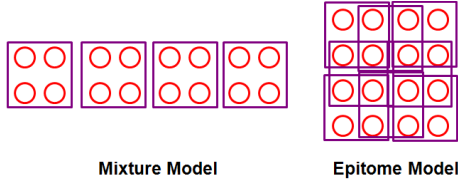


Figure 4. Comparing mixture and epitome models. For the same number of parameters, the epitome is more expressive than a simple mixture. The mixture above models 4 patches separately, while the epitome extracts 9 patches by arranging the variables in a 2-d array. 16 patches can be extracted if a toroidal structure is assumed.

Figure 4 illustrates the advantages of this parameterization. The epitome enables us to build expressive mixture models using a restricted number of parameters due to the effective sharing of  $\theta$  elements between mixture components. Moreover, translation invariance is automatically built into the epitomic representation and the model doesn't waste resources modeling small patch shifts.

### 2.1.1 Training

For training, we use the expectation maximization (EM) algorithm [2] to maximize a lower bound on the log-likelihood of a set of training label patches,  $\mathbf{I}_{1 \dots N}$ , given the epitome parameters,  $e = \{\alpha, \theta\}$ . For the E-step, we calculate the posterior probabilities  $\gamma_{nt}$  that training patch  $n$  was generated from the  $t$ 'th patch in the epitome:

$$\gamma_{nt} = \frac{\alpha_t \prod_i \theta_{q_t(i), l_{p_n(i)}}}{\sum_{t'} \alpha_{t'} \prod_i \theta_{q_{t'}(i), l_{p_n(i)}}} \quad (3)$$

In the M-step we update the epitome parameters:

$$\alpha_t = \frac{\sum_n \gamma_{nt}}{N} \quad (4)$$

$$\theta_{tk} = \frac{\sum_{n, i \in \Omega_t} \gamma_{ni} \cdot [l_{p_n(\phi_t(i))} = k]}{\sum_{n, i \in \Omega_t, k'} \gamma_{ni} \cdot [l_{p_n(\phi_t(i))} = k']} \quad (5)$$

where the set  $\Omega_t = \{i \in T | t \in q_i\}$  represents sites  $t$  in the epitome whose patches  $q_i$  include the given site  $i$ , and  $\phi_t(i) = j$  s.t.  $q_i(j) = t$  is the position of site  $t$  in epitome patch  $q_i$ .  $[\cdot]$  is the zero-one indicator function. The

weighted sums in Equation 5 thus consider all the possible epitome patches that overlap with a given site  $t$ , combining the evidence for class  $k$  at  $t$  from across the training data by looking at its frequency of occurrence at the relevant positions in the training patches.

### 2.1.2 Inference

For inference in multi-labeling tasks we combine the EMoM prior with a unary classifier (see Figure 2). The unary classifier takes the feature vector,  $\mathbf{x}_s$ , and gives a distribution  $\psi_{s, k=1 \dots K}$  over the class label at  $l_s$ . To combine these unary estimates  $Pr_{un}(\mathbf{l}|\mathbf{x}) = \psi$  with our learned prior  $Pr_{pr}(\mathbf{l})$ , we treat them as CRF potentials via:

$$Pr(\mathbf{l}|\mathbf{x}) = \frac{1}{Z} Pr_{un}(\mathbf{l}|\mathbf{x})^\lambda \cdot Pr_{pr}(\mathbf{l}) \quad (6)$$

where  $\lambda$  is a weighting factor. We could in principle learn the unary classifier and prior simultaneously. However, the piecewise approach makes training more tractable, and has been shown to give good results (eg. [6, 15, 16]).

It is difficult to directly maximize the log posterior  $\log[Pr(\mathbf{l}|\mathbf{x})]$  of the labels given the observed data because of the hidden variables in the prior. Instead we use a posterior EM approach in which we maximize a lower-bound on the log-posterior. Due to the assumption of independence, inference is performed for each patch in the test image  $\mathbf{I}_{1 \dots N}$  separately. In the E-step, we calculate the posterior probabilities  $\gamma_{nt}$  that the current assigned labels in the patch were generated from each site  $t$  in the epitome (exactly as in Equation 3). In the M-step we maximize the current bound over  $\mathbf{l}$ . To this end, we update  $l_{p_n(i)}$  as:

$$l_{p_n(i)} = \operatorname{argmax}_k \left\{ \sum_t \gamma_{nt} [\lambda \log \psi_{p_n(i), k} + \log \theta_{q_t(i), k}] \right\} \quad (7)$$

## 2.2. Conditional Epitome

The above model uses independent non-overlapping patches, limiting its representation power. Jojic et al. [8] proposed to resolve this problem by using independent overlapping patches. This was adequate for their applications but unattractive in the context of a prior as it will lead to overconfidence. In this subsection and the following, we propose two alternative extensions to transform the epitome into a model of the entire label field.

The conditional epitome was inspired by the directed model priors of Domke et al. in [3]. In 1-d these take the form of an  $n$ 'th order Markov chain (see Figure 5a). In 2-d, Domke et al. propose parent-child relationships such that for a  $5 \times 5$  patch, the central (child) site is conditioned on (parent) sites above and to the left (Figure 5b). Samples can be drawn from the distribution by scanning the sites in

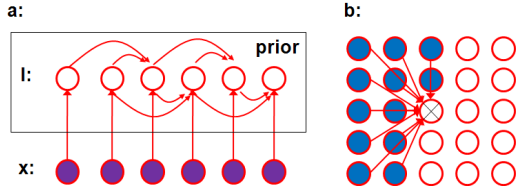


Figure 5. a) Conditional Epitome Model. Each label  $l$  is conditioned both on observations  $x$ , and nearby preceding  $l$ 's. The factors within the prior are modeled as conditional mixture or epitome distributions. b) Parent (blue) and child (cross) nodes in the 2-d version of the model are arranged as shown.

column-major order. Domke et al. argue this model is simpler than undirected forms, such as MRFs and CRFs, but the Markov blanket over individual pixels is similar.

A logical choice for describing the dependencies between parents and child in our case would be a conditional mixture of multinomials (CMoM). This is formed by dividing a joint MoM distribution over parents and child with the marginal over just the parents. In our proposed model we go a step further, and substitute the epitome to create a conditional epitome model (or ECMoM).

For each image site  $s$  we have parents  $pa(s) \in S$ . Unlike the independent patches model which had non-overlapping patches in the image, now the patches are densely extracted from both image and epitome. These are denoted by  $p_s$  and  $q_t$ , where  $p_s = \{pa(s), s\}$ , and  $q_t$  is defined as before (but with an irregular patch shape to match the parent/child combinations in the image). As well as these patches of size  $I = |\{pa(s), s\}|$ , we also extract patches of size  $I - 1$  which are notated  $p_s^*$  and  $q_t^*$ , where  $p_s^* = \{pa(s)\}$  and  $q_t^* = q_t(1 \dots I - 1)$ . Using this notation, the conditional distribution for the ECMoM model is defined as:

$$Pr(l_s | \mathbf{l}_{pa(s)}) = \frac{Pr_{pr}(l_s, \mathbf{l}_{pa(s)})}{Pr_{pr}(\mathbf{l}_{pa(s)})} = \frac{\sum_t \alpha_t \prod_i \theta_{q_t(i), l_{p_s(i)}}}{\sum_t \alpha_t \prod_{i^*} \theta_{q_t^*(i^*), l_{p_s^*(i^*)}}} \quad (8)$$

where  $i \in \{1 \dots I\}$  and  $i^* \in \{1 \dots I - 1\}$ . Because of the directed form of the model, the joint distribution across the image is formed by simply taking the product at all sites:

$$Pr(\mathbf{l}) = \prod_s Pr(l_s | \mathbf{l}_{pa(s)}) \quad (9)$$

### 2.2.1 Training

Since we are now modeling conditional rather than joint distributions, we maximize the conditional log likelihood of the data given the parameters. Following [3], this is achieved by gradient ascent, as the standard E-M bound is no longer applicable. Training data comes in the form of  $n = 1 \dots N$  training patches, where  $p_n(1 \dots I - 1) =$

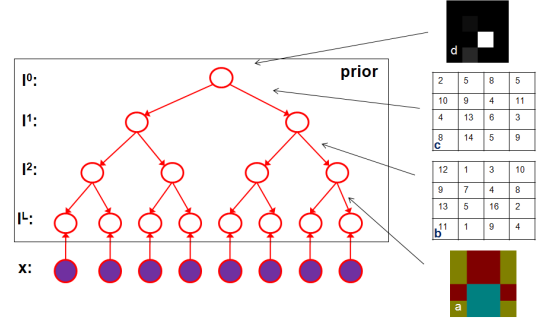


Figure 6. Epitome Tree. Class labels  $l^L$  are dependent on the observations,  $x$ , and a tree of unobserved labels  $l^{0 \dots L-1}$ , which forms the prior. Links from levels 2 to  $L$  are modeled as an epitome over class labels (a). Links from levels 0 to 1, and 1 to 2 are modeled as epitomes over patch indices at the next level down (b,c). A prior is also placed across the labels at the tree root (d).

$pa(p_n(I))$ . The conditional log-likelihood is then written:

$$\begin{aligned} L &= \sum_n Pr(l_{p_n(I)} | \mathbf{l}_{p_n(1 \dots I-1)}) \\ &= \sum_n \log \sum_t \alpha_t \prod_i \theta_{q_t(i), l_{p_s(i)}} - \\ &\quad \sum_n \log \sum_t \alpha_t \prod_{i^*} \theta_{q_t^*(i^*), l_{p_s^*(i^*)}} \quad (10) \end{aligned}$$

We reparameterize  $\alpha$  and  $\theta$  by  $a$  and  $b$  where  $\alpha = \text{softmax}[a]$  and  $\theta_t = \text{softmax}[b_t]$  to ensure that the mixture weights and multinomial parameters sum to one. Then we calculate gradients with respect to  $a$  and  $b$  and perform unconstrained optimization.

### 2.2.2 Inference

As for the EMoM model, we treat the prior as a component in a CRF framework (see Equation 6 and Figure 5a). Inference proceeds by sampling from  $Pr(\mathbf{l} | \mathbf{x}, \mathbf{e})$ , and estimating the maximum posterior marginals (MPM) at each site  $s$ . Samples are easily drawn from the directed model with ancestral sampling across all sites  $s = 1 \dots S$ , where:

$$\begin{aligned} Pr(l_s = k | \mathbf{x}_s, \mathbf{l}_{pa(s)}, \mathbf{e}) &= \frac{1}{Z_s} Pr_{un}(l_s = k | \mathbf{x}_s)^\lambda \cdot Pr_{pr}(l_s = k | \mathbf{l}_{pa(s)}) \\ &= \frac{1}{Z_s} \psi_{sk}^\lambda \cdot \frac{\sum_t \alpha_t \prod_i \theta_{q_t(i), l_{p_s(i)}}}{\sum_t \alpha_t \prod_{i^*} \theta_{q_t^*(i^*), l_{p_s^*(i^*)}}} \quad (11) \end{aligned}$$

### 2.3. Epitome Tree

The conditional epitome does model the entire label field, but unfortunately the learning algorithm relies on straightforward optimization and is hence inefficient and prone to local minima. To resolve this problem we investigate a third model inspired by the Tree Structured Belief

Network prior (TSBN) [4]. In this model, a quad-tree is used to define a prior over labels, where each group of four pixel sites at level  $L$  are dependent on a common parent at level  $L - 1$ , and groups of 4 sites at this level are in turn dependent on a common parent at level  $L - 2$  (see Figure 6 for the 1-d case). The model is completed by defining a prior over the labels at level 0.

In the TSBN, each level takes the same set of  $K$  labels, where  $K$  represents the number of classes at level  $L$ . The relationship between parent and child is the same as that between hidden and observed variables in a mixture model. The epitome tree (or ETSBN) directly substitutes epitomes for these mixtures and learns a separate epitome for each level in the tree. We note however an important difference to the TSBN: In the epitome tree, labels at each level are not from a common set. The epitome at level  $L$  represents a distribution across patches of class labels as before, but the epitome at level  $L - 1$  represents a distribution over groupings of patches at level  $L$ , using the patch indices as the labels (see Figure 6).

We thus extend our notation to incorporate level indices. We denote the parameters of the epitomes at each level by  $\theta^j, j \in \{0 \dots L\}$ . The prior at the top of the tree is represented as  $\theta^0$  rather than  $\alpha$ .  $\theta^L$  is an epitome over the class indices, and hence is indexed by  $\theta_{tk}^L, t \in \{1 \dots T\}, k \in \{1 \dots K\}$ . The remaining  $\theta^{j=1 \dots L-1}$  are epitomes over patch indices, and hence are indexed by  $\theta_{t_1 t_2}^j, t_1, t_2 \in \{1 \dots T\}$  (assuming the same number of epitome sites at all levels), while the prior  $\theta^0$  is indexed only as  $\theta_t^0$ . We also need to extend the sites in the image domain over all levels  $s^j$ , and to define a network of parent child relationships on the sites via  $\forall s_1 \in s^j, \exists s_2 \in s^{j-1} \text{ s.t. } pa(s_1) = s_2$ . Patches are also defined over each level, giving  $p_n^j \in s^j$ , and associated labels are written  $l_{p_n^j(i)}$ .

The joint distribution over a particular arrangement of labels in the tree is:

$$Pr(\mathbf{l}) = \theta_{l_{s_0}}^0 \prod_{j=1 \dots L} \prod_{s \in s^j} \theta_{l_{pa(s)}, l_s}^j \quad (12)$$

The probability of a label map at level  $L$  is found by summing over all possible label configurations at higher levels.

### 2.3.1 Training

We assume the training data consists of  $1 \dots N$  training label maps  $\mathbf{l}_{train}^n$ . For each training example we will have an associated tree of labels  $\mathbf{l}^{n,j}$ , built across sites  $s^j, j \in \{1 \dots L\}$ . The labels at level  $L$  are set to the observed label maps, hence  $\mathbf{l}^{n,L} = \mathbf{l}_{train}^n$ . We aim to estimate tree parameters  $\theta$  based on the observed labels, while marginalizing over the hidden labels higher in the tree. This is accomplished using a combination of EM and belief propagation (BP). In the E-step, a message passing scheme is used to calculate

the expected values of the hidden variables given the current estimates of  $\theta$ . In the M-Step we then update  $\theta$  using the expected transition frequencies between labels.

In the E-step, for each training example we run an upward and downward message-pass, denoting the respective messages as  $\mu^1$  and  $\mu^2$ . In the former, messages at level  $L$  are initialized to  $\mu_s^1(k) = 1$  if  $\mathbf{l}^L = k$ , and to 0 otherwise. We pass messages from levels  $j = L - 1 \dots 0$ :

$$\mu_s^1(t) = \prod_i \sum_k \theta_{qt(i),k}^{j+1} \cdot \mu_{ch_i(s),k}^1 \quad (13)$$

where  $s \in s^j$ , and  $k \in \{1 \dots K\}$  if  $j = L - 1$ , and  $k \in \{1 \dots T\}$  otherwise. The function  $ch_i(s)$  returns the  $i$ 'th child of site  $s$ . In the downward pass, we initialize  $\mu_{s_0}^2$  to  $\theta^0$ , i.e. the prior across the tree. Messages are then passed from levels  $j = 1 \dots L$ :

$$\mu_s^2(k) = \sum_t \mu_{pa(s)}^2(t) \cdot \theta_{qt(i_s),k}^j \cdot \prod_{i_{sib(s)}} \sum_{k'} \theta_{qt(i),k'}^j \cdot \mu_{sib_i(s),k'}^1 \quad (14)$$

where again  $s \in s^j$ , and  $k \in \{1 \dots K\}$  if  $j = L - 1$ , and  $k \in \{1 \dots T\}$  otherwise. The function  $sib_i(s)$  returns the  $i$ 'th sibling of site  $s$ , and  $i_s$  denotes  $s$ 's patch relative index.

The M-step then updates the parameters based on the expectations of the transition frequencies:

$$\theta_{tk}^j = \frac{\sum_n \sum_{s \in s^j} Pr(l_{pa(s)}^n = t' \text{ s.t. } \phi_{t'}(i_s) = t, l_s^n = k | \mathbf{l}^n, \theta)}{\sum_n \sum_{s \in s^j} \sum_{k'} Pr(l_{pa(s)}^n = t' \text{ s.t. } \phi_{t'}(i_s) = t, l_s^n = k' | \mathbf{l}^n, \theta)} \quad (15)$$

where  $\phi$  is defined as in section 2.1.1. These can be calculated from the messages of the E-step by:

$$Pr(l_s = k, l_{pa(s)} = t | \mathbf{l}^L, \theta) = \mu_s^1(k) \cdot \mu_{pa(s)}^2(t) \cdot \theta_{t,k}^j \cdot \prod_{i_{sib(s)}} \sum_{k'} \theta_{qt(i),k'}^j \cdot \mu_{sib_i(s),k'}^1 \quad (16)$$

### 2.3.2 Inference

In inference we again embed the epitome tree prior within a CRF of the form given in Equation 6 (see Figure 6). Given a test image,  $\mathbf{x}$ , we then use BP to estimate the maximum posterior marginals of the labels at each site,  $Pr(\mathbf{l}_s = k | \mathbf{x}), k = 1 \dots K$ . The upward message pass is initialized from the outputs of the unary classifier, by letting:

$$\mu_s^1(k) = \psi_{s,k}^\lambda \quad \forall s \in \mathbf{s}^L \quad (17)$$

Message passing proceeds as in the training E-step and the required marginals are evaluated by setting:

$$Pr(\mathbf{l}_s = k | \mathbf{x}) = \frac{\mu_s^1(k) \cdot \mu_s^2(k)}{\sum_{k'} \mu_s^1(k') \cdot \mu_s^2(k')} \quad \forall s \in \mathbf{s}^L \quad (18)$$

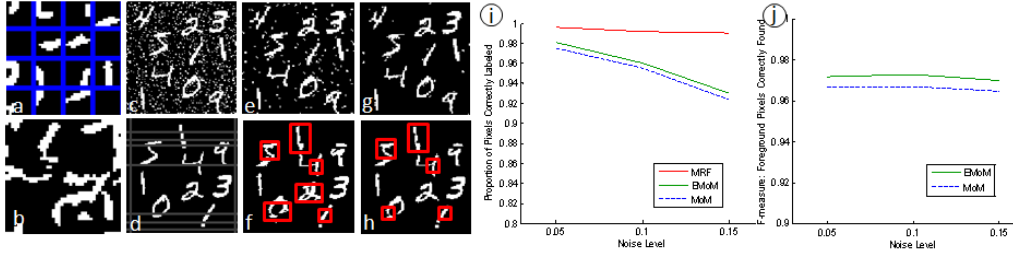


Figure 7. Handwritten digits results. a-b) MoM and EMoM models learned during training, showing  $\text{argmax}_k \{\theta_k\}$  for each mixture component/epitome location. c-d) Pixel noise and occluding bars test images (NB bars are shown in gray here, but during testing are set to black). e-f) EMoM results. g-h) ETSBN results. Notice the improvement from e-g, and f-h (while the occlusions in h are all spotted in f, extra noise is added in f; the points of interest are highlighted in boxes). i) Results across noise levels for pixel noise, and (j) for occluding bars: The EMoM out-performs the MoM throughout, but the MRF cannot denoise the occluded bars due to its locality assumption.

### 3. Denoising Handwritten Digits

Having defined several models, we first test them here in a binary context on denoising, and then in section 4 on multi-class segmentation. For denoising, we formed binary images of handwritten digits by copying thresholded images from the MNIST database (see [11]) to random positions in a  $100 \times 100$  image. We created 60 such images for training, and 40 for testing. For test data, we add two types of noise: pixel noise, in which a proportion of the pixels are reversed, and occluding bars, in which 2 pixel wide horizontal and vertical stripes are set to background pixels. We investigated flipping 5%, 10% and 15% of the pixels.

We let  $\mathbf{l}$  and  $\mathbf{x}$  take binary values representing the true image and the noisy observed image respectively. Since we know the noise model, we have an exact description of the unary term  $Pr(\mathbf{l}|\mathbf{x})$ . We learned priors using each of the methods described in section 2 combined with mixture and epitome models for comparison. A range of parameter settings were tried, and the best results for each model are reported. Patch sizes between  $2 \times 2$  and  $10 \times 10$  were used, epitome sizes between  $4 \times 4$  and  $50 \times 50$ , and values of  $\lambda$  between 0.01 and 5 (see Equation 6). Mixture models were learned with comparable numbers of parameters to ensure direct comparison. All models were tested at the 0.1 noise level for both kinds of noise, while the MoM and EMoM models were compared across all noise levels. In addition, a Markov Random Field model (MRF) with a single cost  $k$  for different labels at neighboring sites was tested across all conditions (setting  $k$  between 0.01 and 5).

#### 3.1. Results and Discussion

Figures 7a and b show the MoM mixture components and EMoM learned during training. We observe both to include characteristic line-segments from the digits data. Figures 7c and d give example test images with pixel noise and occluding bars added respectively. Figures 7e and f then show the EMoM results on these images, and 7g and h the ETSBN results (with highlights to show the alterations

	Pixel Noise (% correct)			Occluding Bars (F-values)		
	MoM	CMoM	TSBN	MoM	CMoM	TSBN
Mixture	95.5%	95.8%	97.0%	0.967	0.944	0.949
Epitome	96.8%	96.5%	<b>98.5%</b>	0.973	0.973	<b>0.977</b>

Table 1. Results for denoising handwritten digits across models. The epitomized models consistently outperform the mixture models, with the epitome tree (ETSBN) performing best overall.

made). In both cases, we can see the improvements afforded by the ETSBN over the EMoM. This is borne out quantitatively in Table 1 for the 0.1 noise level. For the pixel noise, we use the percentage of correctly labeled pixels as the measure. For the occluding bars we use an F-measure<sup>1</sup> as the small magnitudes of change prevent meaningful comparison of percentages. The results show two things. First, for each type of prior the epitomized version outperforms the mixture model. Secondly, the overlapping of patches in the CMoM/ECMoM and longer range connections in the TSBN/ETSBN tend to improve results over the MoM/EMoM, with the tree models performing best.

Figures 7i and j show the performance of the MoM and EMoM models across all noise levels, where we see the epitome model offers a consistent improvement over the mixture. The MRF model outperforms all models on the pixel noise (7i), but cannot perform the occluding bars task, as the 2-pixel gaps exceed its neighborhood model (it adds no foreground pixels for any setting of  $k$ , hence giving an undefined F-value). The mixture and epitome models however give a similar pattern of performance to the pixel noise, suggesting they have a larger notion of shape (7j).

### 4. Multi-class Scene Segmentation

We used the Corel and Sowerby databases (see [6, 15]) to test our models on multi-class segmentation. We created a

<sup>1</sup>F =  $(1 + \beta^2)RP / (\beta^2P + R)$ , where  $R = TP / (TP + FN)$  and  $P = TP / (TP + FP)$ . We set  $\beta = 0.1$  and normalize the true positives and false negatives by the number of pixels in the ground truth positive class, and the false positives by those in the ground truth negative class.

random train/test split of 60/40 images for Corel, and 70/34 for Sowerby. We used the training data to train both a unary classifier (Textonboost, see [15]) and priors separately. For speed, we chose to first over-segment all images, and learn priors across the ‘superpixels’ thus formed. We use the approach of Moore et al. [12] for this purpose, which returns a regular grid of  $20 \times 20$  superpixels. These are treated exactly as if they were pixels.

We used patch size  $5 \times 5$ , epitome size  $25 \times 25$ , and 25 mixture components across all models. Values of  $\lambda$  between 0.1 and 10 were tried, with results reported using the best values for each model (on Corel, TSBN  $\lambda = 0.5$ , other models  $\lambda = 2$ , on Sowerby, all models  $\lambda = 5$ ). For the CMoM and ECMoM, we rotated the training set by  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  and  $270^\circ$ , and learned a prior at each orientation. We perform inference by using each prior in turn, and using the maximum posterior marginal (MPM) labels at each site. For the TSBN, we learned overlapping trees with 4 levels at each corner of the image (with  $16 \times 16$  sites in each). Again, inference uses each prior in turn, and MPM labels are chosen in the overlapping regions. For the ETSBN we adopt a simple 2-level tree structure, with  $5 \times 5$  patches and a full epitome at level 2, and  $25 \times 25$  patches and a 25 component mixture model at level 1. We initialized the multinomial parameters in the MoM and EMoM models by adding random noise vectors to the overall frequencies of the labels across the training data. The CMoM, ECMoM and ETSBN models were initialized using the MoM and EMoM models already learned, and the TSBN model used sub-sampling to estimate transition frequencies between levels as in [4].

We clustered the training set ground-truth label maps by fitting a further mixture of multinomials directly to their histograms, and learned a separate prior per model for each cluster (see He et al. [7] for a similar approach). During testing, the initial classifier output was then assigned to a cluster by its histogram, and inference proceeded according to the relevant prior. We also imposed a ‘spatial constraint’ on all epitome models, where the prior  $\alpha$  is weighted by a 2-d Gaussian whose center varies with position in the image (encouraging, for instance, patches at the top of an image to be taken from the top of the epitome). Because of the reduction in patch choice which therefore results, we expand the epitome to include 3 ‘layers’, all of identical size and with the same spatial constraints. Patches can therefore come from the corresponding region in any one of these layers. We test the EMoM both with and without these constraints.

For inference, we used an iterative approach which alternates between using an epitome/mixture model at the superpixel level, and a ‘grab-cut’ refinement at the pixel level (based on per-image class color models and a context-sensitive edge term, see [14, 15]). We add a refinement to the grab-cut step, by weighting the unary cost for a given label by its Mahalanobis distance from the center of all su-

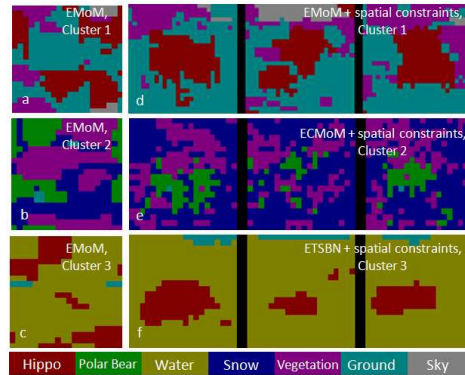


Figure 8. Example epitomes learned on the Corel dataset. The most probable class,  $\text{argmax}_k \{\theta_{tk}\}$ , is plotted at each epitome site  $t$ . a-c) show 3 basic EMoM models, each learned on a different subset of the data. The subsets were formed through clustering so that each includes a restricted number of object classes. d-f) show models for the same clusters where the epitome has been enlarged to 3 ‘layers’, and spatial constraints have been added. Now, patches in the center of an image are more likely to be taken from the center of any one of these layers etc (via a movable 2-d Gaussian weighting). These constraints are shown with: d) EMoM, e) Conditional Epitome (ECMoM) and f) Epitome Tree at level  $L$  (ETSBN). Note particularly the more disparate nature of e where only the conditional of the central pixel in each patch is modeled.

	Corel			Sowerby		
	MoM	CMoM	TSBN	MoM	CMoM	TSBN
Mixture	73.50%	73.65%	74.04%	83.23%	83.64%	83.58%
Epitome	74.07%	74.49%	<b>75.04%</b>	83.79%	83.98%	<b>84.13%</b>

Table 2. Performance of models on the Corel and Sowerby databases in terms of pixels correctly classified. The epitomized models outperform the mixture models on both databases, and the epitome tree (ETSBN) performs best overall.

perpixels with the same label. Experimentally, we found 3 iterations of superpixel/pixel inference to be sufficient for convergence. Finally, we also implement a smoothing CRF prior at the superpixel level to compare with our models. This has a single parameter  $k$  to penalize different neighboring labels (setting  $k = 1$  gave the best performance).

#### 4.1. Results and Discussion

Training times were 3, 15 and 25 minutes per EM iteration for the EMoM, ECMoM and ETSBN models respectively (we used 3 iterations), and 10s, 20s, 60s per image for testing (mixture model times were shorter, but comparable).

Figure 8 shows examples of some of the epitomes learnt from the Corel dataset. Epitomes for different clusters, with and without spatial constraints, and EMoM ECMoM and ETSBN models are shown, giving a qualitative impression of the different types of information captured. Figure 9 then compares the results for EMoM, ECMoM and ETSBN

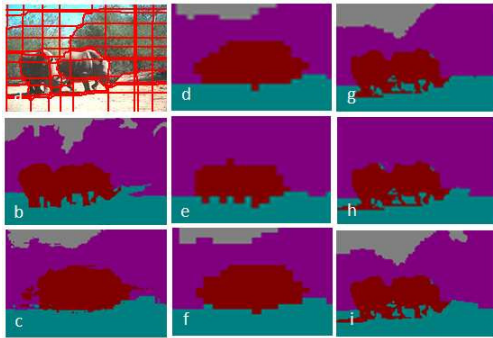


Figure 9. Comparing the segmentation of a single image from Corel by the three epitome models. a) The original image in superpixels. b) The ground truth. c) The initial per-pixel classification result. d-f) The result of inference at the superpixel level using the EMoM, ECMoM and ETSBN models respectively. g-i) The final results, enhancing d-f iteratively in combination with grabcut. Notice how the ECMoM smooths out the sky, but gets all the trees, while the ETSBN finds the extra portion of sky.

models on a single test image, showing the ability of the ECMoM and ETSBN to capture larger-scale dependencies.

Table 2 compares the performance of all models on both databases. As in section 3, we see that the epitomized models offer a performance gain over the mixture models. We also see that the CMoM/TSBN and ECMoM/ETSBN perform better than the MoM and EMoM respectively, reflecting the fact that they can take advantage of larger-scale dependencies. The ETSBN gives the best performance here, as it did on the handwritten digits, possibly because, unlike the ECMoM, it more effectively models minority classes (see Figure 9h and i). The EMoM that was tested without the spatial constraints performed at 73.85% on Corel and 83.71% on Sowerby, giving lower performance than the results with spatial constraints (see Table 2) as expected. All models outperformed the superpixel smoothing CRF, which gave 72.99% on Corel and 83.13% on Sowerby, and the initial classifier gave respectively 69.73% and 79.54%.

In comparing with the state of the art, we note that since training and test sets are not fixed on these databases, differences in results should be treated with some caution. Our result of 75.04% on Corel compares favorably with Shotton et al. [15], who report 74.6% for their full model. Although they report 88.6% for their full model on Sowerby, initial classification is at 85.6%, so the gain of 4.59% produced by our best prior is slightly better. He et al. [6] report 80.0% on Corel and 89.5% on Sowerby, while Feng et al. report 90.68% on Sowerby. We point out that, while 5-6% higher than us, He et al's prior is undirected and must use sampling in both training and inference. Further, Feng et al's approach combines a TSBN prior with a neural-network classifier, indicating that the ETSBN prior might perform around this level given the same initial conditions.

## 5. Summary

We have developed three models, the Epitomized Mixture of Multinomials (EMoM), Conditional Epitome (ECMoM) and Epitome Tree (ETSBN), which demonstrate the epitome can be used as an effective prior for image labeling. The ETSBN has been shown to perform best in a range of circumstances, and we put forward this model in particular as an alternative to undirected models, with reduced computational requirements and similar representational ability. Future directions include applications to other tasks, epitomization of other models and incorporating features such as rotation and scale invariance into the current models.

**Acknowledgments.** Financial support from EPSRC grant EP/E013309/1 is gratefully acknowledged.

## References

- [1] C. Bouman and M. Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Trans. Image Processing*, 3(2):162-177, 1994.
- [2] A. Dempster, N. Laird and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1-38, 1977.
- [3] J. Domke, A. Karapurkar and Y. Aloimonos. Who killed the directed model? *CVPR*, 2008.
- [4] X. Feng, C.K.I. Williams, and S.N. Felderhof. Combining Belief Networks and Neural Networks for Scene Segmentation. *PAMI*, 24(4):467-483, 2002.
- [5] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *PAMI*, 6, 1984.
- [6] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale Conditional Random Fields for Image Labeling. *CVPR*, 2, 2004.
- [7] X. He, R.S. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. *Lecture Notes in Computer Science*, 3951:338-351, 2006.
- [8] N. Jovic, B. Frey, and A. Kannan. Epitomic analysis of appearance and shape. *ICCV*, 2003.
- [9] P. Kohli, M.P. Kumar, and P.H.S. Torr. P 3 & Beyond: Solving Energies with Higher Order Cliques. *CVPR*, 2007.
- [10] S. Kumar and M. Hebert. Discriminative random fields: a discriminative framework for contextual interaction in classification. *ICCV*, 2:1150-1157, 2003.
- [11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278-2324, 1998.
- [12] A.P. Moore, S.J.D. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel Lattices. *CVPR*, 2008.
- [13] K. Ni, A. Kannan, A. Criminisi, and J. Winn. Epitomic Location Recognition *CVPR*, 2008.
- [14] C. Rother, V. Kolmogorov and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts *ACM Trans. on Graphics*, 23(3):309-314, 2004.
- [15] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. *ECCV*, 1:1-15, 2006.
- [16] C. Sutton and A. McCallum. Piecewise training of undirected models. *Proc. of conference on uncertainty in AI*, 2005.
- [17] Z. Tu, X. Chen, A.L. Yuille and S.C. Zhu. Image Parsing: Unifying Segmentation, Detection, and Recognition. *IJCV*, 63(2):113-140, 2005.