Learning Class-Specific Affinities for Image Labelling

Dhruv Batra[1] Rahul Sukthankar[2,1] Tsuhan Chen[1]

www.ece.cmu.edu/~dbatra rahuls@cs.cmu.edu tsuhan@cmu.edu

[1]Carnegie Mellon University [2]Intel Research Pittsburgh

Abstract

*Spectral clustering and eigenvector-based methods have become increasingly popular in segmentation and recognition. Although the choice of the pairwise similarity metric (or affinities) greatly influences the quality of the results, this choice is typically specified outside the learning framework. In this paper, we present an algorithm to learn class-specific similarity functions. Mapping our problem in a Conditional Random Fields (CRF) framework enables us to pose the task of learning affinities as parameter learning in undirected graphical models. There are two significant advances over previous work. First, we learn the affinity between a pair of data-points as a function of a pairwise feature and (in contrast with previous approaches) the classes to which these two data-points were mapped, allowing us to work with a richer class of affinities. Second, our formulation provides a principled probabilistic interpretation for learning **all** of the parameters that define these affinities. Using ground truth segmentations and labellings for training, we learn the parameters with the greatest discriminative power (in an MLE sense) on the training data. We demonstrate the power of this learning algorithm in the setting of joint segmentation and recognition of object classes. Specifically, even with very simple appearance features, the proposed method achieves state-of-the-art performance on standard datasets.*

1. Introduction

Spectral clustering and eigenvector-based methods have become the focus of significant recent research in computer vision, particularly in clustering and image segmentation [2, 4, 18, 21, 26, 29]. An important benefit of these methods is that they offer good, computationally-efficient approximations to combinatorial problems. The typical approach can be summarized as follows. First, a weighted graph is constructed, where each node corresponds to either a data element (in clustering) or a pixel (in segmentation); and the undirected edge weight between two nodes is defined by a pairwise similarity metric between the nodes. For example, Felzenszwalb and Huttenlocher [6] employ the Euclidean (L_2) distance between the colors of connected pixels as a measure of their dissimilarity.[1] On the other hand, in their work on normalized cuts (Ncut), Shi and Malik [26] define the edge affinity using an exponential kernel of the distance between two nodes in feature space. As has been observed by several authors [2, 4, 25], the quality of results achieved by such methods is strongly dependent on the choice of the affinity function. Thus, it a natural to seek principled ways for selecting these affinities.

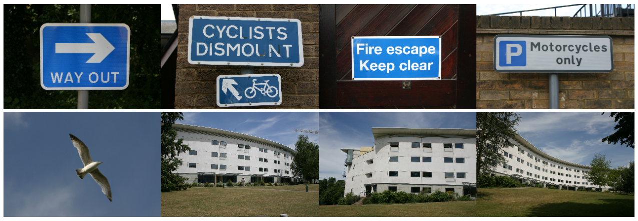


Figure 1: The need for class-specific affinities [best viewed in colour]: the affinity between "blue" and "white" regions should be high for images in the top row (those colors occur together in street signs); the same affinities should be low for images in the bottom row to enable white buildings and birds to be segmented from blue sky.

For some problems, such as unsupervised, task-independent bottom-up segmentation, it may be impossible to propose an optimal affinity function for all possible images; indeed the current trend in computer vision is to treat this type of segmentation simply as a pre-processing step that generates over-segmentations [5, 9]. However, in cases where segmentation is more closely tied to a specific task (such as object category recognition), could we exploit the availability of ground-truth segmentations by placing segmentation within a supervised learning framework? Meila and Shi [18] explore this problem in the random walk interpretation of Ncuts, by minimizing the KL-divergence be-

[1] Dissimilarity and similarity are both employed in related work. This paper consistently defines affinities as similarity.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

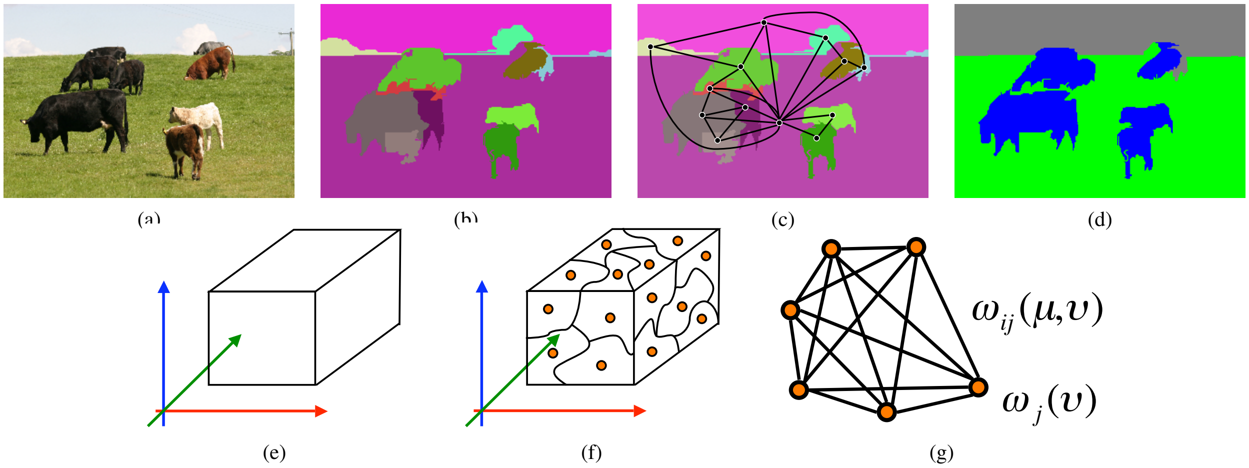


Figure 2: Overview of our approach [best viewed in colour]: (a) an input image; (b) superpixels extracted from this image; (c) region graph G constructed over those superpixels; (d) optimal labelling of the image; (e) visualization of raw feature space $\mathcal{F}$; (f) visual words extracted in this feature space; (g) shows the complete graph G_f over these visual words, along with weights on nodes and edges. Unlike previous work, we employ *class-specific* edge weights.

tween the transition probabilities derived from the affinity matrix and ground-truth segmentation. Bach and Jordan [2] define a cost function measuring error between the Ncut eigenvector and the ground-truth partition. They use a differentiable approximation of the eigenvector to derive the affinity matrix that is optimal under this cost function. Cour *et al.* [4] derive an analytic form for the derivative of the Ncut eigenvector and select the affinity matrix that minimizes the L_2 distance between the Ncut eigenvector and the target partition. Shental *et al.* [25], reformulate the typical cuts criterion [3, 7] as inference in an undirected graphical model, and learn the affinities as the "best" (in an MLE sense) linear combination of features. This last work is the one that is most closely related to our proposed method.

Fundamentally, all of these methods attempt to learn a mapping from the features derived at a pair of data points (or pixels) to *that* affinity that best mimics the segmentation provided in the training set. However, this is inherently an ill-posed task. Consider, for example, the images shown in Figure 1, where the ground-truth segmentation separates the foreground object (sign, bird or building) from the background. Suppose that our (weak) features are colour, and that we would like to learn the affinity between "blue" and "white". The images in the top row would suggest that we should like to keep the affinity between "blue" and "white" high in order to penalize any cuts that separate the two. On the other hand, the images in bottom row suggest that the affinity between "blue" and "white" should be low in order to encourage cuts that separate the two. We claim that these conflicting notions can both be simultaneously incorporated by learning *class-specific* affinities, *i.e.*, affinities that are not just a function of the features measured at the data-points, but also the classes to which these two data-points were mapped. In our framework, we no longer pose questions like "What is the affinity between 'blue' and 'white'?", but rather "What is the affinity between 'blue' and 'white' when 'blue' corresponds to sky and 'white' to building?".

This paper makes two significant contributions. First, by learning class-specific affinities we employ a richer family of dependencies and inter-relations between parts. Second, formulating our problem in the framework of Conditional Random Fields (CRFs) enables us to pose the task of learning affinities as parameter learning in undirected graphical models (in a manner similar to [25]). Thus we provide a probabilistic interpretation for learning affinities, and learn *all* parameters defining these affinities from data. We demonstrate this learning framework on the task of joint segmentation and recognition of multiple object classes in images, where the goal is to output a K-way partition of the image, and to assign to each partition a class label.

2. Proposed Approach

Figure 2 shows an overview of our approach and Table 1 defines the notation used throughout this paper.

2.1. Superpixels

In a manner similar to previous efforts [5, 9], we first over-segment the image and treat the resultant superpixels (rather than the pixels) as our elementary units. This ensures a locally-smooth labelling, and also speeds up the algorithm.

Table 1: Summary of notations used in the paper

G_f	complete graph (with loops) on visual words
G	neighbourhood graph (without loops) on superpixels
W	number of visual words, *i.e.* $\lvert V(G_f)\rvert$
N	number of object classes
S	number of superpixels in an image, *i.e.* $\lvert V(G)\rvert$
$[N]$	$\{1, 2 \ldots N\}$
$\mathcal{F}$	raw feature space
f_k	raw feature vector extracted from superpixel k
μ, υ	visual words
$\omega_j(\upsilon)$	reward for labelling visual word υ as class j
$\omega_{ij}(\mu, \upsilon)$	affinity between visual words μ (labelled i) and υ (labelled j)
$\widetilde{\boldsymbol{\omega}}_j$	$[\, \omega_j(\upsilon_1)\ \ldots\ \omega_j(\upsilon_W)\,]^T_{(W\times 1)}$
$\widetilde{\boldsymbol{\omega}}_{ij}$	$[\, \omega_{ij}(\mu_1, \upsilon_1)\ \ldots\ \omega_{ij}(\mu_W, \upsilon_W)\,]^T_{(W^2\times 1)}$
$\pi_k(\upsilon)$	posterior probability of the visual word υ given feature vector f_k
$\widetilde{\boldsymbol{\pi}}_k$	$[\, \pi_k(\upsilon_1)\ \ldots\ \pi_k(\upsilon_W)\,]^T_{(W\times 1)}$
$\widetilde{\boldsymbol{\pi}}_{kl}$	$[\, \pi_k(\mu_1)\, \pi_l(\upsilon_1)\ \ldots\ \pi_k(\mu_W)\, \pi_l(\upsilon_W)\,]^T_{(W^2\times 1)}$
$\mathrm{E}_{\pi_k}[\,\cdot\,]$	expected value of $(\cdot)$ under the distribution π_k
$\boldsymbol{X}$	$\{X_1, X_2 \ldots X_S\}$, image represented as a collection of superpixels
$\boldsymbol{Y}$	$\{Y_k : k \in [S], Y_k \in [N]\}$, a valid image labelling
D	$\left\{\left(\boldsymbol{Y}^{(1)}, \boldsymbol{X}^{(1)}\right), \ldots, \left(\boldsymbol{Y}^{(m)}, \boldsymbol{X}^{(m)}\right)\right\}$, dataset of image, ground-truth segmentation pairs
m	number of training images

2.2. Non-parametric Class-Specific Affinities

Our next step is to compute features for each superpixel. These can include intensity, colour, texture or gradient cues. We discuss our choice of features in Section 3, but since our approach is agnostic to the exact form of the chosen features, assume for the following discussion that there are some image features[2] (in a feature space $\mathcal{F}$) that can be extracted over these superpixels. Following the popular bag-of-words model [22], we discretize/quantize this feature space to get "visual words". We construct a complete graph (G_f) with the visual words as the nodes. Now, we define a family of weights on the node and edges parametrized by the class labels, i.e.,

$$\{\omega_j(\upsilon) : \upsilon \in V(G_f), j \in [N]\} \text{ and,} \tag{1}$$

$$\{\omega_{ij}(\mu, \upsilon) : \mu, \upsilon \in V(G_f), i, j \in [N]\}, \tag{2}$$

where we define $[N] \overset{\text{def}}{=} \{1, 2, \ldots, N\}$ to be the set of class labels, $W \overset{\text{def}}{=} \lvert V(G_f)\rvert$ to be the number of visual words . Intuitively, we can think of $\omega_j(\upsilon)$ as the reward for labelling visual word υ class label j, and $\omega_{ij}(\mu, \upsilon)$ the affinity between visual words μ and υ when they were labelled as i and j respectively. This is the quantity discussed in the example in Figure 1.

2.3. Visual Word Posteriors

Instead of working with the raw features extracted over the superpixels, we compute the posterior visual word distributions, which can be thought of as a soft assignment of our feature vector to the quantized cluster centers. Formally, if f_k is the feature vector extracted at superpixel k,

$$\pi_k(\upsilon) = \Pr\left(\upsilon \mid f_k\right) \tag{3}$$

$$\propto \Pr\left(f_k \mid \upsilon\right) \Pr\left(\upsilon\right), \tag{4}$$

where the first term is the likelihood of a feature vector given a visual word, and is modelled using an exponential kernel of euclidean distance:

$$\Pr\left(f_k \mid \upsilon\right) \propto e^{-d(f_k, \upsilon)}. \tag{5}$$

It should be noted that in the above relation we have overloaded the term υ to be both the index of the visual word, and the corresponding feature vector in $\mathcal{F}$. The second term in Equation 4, *i.e.*, the marginal over the visual words could be assumed to be uniform. However, since this quantization is usually the result of a clustering process, we model this marginal by the observed "popularity" of visual words at the end of the clustering process:

$$\Pr\left(\upsilon\right) = \frac{\#\,\text{members in cluster } \upsilon}{\#\,\text{data points}}. \tag{6}$$

2.4. Expected Affinities

Now that we have extracted visual word posteriors, we can talk about *expected* costs and affinities under these posteriors, *i.e.*, we define

$$\widetilde{\boldsymbol{\omega}}_j \overset{\text{def}}{=} [\, \omega_j(\upsilon_1)\ \ldots\ \omega_j(\upsilon_W)\,]^T \text{ and,} \tag{7}$$

$$\widetilde{\boldsymbol{\omega}}_{ij} \overset{\text{def}}{=} [\, \omega_{ij}(\mu_1, \upsilon_1)\ \ldots\ \omega_{ij}(\mu_W, \upsilon_W)\,]^T, \tag{8}$$

as the vectors holding affinities, and

$$\widetilde{\boldsymbol{\pi}}_k \overset{\text{def}}{=} [\, \pi_k(\upsilon_1)\ \ldots\ \pi_k(\upsilon_W)\,]^T \text{ and,} \tag{9}$$

$$\widetilde{\boldsymbol{\pi}}_{kl} \overset{\text{def}}{=} [\, \pi_k(\mu_1)\, \pi_l(\upsilon_1)\ \ldots\ \pi_k(\mu_W)\, \pi_l(\upsilon_W)\,]^T, \tag{10}$$

as the vectors holding marginal visual word posteriors for a superpixel k, and the joint visual word posterior for a pair of superpixels (k and l). Note that this joint distribution can be extracted from our marginal distributions using an independence assumption. Now, under these two distributions the expected affinities can be written as:

$$\mathrm{E}_{\pi_k}\left[\,\omega_j\,\right] = \widetilde{\boldsymbol{\omega}}_j^T\, \widetilde{\boldsymbol{\pi}}_k \text{ and,} \tag{11}$$

$$\mathrm{E}_{\pi_{kl}}\left[\,\omega_{ij}\,\right] = \widetilde{\boldsymbol{\omega}}_{ij}^T\, \widetilde{\boldsymbol{\pi}}_{kl}. \tag{12}$$

[2] For clarity of notation, we will refer to these features as "raw" features.

Intuitively, Equation 11 can be interpreted as the expected reward for labelling superpixel k with class label j, while Equation 12 gives the expected affinity between superpixels k and l, when they are labelled as classes i and j, respectively.

2.5. Model

Our model consists of a CRF [13, 14] defined over a planar graph (G) whose nodes are the superpixel labels and adjacent superpixels in the image correspond to an edge in this graph. Formally, if our image is represented as a collection of superpixels $\boldsymbol{X} = \{X_1, X_2 \dots X_S\}$ (where $S \overset{\text{def}}{=} |V(G)|$, is the number of superpixels), then every valid image labelling is given by a collection of random variables $\boldsymbol{Y} = \{Y_k : k \in [S], Y_k \in [N]\}$. We define a distribution on the space of all such image labellings which defines our model:

$$\Pr(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{\theta}) =$$

$$\frac{1}{Z} \exp\left(\sum_{k \in V(G)} \mathrm{E}_{\pi_k}\left[\omega_{Y_k}\right] + \sum_{(k,l) \in E(G)} \mathrm{E}_{\pi_{kl}}\left[\omega_{Y_k Y_l}\right] \right) \tag{13a}$$

$$= \frac{1}{Z} \exp\left(\sum_{k \in V(G)} \widetilde{\boldsymbol{\omega}}_{Y_k}^T \widetilde{\boldsymbol{\pi}}_k + \sum_{(k,l) \in E(G)} \widetilde{\boldsymbol{\omega}}_{Y_k Y_l}^T \widetilde{\boldsymbol{\pi}}_{kl} \right), \tag{13b}$$

where Z is the partition function and $\boldsymbol{\theta} = \{\widetilde{\boldsymbol{\omega}}_j, \widetilde{\boldsymbol{\omega}}_{ij} : i, j \in [N]\}$ is the set of parameters governing this distribution.

2.6. Parameter Learning and Inference

Our model (Equation 13b) tells us that every valid configuration of affinities parameterizes a distribution over image labellings. We have thus formulated the problem of learning affinities as parameter learning in this undirected model. We learn these parameters from a dataset of segmented and labelled images ($D = \{(\boldsymbol{Y}^{(1)}, \boldsymbol{X}^{(1)}), \dots, (\boldsymbol{Y}^{(m)}, \boldsymbol{X}^{(m)})\}$) by maximizing the conditional log-likelihood of this training dataset, *i.e.*,

$$\widehat{\boldsymbol{\theta}}_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} L(D \mid \boldsymbol{\theta}) \tag{14}$$

$$= \arg\max_{\boldsymbol{\theta}} \log \Pr\left((\boldsymbol{Y}^{(d)})_{d \in D} \mid (\boldsymbol{X}^{(d)})_{d \in D}, \boldsymbol{\theta} \right). \tag{15}$$

The motivation behind this criterion is that we want to choose the parameters that lead to a distribution under which the ground-truth segmentations become most likely, and are thus the most discriminative parameters for this dataset. The properties of CRF log-likelihoods have been well studied and the reader is referred to [1, 12, 14–16] for detailed discussions and background. We perform this maximization by gradient ascent, and in general, the derivative of the conditional log-likelihood with respect to a parameter simplifies to a difference between the expected and observed feature responses. Analytically, we can express the gradient as:

$$\frac{\partial L(D \mid \boldsymbol{\theta})}{\partial \widetilde{\boldsymbol{\omega}}_j} = \sum_{d \in D} \left(\frac{\partial \log \Pr\left(\boldsymbol{Y}^{(d)} \mid \boldsymbol{X}^{(d)}, \boldsymbol{\theta}\right)}{\partial \widetilde{\boldsymbol{\omega}}_j} \right) \tag{16a}$$

$$= \sum_{d \in D} \left(- \sum_{Y_k^{(d)} = j} \widetilde{\boldsymbol{\pi}}_k^{(d)} - \frac{1}{Z^{(d)}} \frac{\partial Z^{(d)}}{\partial \widetilde{\boldsymbol{\omega}}_j} \right) \tag{16b}$$

$$= \sum_{d \in D} \left(- \sum_{Y_k^{(d)} = j} \widetilde{\boldsymbol{\pi}}_k^{(d)} + \sum_{k \in V(G)} \Pr(Y_k^{(d)} = j)\, \widetilde{\boldsymbol{\pi}}_k^{(d)} \right), \tag{16c}$$

where the second term in Equation 16c requires the marginal class label distribution of a superpixel. A similar update rule can be derived for the pairwise weights ($\widetilde{\boldsymbol{\omega}}_{ij}$), which will involve class label distributions for pairs of superpixels. Exact computation of these marginal distributions is in general intractable, and thus approximations like Monte Carlo sampling are commonly employed [8]. In a manner similar to [25], we use Loopy Belief Propagation (LBP) [19] to approximate these marginal beliefs. We can see that gradient ascent will converge (*i.e.*, the gradient will become zero) when our model completely "believes" in the training data. It should be noted that our update rule (Equation 16c) requires us to perform inference on *all* the training images (*e.g.*, 276 in the MSRC dataset) at *each* step of gradient ascent. Thus, in practice, we follow a sequential update (rather than this batch update) rule, where the parameters are updated (in the direction of the gradient) after inference on each image.

At test time, we perform Maximum Posterior Marginal (MPM) inference to estimate the most probable segmentation and labelling of an image.

3. Experiments

We evaluated our algorithm on three standard image collections: the 21-class MSRC [27], the 7-class Sowerby [9] and the 7-class Corel [9] datasets. Our results, summarized below, confirm the benefits of employing a class-specific affinity formulation.

3.1. 21-Class MSRC

The MSRC dataset consists of 591 images (mostly 213 × 320) containing 21 object classes (ignoring 'void', 'horse' and 'mountain'). A number of recent works have reported encouraging results on this dataset: Shotton *et al*. (TextonBoost) [27] introduced novel shape-texture features in a boosted framework, and used a CRF to integrate these with other cues; Verbeek and Triggs [28] explored combining spatial field models (like MRFs) with aspect-based models (like PLSA, LDA); Yang *et al.* [30] combined texture, keypoint spatial co-occurrence and global shape into a meanshift framework to perform multi-class segmentation of images. In order to enable direct comparisons against TextonBoost, we duplicate Shotton *et al.*'s experimental methodology [27] and employ a random split of 45% for training, 10% for validation, and 45% for testing, while maintaining a similar distribution of classes.

Superpixels. For superpixel generation, we experimented with both the Felzenszwalb-Huttenlocher (FH) [6] and the Ncut [26] code, and ultimately chose FH purely for its computational efficiency. No effort was made to identify optimal FH parameters; all of the experiments described here used sigma = 1, k = 150, min = 400 for superpixel generation. The number of superpixels per image varied from 4 (*e.g.*, for images containing mostly grass) to 58 (for cluttered indoor environments) with the average being 27.

Weak Raw Features. While our proposed framework works with arbitrary features, we present experimental results on a set of commonly-used weak colour and texture features to highlight the power of our learning algorithm. Specifically, we adopt the the same colour features as Hoiem *et al.* [11], and filterbank responses proposed by Malik *et al.* [17]. These raw features were clustered to form a visual word dictionary using a publicly-available efficient C implementation of k-means/x-means by Pelleg and Moore [23]. The number of clusters was automatically chosen using the x-means algorithm [24] maximizing a Bayesian Information Criterion (BIC) within a range (10–60).

Baseline 1: Node features only. In order to establish a baseline, and to get an estimate of the "power" in our features, we classified superpixels based on these features alone. To this effect, we trained 21 one-vs-rest binary logistic classifiers [20] to model the probability of a class given the feature vector. Ground-truth label for a superpixel was taken to be the most frequent label among the pixels it contained. At test time, we assigned to each feature vector the label of the most confident of the 21 models. These superpixel level classifications were then used to provide pixel level labels by assigning the same label to all pixels within a superpixel. This method achieved an overall pixel accuracy of 59.3%. As a comparison, using their stronger node features alone (without the CRF) Shotton *et al.* (TextonBoost) [27] were able to achieve 69.6%.

Baseline 2: Class-agnostic affinities. For our second baseline, we explore the use of affinities that unlike our proposed class-specific affinities, are not explicit functions of object classes; we refer to these as "class-agnostic" affinities. Such affinities are common in the computer vision literature, *e.g.*, Kumar and Hebert [13] use a logistic unit to model $\Pr\left(Y_k = Y_l \mid \widetilde{\boldsymbol{\pi}}_{kl}\right)$, *i.e.*, the probability of two nodes having the same class given their pairwise feature vector. In our model, this would correspond to replacing our class-parametrized edge affinities (Equation 2) by a weaker family:

$$\begin{aligned}\{\omega_{ij}(\mu,\upsilon) : \mu,\upsilon \in V(G_f), i,j \in [N]\} \longrightarrow \\ \left\{\omega_{\delta(i,j)}(\mu,\upsilon) : \mu,\upsilon \in V(G_f), i,j \in [N]\right\},\end{aligned} \tag{17}$$

where $\delta(i,j) \overset{\text{def}}{=} I(i=j)$, is defined as the indicator function for the case $(i=j)$. Corresponding to this family of weights our conditional distribution over image labellings becomes:

$$\Pr(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{\theta}) = \\ \frac{1}{Z}\ \exp\left(\sum_{k\in V(G)} \widetilde{\boldsymbol{\omega}}^T_{Y_k}\, \widetilde{\boldsymbol{\pi}}_k + \sum_{(k,l)\in E(G)} \widetilde{\boldsymbol{\omega}}^T_{\delta(Y_k,\,Y_l)}\, \widetilde{\boldsymbol{\pi}}_{kl}\right). \tag{18}$$

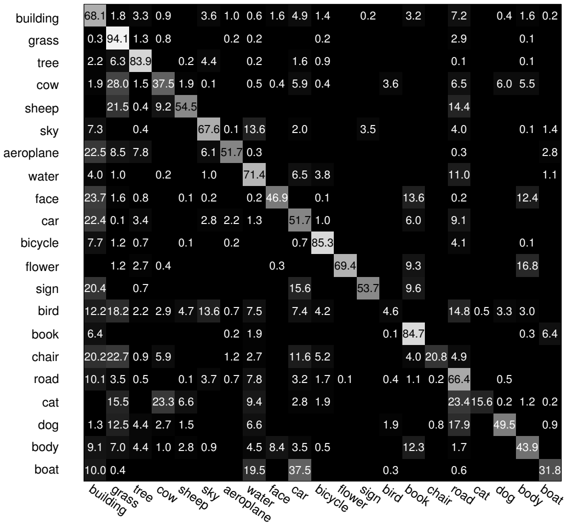
In the context of our example from Figure 1, intuitively this family of pairwise affinities tries to characterize whether a pair of visual words (say "blue" and "white") should belong to the same or different classes. In a manner similar to our discussion in section 2.6, this model was also trained by maximizing the conditional log-likelihood though gradient ascent. We achieved an overall pixel accuracy of 60.9$\%$ with this model.

Comparison with these baselines. Table 2 compares the performance of class-specific affinities (CSA) with these two baselines. We note that our node features are indeed weak, and perform poorly across most classes except the ones easily distinguished by their colour or texture alone (e.g sky, grass, road). Using the CRF framework over these weak node features with agnostic affinities helps a little, but we achieve a significant improvement by using class-specific affinities, thus re-affirming our intuition about their usefulness. An interesting observation is that there exists a certain coupling between classes or correlation in accuracy jumps. This suggests classes that frequently co-occur in the same image (*e.g.*, boat–water, face–body) boost each other's performance though pairwise interactions.

Comparison with existing methods. Table 3 presents a comparison of our method with three previous works: Shotton *et al.* [27] (TextonBoost), Verbeek and Triggs [28] (Verbeek07), and Yang *et al.* [30] (Mspatch). The comparison with TextonBoost is particularly interesting because

	Building	Grass	Tree	Cow	Sheep	Sky	Aeroplane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat	Average
Node Alone	62	95	67	9	19	95	24	43	23	38	63	61	33	3	53	1	75	31	17	19	2	40
Class-agnostic	49	90	67	16	48	88	58	51	7	48	77	90	41	15	40	17	60	44	33	27	0	46
Class-specific	68	94	84	37	55	68	52	71	47	52	85	69	54	5	85	21	66	16	49	44	32	55

Table 2: Classification accuracies for the 21 classes in the MSRC dataset achieved by node feature alone, class-agnostic affinities, and class-specific affinities



	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat
building	68.1	1.8	3.3	0.9		3.6	1.0	0.6	1.6	4.9	1.4		0.2		3.2		7.2		0.4	1.6	0.2
grass	0.3	94.1	1.3	0.8			0.2	0.2			0.2						2.9			0.1	
tree	2.2	6.3	83.9		0.2	4.4		0.2		1.6	0.9						0.1			0.1	
cow	1.9	28.0	1.5	37.5	1.9	0.1		0.5	0.4	5.9	0.4			3.6			6.5		6.0	5.5	
sheep		21.5	0.4	9.2	54.5												14.4				
sky	7.3		0.4			67.6	0.1	13.6		2.0			3.5				4.0			0.1	1.4
aeroplane	22.5	8.5	7.8			6.1	51.7	0.3									0.3				2.8
water	4.0	1.0		0.2		1.0		71.4		6.5	3.8						11.0				1.1
face	23.7	1.6	0.8		0.1	0.2		0.2	46.9		0.1				13.6		0.2			12.4	
car	22.4	0.1	3.4			2.8	2.2	1.3		51.7	1.0				6.0		9.1				
bicycle	7.7	1.2	0.7		0.1		0.2			0.7	85.3						4.1			0.1	
flower		1.2	2.7	0.4					0.3			69.4			9.3					16.8	
sign	20.4		0.7							15.6			53.7		9.6						
bird	12.2	18.2	2.2	2.9	4.7	13.6	0.7	7.5		7.4	4.2			4.6			14.8	0.5	3.3	3.0	
book	6.4						0.2	1.9						0.1	84.7					0.3	6.4
chair	20.2	22.7	0.9	5.9			1.2	2.7		11.6	5.2				4.0	20.8	4.9				
road	10.1	3.5	0.5		0.1	3.7	0.7	7.8		3.2	1.7	0.1		0.4	1.1	0.2	66.4		0.5		
cat		15.5		23.3	6.6			9.4		2.8	1.9						23.4	15.6	0.2	1.2	0.2
dog	1.3	12.5	4.4	2.7	1.5			6.6						1.9		0.8	17.9		49.5		0.9
body	9.1	7.0	4.4	1.0	2.8	0.9		4.5	8.4	3.5	0.5				12.3		1.7			43.9	
boat	10.0	0.4						19.5		37.5				0.3			0.6				31.8

Figure 3: Confusion matrix for 21 class MSRC using class-specific affinities.

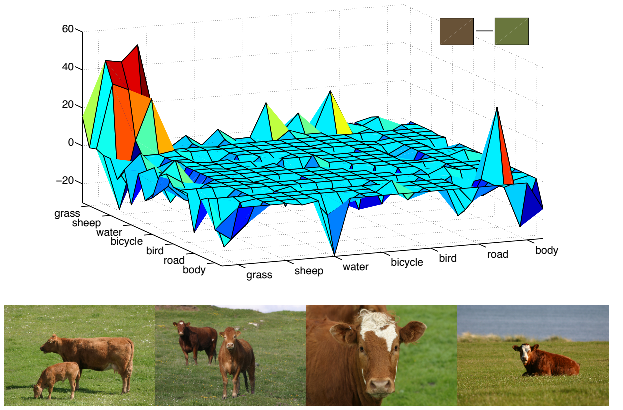


Figure 4: Class-Specific Affinities [best viewed in colour]: plot (top) shows the learnt affinities between a pair of visual words (brown–green) as a function of classes; bottom rows shows images containing the pair of classes (cow–grass) under which these visual words had the highest affinity between them.

the two approaches (Textonboost *vs*. CSA) focus on two different aspects of the overall recognition system (node features *vs*. learning framework). They use an extensive boosting framework (involving 5000 rounds and 42 hours of training) to find powerful node features, which when used alone achieve a competitive classification accuracy (69.9%). However, their learning framework posts only a minor boost ($\sim$ 3%) in performance. On the other hand, while our node features are weak (59.3%), the boost achieved from our learning framework over the node features is significantly higher ($\sim$ 10%). Clearly this shows that our algorithm's performance should primarily be attributed to our learning framework rather than the choice of features.

Analyzing learnt class-specific affinities. As discussed in Section 2.2, our algorithm learns a family of affinities parametrized by a pair of classes. Figure 4 shows this family of learnt parameters for one particular pair of code words (whose colours roughly correspond to brown and green). The figure shows that there is a significant variation in the edge affinity parameters for different classes. The maximum occurs at the joint assignment of "cow-grass" and example images where this pair of code words was assigned to these classes are shown. Class-agnostic affinities are restricted to representing this set of parameters with only a single number. This prevents such techniques from adequately capturing the rich class-specific interactions between features in real-world images.

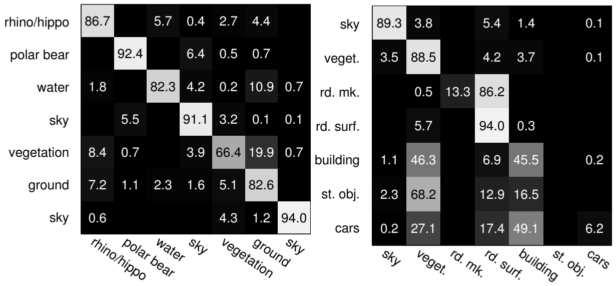
3.2. Sowerby and Corel

The Sowerby dataset introduced by He *et al.* [8] consists of 104 images (64 × 96) containing 7 classes, and the Corel (subset) dataset[3] contains 100 images (120 × 180) also containing 7 classes. We compare our algorithm to TextonBoost and He *et al.* [8] (mCRF) who propose a multiscale CRF defined over the pixels, in a manner similar to

[3] Referred to as "Corel A" by He *et al.* [9].

	21-MSRC				Corel			Sowerby		
	CSA (Us)	TextonBoost	Verbeek07	Mspatch	CSA (Us)	TextonBoost	mCRF	CSA (Us)	TextonBoost	mCRF
Node Alone	59.3	69.6	-	-	63.2	68.4	66.9	84.6	85.6	82.4
With Learning	69.5	72.2	73.5	75.1	82.8	74.6	80.0	87.8	88.6	89.5
Boost	**10.2**	2.8	-	-	**19.6**	6.2	13.1	3.2	3.0	**7.1**

Table 3: Comparison of our method (CSA) with other works; first row holds accuracies achieved by node features alone; second row shows accuracies by using the overall learning framework; and the third row shows the gain.



	rhino/hippo	polar bear	water	sky	vegetation	ground	sky
rhino/hippo	86.7		5.7	0.4	2.7	4.4	
polar bear		92.4		6.4	0.5	0.7	
water	1.8		82.3	4.2	0.2	10.9	0.7
sky		5.5		91.1	3.2	0.1	0.1
vegetation	8.4	0.7		3.9	66.4	19.9	0.7
ground	7.2	1.1	2.3	1.6	5.1	82.6	
sky	0.6				4.3	1.2	94.0

(a) Corel

	sky	veget.	rd. mk.	rd. surf.	building	st. obj.	cars
sky	89.3	3.8		5.4	1.4		0.1
veget.	3.5	88.5		4.2	3.7		0.1
rd. mk.		0.5	13.3	86.2			
rd. surf.		5.7		94.0	0.3		
building	1.1	46.3		6.9	45.5		0.2
st. obj.	2.3	68.2		12.9	16.5		
cars	0.2	27.1		17.4	49.1		6.2

(b) Sowerby

Figure 5: Confusion matrices for the Corel and Sowerby datasets using class-specific affinities.

the product-of-experts model [10]. Following their methodology we also use a random split of 60% for training and 40% for testing on both these datasets.

Table 4 compares our method with the two baselines, and Figure 5 shows the confusion matrices achieved by our method using class-specific affinities. For the Corel dataset, using class-specific affinities improves performance over node features alone and class-agnostic affinities in almost all the classes. Table 3 shows that compared to TextonBoost and mCRF, we work with weaker features but after learning outperform both. For the Sowerby dataset, we post poor performance in classes “Road Marking”, “Street Objects” and “Cars”. This is because that dataset contains low-resolution images (64 × 96) and these objects sometimes occupy as few as 2 pixels in the image. Since our model is based on superpixels, our training set consists of fewer than 5 superpixels from each of these three classes, which predictably leads to poor training of classifiers.

4. Discussion and Conclusions

We introduce the concept of class-specific affinities, *i.e.*, affinities that are not just a function of the features measured at the data-points, but also the classes to which these two data-points were mapped. We present examples of the inherent conflicts that are impossible to resolve when using class-agnostic affinities (Figure 1), and show (Figure 4) how our learnt class-specific affinities incorporate a richer set of dependences and relationships between pairs of classes and visual words. One consequence of employing our proposed family of affinities is that the number of learned parameters scales quadratically with the number of classes (since we examine all pair-wise class interactions). Fortunately, our experiments show that the standard datasets are sufficiently large to allow us to learn these parameters without overfitting. We demonstrate our framework on the task of joint segmentation and recognition of object classes in images, and show our proposed framework for affinities can achieve comparable to state-of-the-art performance even while using weak appearance features. In particular, we show that the improvement obtained over the node-only and class-agnostic baselines is significantly greater than that reported in recent related work. As future work, it would be interesting to combine with the proposed learning framework with more powerful features (*e.g.*, those used by TextonBoost [27]).

Acknowledgments

We thank Denver Dash, Devi Parikh and Martial Hebert for helpful discussions, and acknowledge computing resource support from the VMR lab. Dhruv Batra was partially supported by a summer internship at Intel Research Pittsburgh.

References

[1] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of markov random fields for segmentation of 3d scan data. In *CVPR*, 2005.

[2] F. R. Bach and M. I. Jordan. Learning spectral clustering. In *NIPS*. 2004.

[3] M. Blatt, S. Wiseman, and E. Domany. Data clustering using a model granular magnet. *Neural Computation*, 1997.

[4] T. Cour, N. Gogin, and J. Shi. Learning spectral graph segmentation. In *AISTATS*, 2005.

[5] T. Cour and J. Shi. Recognizing objects by piecing together the segmentation puzzle. In *CVPR*, 2007.

	Corel								Sowerby							
	Rhino	Bear	Water	Snow	Veget.	Ground	Sky	Average	Sky	Veget.	Rd. mk.	Rd. surf.	Building	St. Obj.	Cars	Average
Node Alone	60	43	75	61	80	56	57	62	87	90	15	91	34	0	0	45
Class-agnostic	60	56	72	83	76	47	91	69	95	88	0	93	44	0	6	47
Class-specific	87	92	82	91	66	83	94	85	89	88	13	94	46	0	6	48

Table 4: Class-wise accuracies for the Corel and Sowerby datasets.

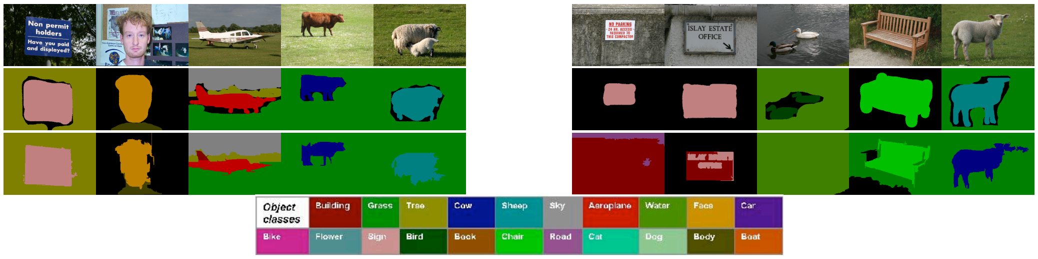


Figure 6: Example successes (left) and failures (right): Top row shows original images; middle row shows ground-truth; and bottom rows shows our results.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[7] Y. Gdalyahu, D. Weinshall, and M. Werman. Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. *PAMI*, 2001.

[8] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *CVPR*, 2004.

[9] X. He, R. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV*, 2006.

[10] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002.

[11] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[12] S. Kumar, J. August, and M. Hebert. Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In *EMMCVPR*, 2005.

[13] S. Kumar and M. Hebert. Discriminative random fields. *IJCV*, 68(2):179–202, 2006.

[14] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[15] Y. LeCun and F. Huang. Loss functions for discriminative training of energy-based models. In *AISTATS*, 2005.

[16] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*, 2006.

[17] J. Malik, S. Belongie, T. K. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 2001.

[18] M. Meila and J. Shi. Learning segmentation by random walks. In *NIPS*, 2000.

[19] T. Meltzer. `http://www.cs.huji.ac.il/~talyam/inference.html`. Inference Package for undirected graphical models.

[20] T. Minka. A comparison of numerical optimizers for logistic regression. Technical report, Machine Learning Department, Carnegie Mellon University, 2004.

[21] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[22] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, 2006.

[23] D. Pelleg and A. Moore. `http://www.cs.cmu.edu/~dpelleg/kmeans.html`. Auton Lab K-means and X-means implementation.

[24] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, 2000.

[25] N. Shental, A. Zomet, T. Hertz, and Y. Weiss. Pairwise clustering and graphical models. In *NIPS*. 2004.

[26] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[27] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[28] J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *CVPR*, 2007.

[29] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *ICCV*, 1999.

[30] L. Yang, P. Meer, and D. Foran. Multiple class segmentation using a unified framework over mean-shift patches. In *CVPR*, 2007.