Recognition of Hand-drawn Sketches with Marcov Random Fields

Chen Yang, Suling Yang*
Department of Computer Science
University of British Colombia
cyang, sulingy@cs.ubc.ca

Abstract

Freehand sketches are complex for recognition. Individual fragments of the drawing are often ambiguous to be interpreted without contextual cues. Markov Random Field (MRF) that ends up with a global model by simply specifying local interactions can naturally suffice the requirement.

In our project, a recognizer based on MRF has been constructed to jointly analyze local features in order to incorporate contextual cues during inference process. Shapes in sketches are detected and matched to a deformable template. Whereas standard belief propagation (BP) is not guaranteed to converge for inference on graphical models with loops, we find that loopy belief propagation (LBP) does converge in our experiment. The final recognition is found as the MAP marginal by global belief propagation.

1 Introduction

Recognition of fragments depends heavily on context in a drawing. For example, the two different sketches in figure 1 contain similar fragments. The fragments can be recognized as the body of wineglass only with the recognitions of a stem and a bottom, or as the body of teacup with the recognition of a handle. Thus the recognition should be treated as a joint classification task instead of independent classifications. To capture the variability and the interactions among different sets of relevant classes, graphical models provide probabilistic approaches to take them as random variables forming a set with joint probability distribution [1]. From a global point of view, all the variables are mutually dependent. Due to the high dimensionality of finding a global solution, graphical models decompose the distribution as product of factors and use graph to capture the Markovian properties among the random variables. In the graph, each node will only directly depend on its neighbors. For inference, Markovian properties also imply the computations to be done remain local. Graphical models use message passing algorithms to only update neighbors at each step of computation. As powerful tools to capture Markovian properties, Graphic models have been widely

* Suling and I have discussed and implemented the system together. Suling cared more about the brute force method while I paid more attention to preprocessing of sketches, feature extraction and inference.

applied to image processing and computer vision. The Markovian modeling for classification can be generally divided into discriminative and generative approaches.

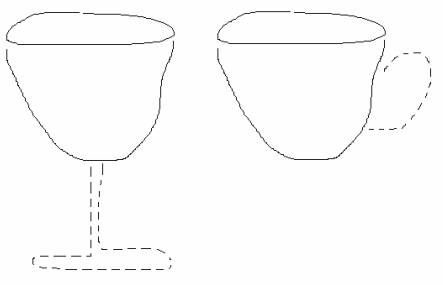

Figure 1: The left wineglass and the right tea cup contain the same fragments (solid parts). This fragment can be recognized correctly only with the recognition of the other fragments (dashed parts)

Conditional Random Fields (CRFs), which can be taken as a network of interacting classifiers, can be generally considered as a discriminative approach. One individual classifier can influence the decisions of its neighbors through the network. Kumar and Hebert [2] applied CRF to the classification of natural image regions by incorporating spatial dependencies in the labels as well as the observed data. Szummer and Qi[3] constructed a CRF-based recognizer on hand-drawn diagrams. In [3], pen strokes were firstly subdivided into small fragments which were small enough to belong to a single container or connector. Then a CRF was constructed on the ink fragments. The site potential refers to the compatibility of the label of a single fragment in its ink context and interaction potential models whether a pair of fragments prefers the same or different labels. In the third step, this CRF is trained and a global compatible solution is given by MAP or MM.

Unlike Szummer and Qi's example, our task is to classify fragments of a sketch to more than two classes which are defined on a template graph. Thus, generative approach is more adequate for our problem. Compared with the discriminative approaches, Coughlan and Ferreir[4,5] found objects in images using deformable templates which were fit with dynamic programming and belief propagation. We have used similar generative modeling since we are more interested in recognition of specified objects in sketches. The generative model is represented as a deformable template of the chosen object. By preprocessing on the sketches, we transform the points of a drawing into a graph. In the training process, the user can arbitrary label the nodes in the graph. MRF is trained and applied to a new sketch which is also preprocessed and represented as a graph. In inference process, standard BP can not be guaranteed to converge to the optimal solution if the graph has loops. We have used LBP[6] and found it to converge in our experiment. MAP marginal [4] is estimated for each node after global belief propagation. The recognition result is a matching between the nodes of trained sketch and test sketch with a probability.

After introduction, we will present how our template is constructed for an example object in section 2. The training process is explained in section 3 followed by explanation of LBP inference in section 4. Experimental results are given in section 5 with conclusions at the last section.

2 Deformable Template

Our deformable template is constructed to recognize and match a given shape in a sketch. The template is defined as MRF with local evidence and edge potentials. We need to find the best match between the template and the shape in a sketch.

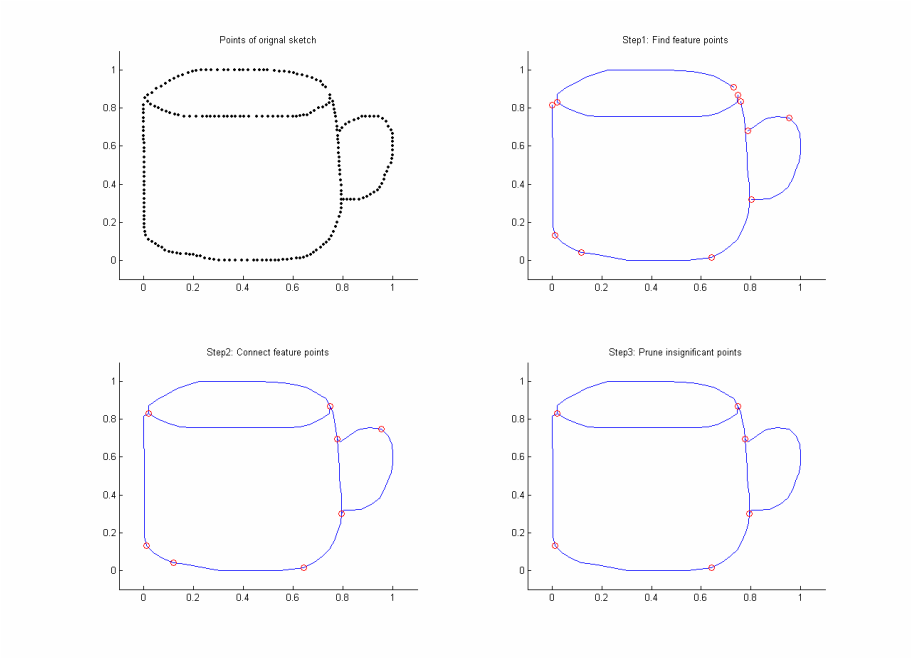

Figure 2.Preprocess of constructing a graph from sketch data.

2.1 Preprocessing

The goal of our preprocessing is to turn free-hand sketches into a graph. The input sketch for our system is represented as an array of time-stamped pixel positions obtained from a sketch interface. A stroke is an array of (x,y,t) values that describe the path that the pen traces between mouse down and mouse up events. The output graph contains key points that are connected by curve segments between pairs of them. The preprocessing has been implemented in three phases.

1. Detection of Feature Points

   A drawing consists of several strokes. We want to find the points that can capture the features of the strokes as the candidates of key points. In most case, the feature points are vertices at corners of a stroke. Having tried finding feature points from the minima of speed, the turn of directions and the maxima of curvature in the presence of noise, we pick out feature points from the curvature data. Average-based filtering [7] is used here.

2. Proximity Linking of Key points

   Having found feature points which also include the start and end points of strokes, we need to link strokes to a graph. A threshold of distance is used here to connect nearby feature points. If there are still unconnected feature points after the first connection step, the nearest normal points within the threshold distance will be flagged as new feature points and connected instead.

3. Elimination of Insignificant Key points

   There are always redundant key points by noise in the data and we need to eliminate them for matching. How to prune those key points depends on the how many key points need to be kept. Our strategy here is just to keep as the same number of key points as that defined by user in training examples. Basically we prune the key point that is near the straight line passing its two neighbors, or which is too near its two neighbors.

After the three preprocessing steps, the original sketches can be turned into key points with curve segments between pairs of them. We take the key points as nodes and add an edge to two nodes if they are directly connected by segments (figure 3). The structure of the graph only depends on the objects, so it can be any random graph and can contain loops.

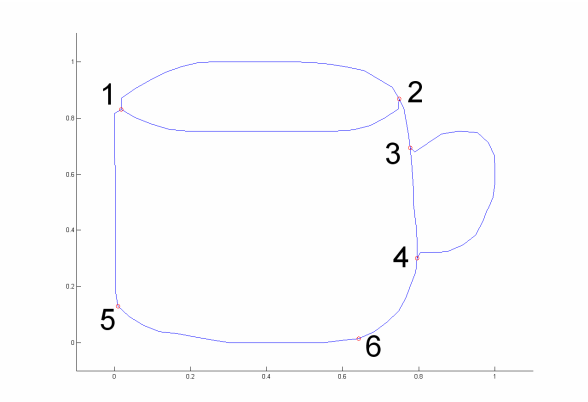

Figure 3. A reference shape of template with labels of key points

2.2 Features

In the template graph, each node has an associated label variable. Our approach is to compute some low-level features and represent them in potentials in the random fields. Our features are based mainly on the connectivity, i.e. we want to recognize the objects which have same topology as the given example.

For each key point, the site features come from the incident curve segments which are incident with another given key point. We can compute the shape context [8] which is basically the histogram of distance and angles to the points of neighboring segments. In this project, only the number of the incident curve segments is considered as site feature.

The interaction features come from the segments between pairs of key points. Curve matching is a common method for recognizing given shape in computer vision. In this project, only connectivity is considered. If two key points are directly connected, the number of edges between them needs to be computed first. We also consider the length of the curve segments to estimate the distance between two key points. If there are multiple connections, the lengths of different segments will be compared to decide whether they have similar length. The ratio of lengths of multiple connections gives us further connective relationship between two key points.

Key Points	1	2	3	4	5	6
Number of Linked Segments	3	3	3	3	2	2

Key Points	1	2	3	4	5	6	Key Points	1	2	3	4	5	6
1		0.80 0.86			0.71		**1**		3			2	
2	0.80 0.86		0.18				**2**	3		1			
3		0.18		0.39 0.78			**3**		1		4		
4			0.39 0.78			0.34	**4**			4			1
5	0.71					0.67	**5**	2					2
6				0.34	0.67		**6**				1	2	

Table 1. Site and interaction features from the example in Figure 3

Top- Number of edges converged at nodes; Bottom Left - Length of edges; Bottom Right - Classification of edges (One connection: 1. short; 2. long (mean length = 0.48). Two connections: 3. similar length; 4. one short, one long (ratio = 1.5))

2.3 Site and Interaction Potentials

Given a reference shape, we need to match a new sketch to it with the similar site and interaction features as in Table 1. NSite is the number of segments linked at one key point which take the value {1, 2, 3,.., Nmax} where Nmax is the maximum number of segments intersected at one key point. NInter is the connection between two key points that can be classified to {1, 2, 3, 4} in this example.

We can express the posterior as a Markov random field with pair wise interactions:

$$P(q_1,...,q_N \mid D) = \frac{1}{Z}\prod_i y_i(q_i)\prod_{i,j} y_{i,j}(q_i,q_j) \quad (1)$$

Where $y_i(q_i)$ is the site potential for q_i, $y_{i,j}(q_i,q_j)$ is the interaction potential between q_i and q_j, and Z is the normalized constant.

3 Training

In the training, the user needs to label the key points in an arbitrary sequence of a reference sketch (interface in figure 4). Then we need to compute the site and interaction potentials of the features, i.e. for each site feature and edge feature we need to compute their probabilities of being specified labels. We can measure the empirical counts as the maximum entropy to

get a standard table potential. Using the same example from section 2, we can compute some site and interaction potentials as below.

For site feature NSite = 3, the site potential is [0.25 0.25 0.25 0.25 0 0]

For interaction feature Ninter = 1, the interaction potential is a 6x6 matrix P.

$$P(i,j)=\begin{cases}0.25, & if\{i,j\}=\{2,3\},\{3,2\},\{4,6\},\{6,4\}\\ 0, & otherwise\end{cases} \tag{2}$$

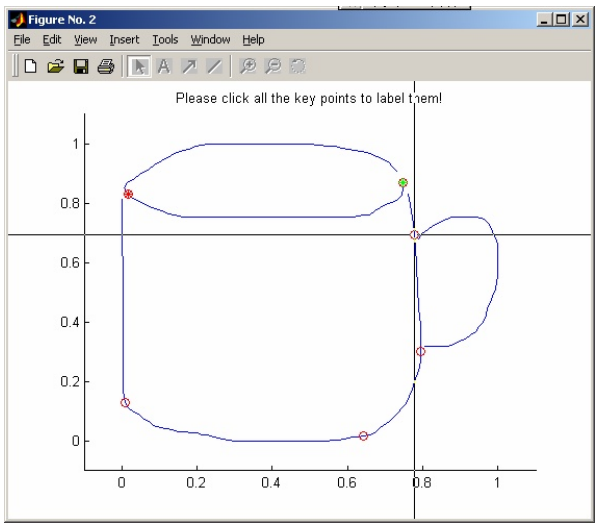

Figure 4. Interface for training

4 Inference

For inference on the graph with loops, LBP is applied to find MAP (maximum a posterior) marginal.

4.1 MAP Marginal

One way to match the shape to the template is to calculate the MAP.

$$Q^{*}=\arg\max_{Q} P(Q\mid D) \tag{3}$$

In our application, we use the MAP marginal [4], which is the MAP applied separately to each variable q_i:

$$q_i^{*}=\arg\max_{q_i} P(q_i\mid D) \tag{4}$$

We estimate the MAP marginal using brute force and LBP[6] methods. The recognition results have proved MAP marginal to be correct in our case.

4.2 Brute-force Approaches

When we were pursuing an adequate recognizer for tea cups, the first thing came into our minds was to use a brute-force method in order to see what features and parameters were suitable for our problem. Through the process of brute-force approaches, we obtained some general ideas about how we could interpret features in a meaningful way.

Towards the tea cup problem, we implemented a program to match key points of a freehand sketch to those that we had pre-defined on the deformable template. In this program, we used the number of incident edges of each key point as site feature, and the number of edges between a pair of key points and the ratio of those edges as interaction feature. However, this didn't give us accurate solution empirically, because these features lacked of information of the whole graph. We'll see later how BP message passing could solve this problem, but for now we need to incorporate more features. Then we made the probabilities of key points based on current features into account, and thus the probabilities of neighbours of each key point are affected. With all these features, this program works well for tea cup examples. Nevertheless we need a more general approach for general graph.

Our second naïve approaches was to develop a training program which took a graph as a template that was represented by the numbers of incident edges of key points, the adjacency matrix, and the lengths of edges. This program learned the features described above from the graph and outputted suitable potentials. A similar program then got the output and inferred on a new graph using these potentials. To infer on a new graph, the program learned the features from that graph, and assigned corresponding probabilities to each key point based on these features and the potentials from previous program. Then, an estimated labeling was obtained by taking the maximum marginal of each key point.

Empirically the second works well for simple asymmetric graph, e.g. tea cup examples. However, for more difficult recognition problems, e.g. mouse examples, this program fails at points that have similar local features. Hence, we looked for some inference methods that can apply global contextual information. CRF and MRF both can effectively use all contextual cues to give correct marginals. Moreover, BP applied on MRF is a powerful and suitable tool for our problem.

4.3 Loopy Belief Propagation

Pearl's belief propagation algorithm [9] is a profoundly powerful and efficient algorithm for finding or estimating posterior probabilities. Yair Weiss et al. [10] explained the correctness of belief propagation in Gaussian graphical models of arbitrary topologies. Furthermore, Murphy [6] showed that with large priors randomly in the range [0, 1] and large weights loopy belief propagation converges and gives an excellent correlation with the correct marginals. Therefore, loopy beliefs can provide desirable estimation for our problem.

In each iteration of Pearl's belief propagation algorithm, we calculate a belief for each node X, $BEL(X) = P(X = x|E)$ where E is the observed evidence and $P(X = x|E)$ depends on the messages from X's parents and children, denoted by $p_X(u_k)$ and $l_{Y_j}(x)$ respectively. More precisely [6],

$$BEL(x) = a\, l(x)\, p(x) \tag{5}$$

Where

$$l^{(t)}(x) = l_X(x) \prod_j l_{Y_j}^{(t)}(x) \tag{6}$$

And

$$p^{(t)}(x) = \sum_u P(X = x \mid U = u) \prod_k p_X^{(t)}(u_k) \tag{7}$$

The message that X passes to its parent U_i is given by:

$$l_X^{(t+1)}(u_i) = a \sum_x l^{(t)}(x) \sum_{u_k : k \neq i} P(x \mid u) \prod_{k \neq i} p_X^{(t)}(u_k) \tag{8}$$

and the message X sends to its child Y_j is given by:

$$p_{Y_j}^{(t+1)}(x) = ap^{t}(x) l_X(x) \prod_{k \neq j} l_{Y_k}^{(t)}(x) \tag{9}$$

5 Experiment Results

We have tested our recognizer on three classes of objects: cup, wineglass and mouse. All the sketches including train and test sketches are drawn with no constraints on stroke sequence or number of strokes for one sketch. Then all the sketches have been preprocessed to graphs with key points as the nodes. For each class of object, we randomly picked one sketch as the training data and labeled the key points. Then all the other sketches were matched to the labeled key points. The matching results in figure 5 show that even with only one training sketch, our recognizer can still find the correct matches invariant to global rotation and translation. This recognizer can also be used to recognize a similar object from multiple objects such as in figure 6 where we match a cup to both cup and wineglass.

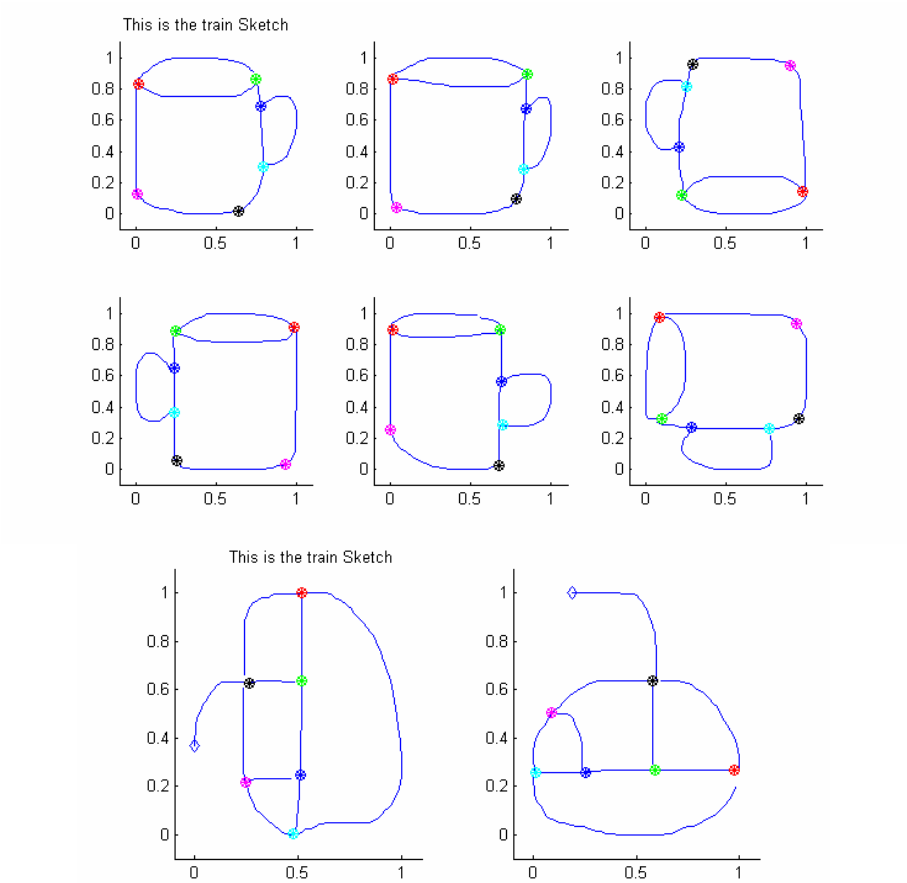

Figure 5. Matching results for same class of objects

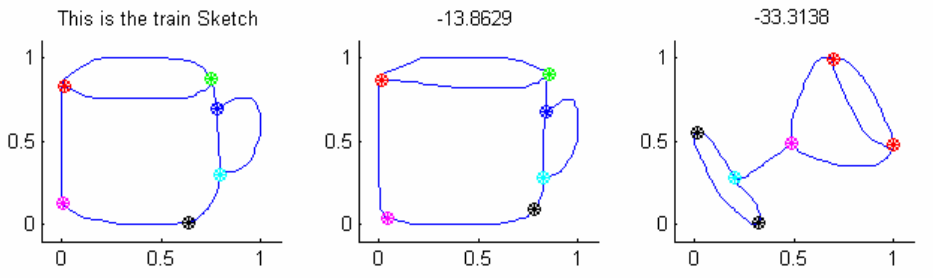

Figure 6. Matching results for different objects

	Number of Iterations	Number of Iterations(1% noise)
Cup	**3**	**5**
Mouse	**7**	**11**

Table 2. Iterations of LBP to converge for cup and mouse examples without and with noise

LBP has been proved to converge for graph with loop in our case. In table 2, we can see that it will take more iteration steps to converge for the mouse example than cup because the mouse has more nodes and states. If some noise is added to the template model, LBP can still guarantee to converge with more iteration steps.

6 Conclusions

In this paper, we have presented a procedure of generative modeling and inference on sketch recognition. The recognition is achieved by correctly matching key points of a sketch to a given template with some energy cost. By preprocessing, a sketch is first transformed into a MRF that captures the individual match and interaction between neighboring matches by site and interaction potentials. Then LBP is used for efficient inference on the graph with loops. The experiment results have shown that LBP can converge and the MAP marginal give us a correct matching of all key points.

Our initial motivation of applying random fields to sketch recognition came from the work by Szummer of applying CRF to recognizing hand-drawn diagrams [3]. Although CRF can capture interactions between labels and take arbitrary features on all the observed data, it requires a crucial training procedure to obtain favorable parameters and much computation for normalizing constant and marginals of densely connected graphs. On the other hand, we are more interested in recognizing specified object and we can also easily construct a template of the object. Hence, we choose generative modeling instead of discriminative modeling. At the same time, LBP provides an efficient approach for inference. MAP marginal from global belief prorogation can give us a correct matching for each node.

Graphical model has been proved to be a strong tool for image processing and computer vision. However, its application has been constrained largely on true imagery instead of sketches. From this project, some worthwhile future work may be done in two aspects. One is from computer vision to construct more meaningful features. The other is to incorporate one-to-one correspondence and junk pruning procedure into modeling and inference of graphical models. The latter work can also be done by dynamic programming and machine learning techniques.

References

[1] Perez, P., Markov random fields and images, *CWI Quarterly*, 1998.

[2] Kumar, S. and Hebert, M., "Discriminative Fields for Modeling Spatial Dependencies in Natural Images," In *Proc. Advances in Neural Information Processing Systems (NIPS)*, December 2003

[3] Szummer, M. and Qi, Y., "Contextual Recognition of Hand-drawn Shapes with Conditional Random Fields," To be Published.

[4] Coughlan, J. and Shen, H., Shape Matching with Belief Propagation: Using Dynamic Quantization to Accommodate Occlusion and Clutter. *To appear in GMBV 2004*

[5] Coughlan, J. and Shen, H., Finding Deformable Shapes Using Loopy Belief Propagation. *Technical report in preparation.*

[6] Murphy, K.P., Weiss, Y., and Jordan, M.I., Loopy belief propagation for approximate: an empirical study. In *Proceedings of Uncertainty in AI*, 1999.

[7] Sezgin, T.M., Stahovich, T. and Davis, R., Sketch Based Interfaces: Early Processing for Sketch Understanding. In *The Proceedings of 2001 Perceptive User Interface Workshop (PUI 01).*

[8] Belongie, S., Malik J. and Puzicha J., "Shape Matching and Object Recognition Using Shape Contexts," accepted for publication in PAMI.

[9] Pearl, J., Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, 1988.

[10] Weiss, Y., Correctness of belief propagation in Gaussian graphical models of arbitrary topology. 1999.