Learning Diagram Parts with Hidden Random Fields

Martin Szummer
Microsoft Research
Cambridge, CB3 0FB, United Kingdom
szummer@microsoft.com

Abstract

Many diagrams contain compound objects composed of parts. We propose a recognition framework that learns parts in an unsupervised way, and requires training labels only for compound objects. Thus, human labeling effort is reduced and parts are not predetermined, instead appropriate parts are discovered based on the data. We model contextual relations between parts, such that the label of a part can depend simultaneously on the labels of its neighbors, as well as spatial and temporal information. The model is a Hidden Random Field (HRF), an extension of a Conditional Random Field. We apply it to find parts of boxes, arrows and flowchart shapes in hand-drawn diagrams, and also demonstrate improved recognition accuracy over the conditional random field model without parts.

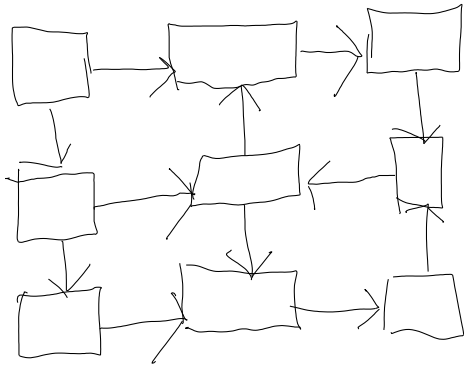

Figure 1. Hand-drawn diagram consisting of boxes and arrows.

1. Introduction

Hand-drawn diagrams consist of objects such as containers and connectors, which are individually composed of parts. For example, a part could be the side of a rectangle or the head of an arrow. A part may be produced as a fragment of a pen stroke, be a full pen stroke, or possibly comprise multiple strokes. Parts combine in versatile ways to form compound objects such as tables, which have varying configurations of rows, columns and cells. In other words, a small set of parts can give rise to rich classes of objects. This compositional aspect of drawings suggests that recognition of objects may be facilitated by identification of parts.

Recognition of hand-drawn objects and parts is challenging because the meaning of individual pen strokes depends on their context. A pen stroke can be simple and indistinctive by itself, and may acquire meaning only by participating in a larger unit. For example, we can recognize arrows by decomposing them into arrow heads and arrow stems, and noting that they occur next to another and to sides of containers (Figure 1). Thus, to exploit context, we need to model parts and relations between parts. We note that the observations about compositionality and context also apply to printed document analysis, and our proposed framework is applicable to that domain.

The complexity and large variability of hand-drawn data and different recognition tasks suggest that machine learning techniques are appropriate. Machine learning systems can be trained to adapt to the user and to novel input. Unfortunately, such training typically requires many examples that must be labeled at the finest-grained level of the system, i.e., the level of individual parts.

We propose a learning framework for complex recognition tasks that consist of multiple interrelated parts, such as hand-drawn diagrams. Our framework automatically discovers these parts, and therefore requires only coarse-grained training labels for compound objects. This has several advantages. First, human labeling effort can be reduced, as only larger objects need to be manually labeled. Second, even if we are not interested in recognizing the parts themselves, the compound object recognizers can become more accurate by modeling parts automatically chosen for the recognition task. Third, the system can also model rich classes that really are unions of distinct subcategories (not strictly parts), and which could not be captured by simpler mod-

els. We can still represent each subcategory as a part, as we do not require all parts to be present in order to recognize a compound object.

Related work has explored recognizers based on manually specified parts in drawings [1]. We are not aware of other techniques that learn parts and relationships between parts in two dimensions, however there is a wealth of hidden Markov models that learn parts in one-dimensional sequences, such as symbol recognizers [6]. Bayesian networks have been used to recognize and generate (but not learn) parts in two dimensions [2]. Contextual parts-based models are also common in the computer vision community [7].

Our model is a conditional hidden random field (HRF) [4], an extension of a conditional random field (CRF) [5] and a hidden Markov random field [10]. Conditional random fields are powerful models of dependencies between items, which use flexible features, and are trained in a discriminative way. They have been successfully applied to recognize hand-drawn diagrams [8] and outlines [9]. Unfortunately, conditional random fields must be trained with fully labeled training data, and cannot discover parts.

Unlike a CRF, our HRF also captures relations between *unobserved* ("hidden") variables, which serve to identify parts. These hidden variables also depend on features of the data (such as lengths and angles of pen strokes), and on observed labels. A simple HRF, modeling restricted spatial structures forming trees, has previously been used to recognize objects in images [7].

We first review CRFs and then propose a full two-dimensional HRF model, including cyclical spatial dependencies. Next, we apply the model to recognize and discover parts in hand-drawn diagrams of boxes, arrows and flowcharts.

2. Conditional Random Fields

Many traditional discriminative classification approaches such as neural networks and support vector machines classify labeled objects examples only independently of one another. In particular, each object or part in a diagram would be classified without considering the labels of the others. In contrast, conditional random fields (CRFs) [5] model dependencies not only between input data and its labels, but also dependencies between labels and neighboring labels, thus allowing us to exploit contextual clues. CRF model this joint distribution over labels, but unlike MRFs, they do not model the distribution over the input but merely condition on it.

A conditional random field can be seen as a network of interacting classifiers: the decision made by one classifier influences the decisions of its neighbors. Edges of a graph denote what interactions are included.

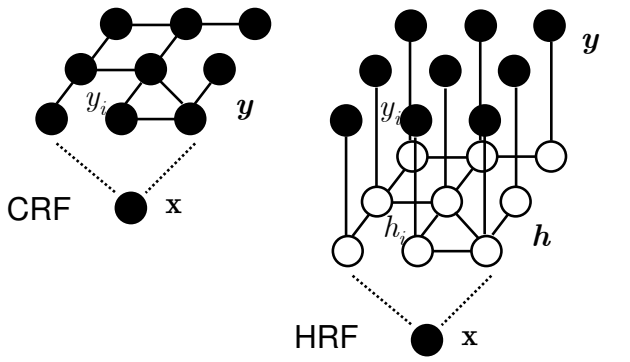

Figure 2. A graphical model for a CRF and an HRF modeling irregular spatial dependencies in two dimensions. At training time, input x and object labels y are observed (indicated by filled circles), but parts h are unobserved (empty circles). All inputs x are connected to the layer above it (indicated by dotted lines).

Formally, let $\mathbf{x} = \{x_i\}$ be an input random vector for the observed data, and $\mathbf{y} = \{y_i\}$ be an output random vector over labels of the corresponding data. The input $\mathbf{x}$ might range over the pen strokes and the output $\mathbf{y}$ range over discrete labels of shapes to be recognized. Interactions are described by an undirected graph $G = (V, E)$ where the nodes V are parts or objects to be classified and the edges E indicate possible dependencies. An example graphical model for a CRF is depicted in Figure 2.

A CRF describes the conditional probability distribution $P(\mathbf{y}|\mathbf{x})$ between the input data and the labels. It has the form of a normalized product of potential functions $\Psi^{(1)}$ and $\Psi^{(2)}$ on nodes and edges of the graph, measuring compatibility between features and labels:

$$P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{i \in V} \Psi_i^{(1)}(y_i, \mathbf{x}; \boldsymbol{\theta}) \prod_{(i,j) \in E} \Psi_{i,j}^{(2)}(y_i, y_j, \mathbf{x}; \boldsymbol{\theta}) \quad (1)$$

$$\text{and } Z(\boldsymbol{\theta}) = \sum_{\mathbf{y}} \Big(\prod_{i \in V} \Psi_i^{(1)}(y_i, \mathbf{x}; \boldsymbol{\theta}) \prod_{(i,j) \in E} \Psi_{i,j}^{(2)}(y_i, y_j, \mathbf{x}; \boldsymbol{\theta}) \Big)$$

$Z(\boldsymbol{\theta})$ is a normalizing constant known as the partition function (for brevity we suppress its dependence on $\mathbf{x}$).

Both types of potentials use a linearly weighted combination of features $\mathbf{g}_i(\mathbf{x})$ or $\mathbf{f}_{ij}(\mathbf{x})$, passed through an expo-

nential nonlinearity:

$$\text{Site} \quad \Psi_i^{(1)}(y_i, \mathbf{x}; \boldsymbol{\theta}) = \exp(\boldsymbol{\theta}^{(1)}(y_i)^{\mathrm{T}} \mathbf{g}_i(\mathbf{x})) \quad (2)$$

$$\text{Interaction} \quad \Psi_{i,j}^{(2)}(y_i, y_j, \mathbf{x}; \boldsymbol{\theta}) = \exp(\boldsymbol{\theta}^{(2)}(y_i, y_j)^{\mathrm{T}} \mathbf{f}_{ij}(\mathbf{x})). \quad (3)$$

Each class has separate site weights $\boldsymbol{\theta}^{(1)}(y_i)$ and pairs of classes have interaction weights $\boldsymbol{\theta}^{(2)}(y_i, y_j)$. One can view the interaction potential (3) as a classifier of pairs of neighboring labels that depends on the data through the features $\mathbf{f}_{ij}(\mathbf{x})$. Note that there are no independence assumptions on features $\mathbf{g}_i(\mathbf{x})$ and $\mathbf{g}_j(\mathbf{x})$, nor on $\mathbf{f}_{ij}(\mathbf{x})$ for different sites i and j. For example, features can overlap, be strongly correlated, and extend over long distances or even depend on all input.

3. Hidden Random Fields

A hidden random field [4] extends the conditional random field by introducing hidden variables $\boldsymbol{h}$ that are not observed during training. These hidden variables provide extra modeling power, allowing the model to uncover an additional layer of structure not explicit in the observed labels. For example, they could provide the model with extra memory to propagate long-range interactions.

In our case, we shall employ the hidden variables to indicate parts of compound objects. During training, we observe an object label y_i at each site i, but we assume that the object consists of unknown parts h_i. For example, a pen stroke labeled as 'arrow' may specifically be an 'arrow stem' or 'arrow head' part. The HRF models dependencies between these parts, e.g. an 'arrow head' may be more likely to occur next to an 'arrow stem' rather than another 'arrow head'. Figure 2 shows the graphical model for the HRF at training time: the unobserved h_i nodes are drawn as empty circles. Edges indicate direct dependencies included in the model.

For simplicity, we fix the relationship between object labels and parts a priori. In particular, we specify the number of parts for each compound class, and do not share parts between classes. In other words, we restrict a part variable h corresponding to a label y to assume only a subset of values, so that h uniquely determines y. We denote this deterministic mapping from parts to objects by $y(h_i)$.

The HRF model averages over the unobserved hidden variables. Mathematically,

$$P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \sum_{\mathbf{h}} P(\mathbf{y}, \boldsymbol{h}|\mathbf{x}, \boldsymbol{\theta}) \quad (4)$$

The joint model over $(\mathbf{y}, \boldsymbol{h})$ is similar to the CRF (with the labels y_i exchanged for parts h_i):

$$P(\mathbf{y}, \mathbf{h}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{i \in V} \Psi_i^{(1)}(h_i, \mathbf{x}; \boldsymbol{\theta}) \Psi_i^{(3)}(y_i, h_i; \boldsymbol{\theta}) \cdot \prod_{(i,j) \in E} \Psi_{i,j}^{(2)}(h_i, h_j, \mathbf{x}; \boldsymbol{\theta}) \quad (5)$$

where the extra potentials $\Psi_i^{(3)}$ are fixed at

$$\Psi_i^{(3)}(y_i, h_i; \boldsymbol{\theta}) = \delta(y(h_i) = y_i) \quad (6)$$

and $\delta(\cdot)$ is an indicator function. Substituting our potentials, the probability becomes

$$P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \sum_{\mathbf{h}} \Big[\exp\big(\sum_{i \in V} \boldsymbol{\theta}^{(1)}(h_i)^{\mathrm{T}} \mathbf{g}_i(\mathbf{x}) \big) \cdot \exp\big(\sum_{(i,j) \in E} \boldsymbol{\theta}^{(2)}(h_i, h_j)^{\mathrm{T}} \mathbf{f}_{ij}(\mathbf{x}) \big) \prod_{i \in V} \delta(y(h_i) = y_i) \Big]. \quad (7)$$

We determine the parameters $\boldsymbol{\theta}$ during training (section 3.2), but first describe inference given such trained parameters.

3.1. Inference in HRFs

We are typically interested in predicting labels for new data $\mathbf{x}$. We predict by averaging out the hidden variables and all label variables but one, to calculate the maximum marginals

$$y_i^{\text{MM}} = \text{argmax}_{y_i} P(y_i|\mathbf{x}, \boldsymbol{\theta}), \quad (8)$$

$$= \text{argmax}_{y_i} \sum_{\mathbf{y} \setminus y_i} \sum_{\boldsymbol{h}} P(\mathbf{y}, \boldsymbol{h}|\mathbf{x}, \boldsymbol{\theta}), \quad \forall i \in V. \quad (9)$$

Alternatively, we can calculate the most likely joint configuration of labels by taking the argmax simultaneously over all $\mathbf{y}$. Although such configurations are globally consistent, the per fragment error tends to be slightly worse. To see what parts the algorithm has learned, we can look at the most likely parts:

$$h_i^{\text{MM}} = \text{argmax}_{h_i} P(h_i|\mathbf{x}, \boldsymbol{\theta}), \quad (10)$$

$$= \text{argmax}_{h_i} \sum_{\mathbf{y}} \sum_{\boldsymbol{h} \setminus h_i} P(\mathbf{y}, \boldsymbol{h}|\mathbf{x}, \boldsymbol{\theta}), \quad \forall i \in V. \quad (11)$$

Both of these tasks require summing over a joint space of $(\mathbf{y}, \boldsymbol{h})$ of exponential size in the number of variables. Fortunately, because of the factorized structure of $P(\mathbf{y}, \mathbf{h}|\mathbf{x}, \boldsymbol{\theta})$ and the assumed sparsity of interactions in the graph G, there is an efficient dynamic programming algorithm to do so [3]. We remove cycles in the original graph G by triangulating the graph and construct a junction tree of cliques, and then apply the junction tree algorithm to calculate all the required marginals. The cost is exponential in the size of the largest clique in the junction tree, which was manageable of size 9 in our experiments.

3.2. Training HRFs

We train the HRF by maximizing the conditional log likelihood $\mathcal{L} = \log P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ of the observed labels $\mathbf{y}$, plus a Gaussian prior on the parameters $P(\boldsymbol{\theta}) = N(\boldsymbol{\theta}; 0, \sigma^2\mathbf{I})$. Since we do not know the assignment of the hidden parts h we have to infer their values. The EM algorithm could be applied here, but we believe that it is faster to maximize the observed likelihood directly. Training via gradient ascent using the BFGS quasi-Newton takes 30 minutes for our dataset. The gradients with respect to the parameters $\boldsymbol{\theta}^{(1)}(h')$ and $\boldsymbol{\theta}^{(2)}(h', h'')$ have simple forms

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}^{(1)}(h')} = \sum_{i \in V} \Big(P(h_i = h' \mid \mathbf{y}, \boldsymbol{\theta}) - P(h_i = h' \mid \boldsymbol{\theta}) \Big) \mathbf{g}_i(\mathbf{x}) \tag{12}$$

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}^{(2)}(h', h'')} = \sum_{(i,j) \in E} \Big(P(h_i = h', h_j = h'' \mid \mathbf{y}, \boldsymbol{\theta}) - P(h_i = h', h_j = h'' \mid \boldsymbol{\theta}) \Big) \mathbf{f}_{ij}(\mathbf{x}). \tag{13}$$

The necessary marginals $P(h_i = h' \mid \mathbf{y}, \boldsymbol{\theta})$ and $P(h_i = h' \mid \boldsymbol{\theta})$ are calculated as during inference via the junction tree algorithm, as are the pairwise marginals for h_i, h_j.

Unlike the log likelihood function for a CRF, the log likelihood of an HRF is not convex, and may have local maxima. To find a good maximum we could constrain the likelihood by training with a few labeled parts h, which we plan to do in future work. The parameter prior σ^2 is important for reducing overfitting and is chosen by cross-validation.

3.3. Parameter structure

Our current interaction features are symmetric so that $\mathbf{f}_{ij} = \mathbf{f}_{ji}$. In this case the interaction parameters will also be symmetric so that $\boldsymbol{\theta}^{(2)}(h', h'') = \boldsymbol{\theta}^{(2)}(h'', h')$. To further reduce the number of parameters, we share parameters between hidden variables corresponding to different labels, such that $\boldsymbol{\theta}^{(2)}(h', h'') = \boldsymbol{\theta}_{\text{shared}}$ when $y(h') \neq y(h'')$. All gradient contributions to shared parameters are summed.

Furthermore, we remove a redundancy in the site and interaction parameters. The parameters $\boldsymbol{\theta}^{(1)}(h_i)$ give an equivalent model to the parameters $\boldsymbol{\theta}^{(1)}(h_i) - \boldsymbol{\theta}^{(1)}(1)$, hence we simply fix the parameter $\boldsymbol{\theta}^{(1)}(1) = \mathbf{0}$. Similarly, we fix $\boldsymbol{\theta}^{(2)}(1, 1) = \mathbf{0}$.

4. Application to Hand-drawn diagram classification

We apply HRFs to classification of online ink, in particular to a classification problem of boxes and arrows, and also to flowcharts. Boxes and arrows can be seen as compound objects with sides, stems and heads, thus we are interested to see what parts the HRF will learn, and whether classification performance is improved over a CRF model without parts.

We break the task into three steps:

1. Subdivision of pen strokes into fragments,
2. Construction of an HRF on the fragments,
3. Training and inference on the random field.

The input is electronic ink recorded as sampled locations of the pen, and collected into strokes. In the first step, strokes are divided into fragments small enough to belong to a single box or arrow. In contrast, strokes can occasionally span more than one shape, for example when a user draws a box and an arrow without lifting the pen. We choose fragments to be groups of ink dots within a stroke that form straight line segments within some tolerance.

In the second step, we construct a hidden random field on the fragments. Each ink fragment is represented by a hidden node in the graph. In successive experiments, we assume that boxes and arrows consist of two, three or four parts each. Once a label node is observed or hypothesized, the hidden variable is constrained to assume only parts corresponding to that label. We also do an experiment assume the objects have only one part each, in which case the model reduces to a CRF.

The HRF potential functions $\Psi^{(1)}$ and $\Psi^{(2)}$ quantify how compatible parts are with the underlying ink and with neighboring parts and labels. Each site potential refers to the part h_i of a particular fragment and its ink context. The context may be any subset of the diagram $\mathbf{x}$, but typically only neighboring fragments are included. Interaction potentials model whether two parts are compatible given pairwise contextual features.

We compute many redundant low-level ink features, and incorporate them into potentials in the random field. The HRF algorithm then learns feature weights discriminative for the task.

Our two simplest features are the length and orientation angle of an ink fragment. These are encoded in site potentials. Next, for interaction potentials, we compute features depending on pairs of fragments i and j. These include the distance and angle between the fragments, and temporal features such as whether the pen was lifted in between them.

Finally, we include template features that detect simple perceptual relations. We employ domain-knowledge to capture parts of hand-drawn diagrams. We employ a basic corner and a T-junction feature, a box-side feature that checks whether corners are present on both ends of a fragment. Some of these features yield real number values, but most are binary. Finally, we include a bias site feature and a bias interaction feature that are both always one. In total, we

Model	Mean error
CRF no interaction	$13.3 \pm 0.7\%$
CRF joint	$2.7 \pm 0.7\%$
HRF 2+2 parts	$2.3 \pm 0.9\%$

Table 1. Number of misclassified fragments for a CRF without interaction potentials, a CRF with interactions but no parts, and an HRF with 2 parts per class.

have 61 site features and 37 interaction features. For other recognition tasks, appropriate features can be added easily.

5. Experiments and Discussion

We used a TabletPC pen computer to collect two small datasets consisting of A) 10 diagrams with boxes and arrows, and B) 40 flowcharts with rectangles, diamonds, ellipses, arrows and straight line connectors. Preprocessing the dataset yielded 800 and 3000 stroke fragments respectively, half of which were used for training, and half for testing. We labeled each training fragment as 'part of container' (i.e. a rectangle, diamond or ellipse), or 'part of connector' (i.e. a line or arrow), and this was the binary classification task. We built an HRF with interaction potentials between all pairs of fragments that were within 5mm of each other (larger values would use more context but more computation)

We compared the recognition performance of two CRF models and an HRF model with 2 parts each per container and connector, all using the same features. The performance for the models was good given the simple features and the small training set. Classification took about 1 second per diagram in our research prototype. The mean test error across 10 splits of the data is summarized in Table 1. The HRF model performed better than both CRFs. The (standard) joint CRF significantly outperformed a CRF without interaction potentials, which classifies items independently of neighboring labels, and is equivalent to a logistic regression classifier applied to the site features.

More interestingly, we examined what parts the HRF algorithm had learned. It determined the most likely parts (shown numbered) and the most likely labels for the box-arrow data in Figure 3 using two parts per class. One arrow head was assigned part 1 (as well as some arrow heads in other diagrams), but most arrow heads and arrow stems were assigned to 2. The algorithm placed horizontal box sides into part 3 and vertical sides in 4. Next, we tried the flowchart data allowing two connector parts (1,2) and four container parts (3–6) in order to capture the differences among container types (Figure 4). Here, parts 5 and 6 turned

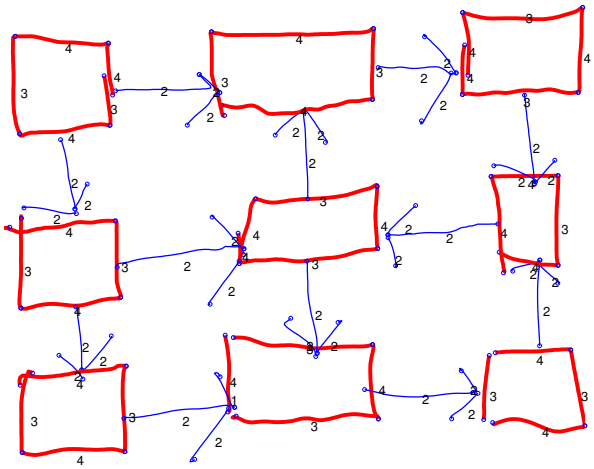

Figure 3. Discovered parts (numbered 1, 2, 3, 4) and labels (connectors and containers are thin and thick lines) in the box-arrow diagram from Figure 1.

out to be horizontal and vertical parts of rectangles and ellipses (ellipses had been fragmented into straight line segments). The algorithm put diagonal sides of diamonds into part 4, and other parts of ellipses in 3. The connector parts were less interpretable, consisting mostly of alternating 1s and 2s.

In general, the types of parts the algorithm discovers depends on the features we provide, and parts may not necessarily correspond to simple semantic parts. Our interaction features were symmetric, but would need to be directed in order to learn asymmetric relations such as 'to-the-right-of' or 'above'.

6. Conclusion

We have proposed a general method that learns parts of compound objects in an unsupervised way. This model is powerful and has many potential applications in handwriting recognition, document analysis, and object recognition, and opens several opportunities for further research. For example, it would be desirable to develop a criterion to automatically determine the number of parts per object class. Moreover, the number of parameters grows quickly as more parts are introduced, necessitating more parameter sharing and regularization to avoid overfitting. Parts may also be learned more easily with partial supervision (labeling some of the parts in addition to labeling the compound objects). Finally, we need more experiments to determine the complexity of parts that can be learned.

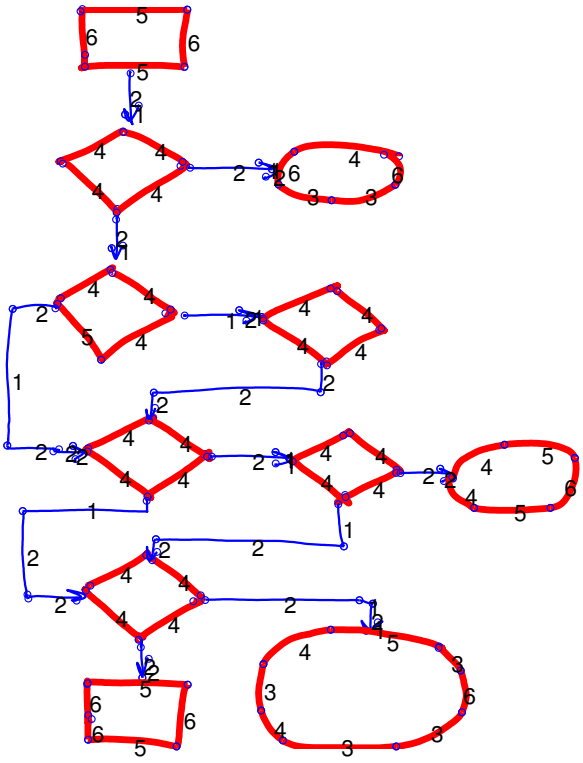

Figure 4. Discovered parts (numbered 1–6) and labels (thin and thick lines) in a flowchart. Small circles indicate fragment ends.

Acknowledgement

I wish to thank Thomas Minka for valuable discussions and for inference software.

References

[1] C. Alvarado, M. Oltmans, and R. Davis. A framework for multi-domain sketch recognition. In *AAAI Spring Symposium on Sketch Understanding*, pages 1–8, Stanford, CA, Mar. 2002. AAAI Press.

[2] S. Cho and J. Kim. Bayesian network modeling of Hangul characters for on-line handwriting recognition. In *Intl. Conf. Document Analysis and Recognition (ICDAR)*, pages 207–211, 2003.

[3] F. Jensen. *Bayesian Networks And Decision Graphs*. Springer, 2001.

[4] S. Kakade, Y. Teh, and S. Roweis. An alternate objective function for Markovian fields. In *Intl. Conf. Machine Learning (ICML)*, pages 275–282, 2002.

[5] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Intl. Conf. Machine Learning (ICML)*, pages 282–289, 2001.

[6] S. Marukatat, T. Artières, and P. Gallinari. A generic approach for on-line handwriting recognition. In *9th Intl. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2004.

[7] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *Neural Information Processing Systems (NIPS)*, volume 17, 2004.

[8] M. Szummer and Y. Qi. Contextual recognition of hand-drawn diagrams with conditional random fields. In F. Kimura and H. Fujisawa, editors, *9th Intl. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 32–37. IEEE, 2004.

[9] M. Ye and P. Viola. Learning to parse hierarchical lists and outlines using conditional random fields. In *9th Intl. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2004.

[10] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Trans. Medical Imaging*, 20(1):45–57, 2001.