Incorporating Context and User Feedback in Pen-Based Interfaces *

Martin Szummer and **Philip J. Cowans**†
Microsoft Research
Cambridge, CB3 0FB, UK
szummer@microsoft.com, pjc51@cam.ac.uk

Abstract

We propose a joint probabilistic model for grouping and labeling hand-drawn ink strokes. We demonstrate that simultaneous grouping and labeling yields superior accuracy to labeling alone. Our probabilistic formulation has many advantages, exact inference is feasible, and we obtain confidence estimates. We show how to incorporate user feedback by conditioning our model, and discuss different types of inference tasks suited for various user interactions.

Introduction

The recent growth in the availability of pen-based input devices, notably in Tablet PCs, Personal Digital Assistants (PDAs) and mobile telephones, has increased the importance of algorithms which process hand-drawn input. One task which falls into this category is that of interpreting text and drawings. In order to perform this task, it is desirable to be able to group ink strokes into perceptually salient objects and label the objects according to their type. These tasks are highly context-dependent; it is difficult to classify a single ink stroke without considering the rest of the drawing.

To improve accuracy in these difficult tasks, systems should exploit user feedback and interaction to update grouping and labeling results. In this paper we develop a general approach that both exploits context and incorporates user feedback to perform grouping and labeling at the same time. When the user corrects errors made by the system, the drawing is reinterpreted to incorporate the new information. We consider explicit feedback such as user corrections to recognition and grouping, but also implicit information gained from operations such as copying and pasting.

We show the significant benefits of using a joint probabilistic model for grouping and labeling ink. Joint modelling considers all ink together, and is a natural way of incorporating context, promising accurate recognition. We also show that accuracy is improved by simultaneous grouping and labeling. Our model is formulated in a probabilistic framework, which allows us to ask the most appropriate questions for the task at hand.

Previously work addressing the issue of perceptual grouping of ink (Saund 2003) has not been formulated within a probabilistic framework. Other work (Mankoff 2001) has discussed the use of uncertainty in user interfaces in a more general context. Conditional probabilistic models have been used for the related task of labeling images (Kumar & Hebert 2003), but their domain required approximate methods. Simultaneous segmentation and labeling has been performed using a generative model (Tu *et al.* 2003), but again it was necessary to resort to approximate inference, in this case using Monte Carlo methods. In contrast, we show how to feasibly employ exact inference in the ink domain.

In this work, we restrict ourselves to the task of interpreting organisational charts of the type shown in Figure 1(a). In this context, perceptual objects are either containers, representing organisational units, or connectors representing the relationships between them. However, our approach is general and may be applied to a wide variety of grouping and labeling tasks.

This paper consists of two parts. First we introduce the probabilistic model used to perform labeling and grouping. We then discuss the application of the model from the perspective of user-interaction, showing how it can be used to incorporate feedback into the interpretation process.

Joint Probabilistic Models

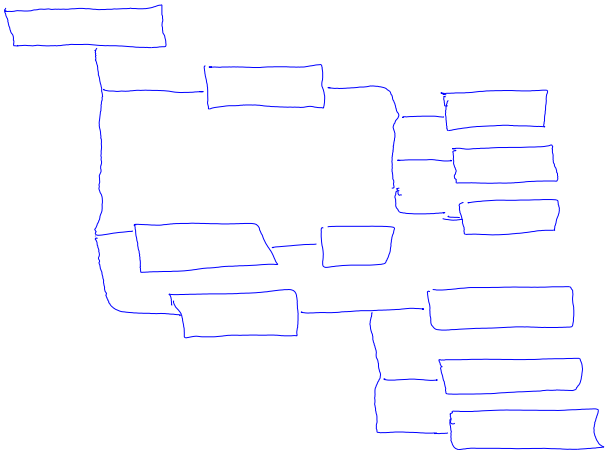
The input data available to our algorithm is a set of strokes, each represented by a set of sampled pen locations. A single stroke is defined as the ink produced between subsequent pen-down and pen-up events. In general, strokes defined in this way are not sufficiently fine-grained for our needs; a single stroke may well span more than one object. The first stage is therefore to split the strokes into smaller fragments, which are assumed to belong to a single object. Fragmentation is performed by dividing each stroke into sections which are straight to within a given tolerance. Each fragment is therefore derived from a single stroke.

We exploit context by employing a probabilistic model defined jointly over all ink fragments,

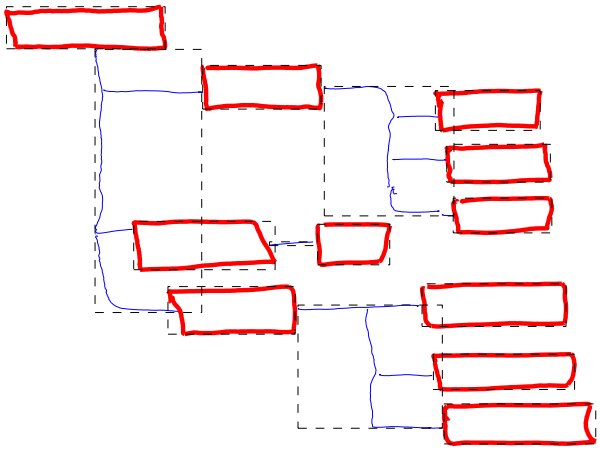
$$P(\mathcal{Y} \mid \mathbf{x}, \boldsymbol{\theta}) \quad (1)$$

*We are grateful to Thomas Minka and Michel Gangnet for valuable discussions and software for inference and ink processing respectively.

†Current address for correspondence: Department of Physics, University of Cambridge, Cavendish Laboratory, Cambridge, CB3 0HE, UK



(a)



(b)

Figure 1: An example of a typical drawing. (a) shows the original ink, (b) shows the resulting labeling and grouping of ink fragments after whole drawing parsing has been performed. Dark lines indicate containers and lighter lines are connectors. Dashed boxes enclose ink fragments which have been grouped as belonging to the same perceptual object.

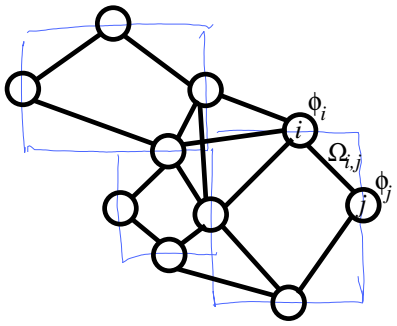


Figure 2: The graph $\mathcal{G}$ constructed from a set of ink fragments. Each vertex represents a fragment, and the edges represent interactions which will be included in the probabilistic model.

where $\mathbf{x}$ is a random vector representing the observed ink and $\boldsymbol{\theta}$ is a vector of model parameters. $\mathcal{Y} = (\mathbf{y}, \mathsf{S})$ represents the output grouping and labeling. $\mathcal{Y}$ consists of labels, $\mathbf{y} = \{y_i\} \in \{\pm 1\}^N$, denoting connector or container and a segmentation matrix S with entries $S_{ij} = 1$ if fragments i and j are in the same group and 0 otherwise. The label and group of each fragment depends on its associated ink as well as on the labels, groups and ink of its spatial and temporal neighbors.

The exact form of the model is similar to the Conditional Random Field (CRF) (Lafferty, McCallum, & Pereira 2001), which is a discriminative classifier using a Markov Random Field (MRF) (Jensen 2001) whose potentials are dependent on the observed ink. The model makes use of an undirected graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, containing one vertex for each ink fragment (see Figure 2(b)). This graph represents the dependencies between ink fragments; for each pair of fragments connected by an edge, an explicit interaction will be included in the model. However, by transitivity implicit dependencies between fragments which are not directly connected will be induced. The graph will also affect the set of possible groupings; groupings will only be considered which are contiguous on $\mathcal{G}$, that is, for any pair of fragments which are in the same group, there must be a path between their associated vertices in $\mathcal{G}$ passing only through vertices which are also in the same group.

Ideally, we would use the fully connected graph so as to incorporate all possible correlations between fragments. Unfortunately such a graph results in computations which are intractable, so a less dense graph must be used as an approximation. The graph must be *triangulated*, that is it must not contain any cycles of length greater than three which are not spanned by a chord. For such graphs, an important quantity is the *maximum clique size*, where a clique is a fully connected subgraph. The computations described below have complexities which scale with this quantity, so it is necessary to restrict ourselves to graphs whose maximum clique size is relatively small. Malvestuto (1991) provides a greedy heuristic for constructing such graphs, whereas Karger and Srebro (2001) provide a more theoretically justified global algorithm.

In this work, we use a simpler approach. In our algorithm, $\mathcal{G}$ is constructed by first producing a candidate graph (which is not necessarily triangulated) constructed by connecting all pairs of fragments which are closer than a maximum threshold distance. Triangulation is then performed, to find the final form for $\mathcal{G}$. Note that interactions are considered for all edges, even those added during triangulation. While our method does not give any guarantees about the maximum clique size of the resulting graph, in general reasonable clique sizes are produced. On our training database, the mean maximum clique size is 5.0. Practically, we found that maximum clique sizes up to 6 could be feasibly handled by our MATLAB implementation of the algorithm.

Our model is defined as a product of potential functions. We employ two types of potentials: observation potentials $\Phi_i(\mathcal{Y}, \mathbf{x}; \boldsymbol{\theta})$ measure the compatibility of one label at node i with its associated ink and interaction potentials $\Omega_{i,j}(\mathcal{Y}, \mathbf{x}; \boldsymbol{\theta})$ measure the compatibility between neighboring groups and labels, depending on their associated ink. We include one observation potential for each ink fragment, and one interaction potential for each pair of fragments connected by an edge in $\mathcal{G}$. Formally, the model assigns probabilities to a joint grouping and labeling according to

$$P(\mathcal{Y} \mid \mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\mathbf{x}, \boldsymbol{\theta})} \prod_{i \in \mathcal{V}} \Phi_i(\mathcal{Y}, \mathbf{x}; \boldsymbol{\theta}) \prod_{(i,j) \in \mathcal{E}} \Omega_{i,j}(\mathcal{Y}, \mathbf{x}; \boldsymbol{\theta}). \tag{2}$$

The constant $Z(\mathbf{x}, \boldsymbol{\theta})$ is required to ensure that the distribution sums to one. Specifically,

$$Z(\mathbf{x}, \boldsymbol{\theta}) = \sum_{\mathcal{Y}'} \prod_{i \in \mathcal{V}} \Phi_i(\mathcal{Y}', \mathbf{x}; \boldsymbol{\theta}) \prod_{(i,j) \in \mathcal{E}} \Omega_{i,j}(\mathcal{Y}', \mathbf{x}; \boldsymbol{\theta}). \tag{3}$$

While at first glance this summation seems difficult to compute owing to the very large number of possible configurations included, it is possible to perform the computation efficiently by exploiting the structure of $\mathcal{G}$ with a *message-passing* algorithm known as a *sum-product* algorithm (Cowans 2004). This algorithm calculates the sum by passing messages between sets of vertices on $\mathcal{G}$ representing the results of local computations.

Rather than working directly with the raw ink data, we define a number of observation and interaction features, $\mathbf{f}_i(\mathbf{x})$ and $\mathbf{g}_{ij}(\mathbf{x})$ for each vertex and edge respectively. The choice of features will be discussed below. The form of the observation potentials is

$$\Phi_i(\mathcal{Y}, \mathbf{x}; \boldsymbol{\theta}) = \exp(y_i \mathbf{w}^{\mathrm{T}} \mathbf{g}_i(\mathbf{x})), \tag{4}$$

and the interaction potentials are given by

$$\Omega_{i,j}(y_i, y_j, \mathbf{x}; \boldsymbol{\theta}) = \begin{cases} \exp(\mathbf{v}_{\mathrm{ss}}^{\mathrm{T}} \mathbf{f}_{ij}(\mathbf{x})) & \mathsf{S}_{ij} = 1 \\ \exp(\mathbf{v}_{\mathrm{sd}}^{\mathrm{T}} \mathbf{f}_{ij}(\mathbf{x})) & \mathsf{S}_{ij} = 0, y_i = y_j \\ \exp(\mathbf{v}_{\mathrm{dd}}^{\mathrm{T}} \mathbf{f}_{ij}(\mathbf{x})) & \mathsf{S}_{ij} = 0, y_i \neq y_j. \end{cases} \tag{5}$$

The parameters are $\theta = (\mathbf{w}, \mathbf{v}_{\mathrm{ss}}, \mathbf{v}_{\mathrm{sd}}, \mathrm{and} \mathbf{v}_{\mathrm{dd}})$. Different parameters $\mathbf{v}_{\mathrm{ss}}$, $\mathbf{v}_{\mathrm{sd}}$, and $\mathbf{v}_{\mathrm{dd}}$ are used depending on whether fragments i and j have the same label or not, and belong to the same group or not. There are only three cases, since fragments in the same group must have the same label, and fragments with different labels must lie in different groups.

No.	Description
1–11	Length
12–20	Angle
21–29	Angles of neighbors
30–40	Relative angles of neighbors
41–51	Distances to neighbors
52	Long fragment indicator
53	Full container template
54–55	T-junction count
56	Neighboring full container template
57	Container edge template
58–60	Container start template
61	Bias dummy feature

Table 1: Observation features

No.	Description
1–11	Angles
12–22	Distance
23–28	Stroke ordering
29	T-junction count
30	Full container template
31	Container side template
32	Container bottom template
33	Neighboring container template
34	Similar direction indicator
35	Fragment alignment indicator
36	Right-angle indicator
37	Bias dummy feature

Table 2: Interaction features

Training

Rather than manually specifying the parameters $\boldsymbol{\theta}$, the probabilistic framework gives us a principled way to learn them from example user drawings with known ground-truth labeling and grouping, $\{\tilde{\mathbf{x}}, \tilde{\mathcal{Y}}\}$. We find the *maximum a posteriori* (MAP) parameter values using Bayes' rule:

$$\boldsymbol{\theta}^{MAP} = \arg\max_{\boldsymbol{\theta}} \left(P\left(\tilde{\mathcal{Y}} \mid \tilde{\mathbf{x}}, \boldsymbol{\theta}\right) \cdot P\left(\boldsymbol{\theta}\right) \right) \tag{6}$$

A gradient ascent algorithm is used to find the parameter values which maximise this probability. $P(\boldsymbol{\theta})$ is a prior distribution over parameter values, which will be discussed below.

Features and priors

We chose features to reflect the spatial and temporal distribution of ink strokes, for example lengths and angles of fragments, whether two fragments were drawn with a single stroke, and the temporal ordering of strokes. We also used a number of 'template' features which were designed to capture important higher level aspects of the ink, such as the presence of T-junctions. A full list of features is given in Tables 1 and 2.

The features fall into three categories: (1) binary features which describe the presence or absence of a particular prop-

erty, for example whether two fragments were drawn in the same stroke, (2) counting features which represent the number of times that a particular event occurs in association with the related ink, for example, the number of T-junctions amongst the neighbors, and (3) continuous features which represent a continuous variable, for example, the angle between two fragments.

While the first two categories may be used directly as features, the third is not well-suited to this purpose as there is not necessarily a linear relationship between the raw feature value and the 'degree of fit' of a particular label or grouping. Instead, a set of histogram features is used, each of which is triggered when the raw value lies within a given range. As we expect the degree of fit to vary relatively smoothly with the raw value, it is appropriate to use a correlated prior on the weights of these features. We choose a non-diagonal Gaussian prior for these weights, with a covariance matrix Σ having components

$$\Sigma_{ij} = \sigma^2 \cdot \exp\left(-\frac{\mathcal{D}\left(x_i, x_j\right)^2}{r^2}\right) \tag{7}$$

corresponding to the i^{th} and j^{th} histogram bins. x_i and x_j are the raw feature values corresponding to the centres of the bins, and $\mathcal{D}\left(x_i, x_j\right)$ is a measure of the distance between them. Here we assume this is linear in $|x_i - x_j|$. Binary and counting features are given independent Gaussian priors with standard deviation σ.

Inference

Having trained the model, it can now be used to make inferences concerning newly observed drawings. The exact nature of the inferences required depends on the task at hand. Several possible user-interaction scenarios are outlined below.

Whole-drawing Parsing

Given a set of ink fragments, one possible task is to fully interpret the whole drawing. This process involves making a 'hard' decision concerning the grouping and labeling of all fragments. Typically, this is necessary in order to perform tasks such as beautification; the conversion of the hand drawn drawing to a machine generated equivalent.

Mathematically speaking, this task involves finding the most probable joint labeling and segmentation:

$$\mathcal{Y}^{\text{MP}} = \text{argmax}_{\mathcal{Y}} P(\mathcal{Y} \mid \mathbf{x}, \boldsymbol{\theta}) \tag{8}$$

This can be found efficiently using the *max-product* algorithm (Cowans 2004); a message passing technique similar to the sum-product algorithm used during training. An example of the typical output from this process is shown in Figure 1(b).

To test the performance of the model on this task, we collected a set of 40 example drawings, consisting of a total of 2157 ink fragments. Three random splits were generated, each consisting of 20 drawings used for training and 20 used for evaluation. Training was performed by finding the MAP weights as described above. A BFGS optimiser was used to perform the maximisation.

Model	Labeling Error			
	1	2	3	Mean
L	8.2%	9.3%	7.9%	8.5%
LI	4.6%	6.8%	1.9%	4.5%
% Δ LI/L	−43.9%	−26.9%	−75.9%	−48.9% ±24.9%
GLI	3.1%	3.8%	0.8%	2.6%
% Δ GLI/LI	−33%	−44%	−58%	−42% ±8%

Table 3: Labeling errors for the three models. Results are given separately for the three cross-validation splits. Relative differences are shown between models L and LI, and between LI and GLI. The mean relative differences are aggregations of the differences for each split, rather than the differences between the means for individual models. This is to reduce the effect of systematic variation between splits.

Model	Grouping Error			
	1	2	3	Mean
GLI	***28***	***49***	***30***	***36***

Table 4: Grouping errors for the GLI model.

We compared three versions of the model. The simplest version, L, does not include interaction potentials, so decisions are made locally without the explicit use of contextual information (although some contextual information is incorporated through the choice of features). This model is not capable of performing grouping. Version LI includes interaction potentials, but only to differentiate between two fragments having the same label or a different label; no grouping is performed. Finally, version GLI is the full model as described above, with grouping and labeling being performed simultaneously.

Results of these experiments are shown in Tables 3 and 4. Labeling errors are quoted simply as the fraction of fragments incorrectly labeled. To evaluate grouping results we used a modified version of the error metric used by Martin *et. al.* (2001). The error for the i^{th} fragment is

$$E_i\left(\mathsf{S}_1, \mathsf{S}_2\right) = \frac{\left|R_i\left(\mathsf{S}_1\right) \setminus R_i\left(\mathsf{S}_2\right)\right| + \left|R_i\left(\mathsf{S}_2\right) \setminus R_i\left(\mathsf{S}_1\right)\right|}{\left|R_i\left(\mathsf{S}_1\right)\right|}, \tag{9}$$

where $R_i\left(\mathsf{S}\right)$ is the set of all fragments in the same group as i, S_1 is the ground-truth segmentation and S_2 is the segmentation being evaluated. Summing over all fragments gives the overall error

$$E\left(\mathsf{S}_1, \mathsf{S}_2\right) = \sum_i E_i\left(\mathsf{S}_1, \mathsf{S}_2\right) \tag{10}$$

Detailed results for all drawings are available online[1].

These results indicate that good performance for both labeling and grouping can be achieved. The improvement in

[1] `http://research.microsoft.com/~szummer/aaai04/`

labeling between the L and LI models indicates that the incorporation of explicit contextual information through the use of interaction features is important, as expected. The fact that a further improvement in labeling performance is seen between the LI and GLI models shows that simultaneously grouping can improve labeling performance too.

Estimation Of Uncertainties

An advantage of the probabilistic framework is that the probability of making errors during a hard decision can be estimated. This information can be used to prompt the user for clarification if the probability of error is too high. Estimation of the probability that a given fragment has been incorrectly labeled can be obtained by computing marginal probabilities, found by summing over all configurations consistent with a particular labeling

$$P\left(y_i = \hat{y} \mid \mathbf{x}, \boldsymbol{\theta}\right) = \sum_{\boldsymbol{y}_{\setminus i}, \mathsf{S}} P\left(y_i = \hat{y}, \boldsymbol{y}_{\setminus i}, \mathsf{S} \mid \mathbf{x}, \boldsymbol{\theta}\right) \quad (11)$$

Where $\boldsymbol{y}_{\setminus i}$ is a vector representing the labels of all fragments other than the i^{th}. As before, this sum can be efficiently computed using the sum-product algorithm. Having computed this, the estimated probability of error is

$$P^{\text{err}} \approx 1 - P\left(y_i = y^{\text{MP}} \mid \mathbf{x}, \boldsymbol{\theta}\right) \quad (12)$$

where y^{MP} is the label assigned during the hard decision process. Figure 3 shows these probabilities for a typical drawing. A similar process is possible for groupings, in which case it is appropriate to calculate the marginal probabilities for each adjacent pair of fragments on $\mathcal{G}$ being in the same group or separate groups, and compare with the most probable configuration.

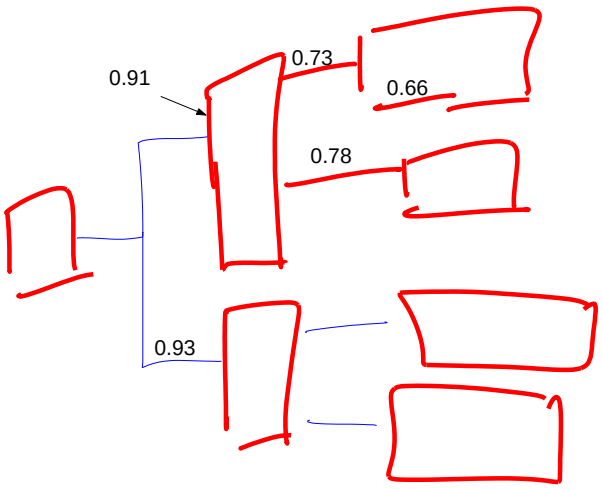
There are a number of ways of selecting a region of the drawing in which to ask the user for clarification. One possible choice is to simply ask for clarification in the region where there is greatest uncertainty. While straightforward, this may not be the optimal method, and several other methods have been proposed in the field of active learning (Cohn, Ghahramani, & Jordan 1995).

Local Parsing

Although whole drawing parsing can be efficiently performed, it is our belief that probabilistic interpretation of drawings should adhere to the following two principles.

1. Hard decisions should be made as late as possible.
2. Where possible, hard decisions should be presented locally rather than globally.

In other words, rather than attempting to make a hard decision as soon as ink is available, the presentation of the ink to the user should remain in as close as possible to the original form until the user explicitly selects an operation (such as beautification) which requires a hard decision to be made. In the meantime, the user is able to edit the raw ink, assisted by temporary, local decisions made by the system. During this process, implicit feedback can be collected which may improve later decisions.



(a)

Figure 3: Result of labeling by marginalizing groupings (11). Estimated confidence (probability of correct recognition) when making hard labeling decisions after whole drawing parsing. The 5 lowest confidences are shown. The second and third lowest-confidence predictions (0.73 and 0.78) do coincide with actual labeling errors (connectors mislabeled as containers). The system can exploit confidence estimates to selectively seek confirmation from the user.

In general, local parsing requires different inferences to processing the whole drawing. The following are two examples of scenarios where this is the case:

1. **Local labeling:** In some cases it is desirable to find the most probable label for a given fragment while marginalising out the labeling of other fragments, as well as the grouping of the whole drawing. For example, if the user makes a change to the properties of a fragment, and wishes to propagate those changes to all other fragments with the same label. In this case, the appropriate inference is to find the marginal probabilities over labels for the selected fragment, and to choose the label for which this value is highest. This does not necessarily give the same result as that obtained from whole-drawing parsing.
2. **Object selection:** The user will often want to manipulate perceptual objects, for example to move or resize them, or to change properties such as colors and line styles. The user indicates the object to be manipulated by selecting one of the component fragments, at which point the whole object is identified and highlighted. In order to identify the set of fragments to be highlighted the following inference task must be performed: find the most probable set of fragments which belong to the same object as the fragment which was selected. This is not the same as the task of finding the most probable global configuration and identifying the perceptual object to which the selected fragment belongs in that case; the former sums over all possible configurations of the fragments not in the selected object, as well as the label of the object which is selected. A branch-and-bound search algorithm may be

suitable for efficiently performing this computation.

User Feedback

Although it is of course desirable to produce a system which is able to perfectly group and label fragments, in practice errors are made. Fortunately, the probabilistic framework allows corrections to be made in a principled way. Suppose that system's initial interpretation was in fact wrong. Explicit feedback can be obtained by the user making corrections, for example, indicating that a particular fragment has been incorrectly included in the target object, or has been assigned the wrong label. Such interactions can be interpreted in terms of constraints and allow the initial interpretation to be re-estimated using conditional probabilities. An advantage of this approach is that correction of a single fragment may result in changes to many other fragments. An example of this is shown in Figure 4.

Further feedback may be gathered implicitly, for example, if having selected an object, the user manipulates that object (moving it, changing the object's properties, performing copy and paste activities and so on), it is possible to interpret this as evidence that the initial interpretation was correct. This information can also be incorporated into future inferences.

Constraints can be placed either for individual fragments (stating that they must have a particular label) or for pairs of fragments (stating, for example, that they must belong to the same object). Constraints involving larger groups can be expressed in this way, for example the statement that a set of three or more fragments must be in the same object can be expressed in terms of pairwise constraints between all pairs within the set (many of which are redundant).

A list of supplied constraints, $\mathcal{C}$, can be maintained by the system, and can be included in future decisions through the use of conditional probabilities. In other words, once the user corrects errors in an interpretation of the drawing, a new interpretation can be generated by finding the most probable configuration conditioned on the constraints being satisfied:

$$\mathcal{Y}^{\text{MAP}} = \text{argmax}_{\mathcal{Y}} P(\mathcal{Y} \mid \mathbf{x}, \boldsymbol{\theta}, \mathcal{C}) \tag{13}$$

The constraint can be inserted as additional terms in the model, in other words:

$$P(\mathcal{Y} \mid \mathbf{x}, \boldsymbol{\theta}, \mathcal{C}) = \frac{1}{Z(\mathbf{x}, \boldsymbol{\theta})} \prod_{i \in \mathcal{V}} \Phi_i(\mathcal{Y}, \mathbf{x}; \boldsymbol{\theta}) \prod_{(i,j) \in \mathcal{E}} \Omega_{i,j}(\mathcal{Y}, \mathbf{x}; \boldsymbol{\theta}) \prod_{c_i \in \mathcal{C}} \delta_{c_i}(\mathcal{Y}), \tag{14}$$

where

$$\delta_{c_i}(\mathcal{Y}) = \begin{cases} 1 & c_i \text{ is satisfied in } \mathcal{Y} \\ 0 & c_i \text{ is violated in } \mathcal{Y}. \end{cases}$$

Unary constraints, and pairwise constraints between fragments connected by an edge in $\mathcal{G}$ can be inserted into the model without changing the complexity of inference. If the constraint involves fragments which are not already connected, an extra edge must be inserted into $\mathcal{G}$ to accommodate this. This necessitates re-triangulation, potentially increasing the maximal clique size and therefore may increase the complexity of inference significantly. In this case it may be necessary to remove other edges to keep the clique size at an acceptable level, or to request a different constraint from the user.

In general it is not necessary to work with 'hard' constraints; if the information supplied by the user is 'soft' in nature, evidence stating that configurations satisfying a particular constraining are simply more probable can also be used. In this case the new potentials can be replaced by

$$\hat{\delta}_{c_i}(\mathcal{Y}) = \begin{cases} 1 & c_i \text{ is satisfied in } \mathcal{Y} \\ \xi & c_i \text{ is violated in } \mathcal{Y}. \end{cases}$$

Where ξ is a small, but non-zero constant specifying the 'strength' of the constraint, with smaller values being 'harder'. In practice it may be necessary to do this in all cases to allow for the case where the user specifies inconsistent constraints. It may also be desirable to use this technique to assign greater importance to more recently supplied constraints, or to reduce the importance of constraints given before new ink was obtained.

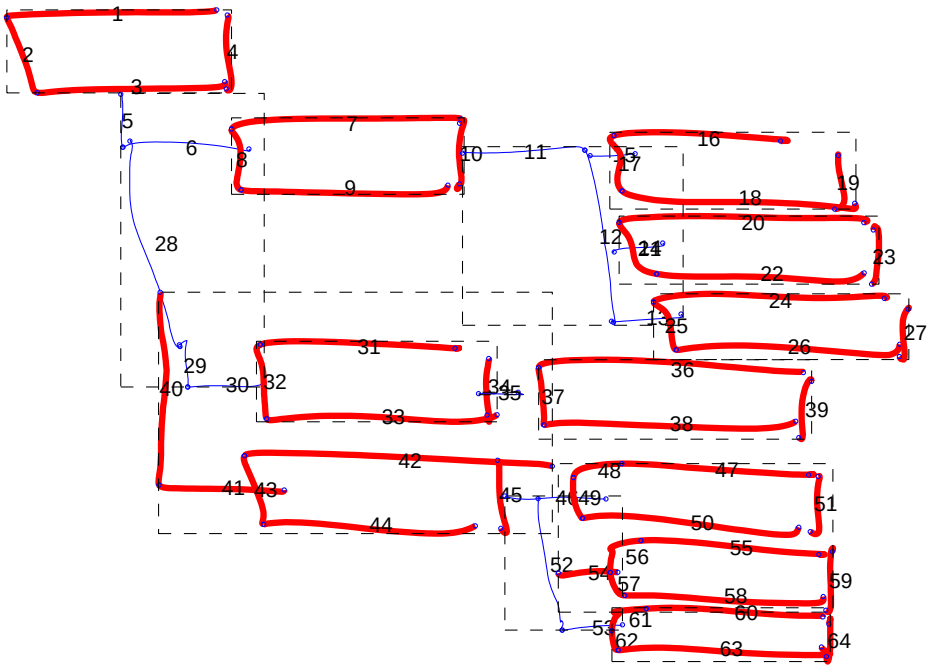
Conclusions And Further Work

We have described a probabilistic model for simultaneously grouping and labeling ink fragments. Experimental results show that the model performs well on the task of interpreting organisational charts consisting of container and connectors. In particular, we have demonstrated that using a joint model to include contextual information improves performance, and that simultaneous labeling and grouping gives increased accuracy even if only the labelings are considered.

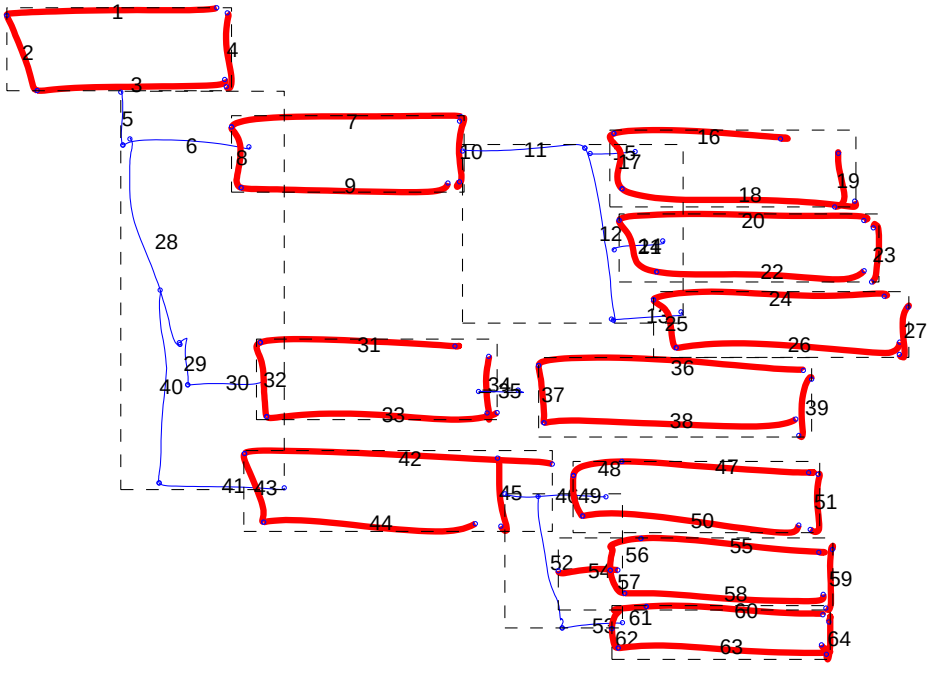
We have also demonstrated that the use of a probabilistic framework provides a number of advantages from a user interface perspective. In particular, it is possible to estimate the probability of errors during the interpretation task, which can be used to prompt the user for feedback. When feedback is obtained, it is possible to incorporate it in a principled way by using constraints.

Rather than asking the user to make specific corrections, an alternative approach is to present the user with a choice of possible interpretations. Nilsson (Nilsson 1998) has presented an efficient algorithm to find the N most probable configurations which is suitable for use in the case where we are only interested in labeling. In the future we would like to investigate the use of these techniques.

In general it may be desirable to use 'factorised' N-most-probable lists. For example, if there are two distinct areas of the drawing in which the interpretation is uncertain, it is better to provide separate lists for the separate areas rather than asking the user to select from a list of all possible combinations. Finding possible factorisations is another area for further work.



(a)



(b)

Figure 4: An example of incorporating user feedback. (a) shows the most probable joint grouping and labeling without feedback, showing three mislabeled fragments (40, 41, 54), as well as incorrect groupings (two containers grouped together at the lower right). The system is the least confident about the labelings of exactly the mislabeled fragments. (b) shows the results after the user indicated that fragments 41 and 43 should belong to different groups, as should fragments 50 and 55. This feedback results in correct grouping and labeling of fragments 40 and 41, as well as correct grouping of the two containers. Fragment 54 is still mislabeled, but can be corrected with one additional step of feedback (either by labeling it as a connector, or by indicating that it does not belong to the same object as fragment 57).

References

[Cohn, Ghahramani, & Jordan 1995] Cohn, D. A.; Ghahramani, Z.; and Jordan, M. I. 1995. Active learning with statistical models. In Tesauro, G.; Touretzky, D.; and Leen, T., eds., *Advances in Neural Information Processing Systems*, volume 7, 705–712. The MIT Press.

[Cowans 2004] Cowans, P. J. 2004. The sum-product and max-product algorithms for graph partitions. Technical report, Microsoft Research Cambridge. `http://research.microsoft.com/~szummer/aaai04/`.

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

[Kumar & Hebert 2003] Kumar, S., and Hebert, M. 2003. Discriminative fields for modeling spatial dependencies in natural images. In *Neural Information Processing Systems (NIPS)*.

[Lafferty, McCallum, & Pereira 2001] Lafferty, J.; McCallum, A.; and Pereira, F. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 282–289.

[Mankoff 2001] Mankoff, J. 2001. *An Architecture and Interaction Techniques for Handling Ambiguity in Recognition-based Input*. Ph.D. Dissertation, Georgia Inst. of Tech.

[Nilsson 1998] Nilsson, D. 1998. An efficient algorithm for finding the m most probable configurationsin probabilistic expert systems. *Statistics and Computing* 8(2):159–173.

[Saund 2003] Saund, E. 2003. Finding perceptually closed paths in sketches and drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(4):475–491.

[Tu *et al.* 2003] Tu, Z.; Chen, X.; Yuille, A. L.; and Zhu, S. C. 2003. Image parsing: Unifying segmentation, detection, and recognition. In *International Conference on Computer Vision*.