Mathematical symbol recognition in the MathBrush system

Scott MacLean
smaclean@cs.uwaterloo.ca

Symbolic Computation Group
University of Waterloo

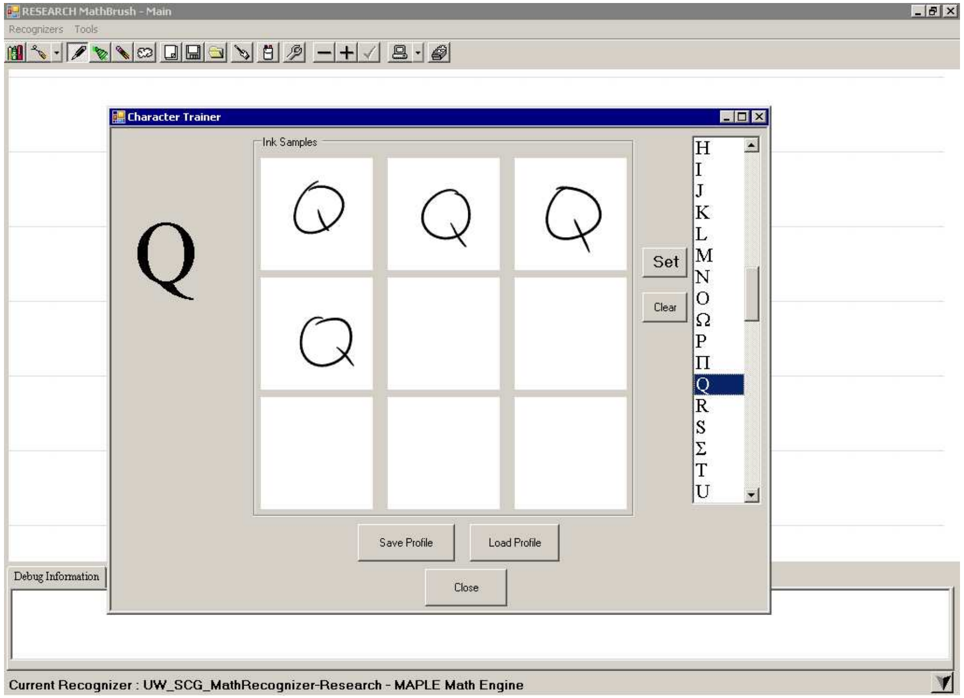
CMS-MITACS Joint Conference 2007

The MathBrush project

- Investigating pen-based interactive mathematics
- Goals:
  - Investigate issues particular to pen-based math by creating a functional prototype
  - Build a pluggable platform for future work in this area
- Components:
  - Symbol recognizer
  - Structural analyzer
  - Matrix analyzer
  - Pretty printer
  - User interface

Requirements

- Accurate
- Reasonably fast (wall-clock time)
- Easily and quickly customizable
  - Model-based matching
  - Training interface in MathBrush UI



Issues in recognition

- Difficulties particular to math recognition
  - Two-dimensional structure
  - Large variations in symbol scale
  - Many more symbol classes
  - No dictionary of all math expressions
  - Peculiar symbol arrangements, eg.

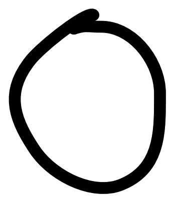
$$\sum_{i=1}^{t} x_1^{e_{i1}} x_2^{e_{i2}} \cdots x_l^{e_{il}}$$

Issues in recognition

- Some shared difficulties:
  - Indistinguishable symbols: 0/o/O, 1/l, 2/z/Z, x/X, etc.

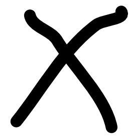
Issues in recognition

- Some shared difficulties:
  - Indistinguishable symbols: 0/o/O, 1/l, 2/z/Z, x/X, etc.



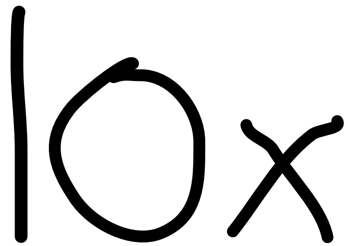
Issues in recognition

- Some shared difficulties:
  - Indistinguishable symbols: 0/o/O, 1/l, 2/z/Z, x/X, etc.



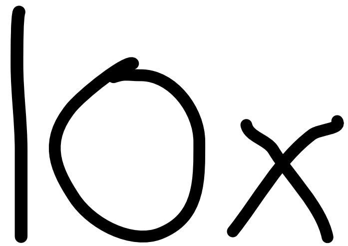
Issues in recognition

- Some shared difficulties:
  - Indistinguishable symbols: 0/o/O, 1/l, 2/z/Z, x/X, etc.

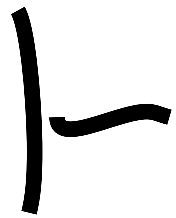


Issues in recognition

- Some shared difficulties:
  - Indistinguishable symbols: 0/o/O, 1/l, 2/z/Z, x/X, etc.



  - Ambiguous grouping: $1-1$ or H?



Recognition in MathBrush

- Input: A sequence of strokes drawn by the user, containing (x,y)-coordinates and timing information.
- Output: A number of bounding boxes, with a list of candidate symbols and confidence values for each.
- Several submodules
  - Preprocessing
  - Matching
  - Grouping
  - Centralized control of the above

Matching

- Problem: Given a group of input strokes and a collection of model symbols, find the n models most resembling the input, ranked by confidence
- General approach:
  - For each model symbol, measure “distance” between the input and that model
  - Rank the models by increasing distance
- Five algorithms, five notions of distance
  - Elastic matching (Tappert)
  - Stroke features
  - Chain code vectors (Chan & Yeung)
  - Direction element method (Kanahori, Suzuki, et al)
  - Deformable templates (Revow, Williams, Hinton)

Elastic matching

- Based on Euclidean distance
- Some notation:
  - (x_i, y_i) : input points ($i = 1, \ldots, n$)
  - (u_i, v_i) : model points ($i = 1, \ldots, m$)
  - θ_i : tangent angle at (x_i, y_i)
  - ϕ_i : tangent angle at (u_i, v_i)
- Uses a pointwise distance function; we chose

$$\begin{aligned} d(i,j) &= \tfrac{2\pi}{N}\left(|x_y - u_j| + |y_i - v_j|\right) \\ &+ \min(|\theta_i - \phi_j|, 2\pi - |\theta_i - \phi_j|) \end{aligned}$$

where N is a constant from normalization.

Elastic matching

- Finds a function M from input points to model points minimizing

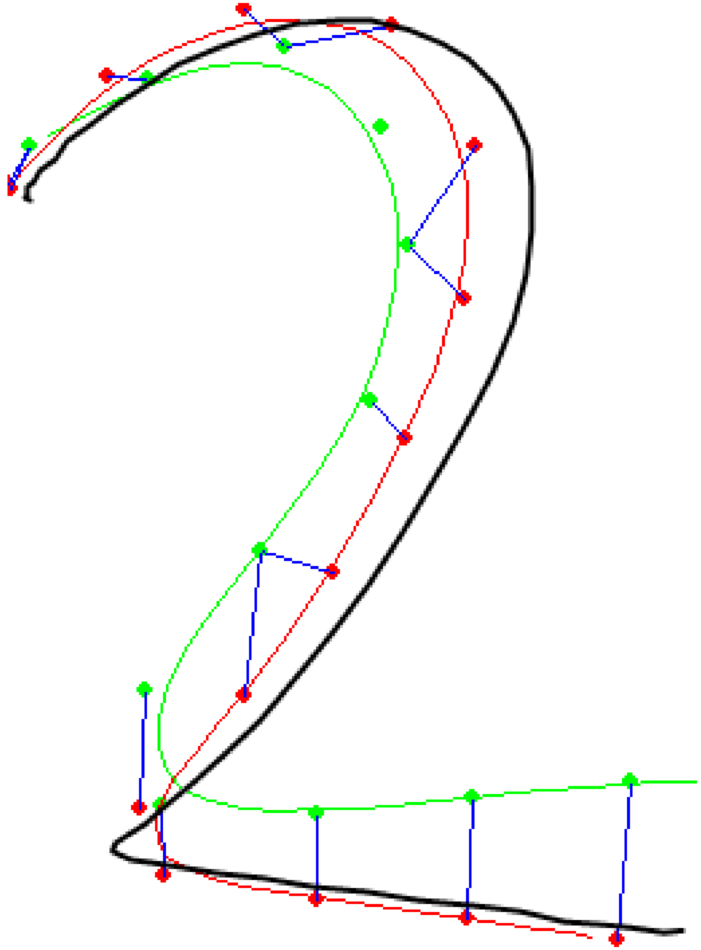
$$\sum_{i=1}^{n} d(i, M(i))$$

such that the following constraints are satisfied:

  - $M(1) = 1$
  - $M(n) = m$
  - If $M(i) = j$, then $M(i+1) \in \{j, j+1, j+2\}$
- Dynamic programming solution

Elastic matching

Example:



Stroke feature matching

- Extract salient features from each stroke
- Create feature vectors for input and model strokes, then take norm
- Features include:
  - topmost coordinate
  - leftmost coordinate
  - width and height
  - first and last points
  - total arclength

Matching

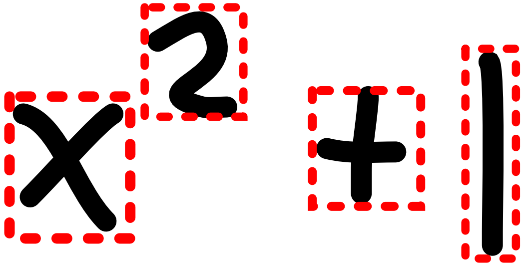
Experimental results for matching isolated symbols:

- A1 and A2 contain symbols from a single writer (625 symbols each)
- B contains symbols from multiple writers (742 symbols)
- C is the training set used in practice

Train/Test sets	Elastic		Features	
	Top1 (%)	Top5 (%)	Top1 (%)	Top5 (%)
A1/A2	92.2	99.8	68.2	93.9
A2/A1	92.8	100	68.5	94.4
B/C	88.3	98.5	54.9	87.7

Grouping

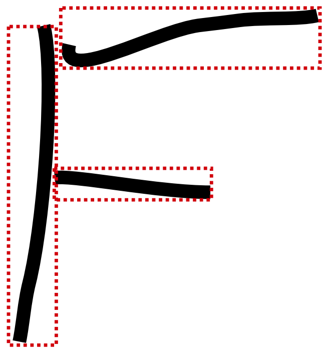
- Problem: Find the size of the next input symbol ("size" means number of strokes)



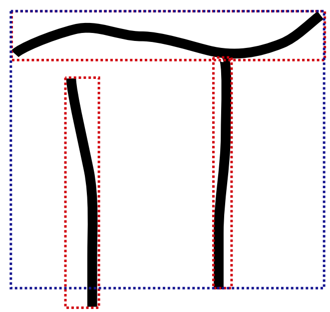
- Recognition fails with improper grouping
- Rule-based methods
- Recognition-based methods
- Time-based methods
- Selective combinations

Rule-based grouping

- Identify conditions in which strokes belong to the same symbol
  - Strokes are near to each other or touching, eg. k, ϕ
  - Bounding box alignment

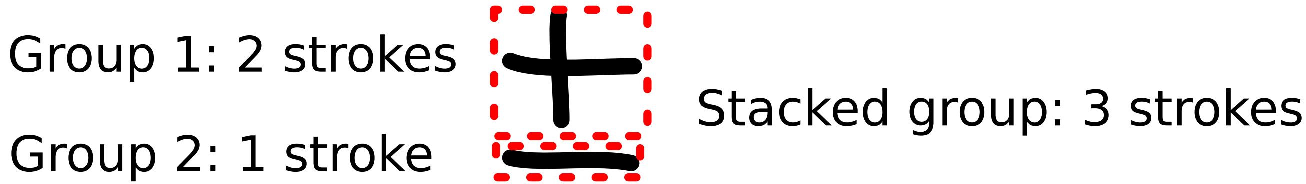


  - Bounding box containment



Rule-based grouping

- Tests fail for symbols like $\pm, \geq, \equiv$
- Many math symbols consist of several symbols stacked up vertically
- Use previous rules and detect stacked structure between groups



- Stacked structure: horizontal overlap & vertical proximity

Recognition-based grouping

- The stroke feature matcher is very fast and reasonably accurate
- Match against every database symbol and use the best match confidence for each symbol size
- Inherent bias toward small symbols
  - Some symbols have smaller symbols “embedded” in them, eg. D contains 1 and $)$
  - Measure how strong these resemblances are
  - Boost the confidence of matching a larger symbol if symbols embedded in it have high confidence

Time-based grouping

- Measure time between pen-up and pen-down
- Exponential distribution model for grouping strokes
- Determine distribution parameters as the user writes
- The user may pause at any time, even in the middle of a symbol
  - Time-based information may not reduce the group sizes produced by other methods, only increase them

Grouping

Experimental results for grouping: (These tests did not use time-based grouping)

- A1 and A2 contain symbols from a single writer (625 symbols each)
- B contains mathematical expressions from multiple writers (742 symbols)
- C is the training set used in practice

Train set	Test set	Rules (%)	Reco. (%)
A1	A2	98.6	90.7
A2	A1	99.4	92.8
C	B	98.9	87.2

System test results

- A1 and A2 contain symbols from a single writer
- B contains mathematical expressions from multiple writers
- C is the training set used in practice

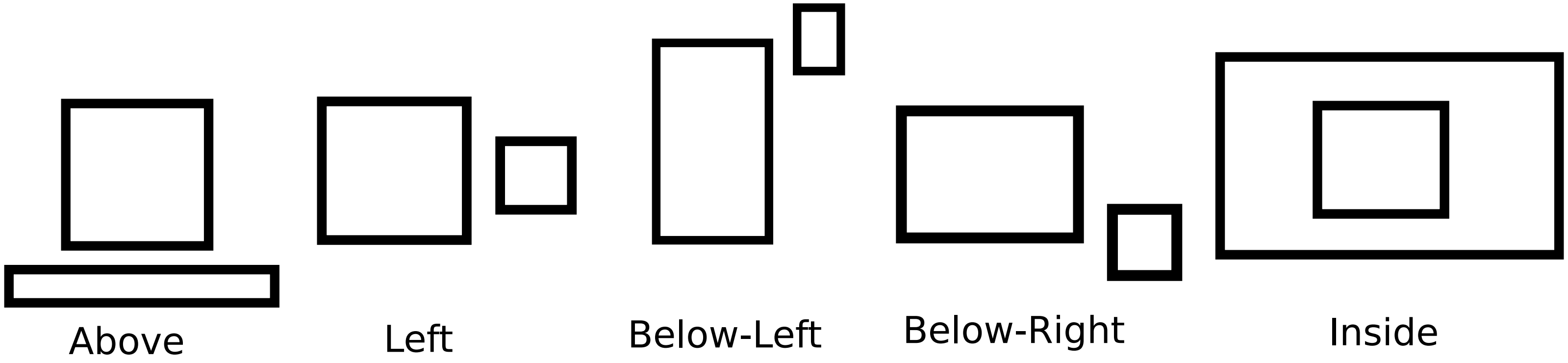
Train set	Test set	# Symbols	Top1 %	Top5 %
A1	A2	625	91.5	99
A2	A1	625	91.5	99.7
C	B	742	85	96.9

Current Issues

- Small symbols
  - Small symbols do not contain much data and are difficult to recognize
  - Disambiguating dots, commas, ticks, etc.
  - Some new recognition scheme may be required
- Disambiguating similar symbols
  - Recall o/O/0, 2/z/Z, etc.
- Often need knowledge of surrounding symbols or the mathematical context
- Idea: combine recognition and structural analysis
- Various approaches:
  - Bayesian belief networks, probabilistic modelling
  - Grammar-based approaches

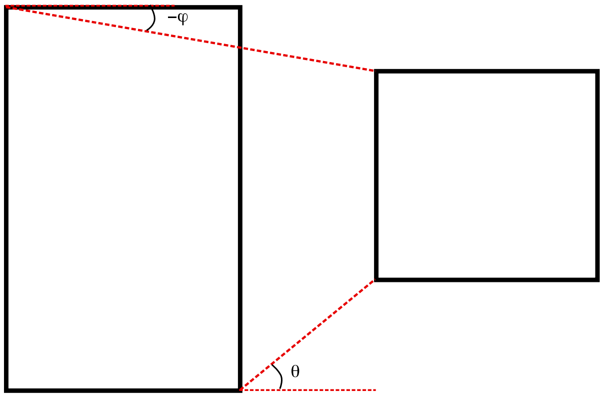
Grammar-based analysis

- Probabilistic, directional grammars
  - Similar to a CFG generating math expressions
  - Set of terminals is augmented by symbols representing directional links
  - Probabilities associated with productions
- A first step: obtaining probabilities for symbol relationships
  - Five possible link types: Left, Above, Above-Left, Below-Left, Inside



Classifying symbol links

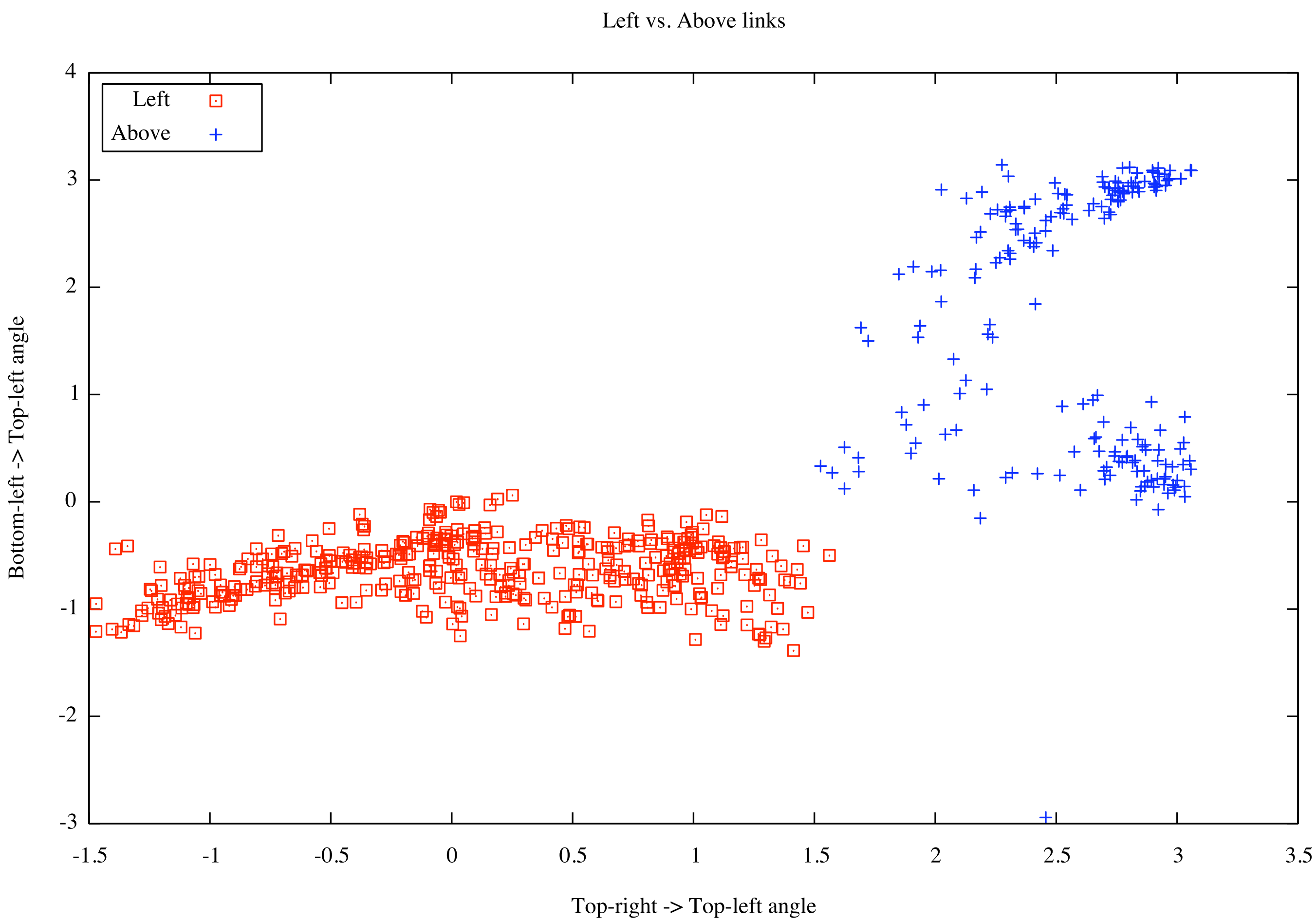
- Empirically measured angles between bounding box corners



- Two measurements are usually sufficient
- Symbol type irrelevent, only bounding boxes used

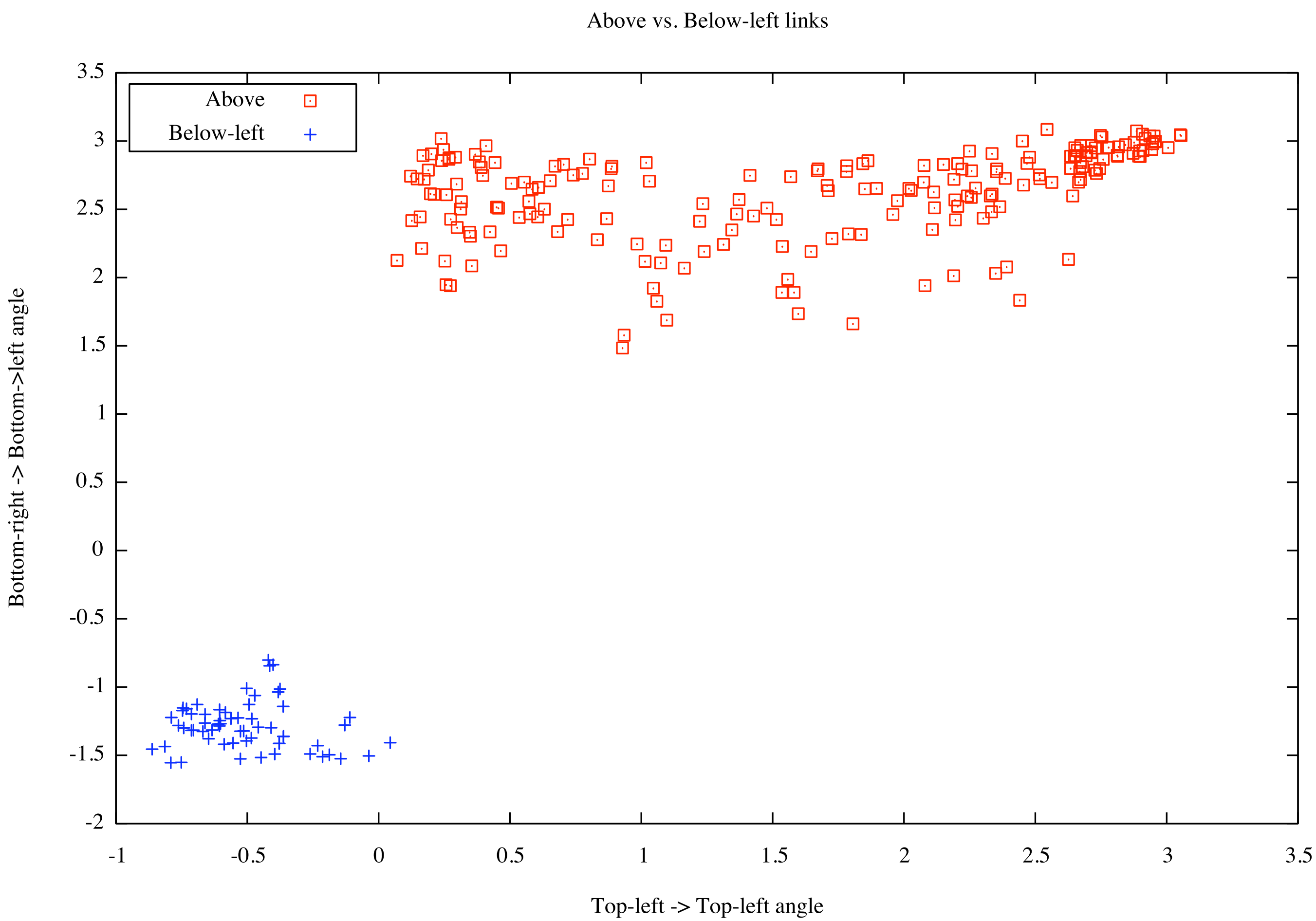
Classifying symbol links

- Many results look promising



Classifying symbol links

- Many results look promising



Classifying symbol links

- Currently building binary classifiers to distinguish between any two link types
- Next step: tournament classifier
- These confidences are necessary for both the grammar model as well as Bayesian models

Conclusions

- MathBrush: a practical tool for doing interactive mathematics with a pen interface
- We implemented several recognition techniques
- Recognizer is easily extendible and accurate in many situations
- Difficulties in recognition are often due to lack of contextual information
- Work ongoing to combine structural analysis and symbol recognition

References

- C.C. Tappert. *Cursive Script Recognition by Elastic Matching*. IBM Journal of Research and Development 26(6) pp.765-771 (1982).
- Stephen M. Watt and Xiaofang Xie. *Prototype Pruning by Feature Extraction for Handwritten Mathematical Symbol Recognition*. Maple Conference 2005 Proceedings, pp.423-437, 2005.

MathBrush

http://www.cs.uwaterloo.ca/scg/mathbrush/