Scaling up: or, how to build a full-ASCII, 100-typeface classifier (+ Tibetan)

• We’re still focused on isolated symbols
• What strategies are good for extending classifiers to:
  – many symbols
  – many typefaces
  – many sizes
  – varied image degradations
• Anatomy of a particular system:
  – how to use ‘structural’ features in a statistical setting
  – how to train (semi-)automatically
  – how to improve systematically
• Versatility: applications to non-ASCII symbol sets

Polyfont Classification

Image recognition algorithm:
  printed & typewriter text
>100 fonts of ASCII symbols

Performance analysis:
  accuracy
  uniformity: symbols × fonts × sizes × defects

Assess “generalization”
  accuracy on fonts not trained on
  rich variety of font-symbol shapes
  large-scale experimental data
Next lecture....

- Is there a dominant classifier?
  - how costly is it to build the best?
- How do classifiers compare?
- Combining classifiers:
  - voting methods
- When are combinations better?
- Randomized classifiers

The Fonts of Greatest Interest

"Fonts" are distinguished by shape:
- roman and italic are different
- size variations don’t count
- condensed width and weight variations don’t count:
  - ne he el
  - ne he el
  - ne he el
  - ne ne el

Body text fonts
- not display/decorative/script/heading

Widely used for English prose
- not classified ads
- 100 fonts = 50 “font-families”

Most Commonly Occurring Fonts

<table>
<thead>
<tr>
<th>For maximum &quot;coverage&quot; of fonts, magazines, newspapers, letters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scholars:</strong></td>
</tr>
<tr>
<td>- Alexander Cazalis, &quot;The Anatomy of a Typeface&quot;</td>
</tr>
<tr>
<td>&quot;The growth of typeface has revolutionized shape of each page.&quot;</td>
</tr>
<tr>
<td><strong>Manufacturers / Designers:</strong></td>
</tr>
<tr>
<td>- Linotype - Bruce Licher</td>
</tr>
<tr>
<td>- Letraset - Ned Meda</td>
</tr>
<tr>
<td>- IBM - John Langley (Condol, etc.)</td>
</tr>
<tr>
<td>- Chalk Figure (Lewis, etc.)</td>
</tr>
<tr>
<td><strong>Industry watch:</strong></td>
</tr>
<tr>
<td>- American Institute of Graphic Arts</td>
</tr>
<tr>
<td>- Anita Bickels Books (1991-98)</td>
</tr>
<tr>
<td>- Type World Newsletter - Pufel Resource</td>
</tr>
<tr>
<td><strong>Consultants:</strong></td>
</tr>
<tr>
<td>- Typo Specimen Co. &amp; Questioned Documents Ltd. (1995)</td>
</tr>
</tbody>
</table>

'Structural' Shape Analysis

Hypothesis: simple ‘shapes’ may:
- capture human intuition
- generalize across font styles

‘Structural’ analysis was popular in late 70’s through early 90s
However, not clear how to learn them (don’t want to hand-code)
Competing statistical methods seemed to provide a better framework for coping with noise or all sorts
### Structural & Statistical

- **Structural shape analysis**
  - yields: set of component shapes
  - "Feature identification" —
    - find a mapping from sets of shapes to bit-vectors

- **Statistical classification**
  - accepts: bit vector

### Combining structural features with statistical classification

- **Structural shape analysis**
  - represents by list of simpler shapes
  - "flexible" good features
  - information-preserving
  - unordered, variable-length list

- **Statistical classification**
  - represents by list of a simple (bit)
  - requires a few, independent features
  - ordered, fixed-length list

← This is the glue between the two approaches

### Choosing a parameterization of the shapes

- Examples (strokes):
  - □
  - □
  - □
  - □

- Criss cross (natural)
  - Exploit the fact that, for all shape types:
    - can find a low-dimensional parameterisation
    - with certain "invariance" properties under a variable metric

- Visually similar shapes lie close together.

- Dissimilar shapes lie far apart.

### Clustering Shapes in their parameter spaces

- Parameterization of strokes of a "P."

- Superimposed "P" strokes, showing clusters.

Note how tightly strokes cluster in this space
The variation of strokes due to noise of all kinds is easier to analyze
Idealizing the shape clusters

Clusters approximated by $L_\infty$-spheres
(Composite results w/ Euclidean norm)

Idealized shape clusters seem to capture some intuition

Most classes are as familiar as the ‘A’ shown here
Roughly correspond to intuition, but there are irritating anomalies
Similar results for the other shape types: holes, endpoints, etc

Causes of anomalies

Some shapes, such as vertical stroke in ‘K’, are difficult to extract reliably without ‘higher-level’ models
So, even intuitively ‘low-level primitives’ challenge the earliest stages of feature extraction
As image degradations worsen, these instabilities become persuasive

Merging clusters

There are lots of similar clusters among the classes
Reduce their number by a greedy merging heuristic:
• merge the next pair of clusters whose merged volume increases the least,
• until have a target number of clusters.
Each final shape-cluster gives a ‘feature’ for classification

- Assign a binary feature to each shape-cluster
- Given an unknown isolated symbol image,
  - perform shape analysis into strokes, holes, etc
  - if any of these ‘matches’ (falls inside) a shape-cluster (in its parameter space), then set that shape-cluster’s feature bit to 1
  - if no shape matches a shape-cluster, its feature bit remains 0
- So, have a bit-vector each bit of which indicates the presence (one or more times) of a shape-cluster
- Finally, classify this vector
  - using, e.g., the linear classifier assuming class-conditional independence among the features

Some simpler feature mappings

Cheaper to compute than the shape-clustering feature mapping
But, gives many more features
Which is better? Must run an experiment

Small-scale experiments: single-font

![Graph showing small-scale experiments: single-font](image)

Better than 99.5% top choice, above 12 point.

Top choice performance of classifiers trained on the Times Roman font and mixed sizes (6-18 points), and tested on the same (not at the specific sizes shown (2100 samples per dataset).

Small-scale experiments: mixtures of similar fonts

![Graph showing small-scale experiments: mixtures of similar fonts](image)

Clustering superior to k-partitions for similar fonts.

Top choice performance of classifiers trained on a mixture of three similar but not identical fonts (Times Roman, Palatino, and Mathis Roman), and on mixed sizes (6-18 points), then tested on the same font mixtures at the specific sizes shown (for 5-18, and 5-partitioning and clustering character-identification methods, 6000 samples per dataset).
Small-scale experiments: mixtures of dissimilar fonts

Top choice bad for mixtures of dissimilar fonts.

Top choice performance of classifiers trained on a mixture of six dissimilar fonts (Times Roman, Times Italic, Times Bold, Helvetica Light, Helvetica Italic, and Helvetica Black), and on mixed sizes (8 14 points), then tested on the same font-mixture as the specific sizes shown (12,600 samples per dataset).

Small-scale experiments: top-3 correct

Top 3 performance is still good.

Performance (correct in top 3 choices) of classifiers trained on a mixture of six dissimilar fonts (Times Roman, Times Italic, Times Bold, Helvetica Light, Helvetica Italic, and Helvetica Black), and mixed sizes (8-14 points), then tested on the same font-mixture as the specific sizes shown (12,600 samples per dataset).

How to improve classifier?

‘split’ classes into sets of similar shapes

- 1 a a a a a etc.
- 2 a a a a a etc.
- 3 a a a a a etc.
- 4 a a a a a etc.
- 5 a a a a a etc.
- 6 a a a a a etc.

"Variant =" Bayesian "class =" statistical prototype

Splitting classes

more variants:
- larger, slower classifier
- more accurate on given fonts
- less accurate on other fonts

We manually chose 612 variants:
- 1 variant (no splitting)
- 18 variants
- 6.5/symbol on average
Best & Worst....

Points:

- Brilliant Beauty: Pack my box with four dozen sugar lumps.
- Harsh Hell: Pack my box with four dozen sugar lumps.
- Player Light Shade: Pack my box with four dozen sugar lumps.
- Player Plain Shade: Pack my box with four dozen sugar lumps.

Average:

- Hypothesis Banks: Pack my box with 20 dozen sugar lumps.
- Kernel Stocks: Pack my box with 20 dozen sugar lumps.
- Defense Coax: Pack my box with 20 dozen sugar lumps.
- Cocktail Call: Pack my box with 20 dozen sugar lumps.

Visions:

- Angled Stock: Pack my box with 20 dozen sugar lumps.
- Hot Line: Pack my box with 20 dozen sugar lumps.
- voter: Pack my box with 20 dozen sugar lumps.
- Market Order: Pack my box with 20 dozen sugar lumps.

Characters:

- e > i > e > n > a...
- ... 

Accuracy by symbol

Top-1 accuracy of each of the 94 symbols, averaged over all 100 fonts and over the classes 1ip, 12p, and 14p (at 300 pp), ensuring that all fonts and symbols are approximately. The 94 symbols are plotted in ascending order of accuracy. (About 1000 samples per curve; fewer for symbols not present in all 94 fonts.)

75% of top-1 confusions:

Accuracy by font-symbol

1. Ca - 4.
2. CaT - 4.
5. ClM - 4.
Accuracy by font-symbol

Space requirements

Space required to represent all:

- fonts × symbols × sizes × defects

Space ≈ no. shape variants

Each shape variant represents:

- a set of font-symbols, e.g. \{a, b, a, a, a\}
  - all sizes × all defects

‘Compression’

Compression = no. font-symbols / no. variants

9400 / 612 = 15.4

- 15 font styles, on average, are represented by one statistical prototype,
  - in spite of size and defect variations

Implies that larger classifiers can be built with only a fractional increase in computing resources

Generalization

Extrapolation to fonts not trained on

We trained on only 3416 of the 9400 font-symbols, and still did well on all of them

“Generalization factor” = 3.9

This has held roughly constant:

- 25-, 40-, 60-, 80-, 100-font trials

Implies that a 100-font classifier will do well on many more than 100 fonts
Feature identification

Choosing a mapping —

- automatic utilitarian from training data
- universally applicable to all shape types
- "simple" —
  - controlled by a few, innate parameters
  - most values of parameters are critical

Applying a mapping —

- unify fast explicit multi-arrangement search algorithms

Effect on classification —

- superior to simpler hand mappings
- demonstrated in large-scale experimental trials

Summary of characteristics

- Simple, fully automated method —
  - a minimum of hand-sold rules
  - applied uniformly to all shape types
  - guided by a few parameters
  - "natural" statistical test-
    - shows when the system

- Tested in large-scale trials —
  - results yield 95% initial accuracy
  - showed improvement over 90%-95% (hand)

- Fairly general techniques —
  - requires that shape types be elementary
  - "constant" parameterization
  - not strongly specialized in OCR

Training on size ranges