Compact Representation and Recognition for Handwritten Mathematical Characters

Stephen M. Watt

Ontario Research Centre for Computer Algebra

University of Western Ontario

London Ontario, Canada

Joint work with Oleg Golubitsky

ACA Conference, RISC Linz, 29 July 2008

What this talk is about

- The main problem:
  In handwriting recognition, the human and the computer take turns thinking and sitting idle. Small devices therefore challenged by MHR.

- We ask:
  Can the computer do useful work while the user is writing and thereby get the answer faster after the user stops writing?

- We show:
  The answer is "Yes"!

Outline

- Mathematical handwriting recognition
- Representing curve traces using truncated orthogonal series
- A complexity model for on-line computation
- An on-line algorithm to compute series coefficients

Math Handwriting Recognition

- Considered natural and desirable by many

- Different than natural language recog:
  - 2-D layout is similar to a combination of writing and drawing.
  - No fixed dictionary.
  - Many more similar few-stroke characters.

Usual Character Recog Methods

- Smooth and then re-sample data

- Match against *N* models,   *OR....*

- Identify "features", such as
  - Coordinate values of sample points, Number of loops, cusps, Writing direction at selected points, Center of mass, *etc*

  Use a classification method, such as
  - Nearest neighbour, Subspace projection, Cluster analysis, Support Vector Machines

- Rank choices by consulting dictionary

Difficulties

- Many characters in mathematics means comparison against all models slow.
- Determining features from points
  - Requires many *ad hoc* parameters.
  - Replaces measured points with interpolations
  - It is not clear how many points to keep, and most work depends on number of points

Series Representation

Reference Char and Watt [ICDAR 2007]

- **Main idea**: represent *x* and *y* coordinate curves as truncated series.
- **Advantages**:
  - *Compact* – few coefficients needed
  - *Geometric* – the truncation order is a property of the character set, not the device
  - *Algebraic* – desired properties of curves can be computed algebraically
  - *Numerically stable*

Series Representation – Details

- Functional inner product:

$$\langle f, g \rangle = \int_a^b f(t) g(t) w(t) \mathrm{d}t$$

- Given *a, b, w*, the basis functions can be obtained from monomials by Gram Schmidt orthogonalization and can write:

$$f(t) = \sum_{i=0}^{\infty} c_i b_i(t), \quad c_i \in \Re, b_i \in B$$

Series Representation – Details

- Coefficients may be obtained by:

$$c_i = \langle f, b_i \rangle / \langle b_i, b_i \rangle, \quad i \in \mathbb{N}_0$$

- Choosing weight fn $w(t) = 1/\sqrt{1-t^2}$ gives Chebyshev polynomial basis.

$$T_n(t) = \cos(n \arccos t)$$

- Series may be truncated, leaving a residual error that decreases with order.

Advantages of Series Rep.

- The representation is compact
- Device and scale invariance
- Other features can be computed
- Degree determined intuitively (# cusps and turns)

Disadvantages of Series Rep.

- All the work to compute the series coefficients must be done after the series is written [non-linearity of $w(t)$].
- Many other recognition techniques share this problem.
- Can we somehow be more efficient by computing the series coefficients as the data is collected?

An On-Line Complexity Model

- Input is a sequence of *N* values received at a uniform rate.
- Characterize an algorithm by
  - $T_\Delta(n)$ no. operations as *n*-th input is seen
  - $T_F(n)$ no. operations after last input is seen
- Write on-line time complexity as

$$\mathrm{OL}_n[T_\Delta(n), T_F(n)]$$

- E.g., linear insertion requires time

$$\mathrm{OL}_n[O(n), 0]$$

On-Line Complexity Model (2)

- An algorithm that takes on-line time

$$\mathrm{OL}_n[T_\Delta(n), T_F(n)]$$

takes total time

$$\sum_{i=0}^{N} T_\Delta(i) + T_F(N)$$

- That is, a factor of N can come for free.

On-Line Algorithm for Series Coeffs

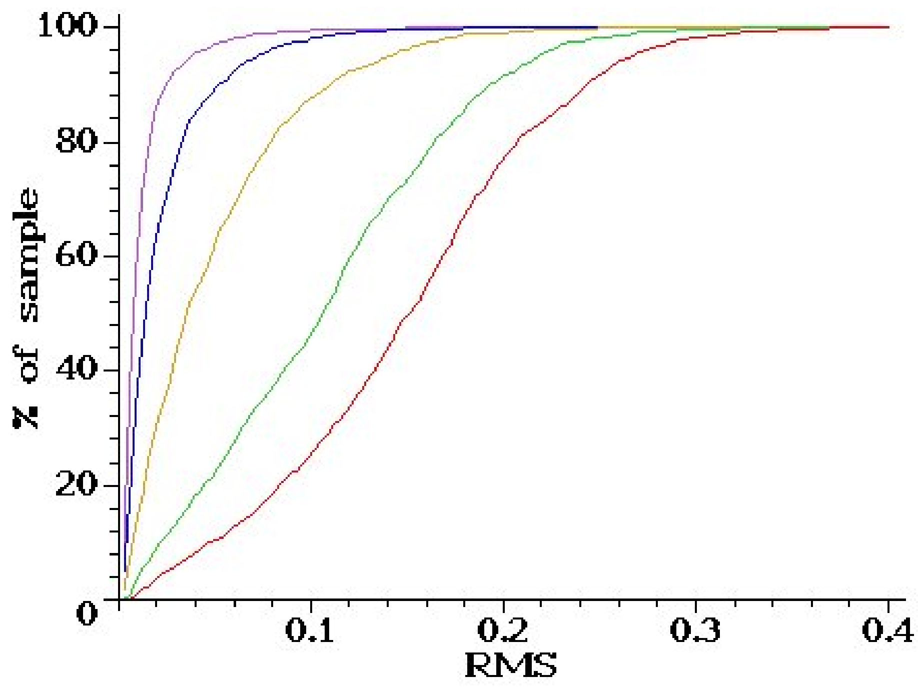
- If we choose a series whose weight function is linear, then the series coefficients can be computed on line.
- The series coefficients are linear combinations of the moments, which can be computed by numerical integration as the points are received.
- This is the Hausdorff moment problem (1921) and has been shown to be unstable by Talenti (1987).
- It is just fine, however, for the orders we need.

On-line Series Coeffs – Outline

- Use Legendre polynomials P_i as basis on the interval $[-1,1]$, with weight function 1.
- Collect numerical values for $f(\lambda)$ on $[0, L]$. L is not known until the pen is lifted.
- As the numerical values are collected, compute the moments $\int \lambda^i f(\lambda)\, \mathrm{d}\lambda$.
- After last point, compute series coeffs for f with domain and range scaled to $[-1,1]$. These will be linear combinations of he moments.

Quality of Legendre Series

- Legendre series give the same order of RMS error as the Chebyshev series representation, but can be computed on-line.



On-line Series Coeffs – Details

$$\hat{f}(\tau) = f\big((\tau+1)L/2\big) = \textstyle\sum_{n=0}^{\infty} \hat{\alpha}_n P_n(\tau)$$

$$\begin{aligned}
\hat{\alpha}_n &= (n+\tfrac{1}{2}) \int_{-1}^{1} \hat{f}(\tau) P_n(\tau)\, \mathrm{d}\tau \\
&= \frac{2n+1}{L} \int_{0}^{L} f(\lambda) P_n(2\lambda/L - 1)\, \mathrm{d}\lambda \\
&= \frac{2n+1}{L} \sum_{i=0}^{n} [t^i] P_n(2t-1) \int_{0}^{L} f(\lambda)\, (\lambda/L)^i\, \mathrm{d}\lambda \\
&= \frac{2n+1}{L} \sum_{i=0}^{n} \frac{[t^i] P_n(2t-1)}{L^i} \mu_i(f, L).
\end{aligned}$$

$$\hat{\alpha}_n = (-1)^n \frac{2n+1}{L} \sum_{i=0}^{n} \left(\frac{-1}{L}\right)^i \binom{n}{i} \binom{n+i}{i} \mu_i(f, L)$$

On-line Series Coeffs – Details

- A specialized numerical integration scheme is required for the moments.

$$\int_0^L \lambda^n f(\lambda) d\lambda \approx$$

$$\sum_{i=1}^{L-1} \frac{i^{n+1}}{n+1} \times \frac{f(i-1) - f(i+1)}{2} +$$

$$\frac{L^{n+1}}{n+1} \times \frac{f(L-1) + f(L)}{2}.$$

On-line Series Coeffs – Details

- The range of f can be scaled to any desired range $[a, b]$

$$\hat{\hat{f}}(\tau) = \frac{b-a}{f_{\max} - f_{\min}} \hat{f}(\tau) + \frac{a f_{\max} - b f_{\min}}{f_{\max} - f_{\min}}$$

$$= \sum_{i=0}^{\infty} \hat{\hat{\alpha}}_i P_i(\tau),$$

Complexity

- The on-line time complexity to compute coefficients for a Legendre series truncated to degree d is then

$$T_\Delta = 2(d+2)$$

$$T_F = \frac{3}{2}d^2 + \frac{11}{2}d + 10$$

- That is, the time at pen up is *constant* with respect to the number of points.

Practical Cost

- The construction of the series coefficients at the end is the real time limiting operation.
- This should take on the order of 200 to 500 machine instructions for a series of order 10 to 15.

Comparison with Models:
Distance between curves

- Some classification methods compute the distance between the input curve and models.
  E.g. Elastic matching, which takes time quadratic in the number of sample points.
- With orthogonal series representation, we have a much less expensive comparison:

$$\rho^2(C,\bar{C}) = \int_0^1 \left[x\left(\frac{t}{N}\right) - \bar{x}\left(\frac{t}{\bar{N}}\right)\right]^2 + \left[y\left(\frac{t}{N}\right) - \bar{y}\left(\frac{t}{\bar{N}}\right)\right]^2 dt$$

$$\approx \sum_{i=0}^{d} [c_i^x - c_i^{\bar{x}}]^2 + [c_i^y - c_i^{\bar{y}}]^2$$

- Linear in d. About 60-100 machine instructions!

Conclusions

- It is possible to compute series representation of characters quickly and compactly.
- This representation is compact, high-fidelity, device independent, numerically robust and allows algebraic treatment of the curves.
- The work to compute this rep. at pen-up is minimal, on the order of a few hundred machine instructions.
- Likewise, the cost to compute the distance between two curves is on the order of 100 machine instructions.
- The computations are straightforward and are suitable for hardware implementation.

References

- F. Hausdorff. "Summationsmethoden und Momentfolgen. I" Mathematische *Zeitschrift*, 9:74-109, 1921.
- F. Hausdorff. "Summationsmethoden und Momentfolgen. II" Mathematische *Zeitschrift*, 9:280-299, 1921.
- G. Telenti. "Recovering a function from a finite number of moments." 3:501-517, 1987.