Polynomial Approximation in Handwriting Recognition

Stephen M. Watt
University of Western Ontario

International Workshop on Symbolic-Numeric Computation (SNC 2011)
7-9 June 2011, San Jose, California

The Pen as an Input Device

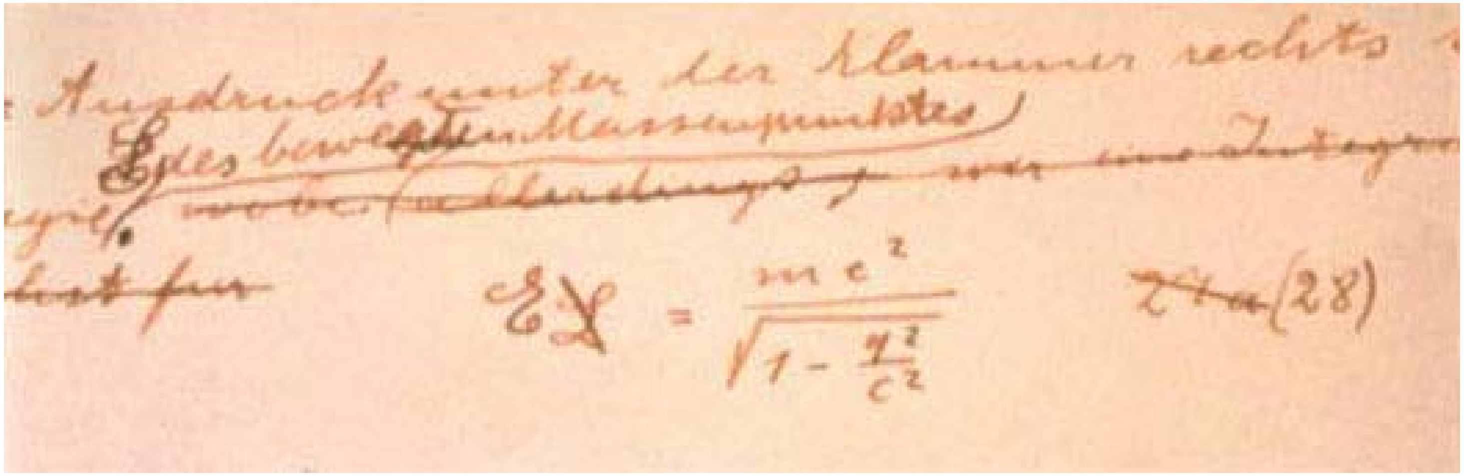
- Pen input for electronic devices is becoming important as an input modality.

- Pens can be used where keyboards can't, on very small or very large devices, in wet or dirty environments, by people with repetitive stress injuries.

- They also allow much easier 2-dimensional input, *e.g.* for drawings, music or mathematics.

$$e^x = \int e^x dx = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

Pen-Based Math



- Input for CAS and document processing.
- 2D editing.
- Computer-mediated collaboration.

Pen-Based Math

- Different than natural language recognition:
  - 2-D layout is a combination of writing and drawing.
  - Many similar few-stroke characters.
  - Many alphabets, used idiosyncratically.
  - Many symbols, each person uses a subset.
  - No fixed dictionary for disambiguation.

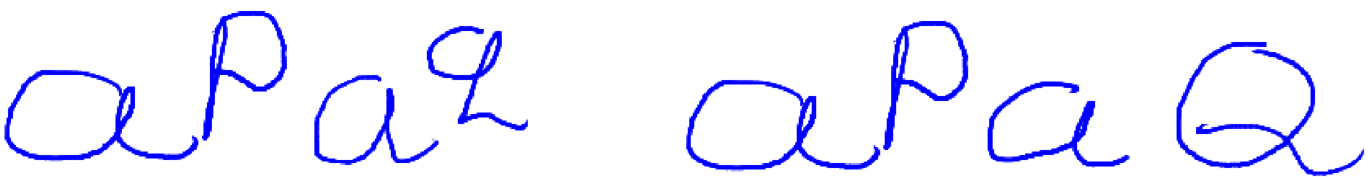
Similar Few-Stroke Characters

Character Ambiguities

$$\sum_{i} i^2$$

$$i + z = \sin \omega t$$

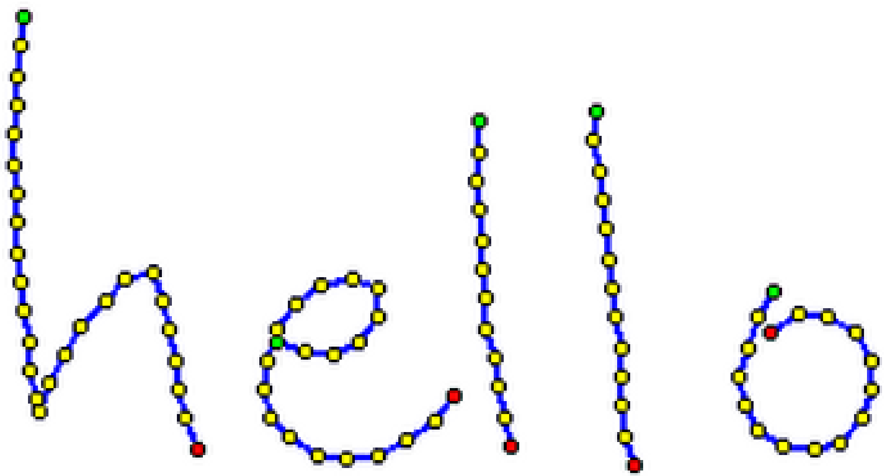
Character Ambiguities



Character Recognition

- A story about three statisticians
- Will concentrate on character recognition
- Several projects ignoring this problem

Digital Ink Formats



- Collected by surface digitizer or camera
- Sequence of (***x,y***) points sampled at some known frequency
- Possibly other info (angles, pressure, etc)
- Grouping into traces, letters, words + labelling

W3C Proposed Recommendation

Ink Markup Language (InkML)

W3C Proposed Recommendation 10 May 2011

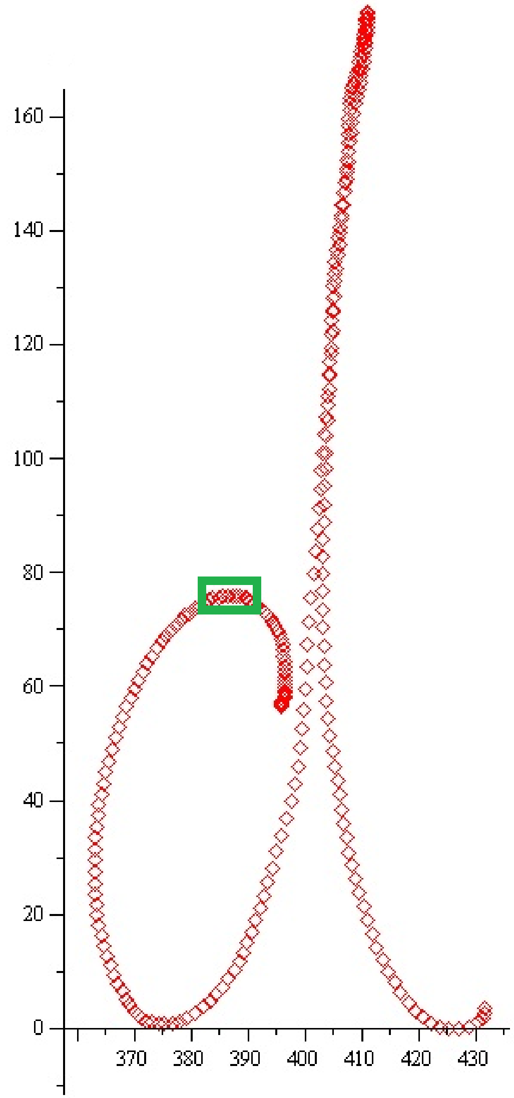
This version:
http://www.w3.org/TR/2011/PR-InkML-20110510/
Latest version:
http://www.w3.org/TR/InkML
Previous version:
http://www.w3.org/TR/2011/CR-InkML-20110111/

Editors:
Stephen M. Watt, University of Western Ontario
Tom Underhill, Microsoft
Authors:
Yi-Min Chee (until 2006 while at IBM)
Katrin Franke (until 2004 while at Fraunhofer Gesellschaft)
Max Froumentin (until 2006 while at W3C)
Sriganesh Madhvanath (until 2009 while at HP)
Jose-Antonio Magaña (until 2006 while at HP)
Grégory Pakosz (until 2007 while at Vision Objects)
Gregory Russell (until 2005 while at IBM)
Muthuselvam Selvaraj (until 2009 while at HP)
Giovanni Seni (until 2003 while at Motorola)
Christopher Tremblay (until 2003 while at Corel)
Larry Yaeger (until 2004 while at Apple)

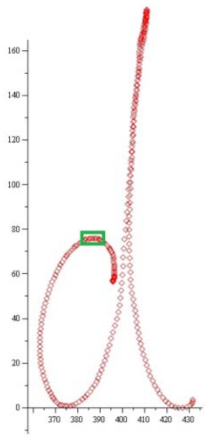
InkML Concepts

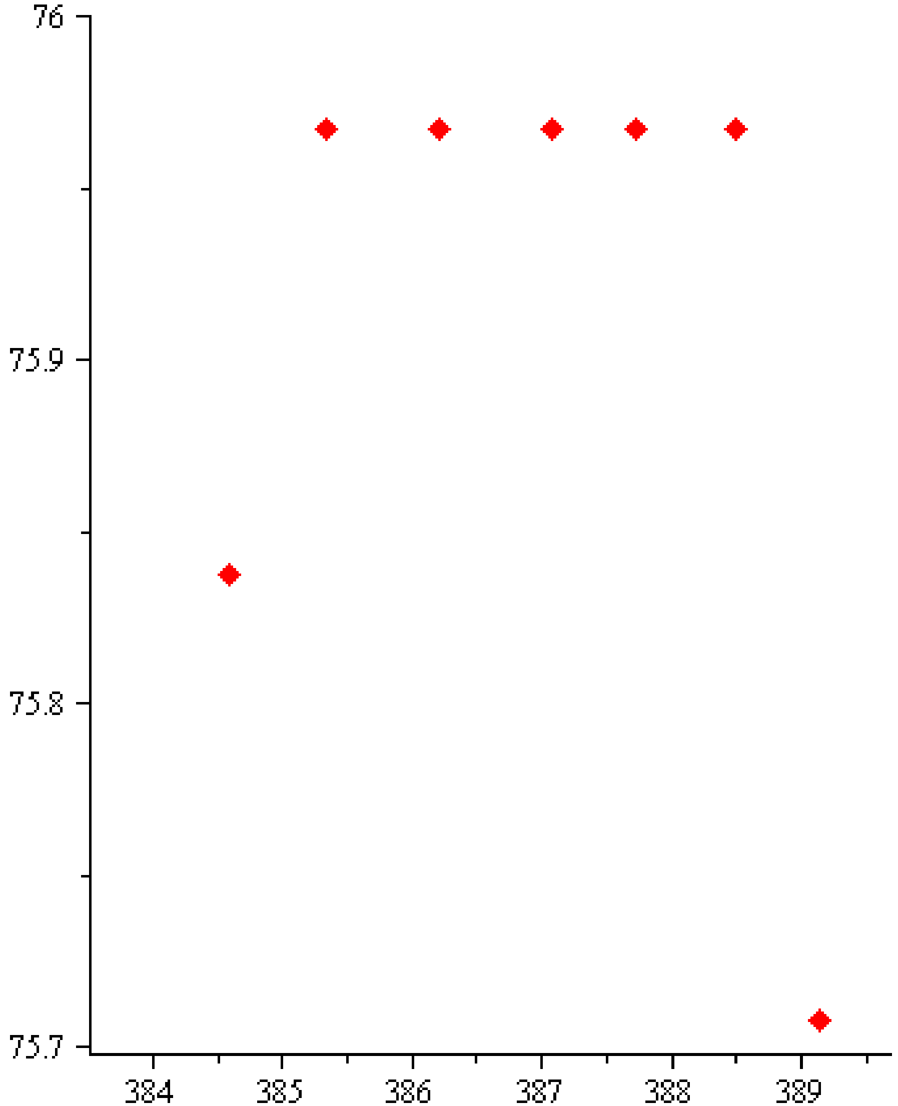
- Traces, trace groups
- Device information: sampling rate, resolution, etc.
- Pre-defined and application defined channels
- Trace formats, coordinate transformations
- Annotation text and XML

What the Computer Sees



What the Computer Sees





Usual Character Reco. Methods

- Smooth and re-sample data *THEN*
- Match against *N* models by sequence alignment
  OR
- Identify “features”, such as
  - coordinate values of sample points, Number of loops, cusps, Writing direction at selected points, *etc*

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

  THEN
- Rank choices by consulting dictionary

Difficulties

- Having many similar characters (e.g. for math) means comparison against all possible symbol models is 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 methods depend on number of points
  - Device dependent
- What to do since there is no dictionary?

- New ideas are needed!

Two Thoughts

- For HWR do we need all the trace data?
  - Do we need all the points?
  - Do we need full accuracy for all the points?

- What is classification?
  - H (English aitch, Greek eta, Russian en)
  - O (zero, oh, degree, …)
  - P, C, S (R, S, T)

Main Ideas

- Treat traces as curves, not point sets.
- Have to be able to recognize perturbed input.
- There is not always a single “right answer”.

First Axiom of HWR

$\forall$ A*,*
if a sample looks like an A*,*
then it can be an A*.*

First Axiom of HWR

∀ A,
if a sample looks like an A,
then it can be an A.

Implications:

- Classification gives a ***set*** of valid possibilities.
- Must be able to classify ***perturbed inputs***.
- Can use ***approximation*** to represent traces more conveniently.

Orthogonal Series Representation

- **Main idea**:
  Represent *c*oordinate curves as truncated orthogonal series.

- **Advantages**:
  - *Compact* – few coefficients needed
  - *Geometric*
    – the truncation order is a property of the character set
    – gives a natural metric on the space of characters
  - *Algebraic*
    – properties of curves can be computed algebraically
    (instead of numerically using heuristic parameters)
  - *Device independent*
    – resolution of the device is not important

Inner Product and Basis Functions

- Choose a functional inner product, e.g.

$$\langle f, g \rangle = \int_a^b f(t)g(t)w(t)dt$$

- This determines an orthonormal basis in the subspace of polynomials of degree *d.*
- Determine ϕ_i using GS on $\{1, t, t^2, t^3, \dots\}$.
- Can then approximate functions in subspaces

$$A(t) \approx \sum_{i=0}^{d} \alpha_i \phi_i(t) \qquad \alpha_i = \langle A(t), \phi_i(t) \rangle$$

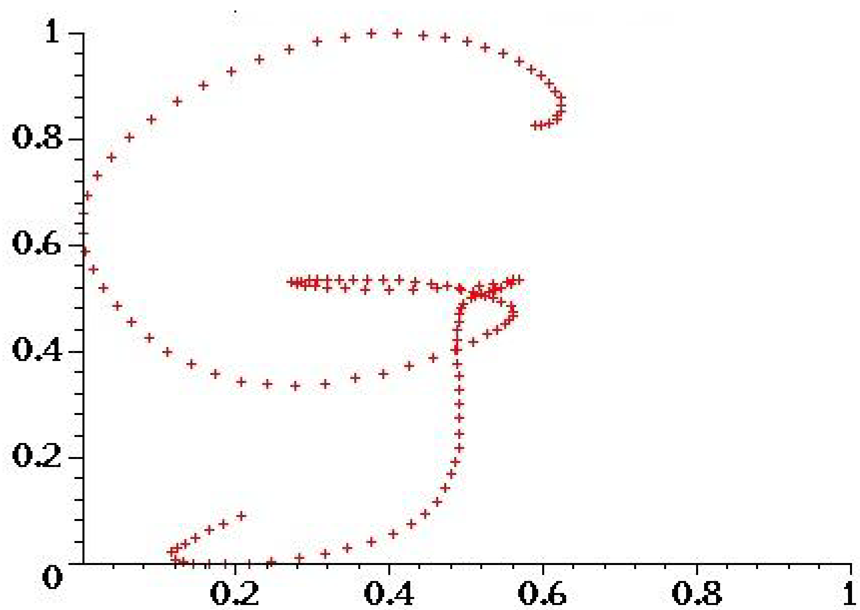
First Look: Chebyshev Series

- Initially used Chebyshev series
  [Char+SMW ICDAR 2007].

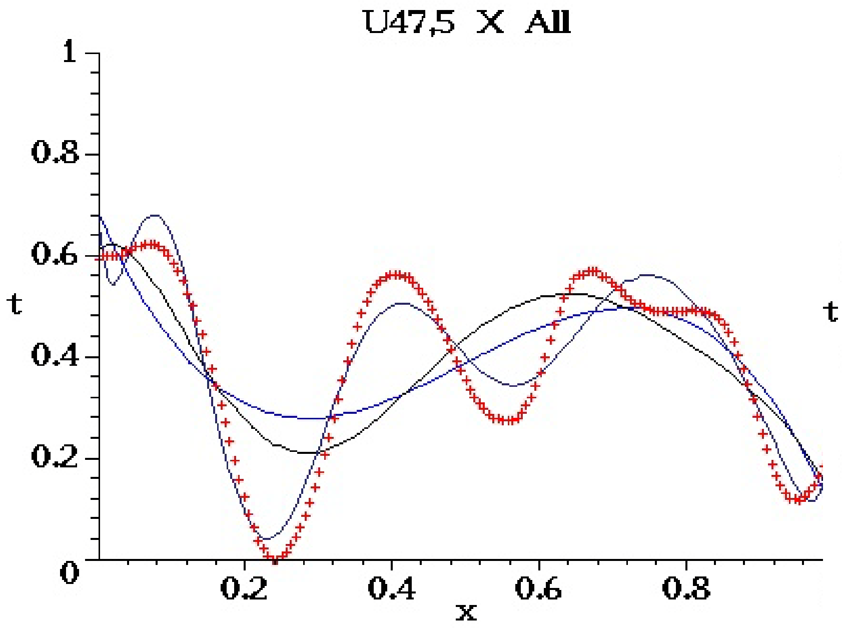
- Found could approximate closely
  (small RMS error) with series of order 10.

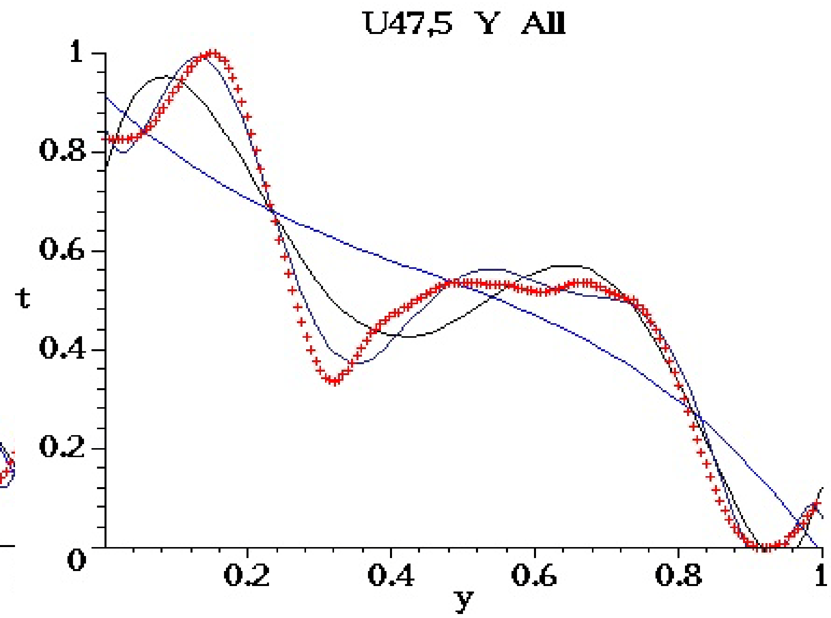
- Like symbols tended to form clusters.

Raw Data for Symbol G

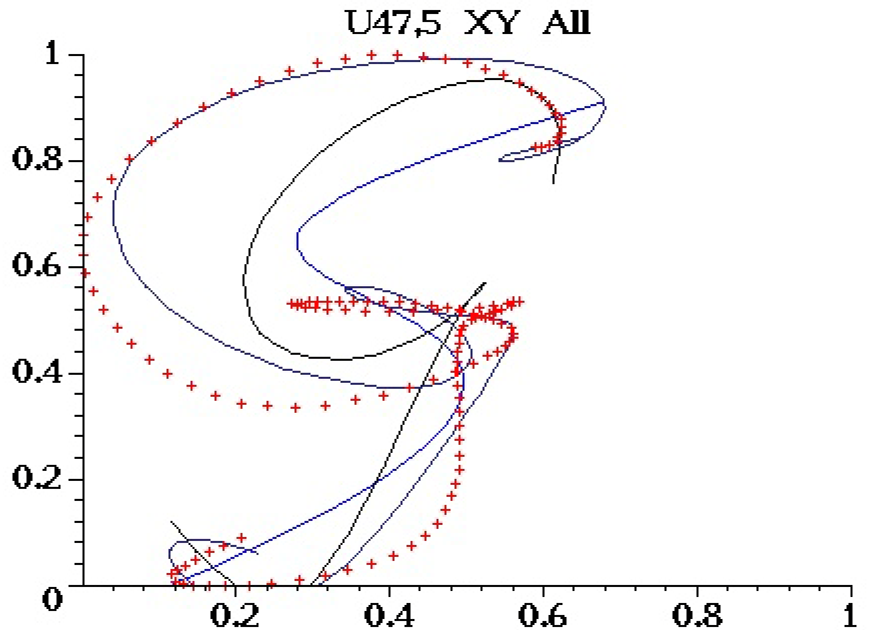


Coordinate fn approximations

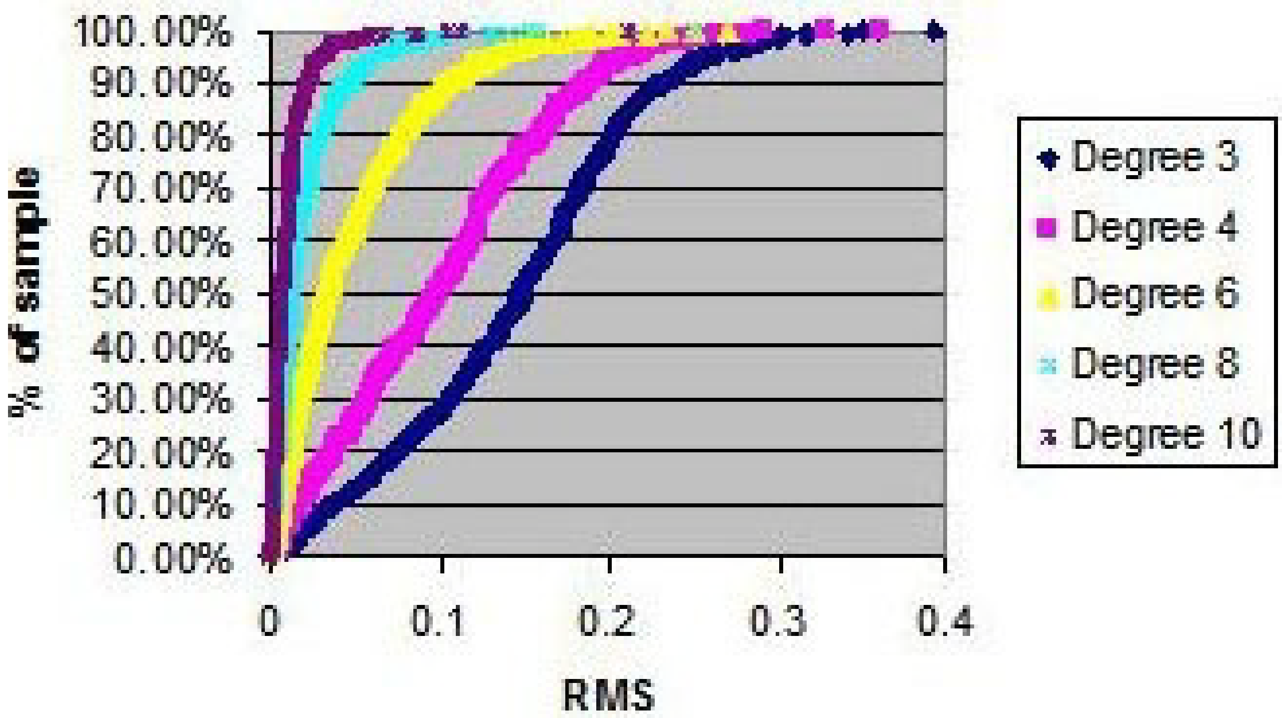




Chebyshev Approx to Character



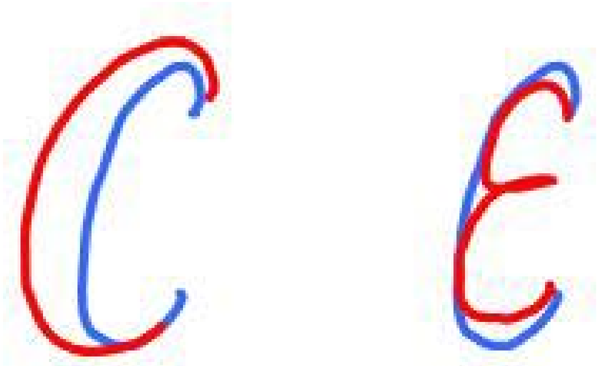
RMS Error


100.00%
90.00%
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
% of sample
0
0.1
0.2
0.3
0.4
RMS
Degree 3
Degree 4
Degree 6
Degree 8
Degree 10

Problems

- Want fast response –
  how to work while trace is being captured.

- Low RMS does not mean similar shape.



Problem 1. On-Line Ink

- The main problem:
  In handwriting recognition, the human and the computer take turns thinking and sitting idle.

- We ask:
  Can we 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"!

On-Line Series Coefficients

- Use Legendre polynomials P_i as basis on the interval $[-1,1]$, with weight function 1.
- Collect numerical values for $f(\lambda)$ on $[0, L]$.
  λ = arc length.
  L is not known until the pen is lifted.
- *As the sample points are collected*, numerically integrate the moments $\int \lambda^i f(\lambda) d\lambda$.
- After last point, compute series coefficients for f with domain and range scaled to $[-1,1]$.
  This uses a simple linear transformation of the moments.

On-Line Series Coefficients

- Transform moments $\mu_i(f, L)$ of $f(\lambda)$ on $[0, L]$ to coefficients of $\hat{f}(\lambda) = \sum_k \hat{\alpha}_k P_k(\lambda)$ on $[-1, 1]$:

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

- Normalize range of f:

$$\hat{\alpha}_k \frac{b-a}{f_M - f_m} + \delta_{i0} \frac{a\, f_M - b\, f_m}{f_M - f_m}$$

On-Line Series Coefficients

- Approach works for any inner product with linear weight function.
- This is the Hausdorff moment problem (1921), shown to be unstable by Talenti (1987).
- It is just fine, however, for the dimensions we need.

An On-Line Complexity Model

- Input is a sequence of values received at a uniform rate.

- Characterize an algorithm by
  - $\Delta(n)$ complexity as n-th input is seen
  - $F(n)$ complexity after last input is seen

- Write on-line complexity as

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

- E.g., linear insertion sort requires time

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

Complexity

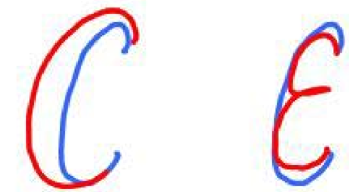
- The on-line time complexity to compute coefficients for a Legendre series truncated to degree d is then

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

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

- The time at pen up is *constant* with respect to the number of points in the trace.

Problem 2. Shape vs Variation



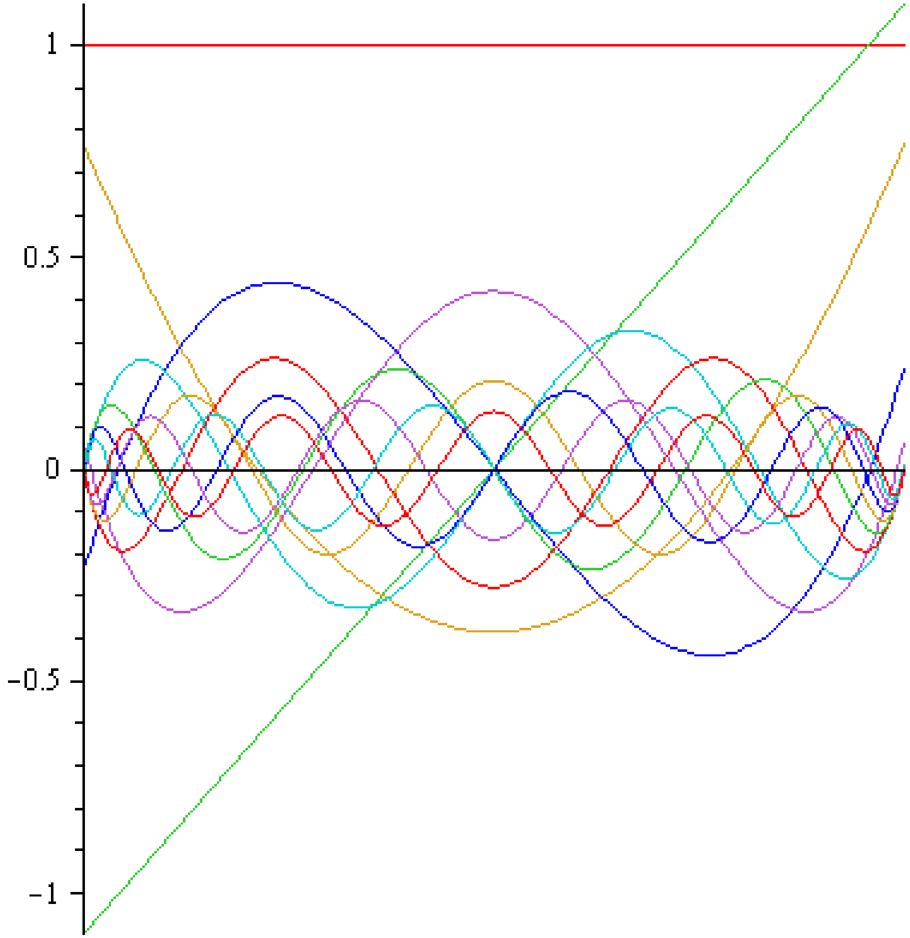
- The corners are not in the right places.
- Work in a jet space to force coords & derivatives close.
- Use a Legendre-Sobolev inner product

$$\langle f, g \rangle = \int_a^b f(t)g(t)dt + \mu_1 \int_a^b f'(t)g'(t)dt + \mu_2 \int_a^b f''(t)g''(t)dt + \cdots$$

- 1st jet space ⇒ set $\mu_i = 0$ for $i > 1$.
  - Choose μ_1 experimentally to maximize reco rate.
  - Can be also done on-line.

  [Golubitsky + SMW 2008, 2009]

Legendre-Sobolev Basis

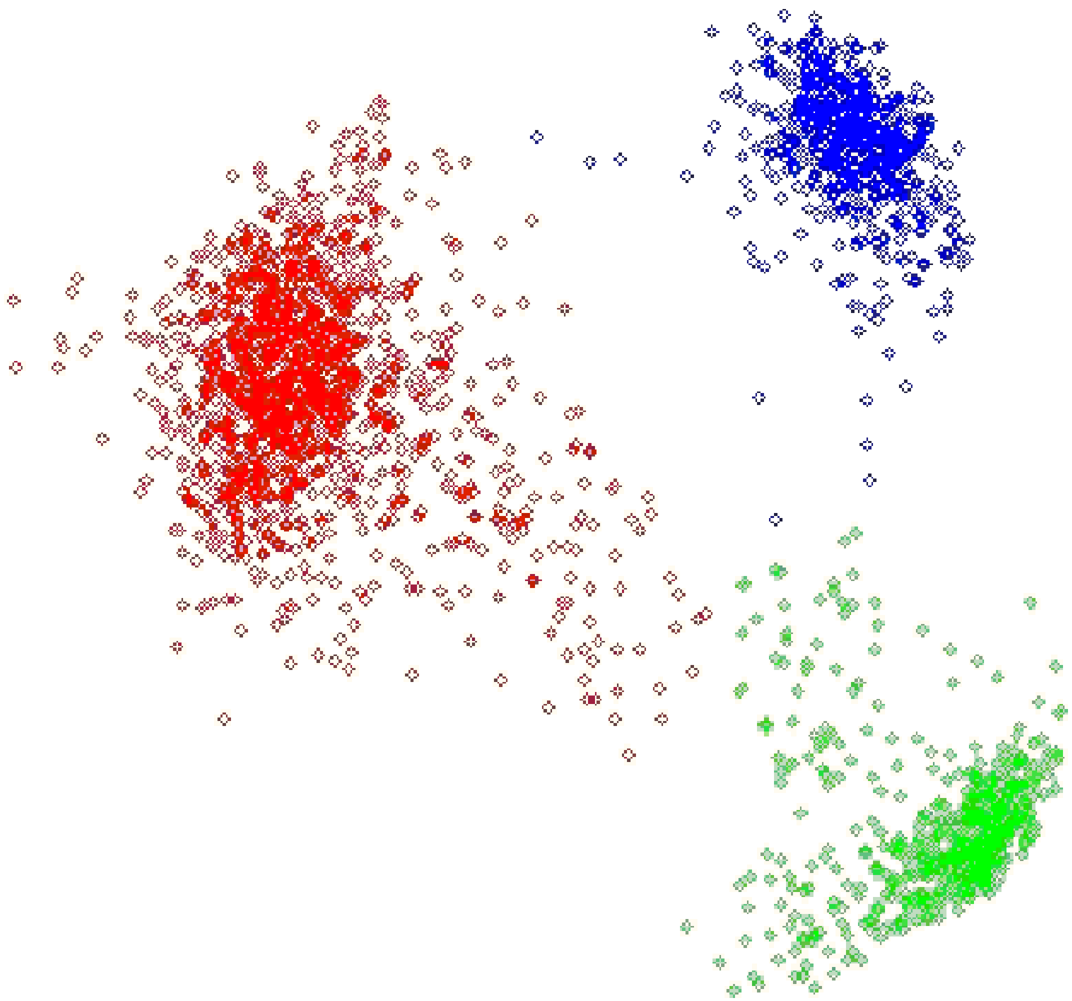


$$a = 0, b = 1, \mu = .125$$

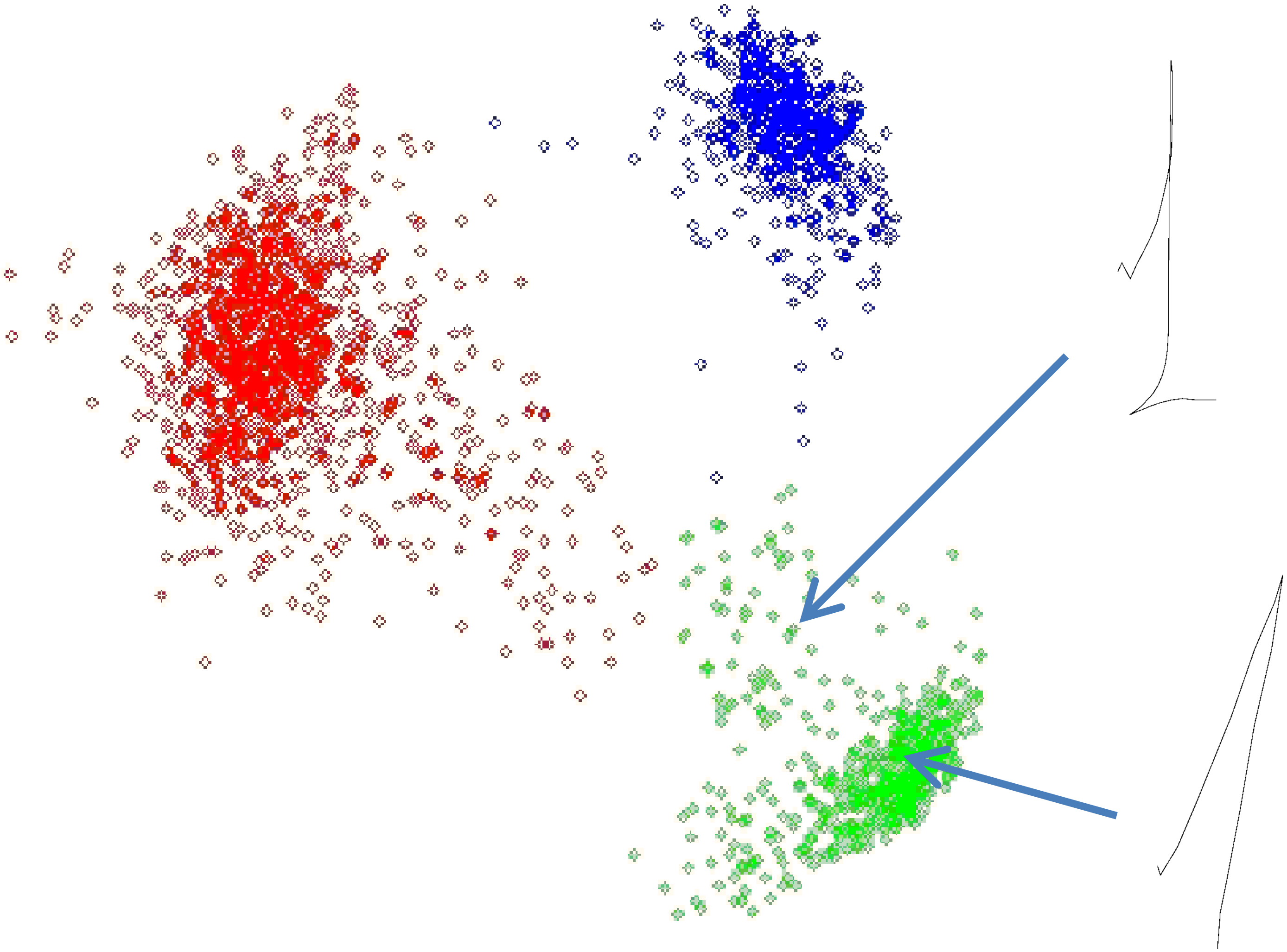
Life in an Inner Product Space

- With the Legendre-Sobolev inner product we have
  - Low dimensional rep for curves (10 + 10 + 1)
  - Compact rep of samples ~ 160 bits [G+W 2009]
  - **>99% linear separability => convexity of classes**
  - A useful notion of distance between curves *that is very fast to compute*

Linear Separability



Linear Separability



Linear Separability

- Can separate N classes with $N(N-1)$ SVM planes.
- Each class is then (mostly) within its own convex polyhedral cell.
- Can classify either by
  - SVM majority voting + run-off elections (96%)
  - Distance to convex hull of *k* nearest neighbours (97.5%).
  - On-line computation.

Recognition

- Some classification methods compute the distance between the input curve and models.

  E.g. Elastic matching with DTW takes time up to quadratic in the number of sample points and linear in the number of models.

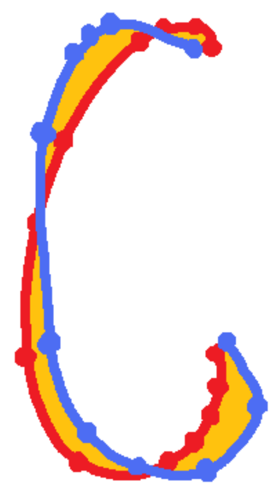
- Many tricks and heuristics to improve on this.

  E.g. Limit amount of dynamic time warping,
  pre-classify based on features, ...

- We can do substantially better.

Distance Between Curves

- **Elastic matching:**
- Approximate the variation between curves by some fn of distances between sample points.
- May be coordinate curves or curves in a jet space.



- Sequence alignment
- Interpolation ("resampling")

- **Why not just calculate the area?**
- This is very fast in ortho. series representation.

Distance Between Curves

$$\bar{x}(t) = x(t) + \xi(t) \qquad \xi(t) = \sum_{i=0}^{\infty} \xi_i \phi_i(t), \qquad \phi_i \text{ ortho on } [a,b] \text{ with } w(t) = 1.$$

$$\bar{y}(t) = y(t) + \eta(t) \qquad \eta(t) = \sum_{i=0}^{\infty} \eta_i \phi_i(t)$$

$$\rho^2(C, \bar{C}) = \int_a^b \left[\left(x(t) - \bar{x}(t)\right)^2 + \left(y(t) - \bar{y}(t)\right)^2\right] dt$$

$$= \int_a^b [\xi(t)^2 + \eta(t)^2] dt$$

$$\approx \int_a^b \left[\sum_{i=0}^{d} \xi_i^2 \phi_i^2(t) + \text{cross terms} + \sum_{i=0}^{d} \eta_i^2 \phi_i^2(t) + \text{cross terms}\right] dt$$

$$= \sum_{i=0}^{d} \xi_i^2 + \sum_{i=0}^{d} \eta_i^2$$

Comparison of Candidate to Models

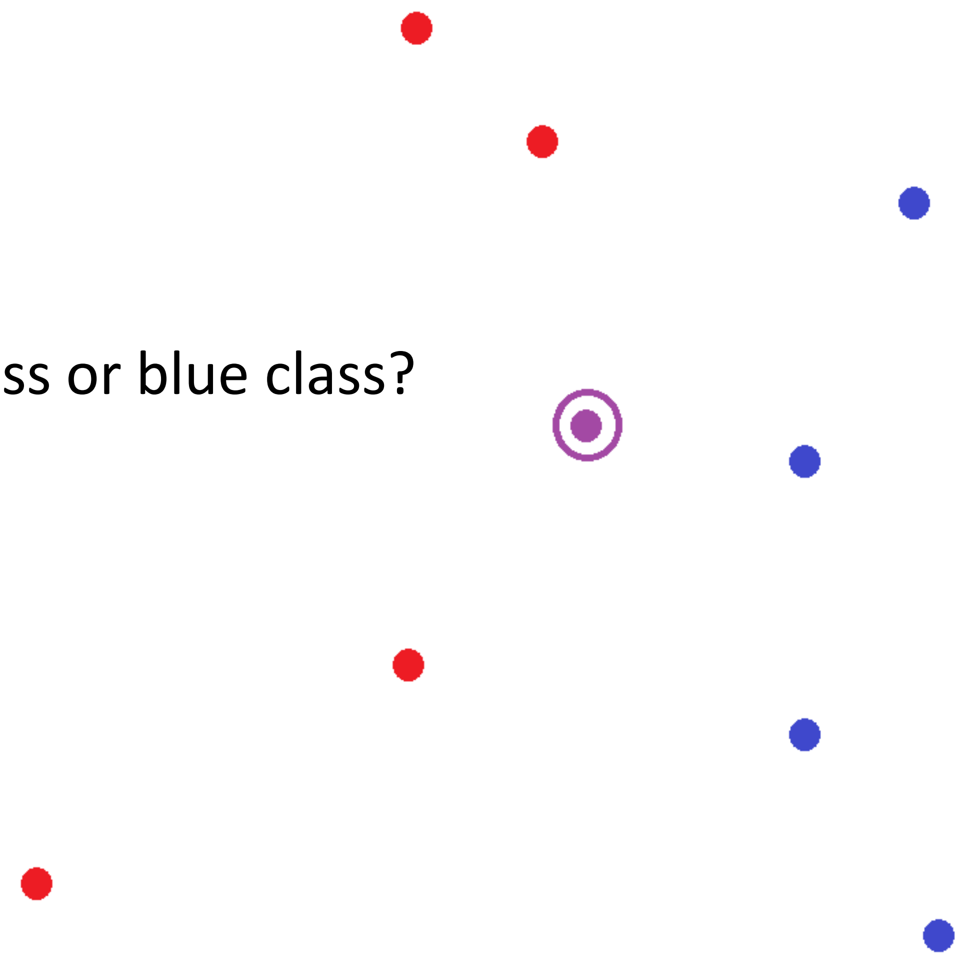
- Use Euclidean distance in the coefficient space.

- *Just as accurate* as elastic matching.
- *Much less expensive.*
- Linear in d, the degree of the approximation.
  < 3 d machine instructions (30ns) *vs* several thousand!
- Can trace through SVM-induced cells incrementally.

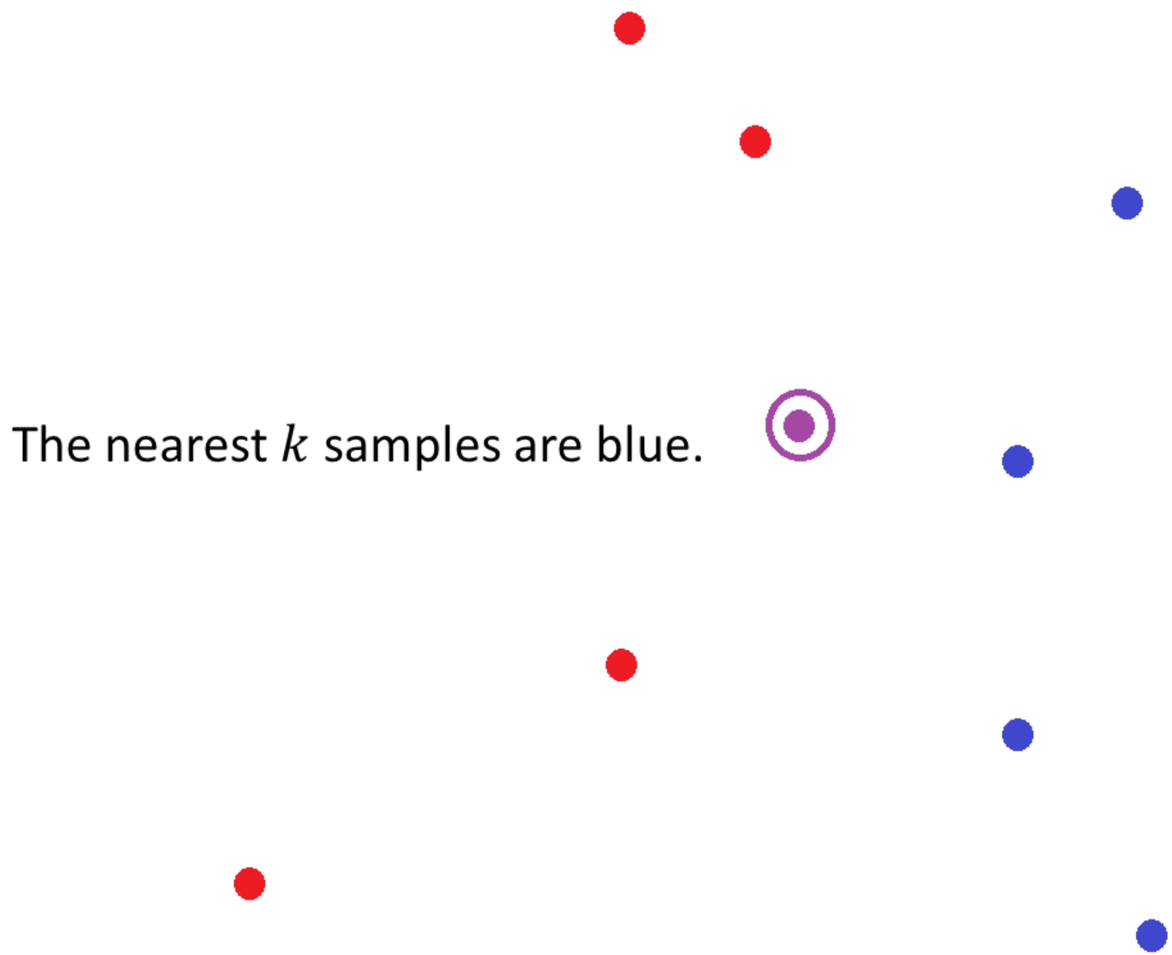
- Normed space for characters gives other advantages.

Choosing between Alternatives

Red class or blue class?

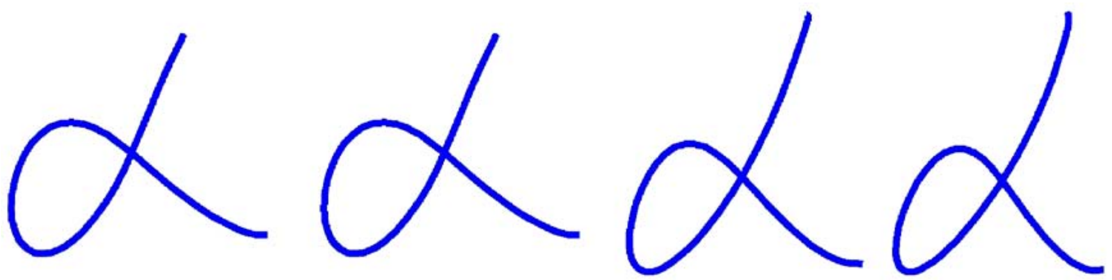


Choosing between Alternatives



The Joy of Convex

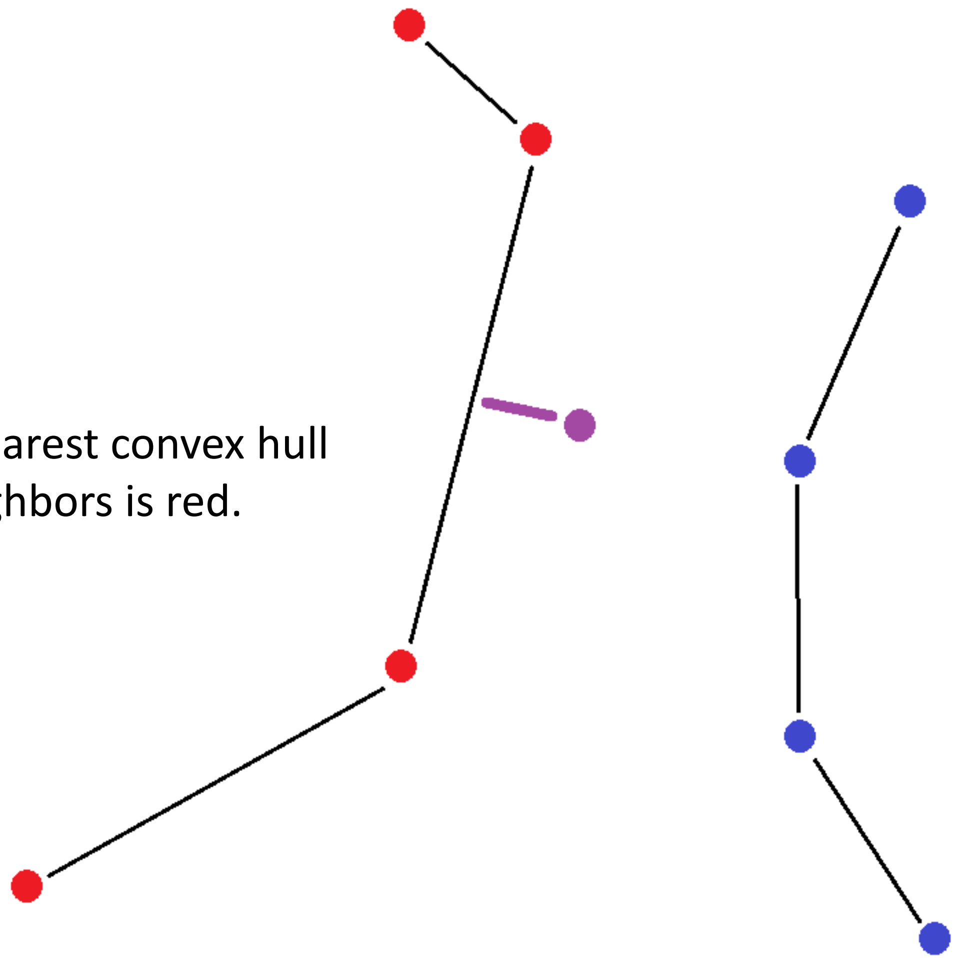
- Convexity $\Rightarrow$ Linear homotopies stay within a class



$$C = (1 - t)\,A + t\,B$$

- Can compute distance of a sample to this line
- Distance to convex hull of nearest neighbors in class gives best recognition [Golubitsky+SMW 2009,2010]

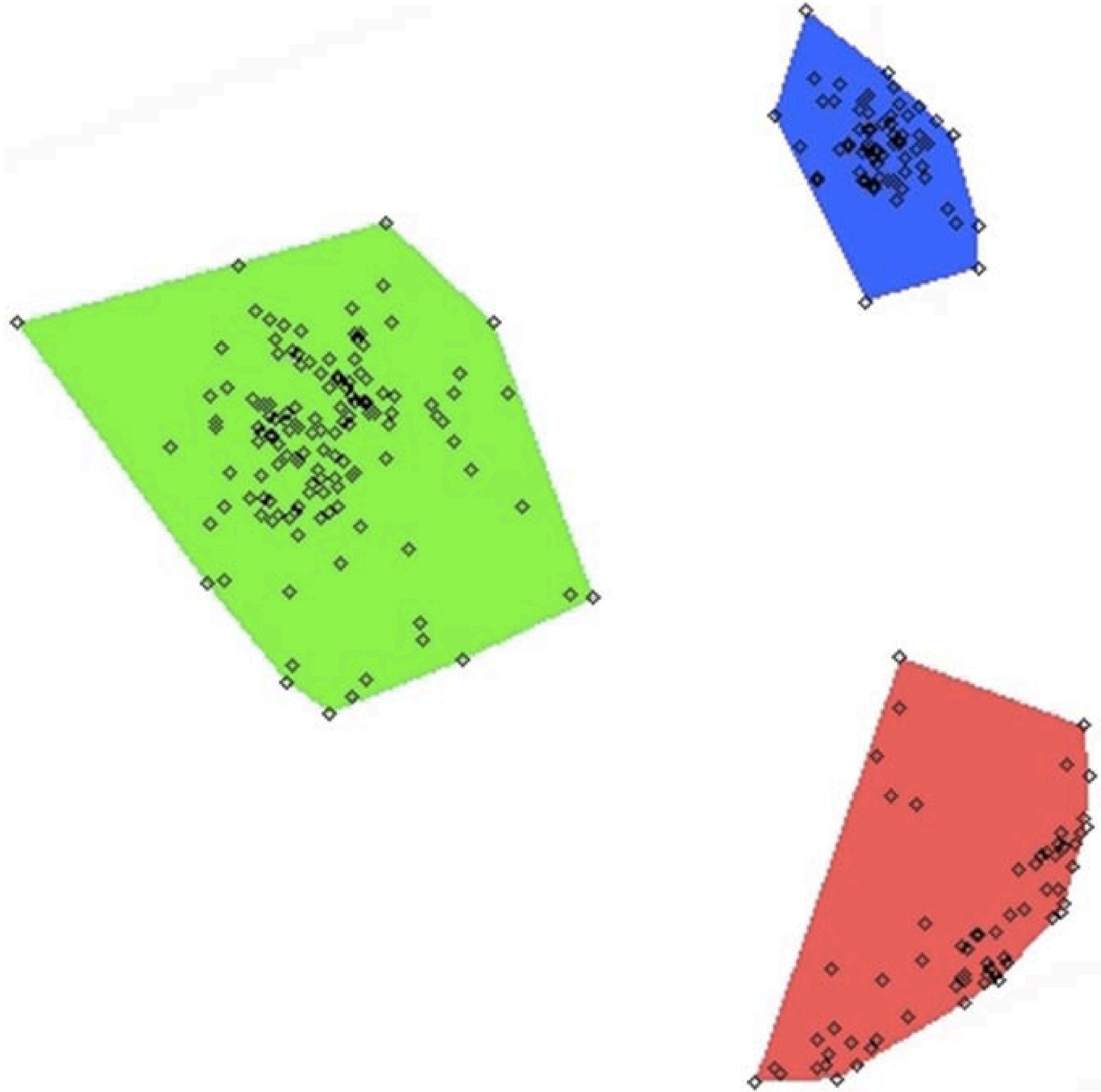
Choosing between Alternatives

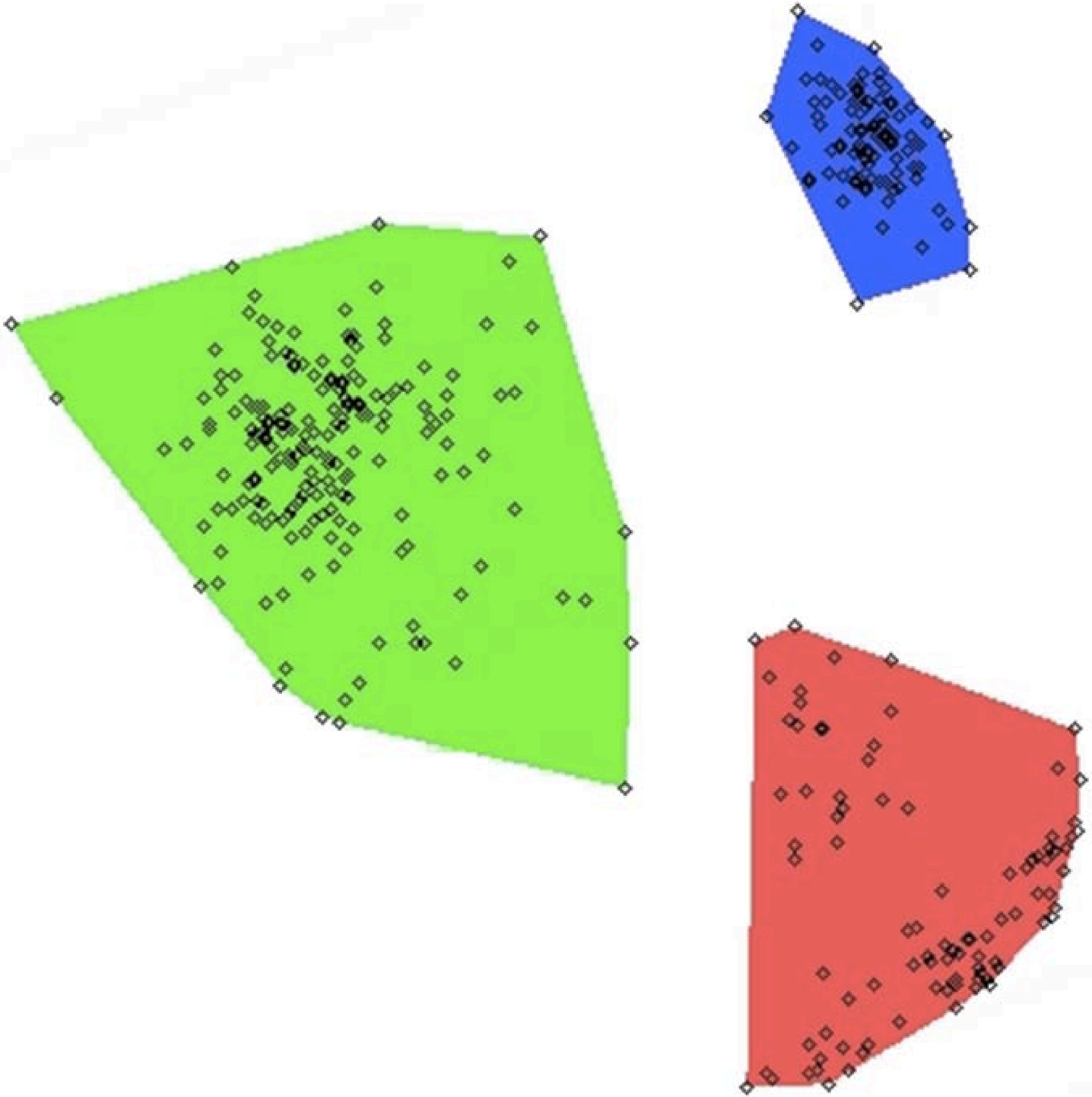


The nearest convex hull
of neighbors is red.

Training

- Using CHKNN allows training with relatively few samples. (Dozens vs Thousands per class)

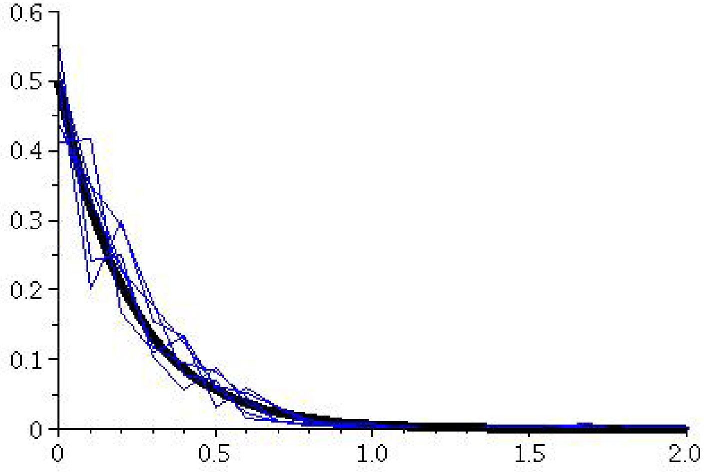




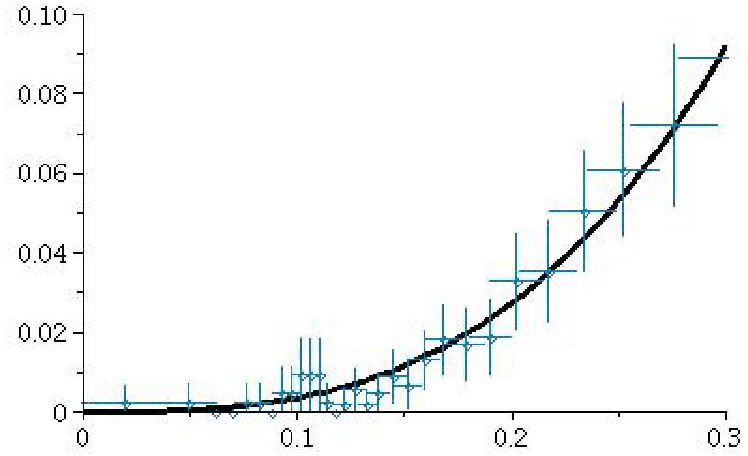
Recognition Summary

- Database of samples ⇒ set of LS points
- Character to recognize ⇒ Integrate moments as being written
  - Lin. trans. to obtain one point in LS space
  - Classify by distance to convex hull of k-NN.

Error Rates as Fn of Distance



SVM



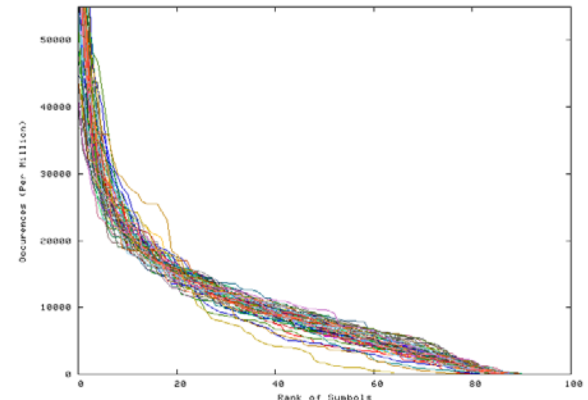
Convex Hull

- Error rate as fn of distance gives confidence measure for classifiers [MKM – Golubitsky + SMW 2009]

Combining with Statistical Info

- Empirical confidence on classifiers allows geometric recognition of isolated symbols
  to be combined with statistical methods.

- Domain-specific *n*-gram information:
  - Research mathematics –
    20,000 articles from arXiv
    [MKM -- So+SMW 2005]
  - 2nd year engineering math –
    most popular textbooks
    [DAS -- SMW 2008]
  - Inverse problem –
    identifying area via *n*-gram freq! [DML -- SMW 2008]



Deciding with Confidence Measure

Symbol X in an Expression E

Symbol Recognizer:
$X \in$ Class$_1$ **with Conf** ξ_1

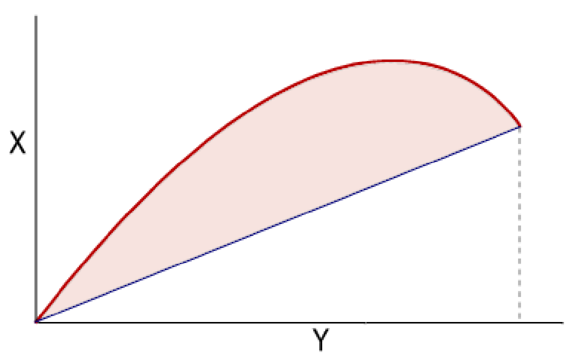
Context-Based Predictor:
$X \in$ Class$_2$ **with Conf** ξ_2

$X \in$ Class$_i$ $\Leftrightarrow$ $\xi_i = \max(\xi_1, \xi_2)$

Orientation and Shear

- Reco when writing at an angle, or with slanted chars.

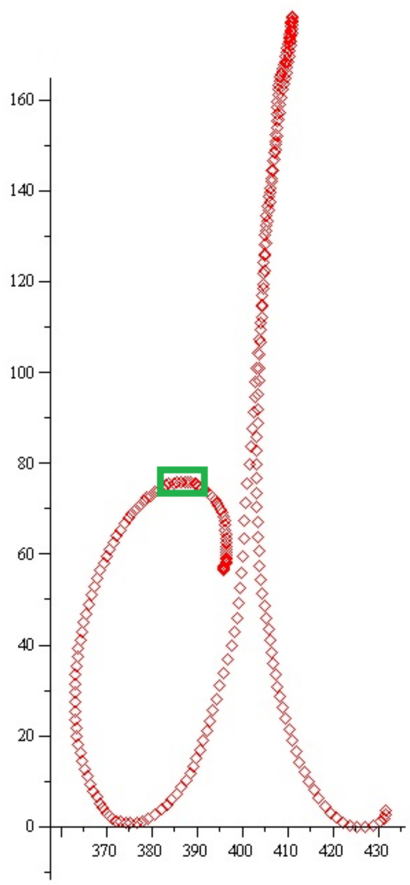
- Instead of taking ortho series of coord fns $x(\lambda)$ and $y(\lambda)$, use ortho series of integral invariants of these.
  [Golubitsky, Mazalov, SMW 2009 rotn, 2010 shear]

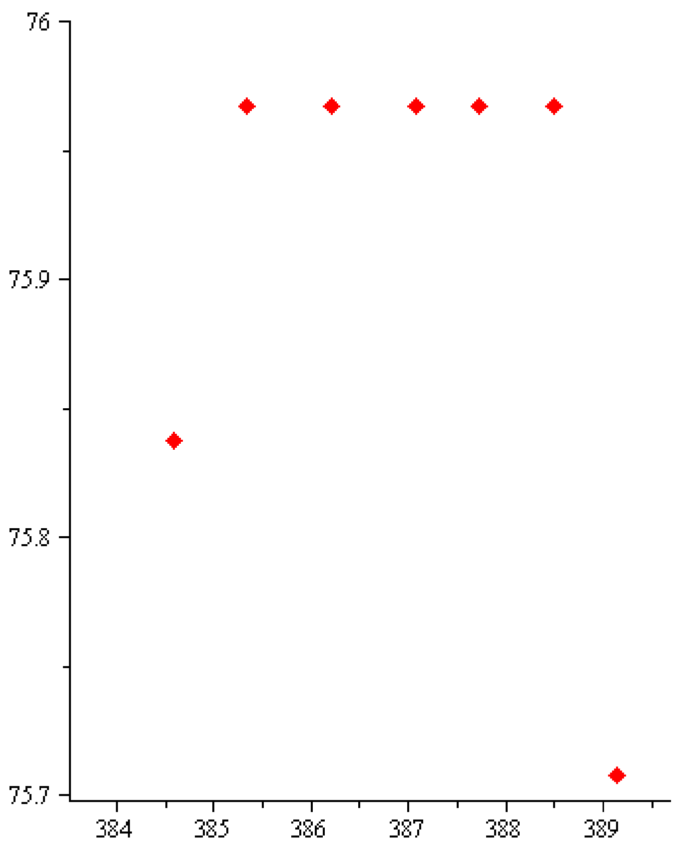


$$I_0(\lambda) = radius \qquad I_1(\lambda) = area$$

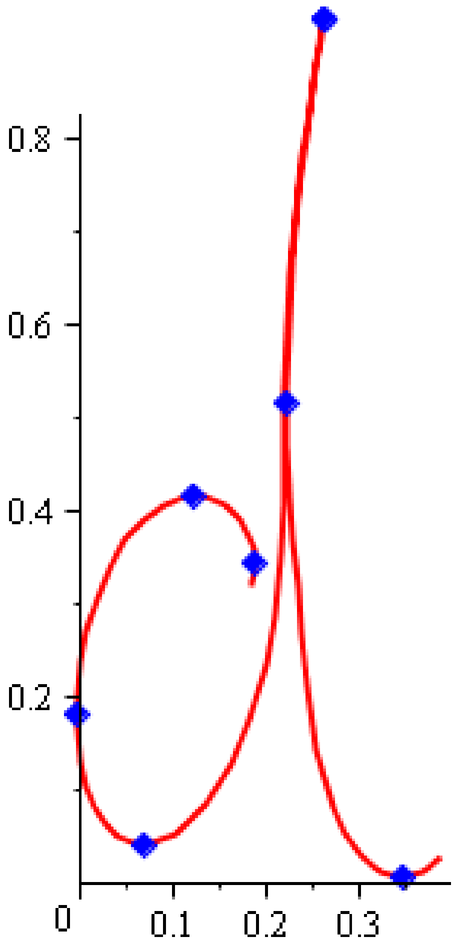
$$I_{k>1}(\lambda) = more\ complicated\ integrals$$

SNC for Features





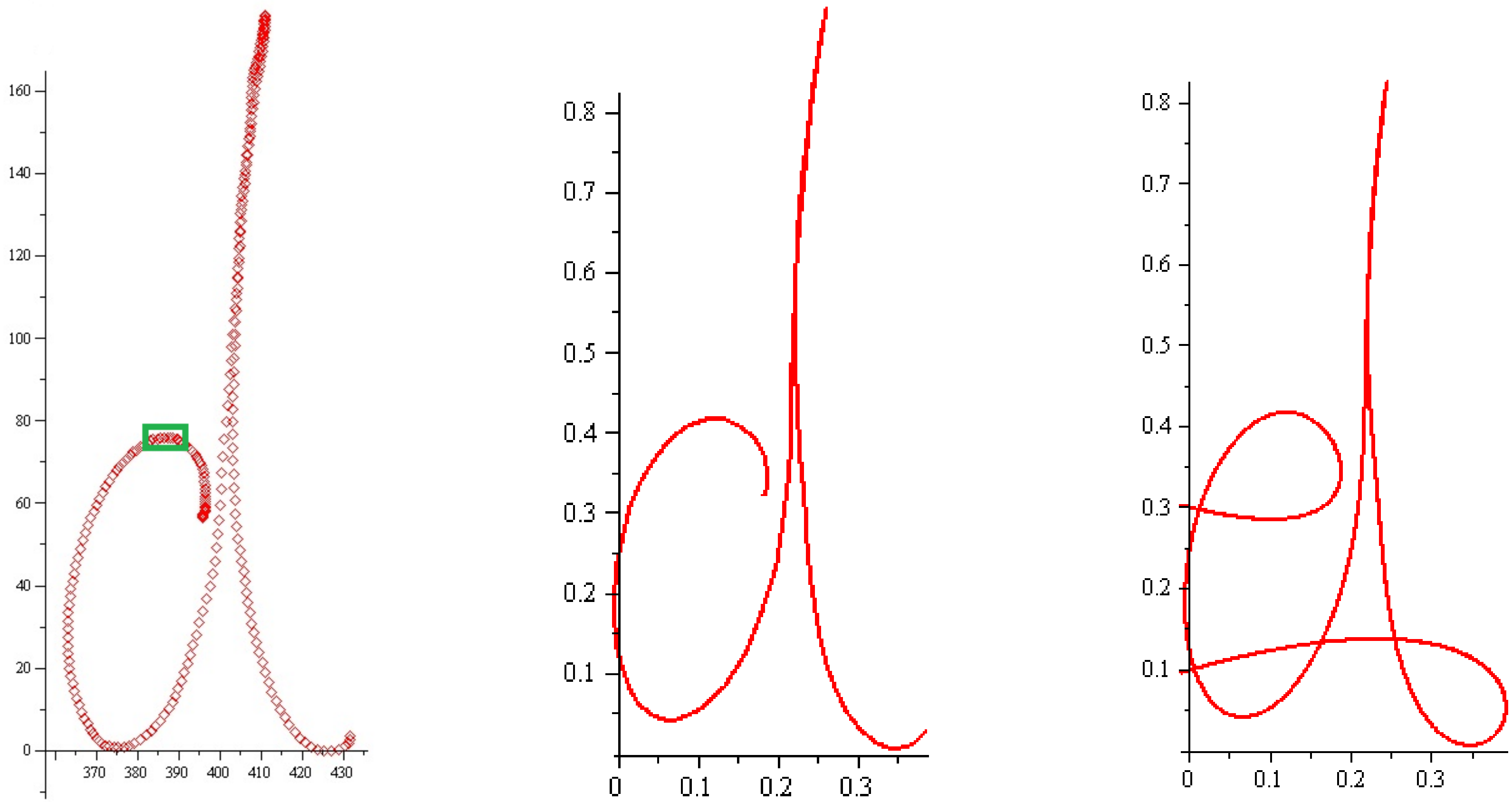
Sensible Critical Points



- Functional approx uses non-local information
- Puts critical points where they should be.
- Univ. polynomial root finding.

$$\frac{dx}{d\lambda} = 0 \quad \frac{dy}{d\lambda} = 0$$

Representations



SNC Problems

- Want small perturbations wrt LS norm.
- Transformation between LS basis and monomial basis ill-conditioned.
- Want to compute resultants, etc, without transforming to monomial basis.
- Can use degree-grading to push some arguments through. How far can we take this?

Conclusions

- Ask what are we really trying to do.
- Work with ink traces as curves,
  rather than as collections of sample points.

- Admits powerful analytic tools.
- Have useful geometry on space of curves.

- Gives device/resolution independence.
- Gives faster algorithms.
- Gives useful insights.

Thanks

Bruce Char
Joseph Choi
Michael Friesen
Oleg Golubitsky
Rui Hu
Vadim Mazalov
Jeliasko Polihronov
Elena Smirnova
Clare So
Coby Viner

Maplesoft
Microsoft
MITACS
NSERC