

Reconocimiento de Escritura Manuscrita (Online Handwriting Recognition)

PABLO SPECIALE

Tesina de Grado
(Septiembre 2011)



*Lic. en Cs. de la Computación
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario*

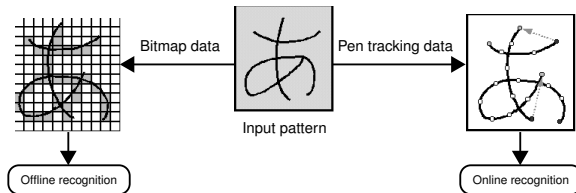
Director: Dr. Juan Carlos Gomez¹

Co-director: Dr. Pablo Granitto²

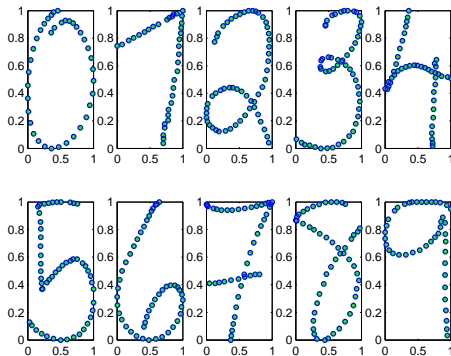
¹ *Procesamiento de Señales Multimedia, CIFASIS*

² *Aprendizaje Automatizado y Aplicaciones, CIFASIS*

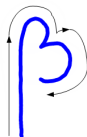
Offline vs Online



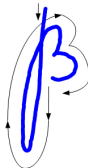
Base de Datos (dígitos)



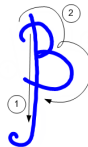
Simple y Multi-Trazos



Single-Stroke

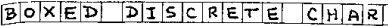


Single-Stroke

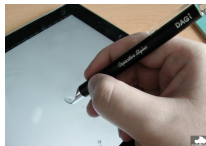
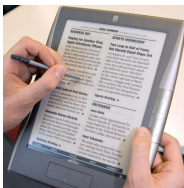


Multi-Stroke

Segmentación

1. 
2. Spaced Discrete Characters
3. Run-on discretely written characters
4. pure cursive script writing
5. Mixed Cursive and Discrete

Motivación



Reconocimiento de escritura para todos



Expresiones Matemáticas

Handwriting

$$\int \frac{(3x^2+2)\sin(x^3+2x-1)}{\cos(x^3+2x-1)} dx$$

Usando L^AT_EX 2_ε

```
\int {\frac { \left( 3\,{x}^{2}+2 \right)
      \sin \left( {x}^{3}+2\,x-1 \right) }
      { \cos \left( {x}^{3}+2\,x-1 \right) }
      } ~ dx
```


Expresiones Matemáticas

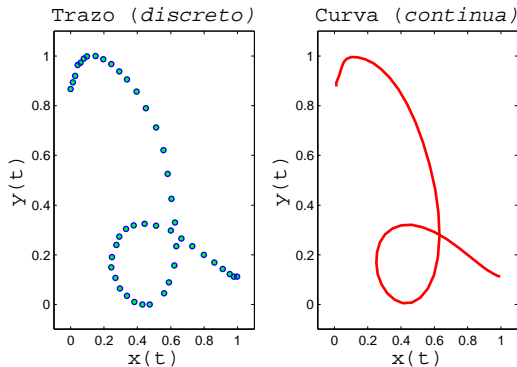
Handwriting

$$\int \frac{(3x^2+2)\sin(x^3+2x-1)}{\cos(x^3+2x-1)} dx$$

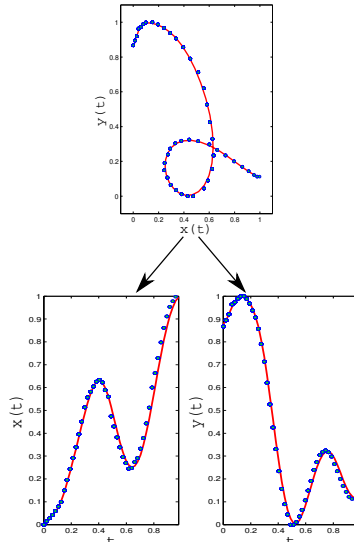
Usando L^AT_EX 2_ε

```
\int {\frac { \left( 3\,{x}^{2}+2 \right) \sin \left( {x}^{3}+2\,x-1 \right) }{ \cos \left( {x}^{3}+2\,x-1 \right) }} ~ dx
```

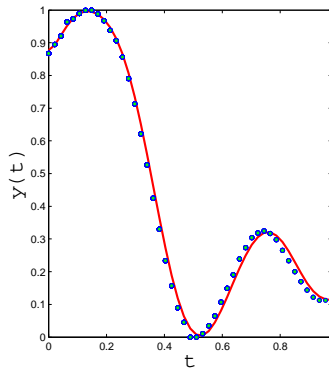
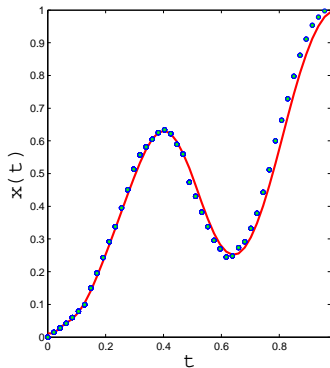
Trazos discretos como curvas continuas



Curvas paramétricas



Curvas paramétricas



Aproximaciones

con Bases de Polinomios Ortogonales

$$\begin{cases} x(t) \approx \sum_{i=0}^d \alpha_i B_i(t) \\ y(t) \approx \sum_{i=0}^d \beta_i B_i(t) \end{cases}$$

donde $\{B_i\}$ es base de polinomios ortogonales

Features

$$(\alpha_0, \dots, \alpha_d, \beta_0, \dots, \beta_d)$$

Aproximaciones

con Bases de Polinomios Ortogonales

$$\begin{cases} x(t) \approx \sum_{i=0}^d \alpha_i B_i(t) \\ y(t) \approx \sum_{i=0}^d \beta_i B_i(t) \end{cases}$$

donde $\{B_i\}$ es base de polinomios ortogonales

Features

$$(\alpha_0, \dots, \alpha_d, \beta_0, \dots, \beta_d)$$

Aproximaciones

con Bases de Polinomios Ortogonales

$$\begin{cases} x(t) \approx \sum_{i=0}^d \alpha_i B_i(t) \\ y(t) \approx \sum_{i=0}^d \beta_i B_i(t) \end{cases}$$

donde $\{B_i\}$ es base de polinomios ortogonales

Features

$$(\alpha_0, \dots, \alpha_d, \beta_0, \dots, \beta_d)$$

Producto interno:

$$\langle B_i, B_j \rangle \doteq \int_a^b B_i(t) B_j(t) w(t) dt$$

donde $w(t)$ es una función peso.

Ortogonalidad:

- $\langle B_i, B_j \rangle = 0, \quad \forall i \neq j \quad \implies \quad B_i \text{ y } B_j \text{ son ortogonales}$
- Si además, $\langle B_i, B_i \rangle = 1 \quad \implies \quad B_i \text{ y } B_j \text{ son ortonormales.}$

Producto interno:

$$\langle B_i, B_j \rangle \doteq \int_a^b B_i(t) B_j(t) w(t) dt$$

donde $w(t)$ es una función peso.

Ortogonalidad:

- $\langle B_i, B_j \rangle = 0, \quad \forall i \neq j \quad \implies \quad B_i \text{ y } B_j \text{ son ortogonales}$
- Si además, $\langle B_i, B_i \rangle = 1 \quad \implies \quad B_i \text{ y } B_j \text{ son ortonormales.}$

Producto interno:

$$\langle B_i, B_j \rangle \doteq \int_a^b B_i(t) B_j(t) w(t) dt$$

donde $w(t)$ es una función peso.

Ortogonalidad:

- $\langle B_i, B_j \rangle = 0, \quad \forall i \neq j \quad \implies \quad B_i \text{ y } B_j \text{ son ortogonales}$
- Si además, $\langle B_i, B_i \rangle = 1 \quad \implies \quad B_i \text{ y } B_j \text{ son ortonormales.}$

Polinomios de Legendre

Producto interno:

$$\langle L_i, L_j \rangle = \int_{-1}^1 L_i(t) L_j(t) w(t) dt$$

Si tomamos como función peso $w(t) = 1$ en la definición anterior

$$\langle L_i, L_j \rangle = \int_{-1}^1 L_i(t) L_j(t) dt$$

se pueden generar los polinomios de Legendre $\{L_j\}$ con el proceso de ortogonalización de **Gram-Schmidt** en el intervalo $[-1, 1]$.

Polinomios de Legendre

Producto interno:

$$\langle L_i, L_j \rangle = \int_{-1}^1 L_i(t) L_j(t) w(t) dt$$

Si tomamos como función peso $w(t) = 1$ en la definición anterior

$$\langle L_i, L_j \rangle = \int_{-1}^1 L_i(t) L_j(t) dt$$

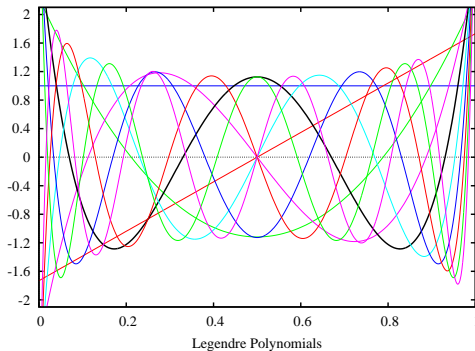
se pueden generar los polinomios de Legendre $\{L_i\}$ con el proceso de ortogonalización de **Gram-Schmidt** en el intervalo $[-1, 1]$.

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i c_{ij} t^j$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i c_{ij} t^j$$



Serie de funciones ortonormales

Serie

$$f(t) = \sum_{i=0}^{\infty} \alpha_i L_i(t)$$

Coeficientes

$$\alpha_i = \langle f, L_i \rangle$$

Aproximación

$$f(t) \approx \sum_{i=0}^d \alpha_i L_i(t)$$

Features

$$\begin{cases} x(t) \approx \sum_{i=0}^d \alpha_i L_i(t) \\ y(t) \approx \sum_{i=0}^d \beta_i L_i(t) \end{cases}$$
$$(\alpha_0, \dots, \alpha_d, \beta_0, \dots, \beta_d)$$

Serie de funciones ortonormales

Serie

$$f(t) = \sum_{i=0}^{\infty} \alpha_i L_i(t)$$

Coeficientes

$$\alpha_i = \langle f, L_i \rangle$$

Aproximación

$$f(t) \approx \sum_{i=0}^d \alpha_i L_i(t)$$

Features

$$\begin{cases} x(t) \approx \sum_{i=0}^d \alpha_i L_i(t) \\ y(t) \approx \sum_{i=0}^d \beta_i L_i(t) \end{cases}$$
$$(\alpha_0, \dots, \alpha_d, \beta_0, \dots, \beta_d)$$

Serie de funciones ortonormales

Serie

$$f(t) = \sum_{i=0}^{\infty} \alpha_i L_i(t)$$

Coeficientes

$$\alpha_i = \langle f, L_i \rangle$$

Aproximación

$$f(t) \approx \sum_{i=0}^d \alpha_i L_i(t)$$

Features

$$\begin{cases} x(t) \approx \sum_{i=0}^d \alpha_i L_i(t) \\ y(t) \approx \sum_{i=0}^d \beta_i L_i(t) \end{cases}$$

$$(\alpha_0, \dots, \alpha_d, \beta_0, \dots, \beta_d)$$

Serie de funciones ortonormales

Serie

$$f(t) = \sum_{i=0}^{\infty} \alpha_i L_i(t)$$

Coeficientes

$$\alpha_i = \langle f, L_i \rangle$$

Aproximación

$$f(t) \approx \sum_{i=0}^d \alpha_i L_i(t)$$

Features

$$\begin{cases} x(t) \approx \sum_{i=0}^d \alpha_i L_i(t) \\ y(t) \approx \sum_{i=0}^d \beta_i L_i(t) \end{cases}$$

$$(\alpha_0, \dots, \alpha_d, \beta_0, \dots, \beta_d)$$

Momentos

Definición

Los momentos de una función f definida en $[0, 1]$ son

$$\mu_k \doteq \int_0^1 f(t) t^k dt$$

Hausdorff Moment Problem

Recuperar a partir de una secuencia finita de momentos $\{\mu_k\}_{k=0,1,2,\dots,d}$ una función f en el dominio $[0, 1]$.

Momentos

Definición

Los momentos de una función f definida en $[0, 1]$ son

$$\mu_k \doteq \int_0^1 f(t) t^k dt$$

Hausdorff Moment Problem

Recuperar a partir de una secuencia finita de momentos $\{\mu_k\}_{k=0,1,2,\dots,d}$ una función f en el dominio $[0, 1]$.

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

$$f(t) \approx \sum_{i=0}^d \alpha_i L_i(t)$$

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

$$f(t) \approx \sum_{i=0}^d \alpha_i L_i(t)$$
$$\alpha_i = \langle f, L_i \rangle$$

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

$$f(t) \approx \sum_{i=0}^d \alpha_i L_i(t)$$
$$\alpha_i = \langle f, L_i \rangle$$
$$= \int_0^1 f(t) L_i(t) dt$$

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

$$f(t) \approx \sum_{i=0}^d \alpha_i L_i(t)$$

$$\begin{aligned} \alpha_i &= \langle f, L_i \rangle \\ &= \int_0^1 f(t) L_i(t) dt \\ &= \int_0^1 f(t) \left(\sum_{j=0}^i C_{ij} t^j \right) dt \end{aligned}$$

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

$$\begin{aligned} f(t) &\approx \sum_{i=0}^d \alpha_i L_i(t) \\ \alpha_i &= \langle f, L_i \rangle \\ &= \int_0^1 f(t) L_i(t) dt \\ &= \int_0^1 f(t) \left(\sum_{j=0}^i C_{ij} t^j \right) dt \\ &= \sum_{j=0}^i C_{ij} \left(\int_0^1 f(t) t^j dt \right) \end{aligned}$$

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

$$\begin{aligned} f(t) &\approx \sum_{i=0}^d \alpha_i L_i(t) \\ \alpha_i &= \langle f, L_i \rangle \\ &= \int_0^1 f(t) L_i(t) dt \\ &= \int_0^1 f(t) \left(\sum_{j=0}^i C_{ij} t^j \right) dt \\ &= \sum_{j=0}^i C_{ij} \underbrace{\left(\int_0^1 f(t) t^j dt \right)}_{\mu_j} \end{aligned}$$

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

$$\begin{aligned} f(t) &\approx \sum_{i=0}^d \alpha_i L_i(t) \\ \alpha_i &= \langle f, L_i \rangle \\ &= \int_0^1 f(t) L_i(t) dt \\ &= \int_0^1 f(t) \left(\sum_{j=0}^i C_{ij} t^j \right) dt \\ &= \sum_{j=0}^i C_{ij} \underbrace{\left(\int_0^1 f(t) t^j dt \right)}_{\mu_j} = \sum_{j=0}^i C_{ij} \mu_j \end{aligned}$$

Producto interno

$$\langle f, g \rangle = \int_0^1 f(t) g(t) dt$$

Polinomios de Legendre

$$L_i(t) = \sum_{j=0}^i C_{ij} t^j$$

Momentos

$$\mu_j = \int_0^1 f(t) t^j dt$$

$$\begin{aligned} f(t) &\approx \sum_{i=0}^d \alpha_i L_i(t) \\ \alpha_i &= \langle f, L_i \rangle \\ &= \int_0^1 f(t) L_i(t) dt \\ &= \int_0^1 f(t) \left(\sum_{j=0}^i C_{ij} t^j \right) dt \\ &= \sum_{j=0}^i C_{ij} \underbrace{\left(\int_0^1 f(t) t^j dt \right)}_{\mu_j} = \sum_{j=0}^i C_{ij} \mu_j \end{aligned}$$

$$\alpha_i = \sum_{j=0}^i C_{ij} \mu_j$$

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_d \end{bmatrix} = \begin{bmatrix} C_{00} & & & \\ C_{10} & C_{11} & & \\ \vdots & \vdots & \ddots & \\ C_{d0} & C_{d1} & \dots & C_{dd} \end{bmatrix} \cdot \begin{bmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_d \end{bmatrix}$$

Matricialmente

$$\alpha = \mathbf{C} \mu$$

$$\alpha_i = \sum_{j=0}^i C_{ij} \mu_j$$

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_d \end{bmatrix} = \begin{bmatrix} C_{00} & & & \\ C_{10} & C_{11} & & \\ \vdots & \vdots & \ddots & \\ C_{d0} & C_{d1} & \dots & C_{dd} \end{bmatrix} \cdot \begin{bmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_d \end{bmatrix}$$

Matricialmente

$$\alpha = \mathbf{C} \mu$$

$$\alpha_i = \sum_{j=0}^i C_{ij} \mu_j$$

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_d \end{bmatrix} = \begin{bmatrix} C_{00} & & & \\ C_{10} & C_{11} & & \\ \vdots & \vdots & \ddots & \\ C_{d0} & C_{d1} & \dots & C_{dd} \end{bmatrix} \cdot \begin{bmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_d \end{bmatrix}$$

Matricialmente

$$\alpha = \mathbf{C} \mu$$

Calculo numérico de los momentos

$$f(t) \approx \sum_{i=0}^d \alpha_i L_i(t)$$
$$\alpha = \mathbf{C} \mu$$

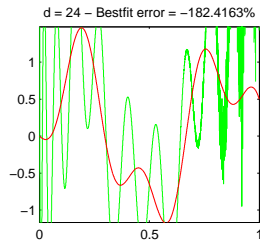
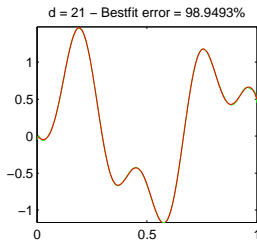
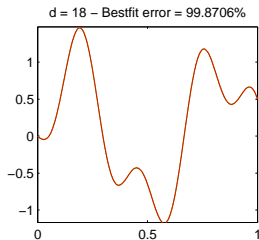
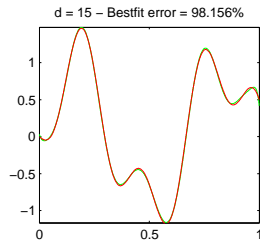
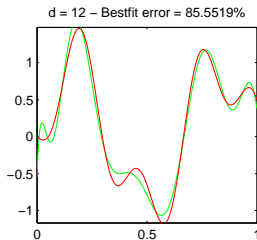
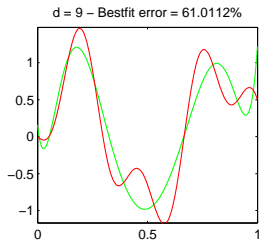
$$\mu_k = \int_0^1 f(t) t^k dt$$

Calculo numérico de los momentos

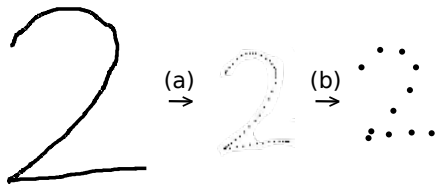
$$f(t) \approx \sum_{i=0}^d \alpha_i L_i(t)$$
$$\alpha = \mathbf{C} \mu$$

$$\mu_k = \int_0^1 f(t) t^k dt$$

Inestabilidad numérica



Preproceso



(a) Suavizado y Resizing, (b) Resampling

Preproceso

Invariante a escala

$$\hat{\alpha} = \frac{\alpha}{\|\alpha\|}$$

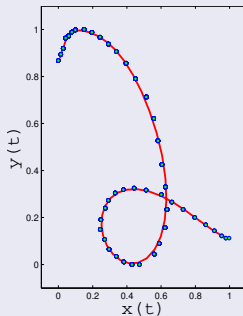
Suavizado

Preproceso

Invariante a escala

$$\hat{\alpha} = \frac{\alpha}{\|\alpha\|}$$

Suavizado



Reparametrización por longitud de arco

Variaciones en la velocidad de escritura

Parametrización por longitud de arco es usualmente preferible, pues **no es afectada por variaciones en la velocidad de escritura**. Esta puede ser expresada como:

$$\text{arc-length}(t) = \int_0^t \sqrt{(x'(\lambda))^2 + (y'(\lambda))^2} d\lambda$$

Pseudo-Inversa

Aproximación por Mínimos Cuadrados

$$f(t) \approx \sum_{i=0}^d \alpha_i B_i(t)$$

Una forma alternativa para calcular los α_i es mediante la Pseudo-Inversa de Moore-Penrose.

Ventajas

- No presenta inestabilidad numérica
- Polinomios cualesquiera

Desventajas

- Cálculo offline
- Implementación más complicada

Pseudo-Inversa

Aproximación por Mínimos Cuadrados

$$f(t) \approx \sum_{i=0}^d \alpha_i B_i(t)$$

Una forma alternativa para calcular los α_i es mediante la Pseudo-Inversa de Moore-Penrose.

Ventajas

- No presenta inestabilidad numérica
- Polinomios cualesquiera

Desventajas

- Cálculo offline
- Implementación más complicada

Pseudo-Inversa

Aproximación por Mínimos Cuadrados

$$f(t) \approx \sum_{i=0}^d \alpha_i B_i(t)$$

Una forma alternativa para calcular los α_i es mediante la Pseudo-Inversa de Moore-Penrose.

Ventajas

- No presenta inestabilidad numérica
- Polinomios cualesquiera

Desventajas

- Cálculo offline
- Implementación más complicada

Pseudo-Inversa

Aproximación por Mínimos Cuadrados

$$f(t) \approx \sum_{i=0}^d \alpha_i B_i(t)$$

Una forma alternativa para calcular los α_i es mediante la Pseudo-Inversa de Moore-Penrose.

Ventajas

- No presenta inestabilidad numérica
- Polinomios cualesquiera

Desventajas

- Calculo offline
- Implementación más complicada

Pseudo-Inversa

Aproximación por Mínimos Cuadrados

$$f(t) \approx \sum_{i=0}^d \alpha_i B_i(t)$$

Una forma alternativa para calcular los α_i es mediante la Pseudo-Inversa de Moore-Penrose.

Ventajas

- No presenta inestabilidad numérica
- Polinomios cualesquiera

Desventajas

- Calculo offline
- Implementación más complicada

Polinomios de Legendre-Sobolev

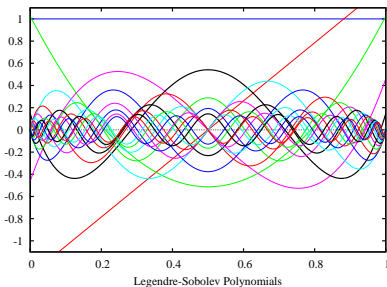
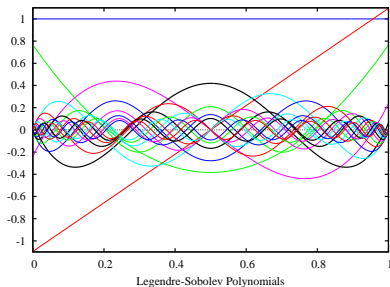
Producto interno (de Sobolev)

$$\langle f, g \rangle_{LS} = \int_0^1 f(t) g(t) dt + \mu \int_0^1 f'(t) g'(t) dt$$

Polinomios de Legendre-Sobolev

Producto interno (de Sobolev)

$$\langle f, g \rangle_{LS} = \int_0^1 f(t) g(t) dt + \mu \int_0^1 f'(t) g'(t) dt$$



k -NN

Distancia Euclidiana

$$\text{dist}(x, y)^2 = \sum_{i=0}^d (x_i - y_i)^2 = (x - y)(x - y)^T$$

Distancia de Hamming o Cityblock

$$\text{dist}(x, y) = \sum_{i=0}^d |x_i - y_i|$$

Distancia Mahalanobis¹

$$\text{dist}(x, y)^2 = (x - y) \Sigma^{-1} (x - y)^T$$

donde Σ es la matriz de covariancia.

¹ A diferencia de la distancia euclidiana, se tiene en cuenta la correlación entre las variables aleatorias

k -NN

Distancia Euclidiana

$$\text{dist}(x, y)^2 = \sum_{i=0}^d (x_i - y_i)^2 = (x - y)(x - y)^T$$

Distancia de Hamming o Cityblock

$$\text{dist}(x, y) = \sum_{i=0}^d |x_i - y_i|$$

Distancia Mahalanobis¹

$$\text{dist}(x, y)^2 = (x - y) \Sigma^{-1} (x - y)^T$$

donde Σ es la matriz de covariancia.

¹ A diferencia de la distancia euclidiana, se tiene en cuenta la correlación entre las variables aleatorias

k -NN

Distancia Euclidiana

$$\text{dist}(x, y)^2 = \sum_{i=0}^d (x_i - y_i)^2 = (x - y)(x - y)^T$$

Distancia de Hamming o Cityblock

$$\text{dist}(x, y) = \sum_{i=0}^d |x_i - y_i|$$

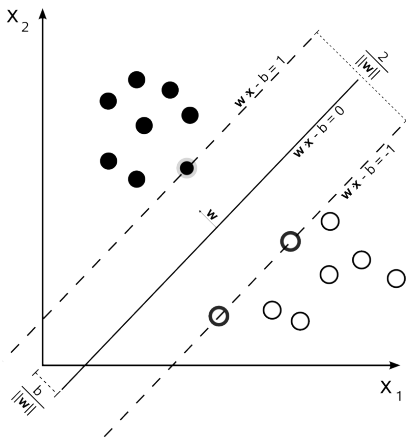
Distancia Mahalanobis¹

$$\text{dist}(x, y)^2 = (x - y) \Sigma^{-1} (x - y)^T$$

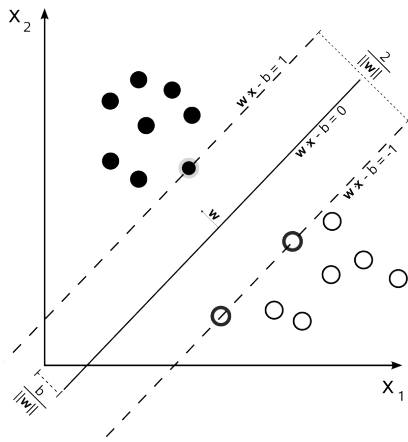
donde Σ es la matriz de covariancia.

¹ A diferencia de la distancia euclidiana, se tiene en cuenta la correlación entre las variables aleatorias

SVM lineal

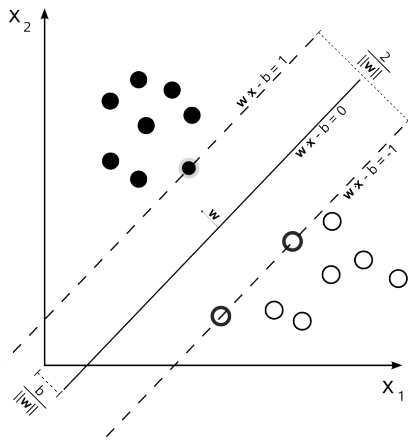


SVM lineal



$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n$$

SVM lineal

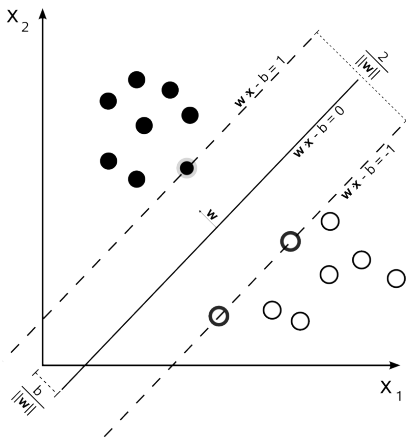


$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n$$

$$\omega^T \mathbf{x}_k - b \geq 1, \quad \text{para } y_k = 1$$

$$\omega^T \mathbf{x}_k - b \leq -1, \quad \text{para } y_k = -1$$

SVM lineal



$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n$$

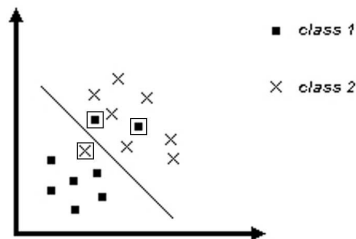
$$\omega^T \mathbf{x}_k - b \geq 1, \quad \text{para } y_k = 1$$

$$\omega^T \mathbf{x}_k - b \leq -1, \quad \text{para } y_k = -1$$

Pueden combinarse,

$$y_k(\omega^T \mathbf{x}_k - b) - 1 \geq 0, \quad k = 1, \dots, n$$

Problema de clasificación no separable



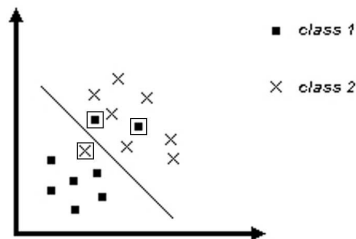
Variable Slack (ξ_k)

$$y_k(\omega^T x_k - b) \geq 1 - \xi_k$$

Formulación

$$\begin{aligned} \min_{\omega, b, \xi} \quad & \frac{1}{2} \|\omega\|^2 + C \sum_{k=1}^n \xi_k \\ \text{s.t.} \quad & y_k(\omega^T x_k - b) \geq 1 - \xi_k, \\ & \xi_k \geq 0, \end{aligned}$$

Problema de clasificación no separable



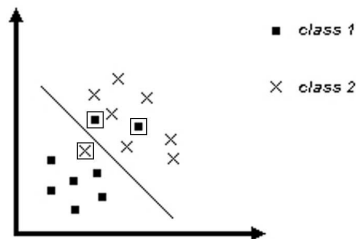
Variable Slack (ξ_k)

$$y_k(\omega^T x_k - b) \geq 1 - \xi_k$$

Formulación

$$\begin{aligned} \min_{\omega, b, \xi} \quad & \frac{1}{2} \|\omega\|^2 + C \sum_{k=1}^n \xi_k \\ \text{s.t.} \quad & y_k(\omega^T x_k - b) \geq 1 - \xi_k, \\ & \xi_k \geq 0, \end{aligned}$$

Problema de clasificación no separable



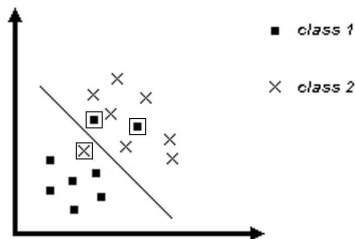
Variable Slack (ξ_k)

$$y_k(\omega^T x_k - b) \geq 1 - \xi_k$$

Formulación

$$\begin{aligned} \min_{\omega, b, \xi} \quad & \frac{1}{2} \|\omega\|^2 + C \sum_{k=1}^n \xi_k \\ \text{s.t.} \quad & y_k(\omega^T x_k - b) \geq 1 - \xi_k, \\ & \xi_k \geq 0, \end{aligned}$$

Problema de clasificación no separable



Variable Slack (ξ_k)

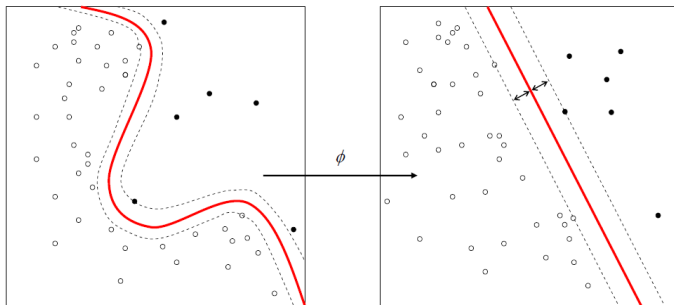
$$y_k(\omega^T x_k - b) \geq 1 - \xi_k$$

Formulación

$$\begin{aligned} \min_{\omega, b, \xi} \quad & \frac{1}{2} \|w\|^2 + \textcolor{red}{C} \sum_{k=1}^n \xi_k \\ \text{s.t.} \quad & y_k(\omega^T x_k - b) \geq 1 - \xi_k, \\ & \xi_k \geq 0, \end{aligned}$$

C: compensación entre errores de entrenamiento y los márgenes rígidos. Permite algunos errores en la clasificación a la vez que los penaliza.

SVM no lineal - Kernels



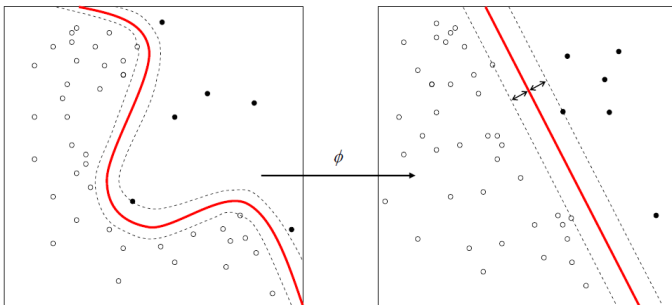
Kernels más usados

$$K_{linear}(x_k, x_\ell) = x_k^T x_\ell$$

$$K_{polinomial}(x_k, x_\ell) = (1 + x_k^T x_\ell)^d$$

$$K_{RBF}(x_k, x_\ell) = e^{-\frac{\|x_k - x_\ell\|^2}{\gamma^2}}$$

SVM no lineal - Kernels



Kernels más usados

$$K_{linear}(x_k, x_\ell) = x_k^T x_\ell$$

$$K_{polinomial}(x_k, x_\ell) = (1 + x_k^T x_\ell)^d$$

$$K_{RBF}(x_k, x_\ell) = e^{-\frac{\|x_k - x_\ell\|^2}{\gamma^2}}$$

SVM no lineal - Kernels

Formulación

$$\begin{aligned} \min_{\omega, b, \xi} \quad & \frac{1}{2} \|\omega\|^2 + C \sum_{k=1}^n \xi_k \\ \text{s.t.} \quad & y_k (\omega^T \phi(x_k) - b) \geq 1 - \xi_k, \\ & \xi_k \geq 0, \end{aligned}$$

Problema dual (clasificador)

$$y(x) = \text{sign} \left(\sum_{k=1}^n \alpha_k y_k K(x, x_k) + b \right)$$

SVM no lineal - Kernels

Formulación

$$\begin{aligned} \min_{\omega, b, \xi} \quad & \frac{1}{2} \|\omega\|^2 + C \sum_{k=1}^n \xi_k \\ \text{s.t.} \quad & y_k (\omega^T \phi(x_k) - b) \geq 1 - \xi_k, \\ & \xi_k \geq 0, \end{aligned}$$

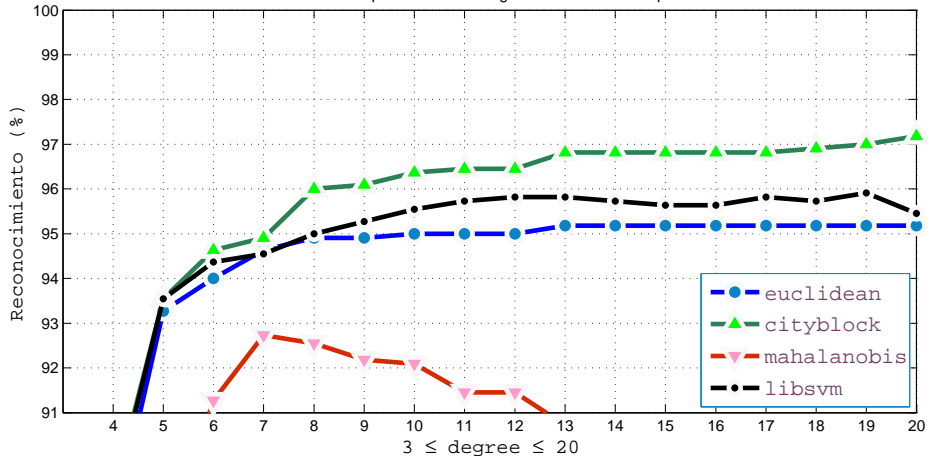
Problema dual (clasificador)

$$y(x) = \text{sign} \left(\sum_{k=1}^n \alpha_k y_k K(x, x_k) + b \right)$$

Resultados

Sin preproceso, Momentos, Polinomios de Legendre

Polynomials: Legendre – Method: Moments – Parameterization: time – Preprocessed: 0
libsvm options = -c 2048 -g 8 -e 0.1 -h 0 -b 0 -q

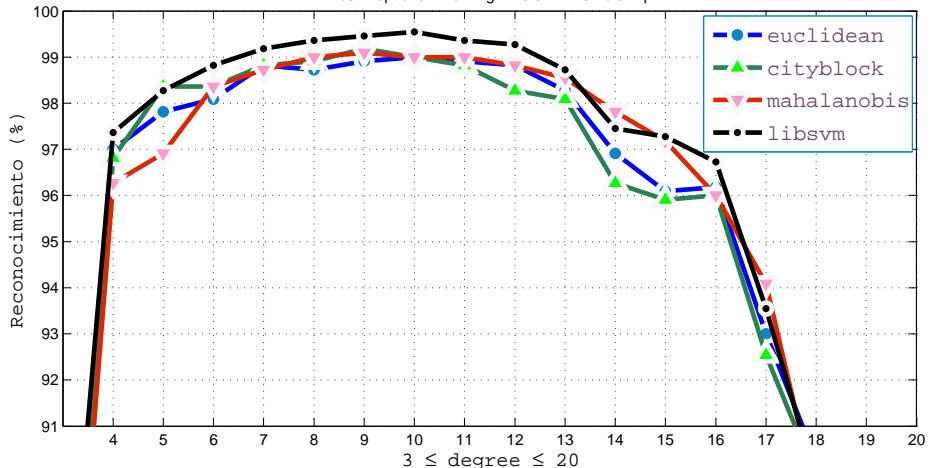


Resultados

Con preproceso, Pseudo-inversa, Polinomios de Legendre-Sobolev

Polynomails: Legendre-Sobolev – Method: Least Square – Parameterization: time – Preprocessed: 1

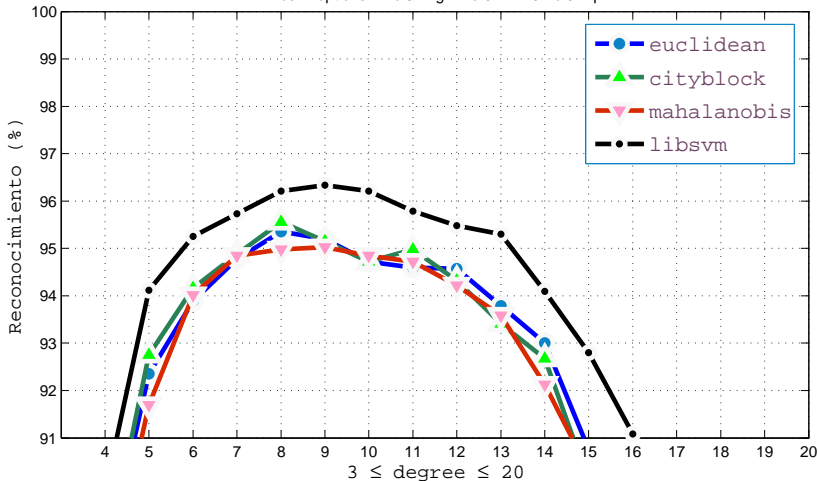
libsvm options = -c 2 -g 2 -e 0.1 -h 0 -b 0 -q



Resultados

Base de datos de letras (3600 muestras)

Polynomails: Legendre-Sobolev - Method: Least Square - Parameterization: time - Preprocessed: 1
libsvm options = -c 32 -g 2 -e 0.1 -h 0 -b 0 -q



Posibles Aplicaciones

$$\int \frac{(3x^2 + 2) \sin(x^3 + 2x - 1)}{\cos(x^3 + 2x - 1)} dx$$

Posibles Aplicaciones

$$\int \frac{(3x^2 + 2) \sin(x^3 + 2x - 1)}{\cos(x^3 + 2x - 1)} dx$$



fontanoviz


DIEGO
(10)



Posibles Aplicaciones

$$\int \frac{(3x^2 + 2) \sin(x^3 + 2x - 1)}{\cos(x^3 + 2x - 1)} dx$$

fontanovoz  DIEGO
(10) 



Conclusiones

- Implementación **eficiente** del cálculo de los momentos (orden constante).
- Representación con polinomios ortogonales caracteriza muy bien a los trazos, permitiendo alcanzar una alta precisión en el reconocimiento. Resultados sobresalientes con **Legendre-Solobev**.
- Se ha mostrado que **SVM** alcanza los mejores resultados de clasificación.
- Se han utilizado métodos modernos para la representación de trazos, diferenciándose de los métodos tradicionales en los cuales éstos son tratados como secuencias de puntos.
- Código disponible:
<https://github.com/pablospe/legendre>

Conclusiones

- Implementación **eficiente** del cálculo de los momentos (orden constante).
- Representación con polinomios ortogonales caracteriza muy bien a los trazos, permitiendo alcanzar una alta precisión en el reconocimiento. Resultados sobresalientes con **Legendre-Solobev**.
- Se ha mostrado que **SVM** alcanza los mejores resultados de clasificación.
- Se han utilizado métodos modernos para la representación de trazos, diferenciándose de los métodos tradicionales en los cuales éstos son tratados como secuencias de puntos.
- Código disponible:
<https://github.com/pablospe/legendre>

Conclusiones

- Implementación **eficiente** del cálculo de los momentos (orden constante).
- Representación con polinomios ortogonales caracteriza muy bien a los trazos, permitiendo alcanzar una alta precisión en el reconocimiento. Resultados sobresalientes con **Legendre-Solobev**.
- Se ha mostrado que **SVM** alcanza los mejores resultados de clasificación.
- Se han utilizado métodos modernos para la representación de trazos, diferenciándose de los métodos tradicionales en los cuales éstos son tratados como secuencias de puntos.
- Código disponible:
<https://github.com/pablospe/legendre>

Conclusiones

- Implementación **eficiente** del cálculo de los momentos (orden constante).
- Representación con polinomios ortogonales caracteriza muy bien a los trazos, permitiendo alcanzar una alta precisión en el reconocimiento. Resultados sobresalientes con **Legendre-Solobev**.
- Se ha mostrado que **SVM** alcanza los mejores resultados de clasificación.
- Se han utilizado métodos modernos para la representación de trazos, diferenciándose de los métodos tradicionales en los cuales éstos son tratados como secuencias de puntos.

- Código disponible:

<https://github.com/pablospe/legendre>

Conclusiones

- Implementación **eficiente** del cálculo de los momentos (orden constante).
- Representación con polinomios ortogonales caracteriza muy bien a los trazos, permitiendo alcanzar una alta precisión en el reconocimiento. Resultados sobresalientes con **Legendre-Solobev**.
- Se ha mostrado que **SVM** alcanza los mejores resultados de clasificación.
- Se han utilizado métodos modernos para la representación de trazos, diferenciándose de los métodos tradicionales en los cuales éstos son tratados como secuencias de puntos.
- Código disponible:
<https://github.com/pablospe/legendre>

¿Preguntas?

¡Gracias!

¡Game Over! :-)

Reconocimiento de Escritura Manuscrita (Online Handwriting Recognition)

PABLO SPECIALE

Tesina de Grado
(Septiembre 2011)



*Lic. en Cs. de la Computación
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario*

Director: Dr. Juan Carlos Gomez³

Co-director: Dr. Pablo Granitto⁴

³ *Procesamiento de Señales Multimedia, CIFASIS*

⁴ *Aprendizaje Automatizado y Aplicaciones, CIFASIS*