Intro a Aprendizaje Automático

Jorge Sánchez

jsanchez@famaf.unc.edu.ar

30-NOV-2013

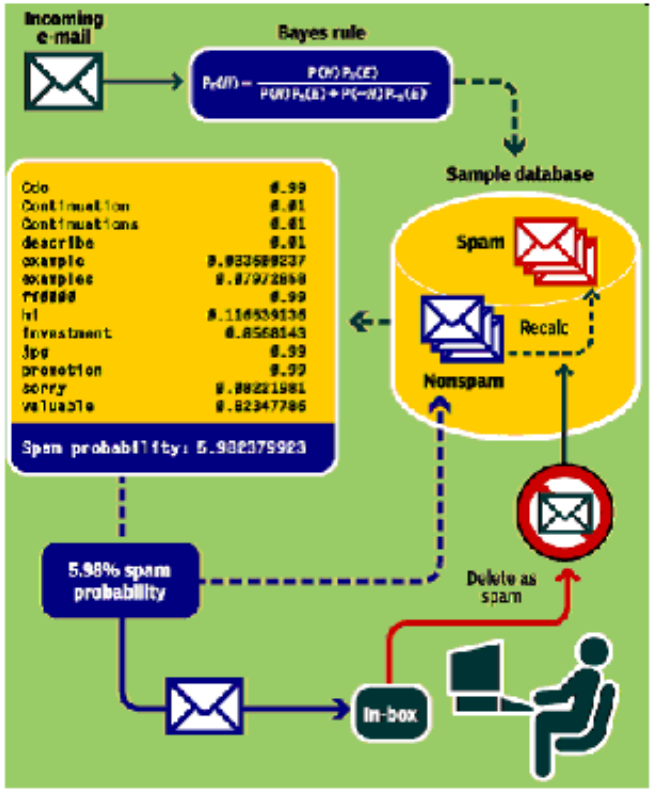
slides: Erik Sudderth, Christopher Bishop, Andrew Ziesserman

Objetivo: *dada una colección de ejemplos de muestra, poder predecir "algo" acerca de ejemplos nuevos.*

- ejemplos de muestra = datos de entrenamiento
- el "algo" define el tipo de problema de aprendizaje
- poder predecir $\Rightarrow$ definir un modelo
  - aprendizaje = "estimar" el modelo a partir de los datos
- ejemplos "nuevos" = ejemplos nunca antes vistos
  - capacidad de generalizar
  - modelo $\neq$ enumeración de datos de entrenamiento

Spam Filtering

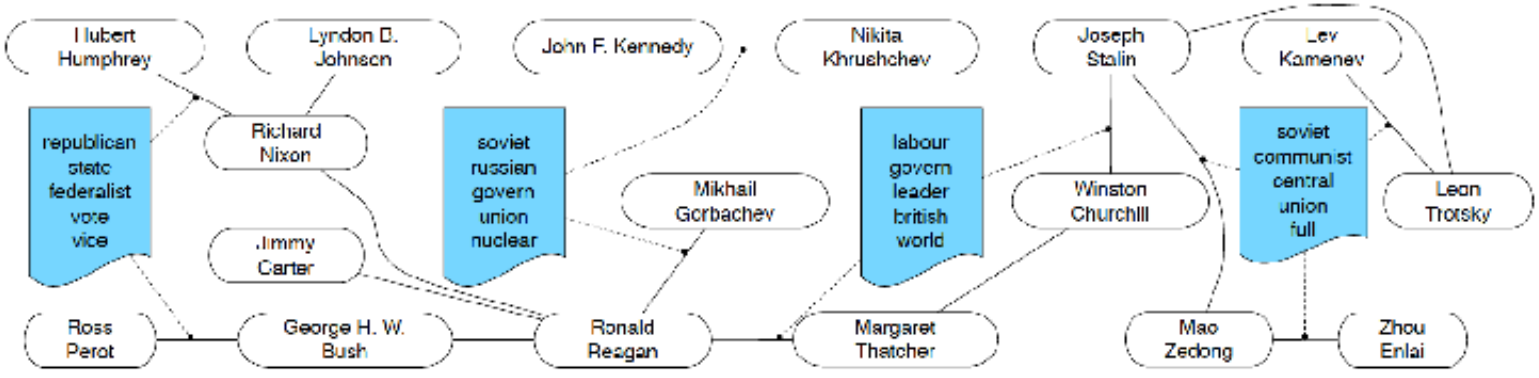
- Binary classification problem: is this e-mail spam or useful (ham)?
- Noisy training data: messages previously marked as spam
- Wrinkle: spammers evolve to counter filter innovations



Spam Filter Express
http://www.spam-filter-express.com/

Social Network Analysis

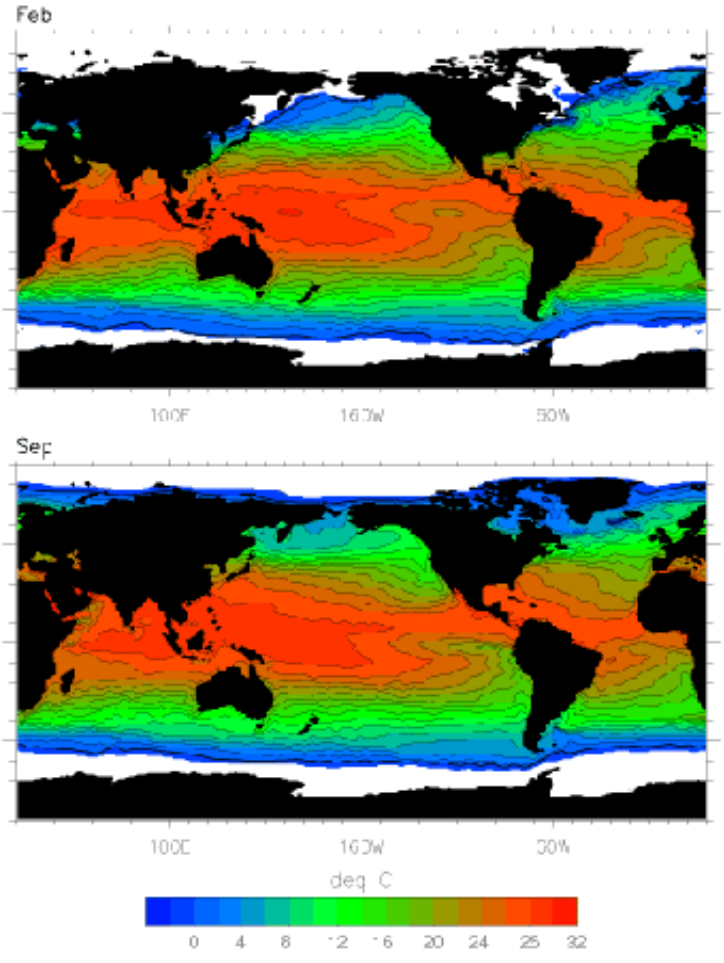
- Unsupervised discovery and visualization of relationships among people, companies, etc.
- Example: infer relationships among named entities directly from Wikipedia entries



Chang, Boyd-Graber, & Blei, KDD 2009

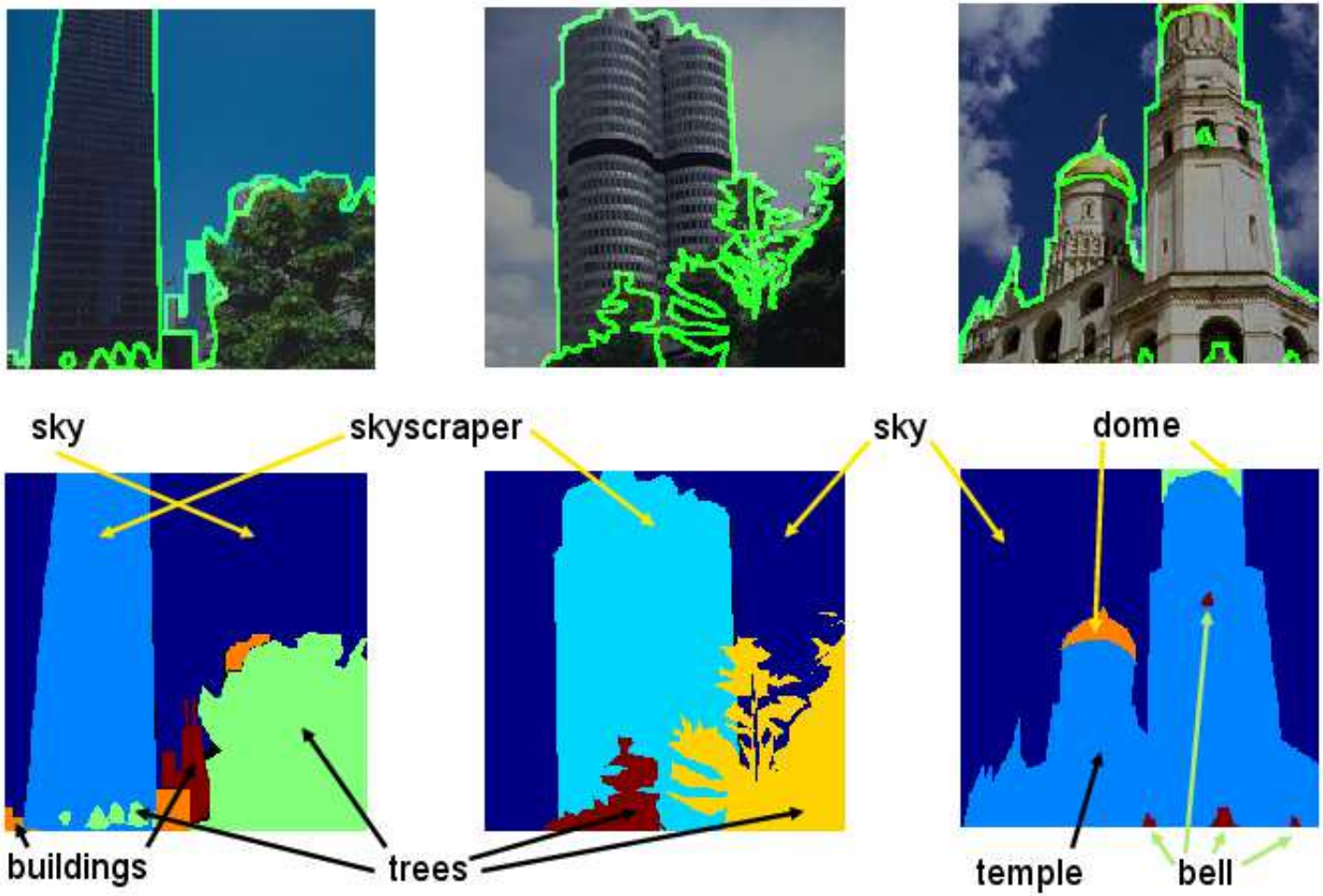
Climate Modeling

- Satellites measure sea-surface temperature at sparse locations
  - ➢ Partial coverage of ocean surface
  - ➢ Sometimes obscured by clouds, weather
- Would like to infer a dense temperature field, and track its evolution

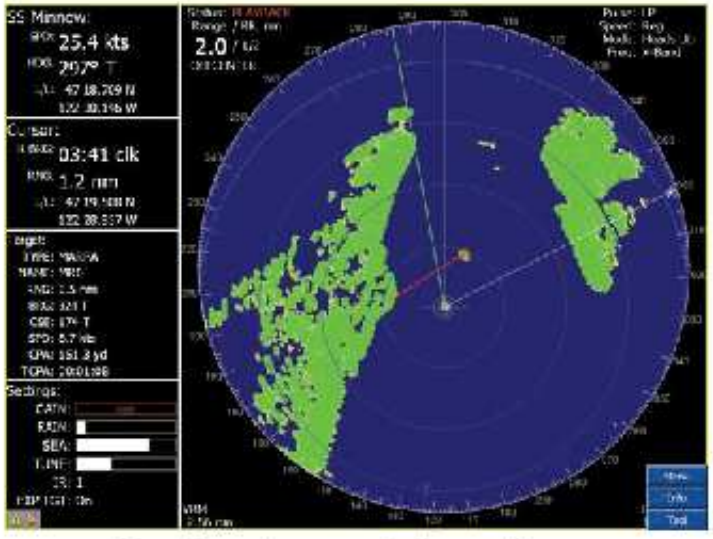


NASA Seasonal to Interannual Prediction Project
http://ct.gsfc.nasa.gov/annual.reports/ess98/nsipp.html

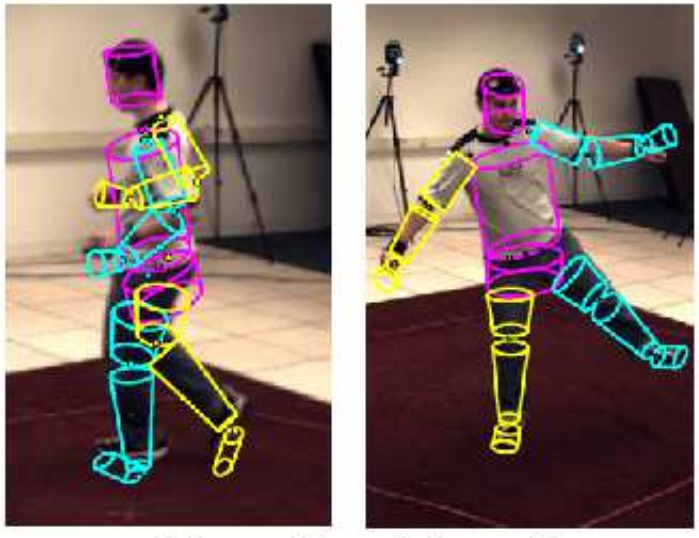
Visual Object Recognition



Target Tracking



Radar-based tracking of multiple targets



Visual tracking of articulated objects

(L. Sigal et. al., 2009)

- Estimate motion of targets in 3D world from indirect, potentially noisy measurements

Robot Navigation: *SLAM*

Simultaneous Localization and Mapping

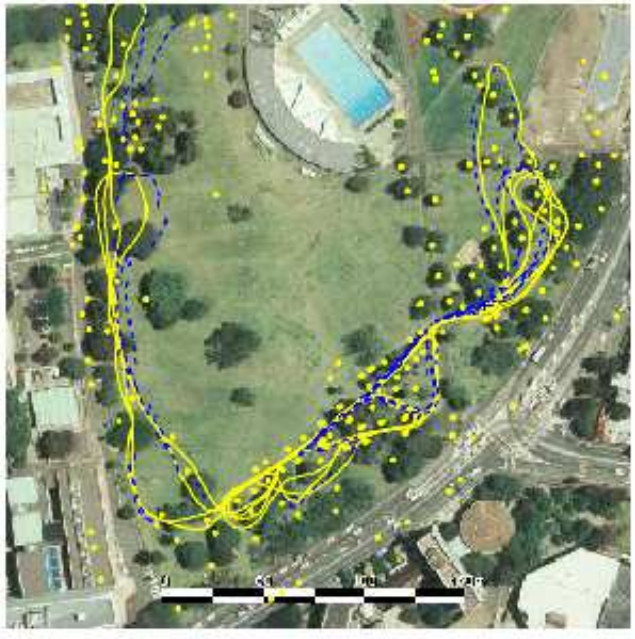
CAD Map

(S. Thrun, San Jose Tech Museum)

Estimated Map

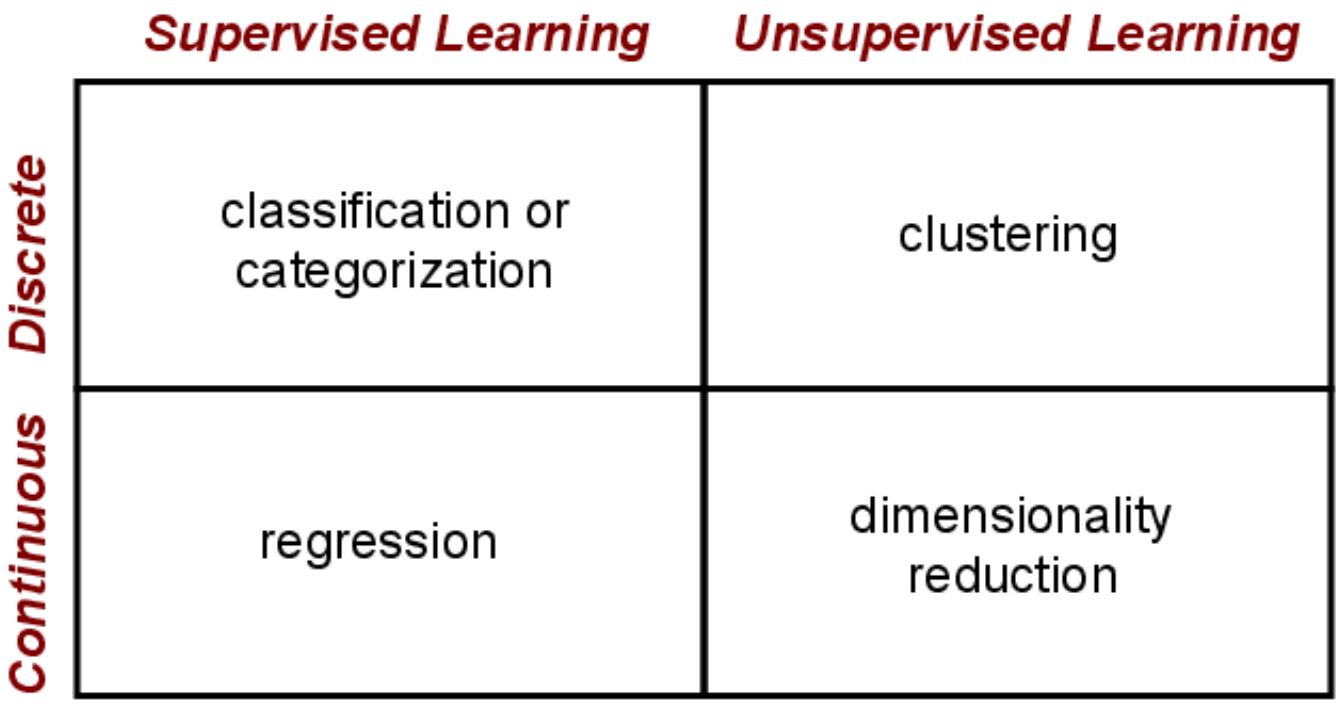
Landmark SLAM

(E. Nebot, Victoria Park)



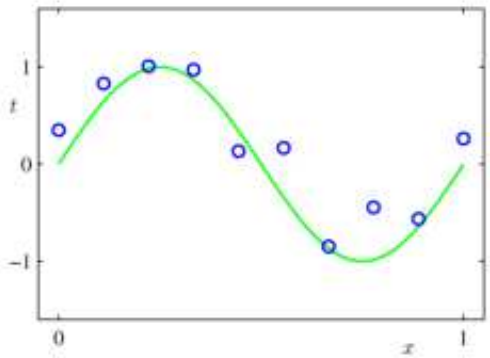
- As robot moves, estimate its pose & world geometry

Machine Learning Problems



Linear Basis Function Models

- Model functions as a linear combination of fixed, typically non-linear basis functions

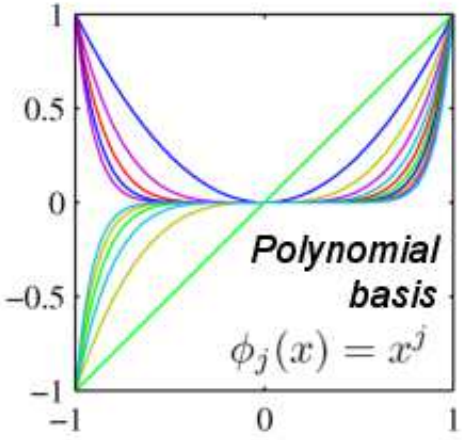


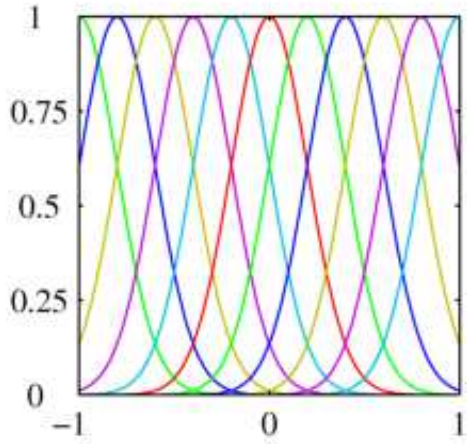
- Observed input-output pairs:

$$(x_n, t_n), \quad n = 1, \ldots, N$$
$$t_n = y_{\text{true}}(x_n) + \text{noise}$$

- Prediction function:

$$t_n \approx y(x_n, w) = \sum_{j=1}^{M} w_j \phi_j(x_n)$$

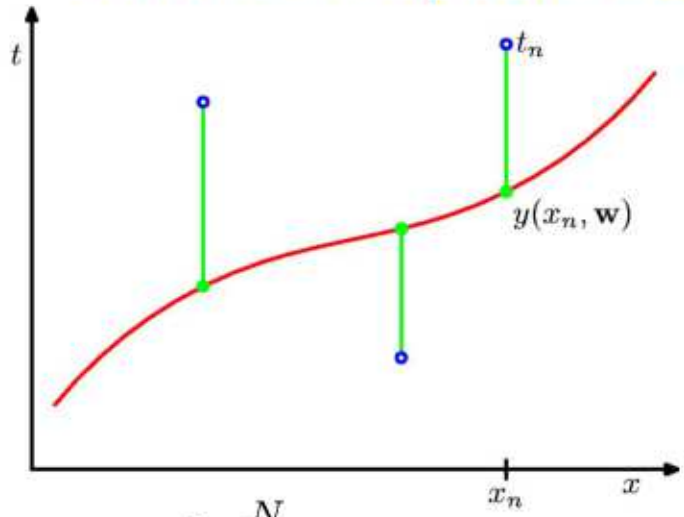




Radial basis functions are non-zero only in a small region of the input space

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

Sum-of-Squares Error Function



$N \longrightarrow$ *number of examples*

$M \longrightarrow$ *number of features*

$t_n \longrightarrow$ *output or response*

$x_n \longrightarrow$ *input or covariates*

$$y(x_n, w) = \phi(x_n)^T w$$

$$\phi(x_n) \in \mathbb{R}^M \qquad w \in \mathbb{R}^M$$

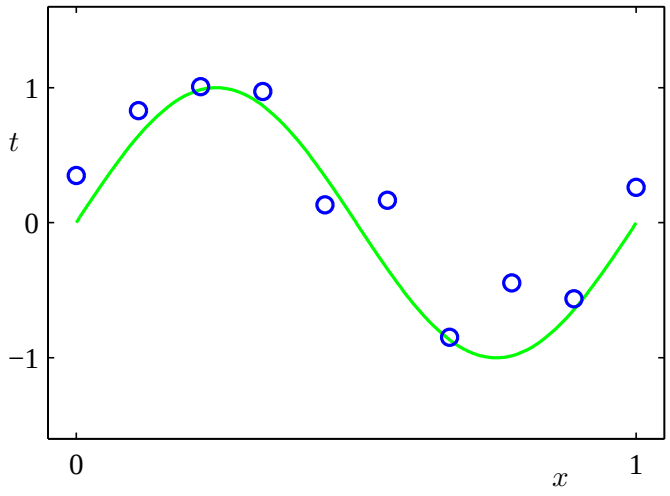
$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - \phi(x_n)^T w)^2 = \frac{1}{2}||t - \Phi w||^2$$

$$t = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix} \in \mathbb{R}^N \qquad \Phi = \begin{bmatrix} \phi(x_1)^T \\ \phi(x_2)^T \\ \vdots \\ \phi(x_N)^T \end{bmatrix} \in \mathbb{R}^{N \times M}$$

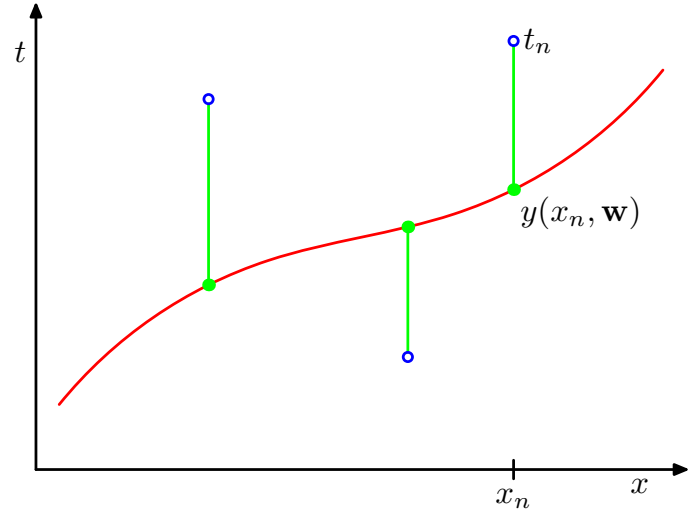
$$\nabla E(\mathbf{w}) = 0 \quad \longrightarrow \quad \mathbf{w}^* = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{t}$$

Aprendizaje supervisado: regresión

Ejemplo: ajuste de curva polinómica



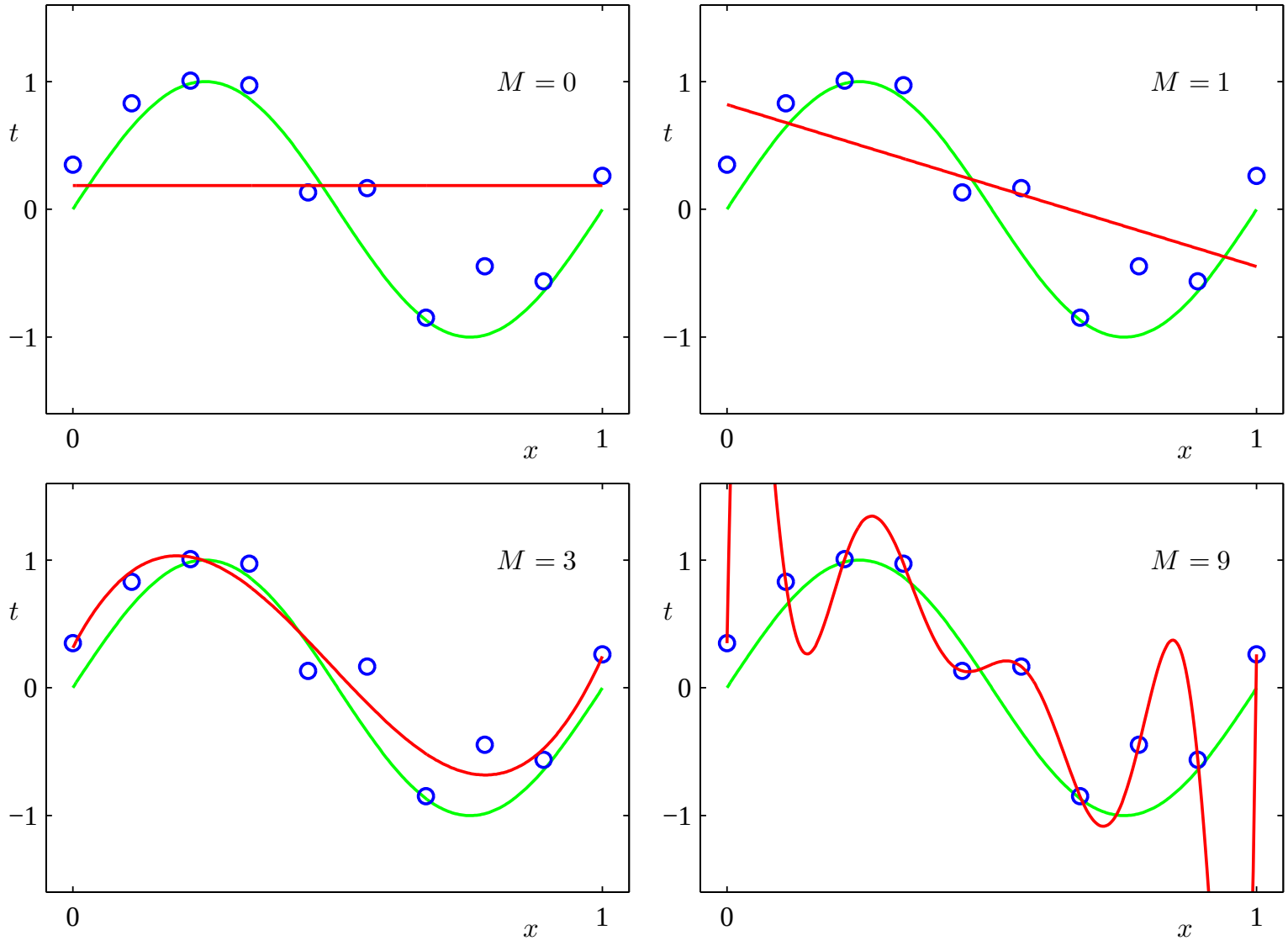
$$y(x, \mathbf{w}) = w_0 + w_1 x + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$



$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}$$

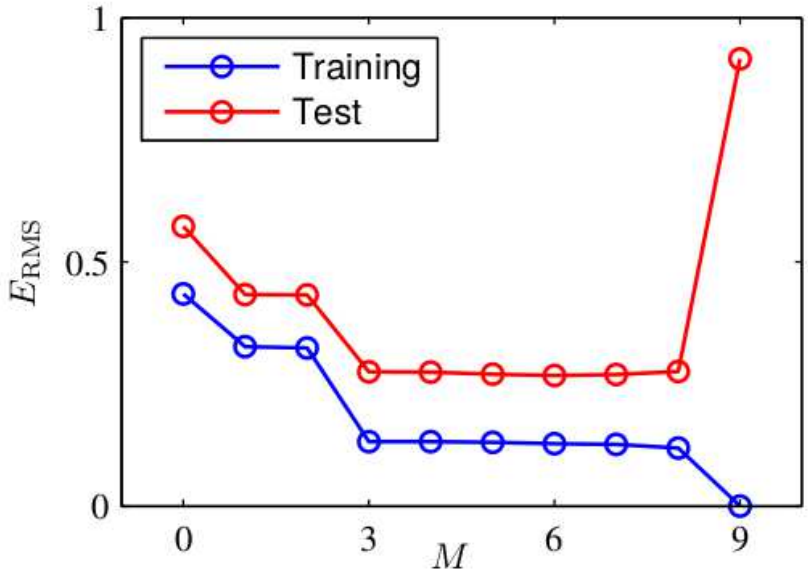
Aprendizaje supervisado: regresión

Ejemplo: ajuste de curva polinómica



Aprendizaje supervisado: regresión

El problema de "overfitting"



$$E_{RMS} = \sqrt{\frac{2E(\mathbf{w})}{N}}$$

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

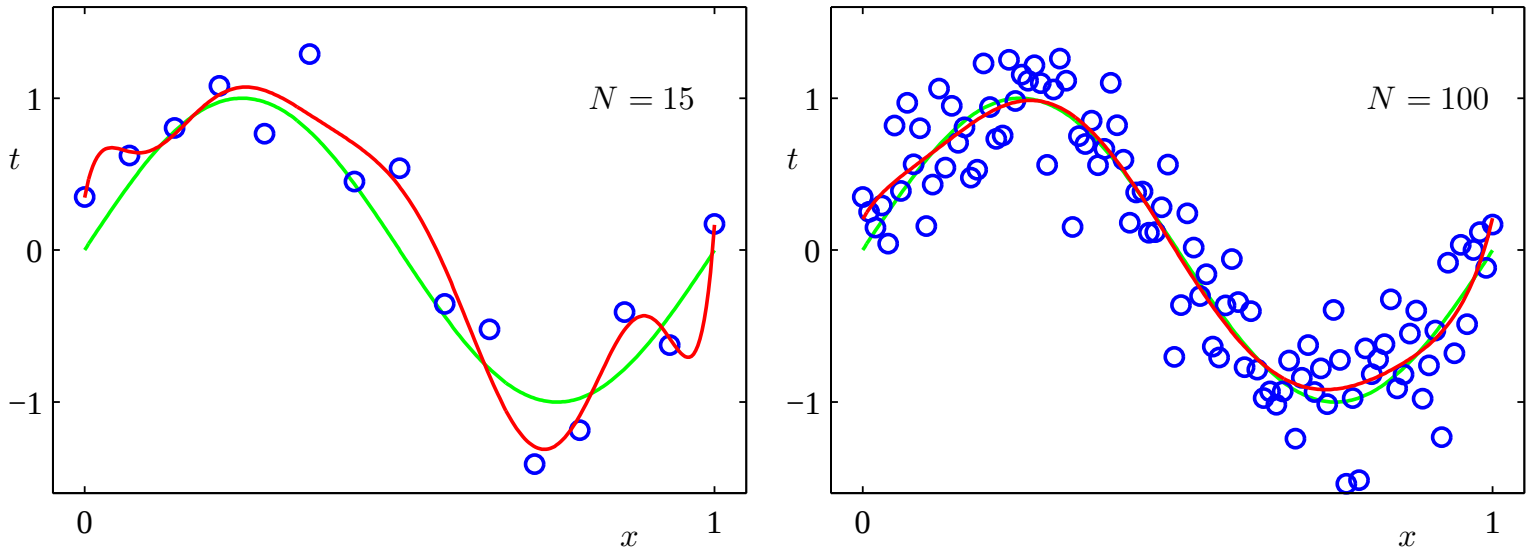
Formas de lidiar con el sobreentrenamiento:

- aumentar el número de muestras de entrenamiento (resp. M)
- **regularización**: penalizar la magnitud de los coeficientes, p.ej.:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\} + \lambda\|\mathbf{w}\|^2$$

Aprendizaje supervisado: regresión
El problema de "overfitting"

Aumentando el número de muestras:



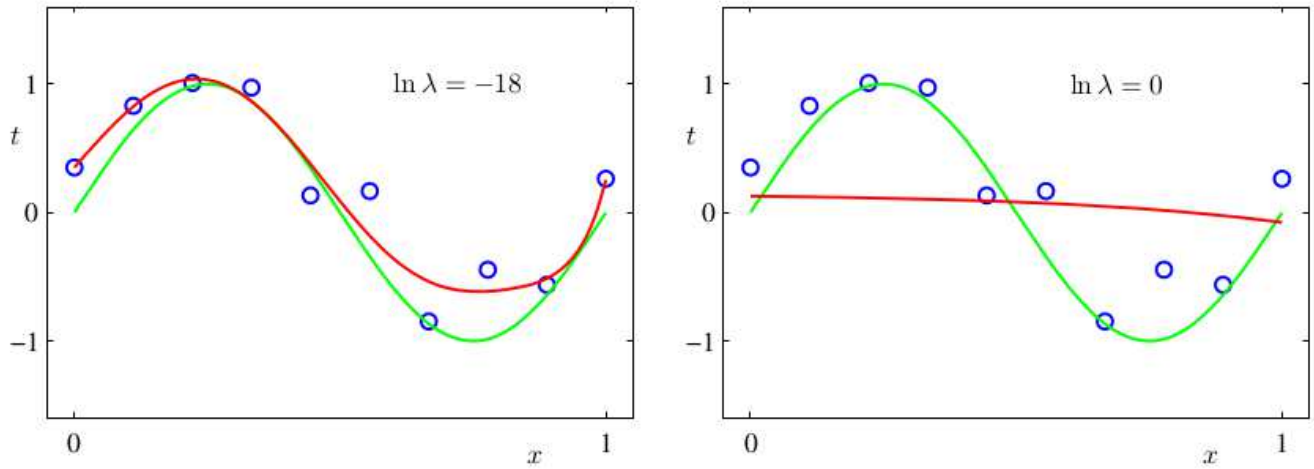
(+) permite que los datos "hablen por si mismos"

(-) no siempre es fácil conseguir más muestras

Aprendizaje supervisado: regresión

El problema de "overfitting"

Usando regularización L_2



	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

(+) el nivel de regularización se puede controlar mediante λ

(-) hay que ajustar λ (*cross-validation*)

Aprendizaje no supervisado: clustering

Clustering: identificar grupos (*clusters*) en un espacio multidimensional = particionar el conjunto de datos de entrenamiento

- es la forma más usual de aprendizaje no supervisado
- necesito definir una noción de similitud / distancia entre muestras
  - alto grado de similitud entre elementos de un mismo grupo
  - bajo grado de similitud entre elementos de grupos distintos
- la definición de "similar" es subjetiva
- debo lidiar con la maldición de la dimensionalidad (*curse of dimensionality*)

Algoritmo k-means:

- datos en $\mathbb{R}^D$ + distancia Euclídea
- se define k *a priori*
- ejemplo de cuantificación vectorial

K-Means Objective: Compression

- Observed feature vectors: $x_i \in \mathbb{R}^d, \quad i = 1, 2, \ldots, N$
- Hidden cluster labels: $z_i \in \{1, 2, \ldots, K\}, \quad i = 1, 2, \ldots, N$
- Hidden cluster centers: $\mu_k \in \mathbb{R}^d, \quad k = 1, 2, \ldots, K$

$$J(z, \mu \mid x, K) = \sum_{k=1}^{K} \sum_{i|z_i=k} ||x_i - \mu_k||^2 = \sum_{i=1}^{N} ||x_i - \mu_{z_i}||^2$$

$$J(z, \mu \mid x, K) = \sum_{k=1}^{K} \sum_{i=1}^{N} r_{ik} ||x_i - \mu_k||^2 \qquad r_{ik} = \mathbb{I}(z_i = k)$$

K-Means alternates between

$$z^{(t)} = \arg\min_{z} J(z, \mu^{(t-1)} \mid x, K)$$

$$\mu^{(t)} = \arg\min_{\mu} J(z^{(t)}, \mu \mid x, K)$$

K-Means Update Equations

- Observed feature vectors: $x_i \in \mathbb{R}^d, \quad i = 1, 2, \ldots, N$
- Hidden cluster labels: $z_i \in \{1, 2, \ldots, K\}, \quad i = 1, 2, \ldots, N$
- Hidden cluster centers: $\mu_k \in \mathbb{R}^d, \quad k = 1, 2, \ldots, K$

$$J(z, \mu \mid x, K) = \sum_{k=1}^{K} \sum_{i=1}^{N} r_{ik} ||x_i - \mu_k||^2 \qquad r_{ik} = \mathbb{I}(z_i = k)$$

$$z^{(t)} = \arg\min_{z} J(z, \mu^{(t-1)} \mid x, K)$$

$$z_i^{(t)} = \arg\min_{k} ||x_i - \mu_k^{(t-1)}||^2$$

Assign to closest cluster center

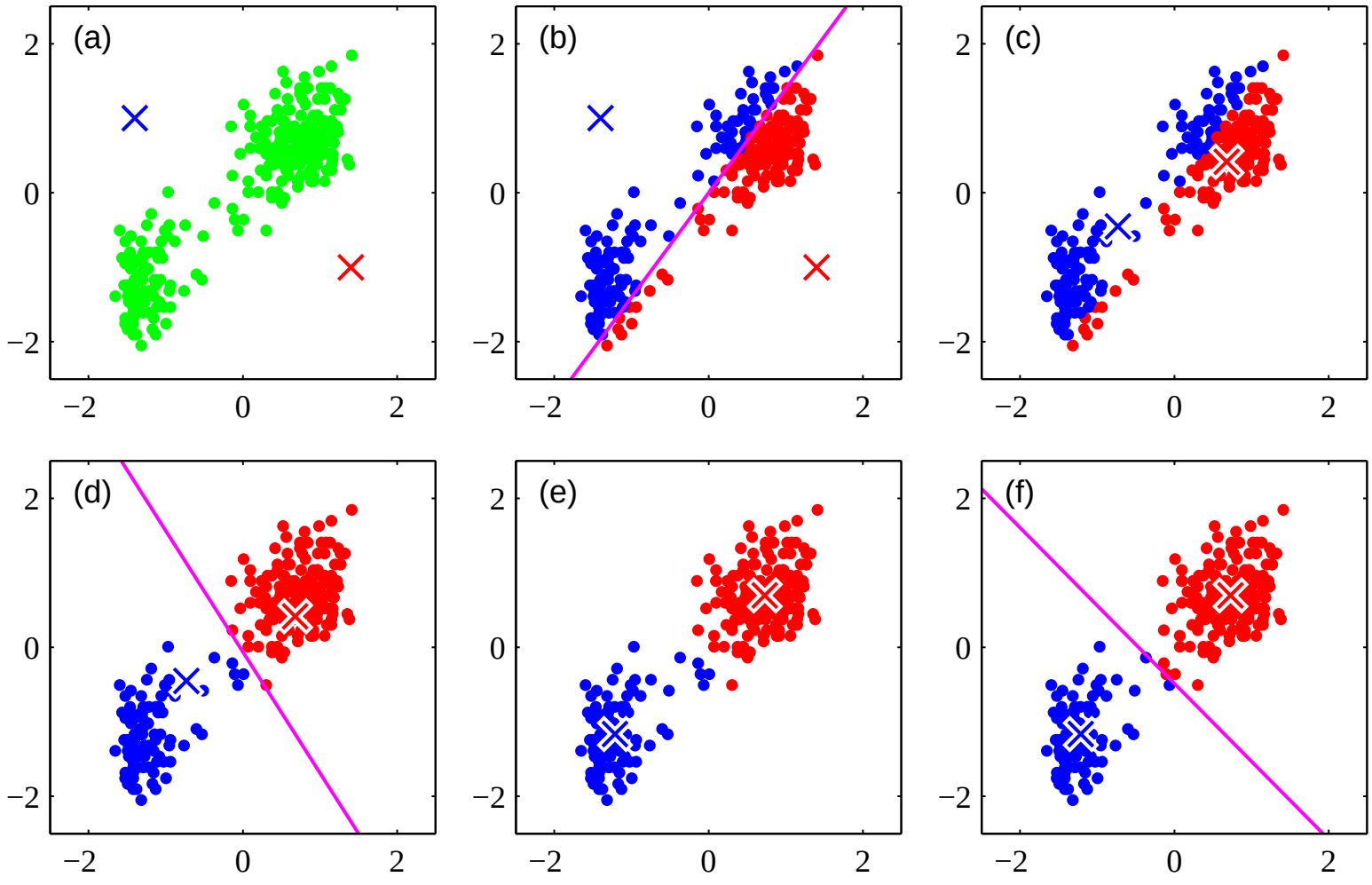
$$\mu^{(t)} = \arg\min_{\mu} J(z^{(t)}, \mu \mid x, K)$$

Mean of assigned data

$$\mu_k^{(t)} = \frac{1}{N_k^{(t)}} \sum_{i=1}^{N} r_{ik} x_i \qquad N_k^{(t)} = \sum_{i=1}^{N} r_{ik}$$

Aprendizaje no supervisado: clustering

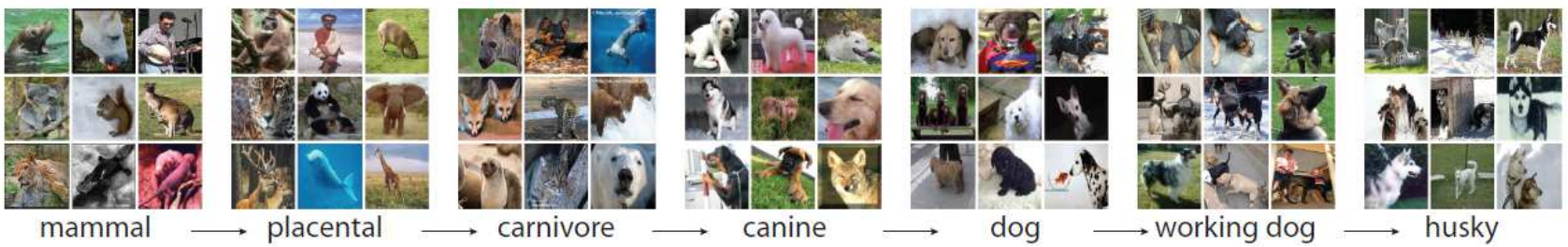
Algoritmo k-means



Aprendizaje supervisado: clasificación

Clasificación: asignar muestras a una o varias categorías/clases de un conjunto predefinido

- muestras de entrenamiento = pares etiquetados: $\{(x_n, y_n)\}_{n=1}^{N}$
  - $x \in \mathcal{X}$, $y \in \{\mathcal{C}_k\}_{k=1}^{K}$
  - clasificación binaria: $y \in \{0, 1\}$, $y \in \{-1, 1\}$
- la definición de las clases de interés es subjetiva
  - el grado de variabilidad intra-clase / entre-clases define en gran medida la dificultad del problema
    - polisemia, ambigüedades, grado de abstracción, etc.
  - involucra conceptos del lenguaje natural
  - lidiar con el "semantic gap": $x \longleftrightarrow \{\mathcal{C}_k\}_{k=1}^{K}$



Aprendizaje supervisado: clasificación

- *Clasificadores probabilísticos*: regla de Bayes

$$p(\mathcal{C}_k|x) = \frac{p(x|\mathcal{C}_k)p(\mathcal{C}_k)}{p(x)} \propto \underbrace{p(\mathcal{C}_k)}_{\text{prior}} \underbrace{p(x|\mathcal{C}_k)}_{\text{likelihood}}$$

$$\hat{k} = \arg\max_k p(\mathcal{C}_k|x)$$

  - **Modelos generativos**: $p(x, \mathcal{C}_k) = p(x|\mathcal{C}_k)p(\mathcal{C}_k)$
  - **Modelos discriminativos**: $p(\mathcal{C}_k|x)$
    - Naïve Bayes ($x \in \mathcal{X}^D$): $p(x|\mathcal{C}_k) = \prod_{i=1}^{D} p(x_i|\mathcal{C}_k)$

- *Función discriminante*: $h : \mathcal{X} \to \{\mathcal{C}_k\}_{k=1}^{K}$
  - En general: $h(x) = g(f(x))$
    - $f : \mathcal{X} \to \mathbb{R}$: función de activación ("clasificador")
    - $g : \mathbb{R} \to \{\mathcal{C}_k\}_{k=1}^{K}$: función de decisión

- Se asume que hay N muestras etiquetadas de entrenamiento: $\{(x_n, y_n)\}_{n=1}^{1}$
- Existe una medida de distancia entre muestras: $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$

$$d(x_i, x_j) \geq 0, \qquad \text{con igualdad sii i=j}$$
$$d(x_i, x_j) = d(x_j, x_i),$$
$$d(x_i, x_k) \leq d(x_i, x_i) + d(x_j, x_k)$$

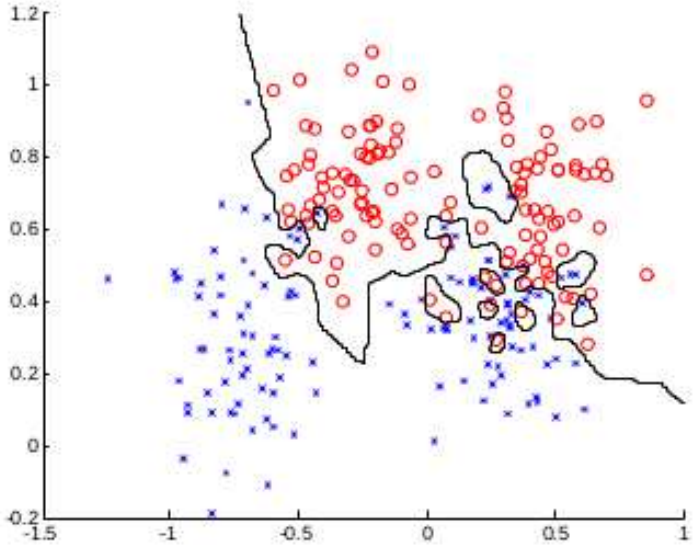
- 1-**NN**: dada una nueva muestra $z \in \mathcal{X}$, se le asigna la etiqueta:

$$\hat{y} = y_k, \qquad \text{donde:} k = \arg\min_{i=1,\ldots,N} d(x_i, z)$$

- k-**NN**: en lugar de asignar la etiqueta del ejemplo mas próximo, tomo los k más próximos y asigno la mayoritaria.
  - elección de k?

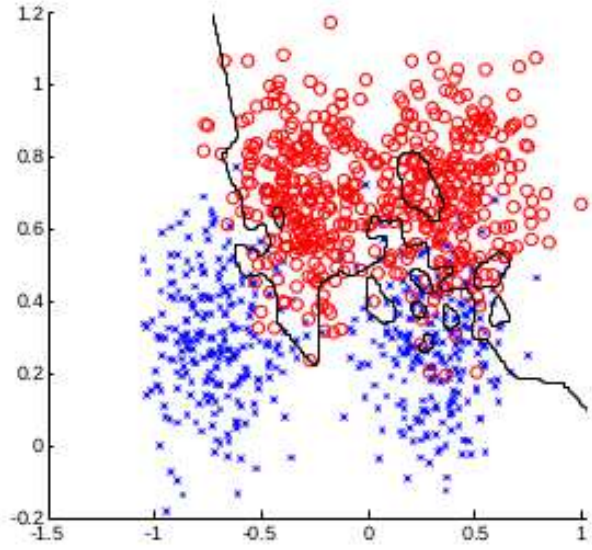
K = 1

Training data



error = 0.0

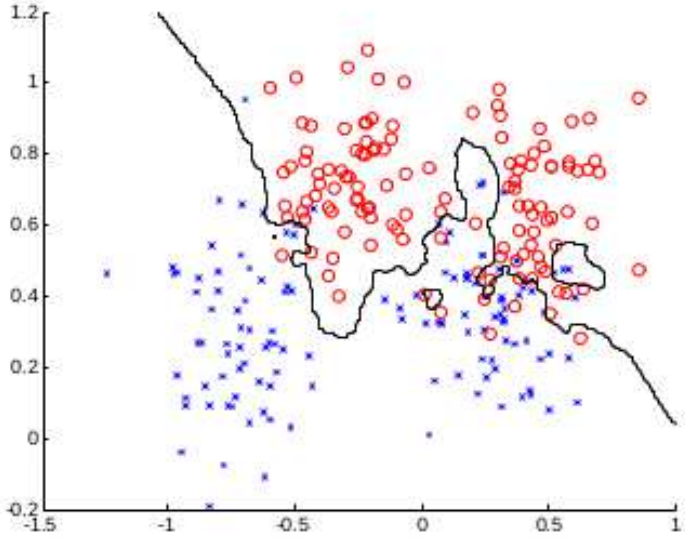
Testing data



error = 0.15

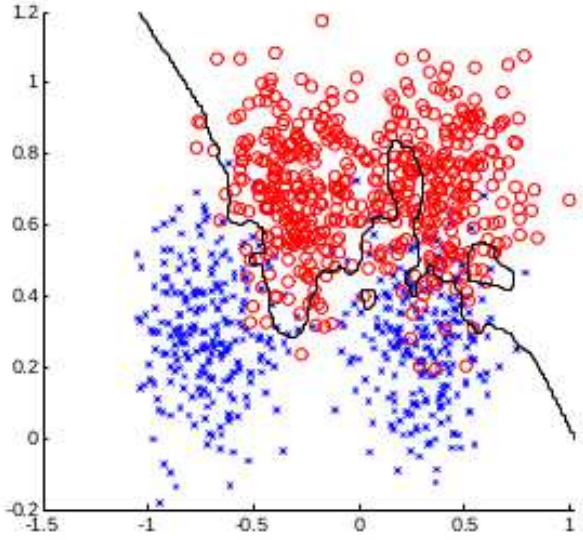
K = 3

Training data



error = 0.0760

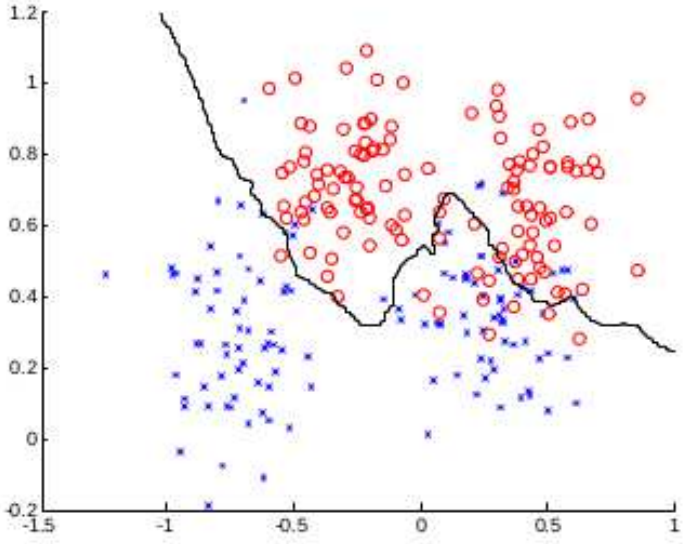
Testing data



error = 0.1340

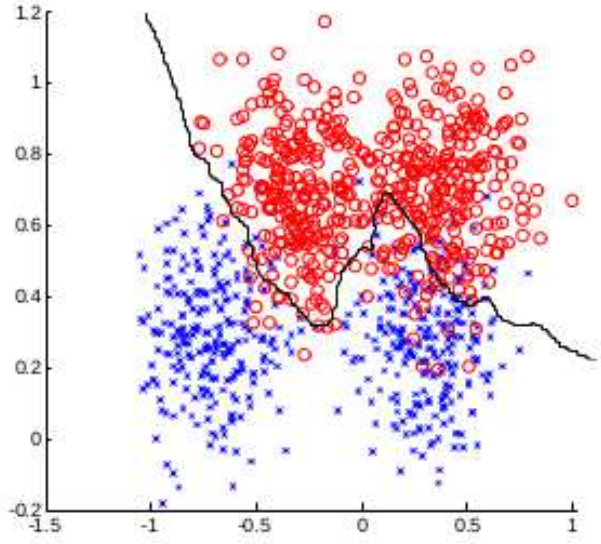
K = 21

Training data



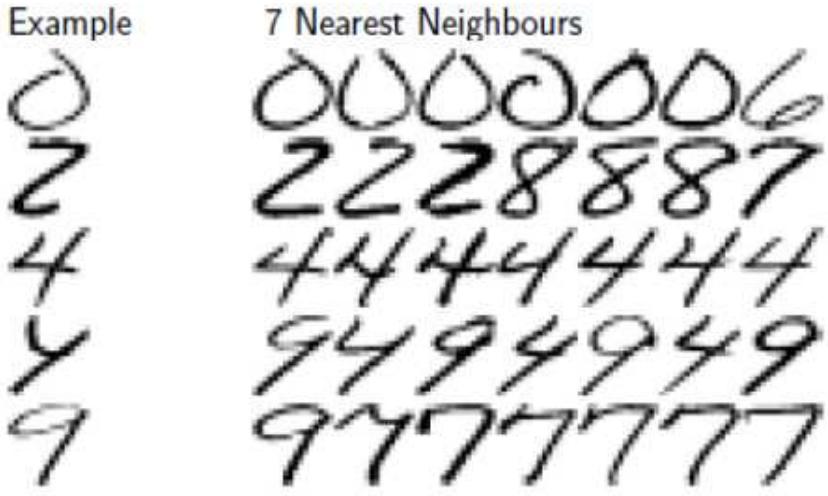
error = 0.1120

Testing data

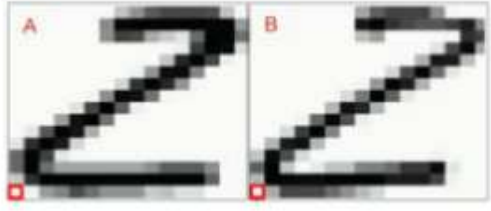


error = 0.0920

Example: hand written digit recognition



- MNIST data set
- Distance = raw pixel distance between images
- 60K training examples
- 10K testing examples
- K-NN gives 5% classification error



$$D(\mathbf{A},\mathbf{B}) = \sum_{ij} \sqrt{\left(a_{ij} - b_{ij}\right)^2}$$

Ventajas:

- simple y efectivo
- permite resolver problemas multi-clase
- superficies de decisión son no lineales
- más muestras $\rightarrow$ mejor performance
- solo un parámetro: k, a determinar por validación cruzada

Ventajas:

- necesita especificar una función de distancia
  - alternativa: *metric learning*
- modelo costoso de evaluar y de almacenar
  - estructuras de datos: KDTrees, etc.
  - esquemas búsqueda aproximada

Slides A. Zisserman: *The SVM Classifier*
`http://www.robots.ox.ac.uk/~az/lectures/ml/`

Selección de modelos:

- errores: train vs. test
  - early stopping
  - conj. validación
- cross-validation
  - *n*-fold CV
  - *leave-one-out* CV
- regularización

Métricas:

- Regresión / Clustering: distorsión
- Clasificación: precisión, matrices de confusión, curvas ROC, AUC, etc. (próximas clases)