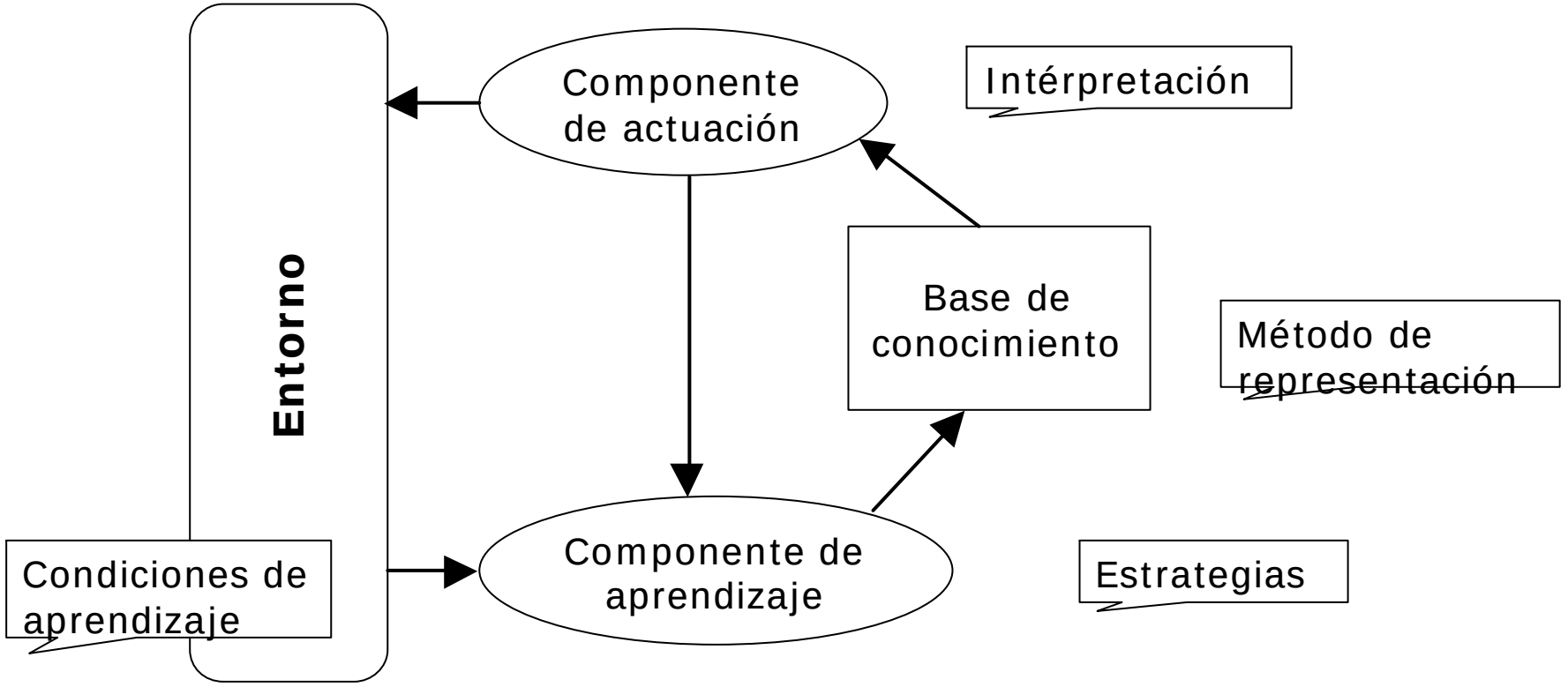
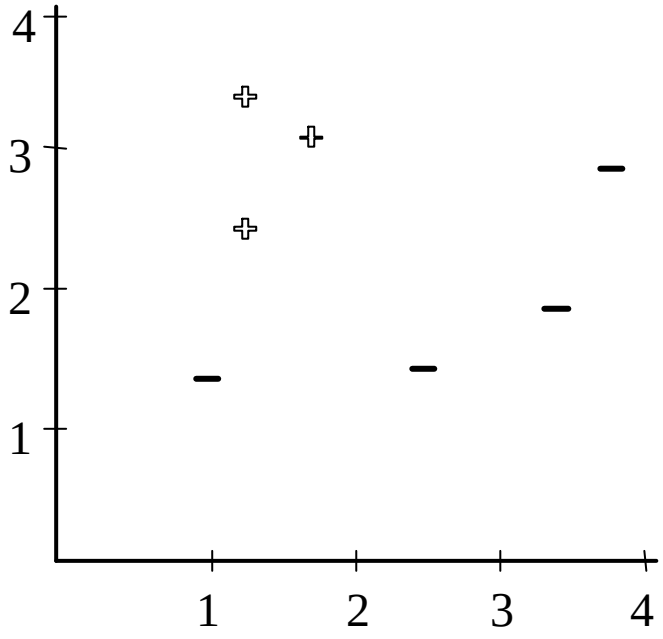
Sistemas reales en aprendizaje automático

Un marco genérico para el aprendizaje



Aprendizaje es la mejora en el **rendimiento** en algún **entorno** a través de la adquisición de **conocimiento** que resulta de la **experiencia** en ese entorno (Langley, 1996).

Nuestro ejemplo sencillo



Tenemos un conjunto de instancias definidas por los atributos X e Y.

Están etiquetadas en dos posibles clases (+,-).

El lenguaje de representación sólo nos permite establecer condiciones sobre números enteros (para simplificar).

La tarea inicial será de **clasificación** (predecir la clase de nuevas instancias).

Conjunciones lógicas

Representación: conjunciones lógicas de pares atributo/valor para atributos nominales o límites para numéricos.

Interpretación: correspondencia completa con los valores de la descripción (*full matching*).

X<2

color = oscuro

Aprendizaje de conjunciones

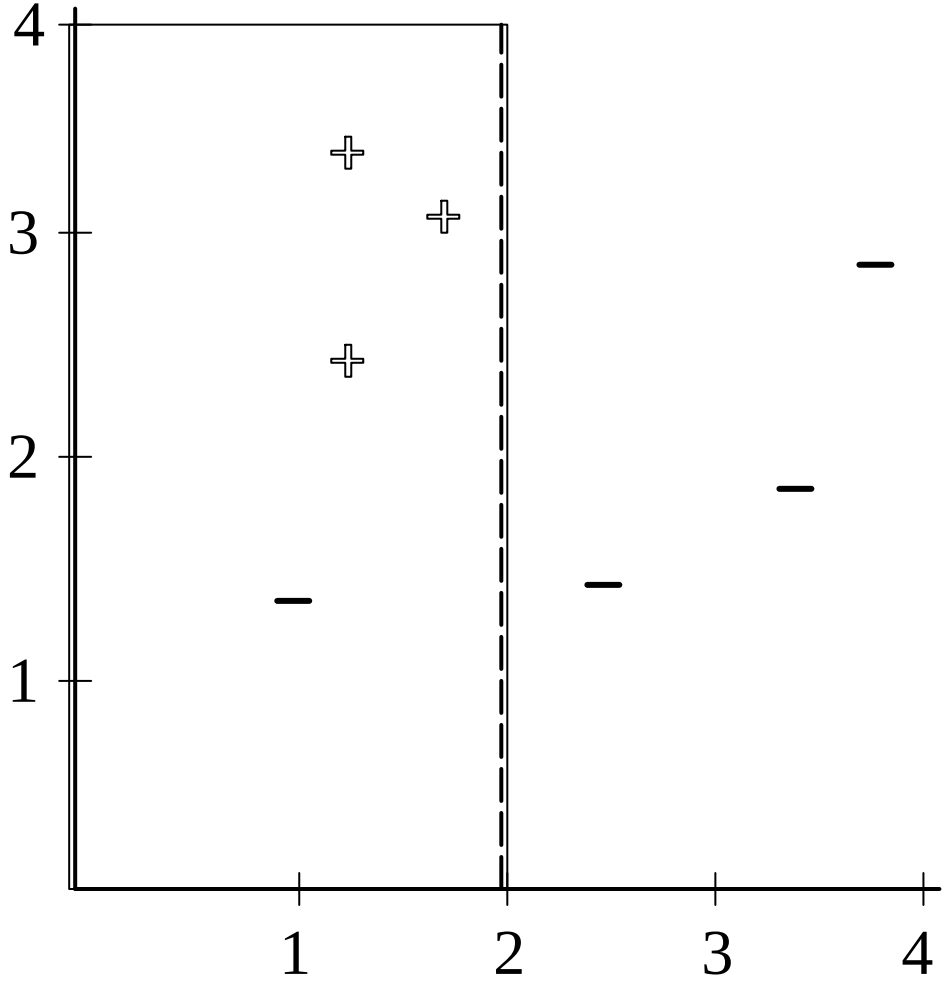
Dirección: de lo general a lo específico o viceversa. También bidireccional.

(ordenación parcial de los conceptos)

Búsqueda exhaustiva o heurística (necesita una función de evaluación de descripciones).

Estrategia batch o incremental.

Heurístico general a específico (I)

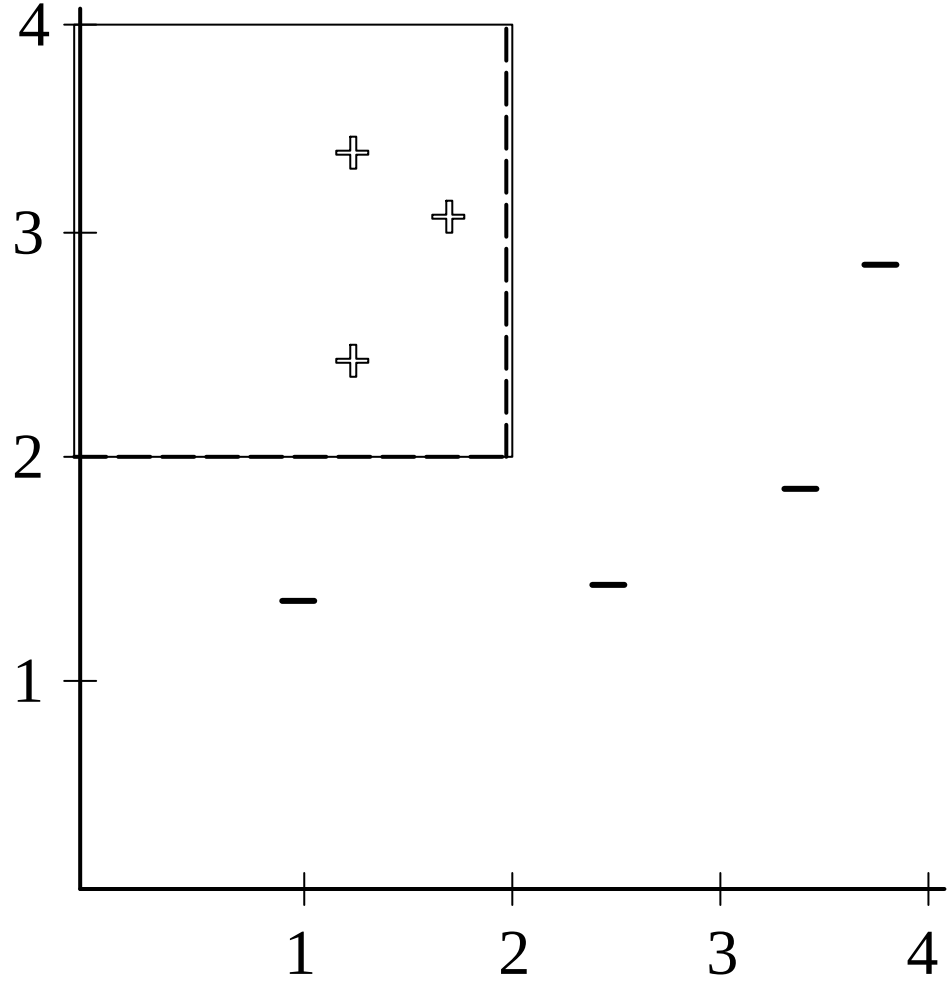


X<2

$h = 3 - 1 = 2$

Como **heurística** *h*, utilizamos la diferencia entre el número de instancias positivas y negativas que cubre la descripción.

Heurístico general a específico (II)



Hacemos la descripción más específica:

X<2 Ù Y>2

h = 3 – 0 = 3

La descripción es consistente y completa y no podemos mejorar *h*.

Búsqueda bidireccional

Método incremental, procesa cada instancia individualmente y modifica las descripciones.

Mantiene dos conjuntos de descripciones: el de las más específicas (S) y el de las más generales (G) que son consistentes con los datos (**espacio de versiones**).

La búsqueda es exhaustiva y puede resultar costosa en tiempo y espacio.

Es muy poco robusto ante el ruido.

Esta implementada en el algoritmo de **eliminación de candidatos** (Mitchell, 1982).

Validación: estimación del error

¿Cómo podemos estimar el error que cometeremos al realizar predicciones sobre nuevas instancias?

Medir el error sobre los datos observados.

Si son representativos de la población es una buena aproximación, pero no es seguro porque disponemos de datos limitados.

En la práctica proporcionan una estimación demasiado **optimista**.

Validación: estimación del error

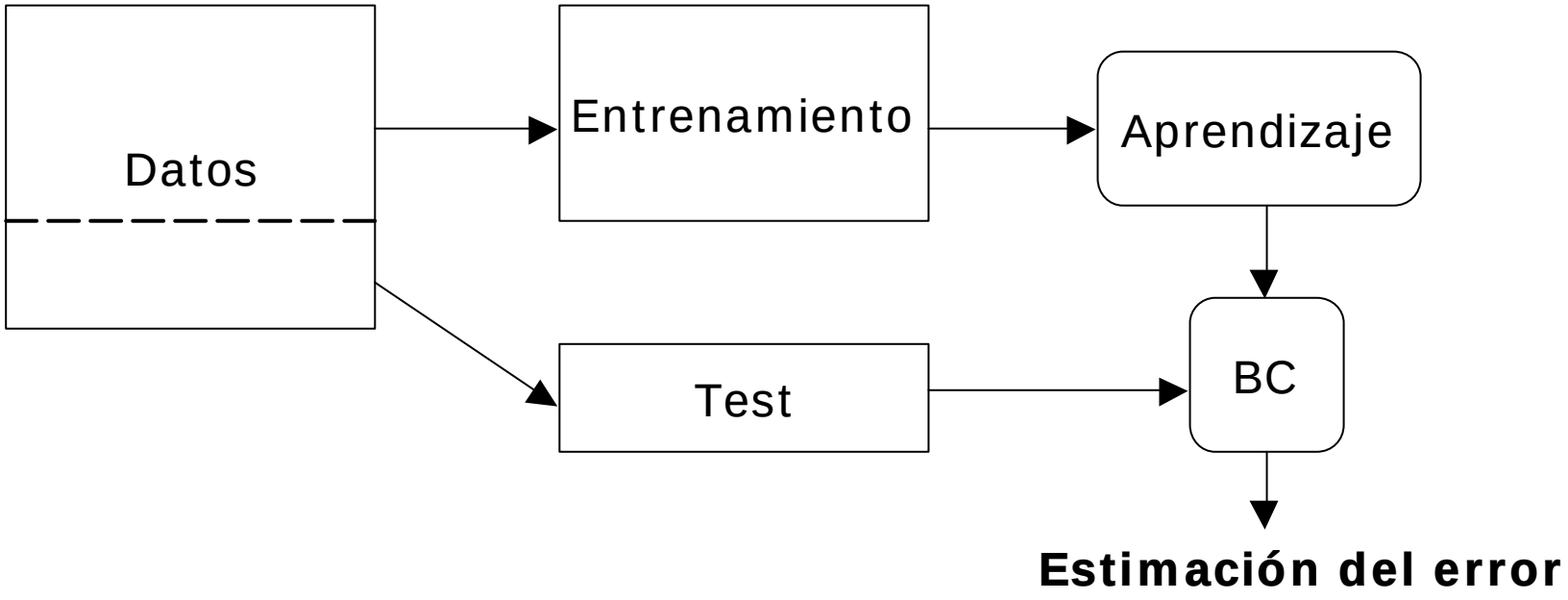
¿Por qué los problemas de examen no son exactamente los mismos de clase?

Porque el alumno podría aprendérselos casi de memoria y el objetivo es que aprenda a aplicar lo que aprende **en general**.

Ciertos alumnos saben hacer los problemas de clase pero cuando se enfrentan a un problema parecido pero nuevo no saben resolverlo.

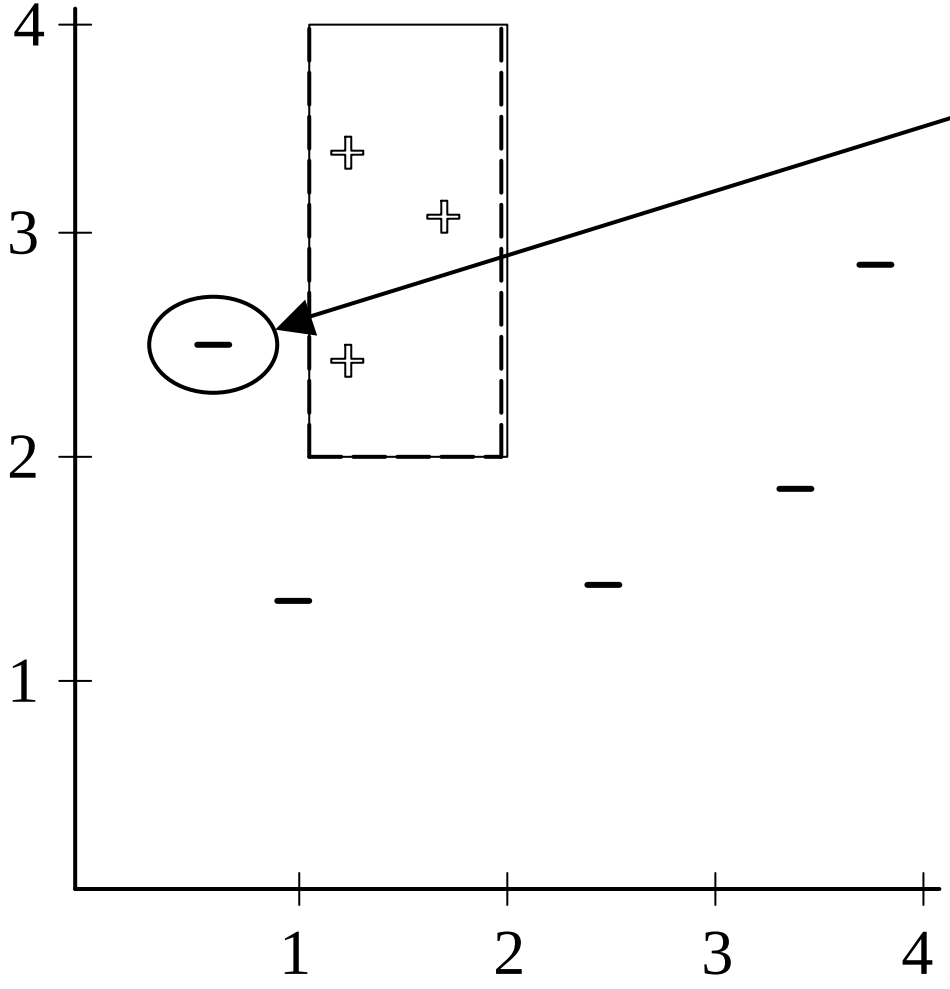
No quiero que mi sistema suspenda, así que...

Validación: estimación del error



Como el conjunto de test es **independiente** del conjunto de entrenamiento, proporciona una estimación más fiable del error cometido.

Un añadido inocente...



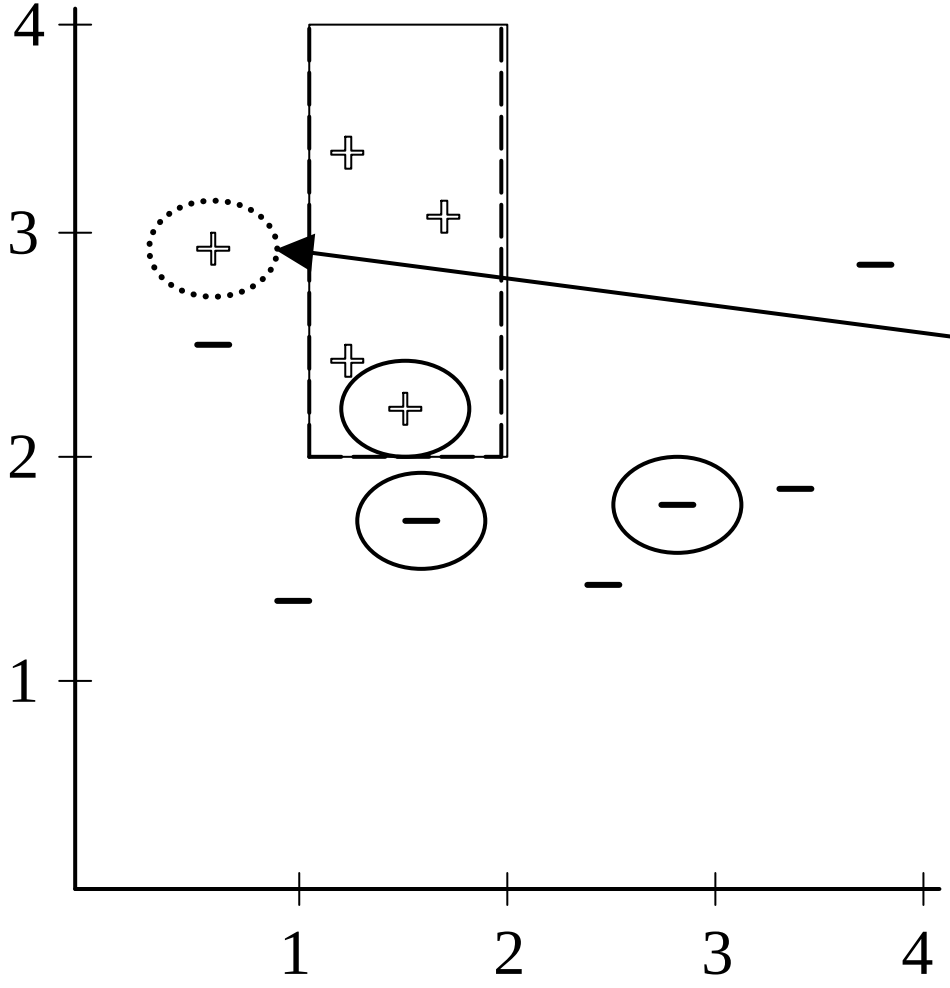
Añadimos otra instancia.

Haciendo la descripción anterior más específica

X<2 Ù Y>2 Ù X>1

el error sobre los datos observados se reduce a 0.

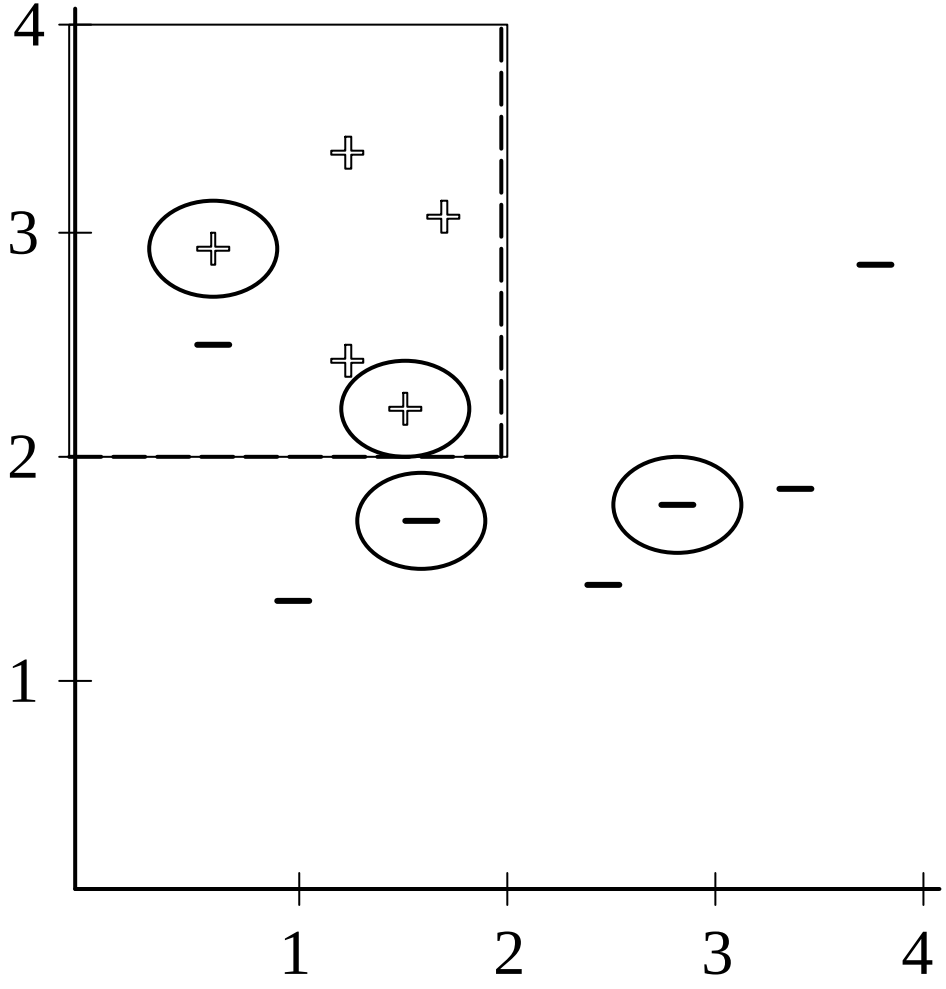
O no tan inocente...



Ahora intentamos predecir nuevos datos en un conjunto de test.

¡Ahora si hay errores!

Sobreajuste (overfitting)



Si no añadimos el último término, permitimos error en el conjunto de entrenamiento.

Pero... ¡no hay error en el conjunto de test!

La descripción anterior causaba un sobreajuste.

Poda

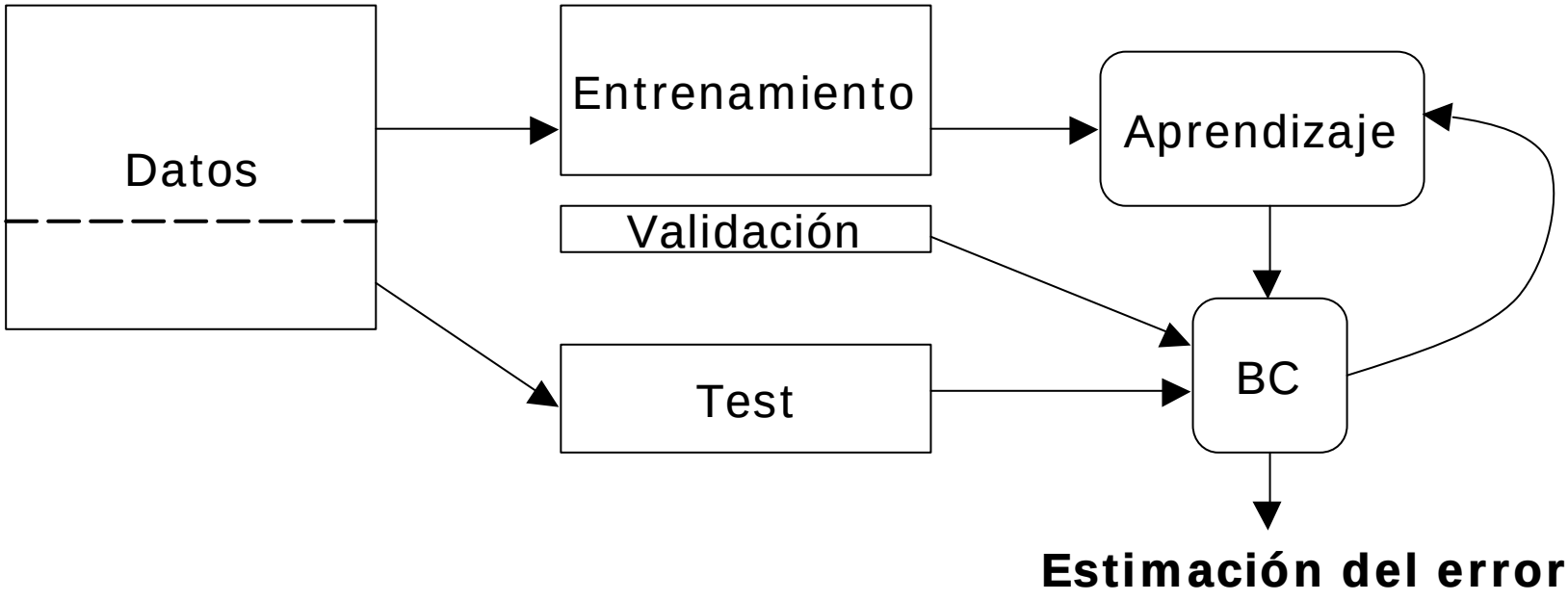
Lo que hemos hecho es “podar” la regla eliminando una condición.

Hemos permitido un error de 0.125 (1/8) en el conjunto de entrenamiento, es decir, relajado la condición de consistencia.

Tenemos un nuevo problema: ¿qué tamaño de descripción (es decir, del árbol) debemos permitir para generalizar correctamente? Es un parámetro a aprender.

(no podemos usar el conjunto de test porque estamos todavía en la parte del proceso de aprendizaje)

Validación interna



El conjunto de validación es independiente del de entrenamiento y permite una estimación del error para diversos valores de los parámetros. Me ayudará así en el proceso de poda.

Datos insuficientes

Si no tenemos suficientes datos puede ocurrir que

- no hayan datos de entrenamiento suficientes para realizar un aprendizaje adecuado.
- hayan pocos datos de test y no resulten representativos.

K-fold cross-validation

Dividimos el conjunto de entrenamiento en k subconjuntos.

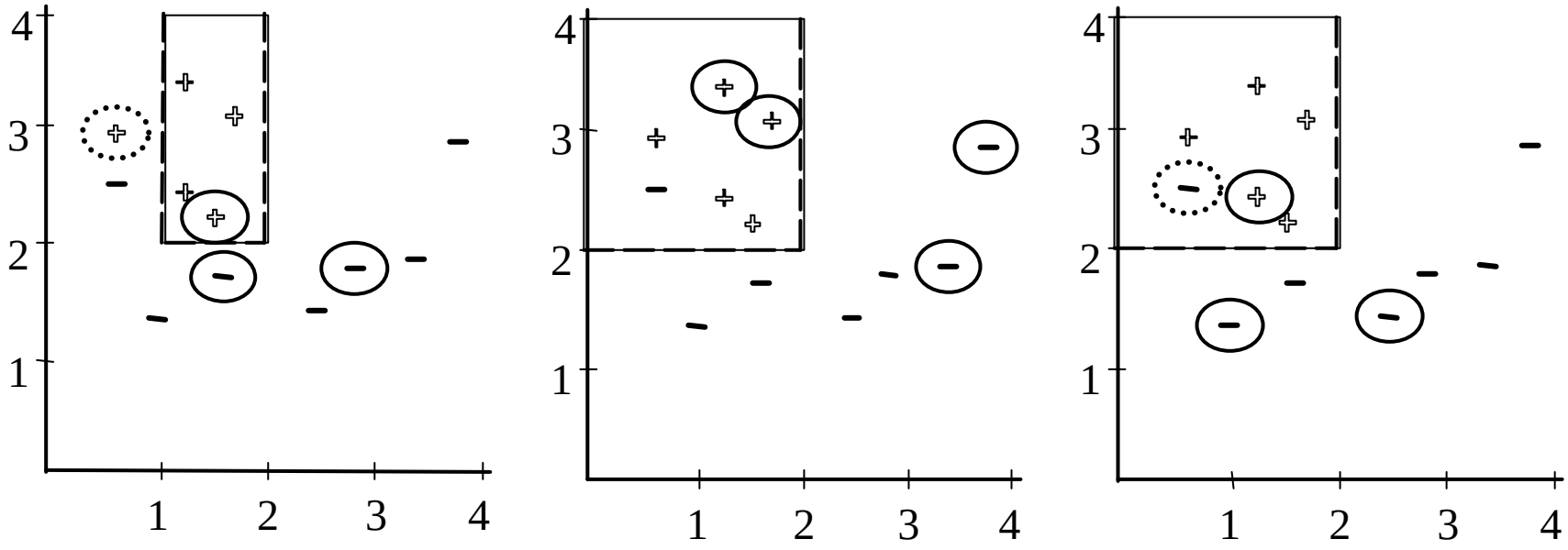
Repetimos k veces:

- Seleccionar un subconjunto distinto cada vez para test y los k-1 restantes para entrenamiento.

La media de error de las k ejecuciones sirve como estimación global.

Muy útil para seleccionar entre diferentes parámetros o entre distintos métodos de aprendizaje con datos limitados.

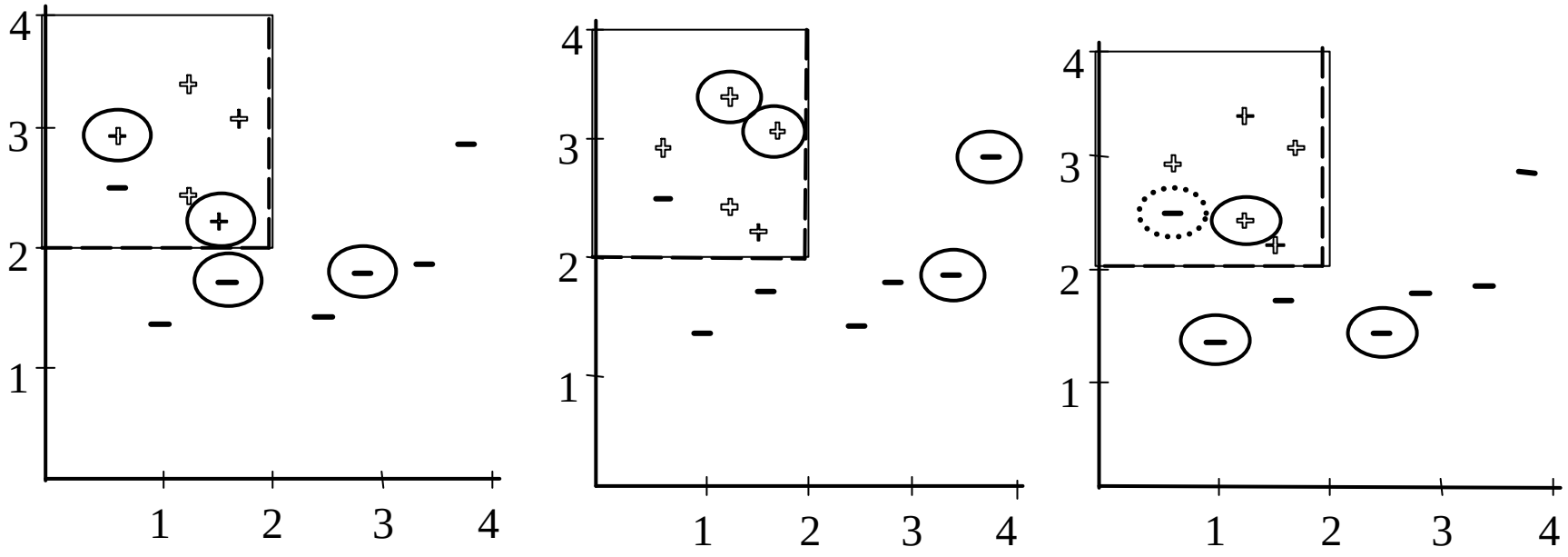
3-fold CV sin poda



error = 0.25 error = 0.0 error = 0.25

Error medio = 0.5 / 3 = 0.167

3-fold CV con número términos <=2



error = 0.0 error = 0.0 error = 0.25

Error medio = 0.25 / 3 = 0.083

Una nota sobre la poda

Podemos realizarla durante el aprendizaje, p.e., no especializo la regla cuando llego a cierto punto (*pre-pruning*, *prospective pruning*).

Podemos hacerla una vez se ha aprendido y simplificar entonces la estructura (*post-pruning*, *retrospective* pruning).

Matriz de confusión (del modelo)

A veces es necesario desglosar los resultados de la predicción para cada valor de la clase.

	Valor real	
Predicción	**NO**	**SI**
NO	2320	290
SI	260	170
Precisión	0.90	0.37

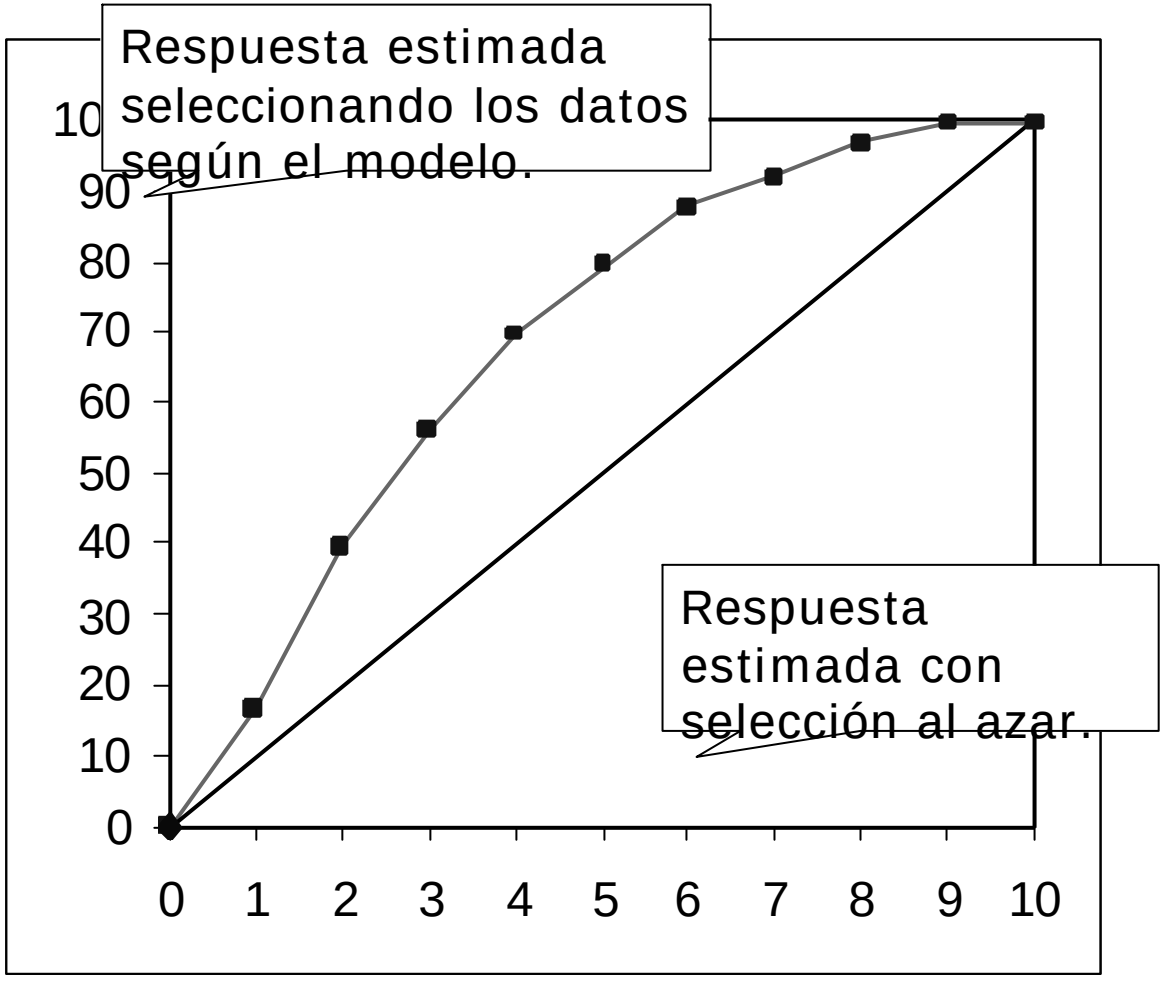
La precisión total es de 0.82 pero sólo acertamos un 37% de las respuestas positivas.

En ciertas aplicaciones es interesante distinguir entre falsos positivos y negativos.

Es muy importante analizarla si los valores de las clases tienen distribuciones muy distintas.

Gráfico de ganancia

Mide en qué proporción una ordenación de los datos según el modelo acierta uno de los valores de la clase.



Campaña de marketing de ACME

La compañía ACME especializada en equipamiento para capturar animales ficticios tiene un nuevo producto “el cazador de correcaminos”. Está dirigido a su fiel audiencia de coyotes y quiere realizar una campaña por correo. Dispone de 60.000€.

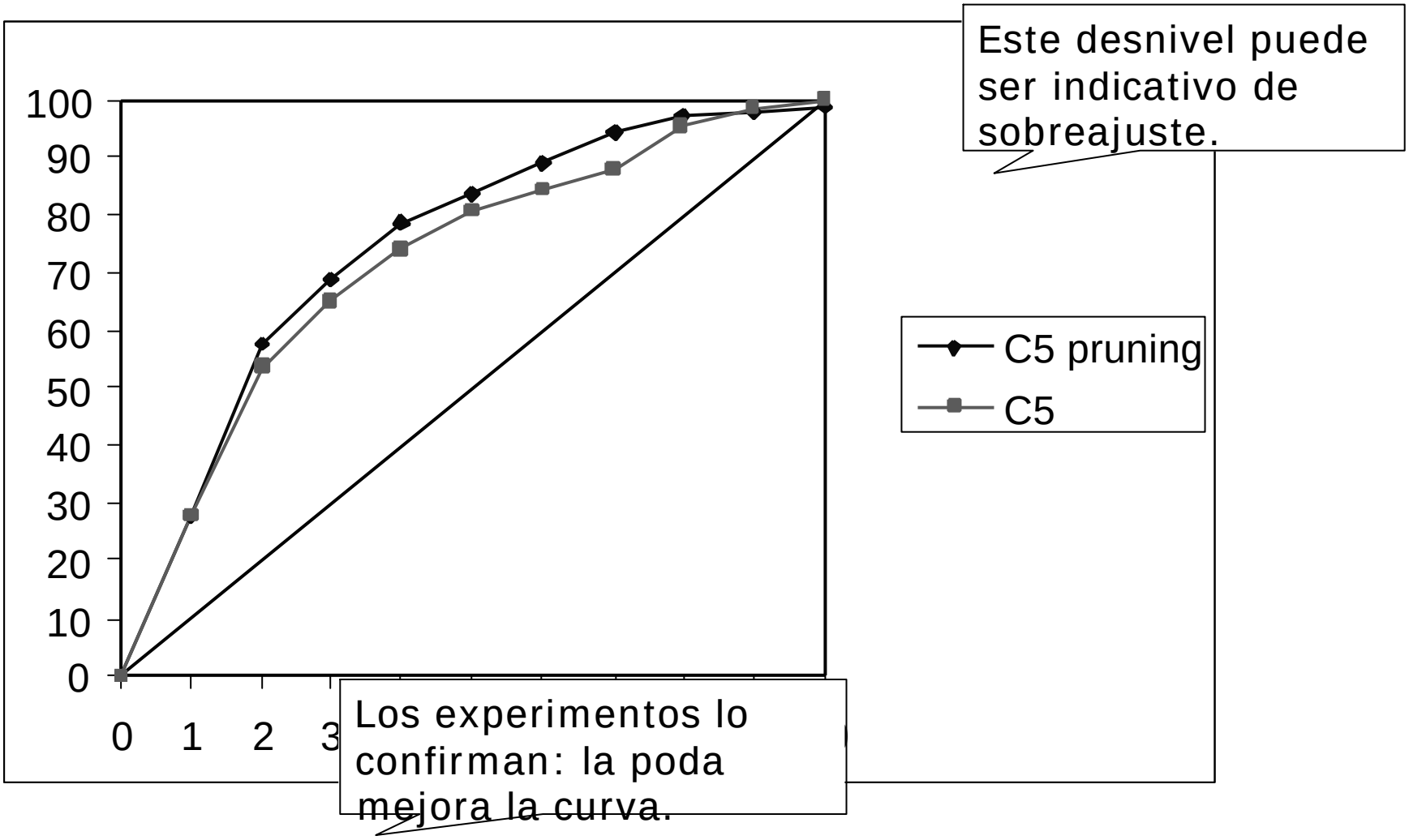
Coste de un envío: 2€

Puede enviar 30.000 ofertas. Pero en su base de clientes hay 100.000 coyotes.

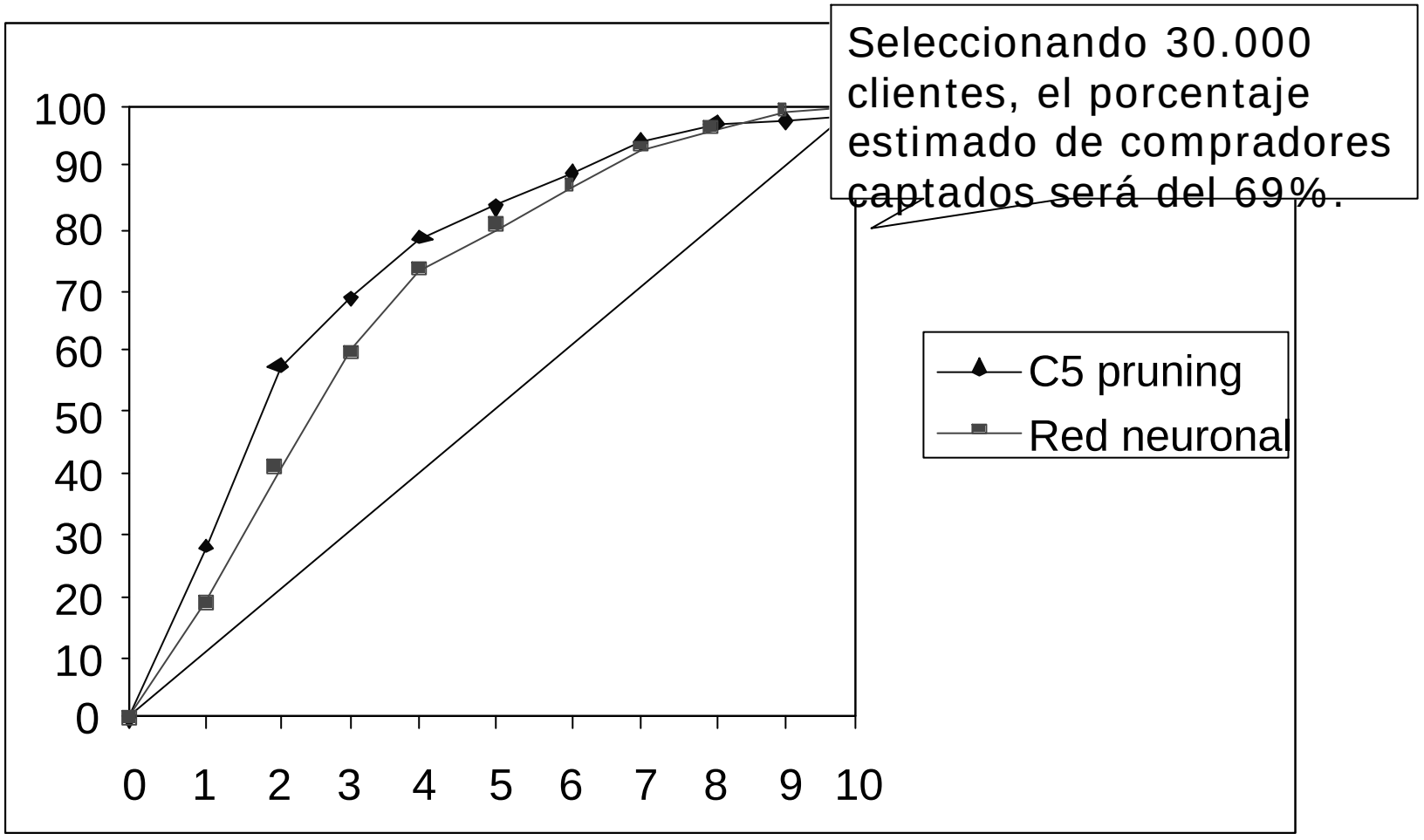
Soluciones para ACME

- Seleccionar 30.000 clientes al azar.
- Realizar un análisis RFM y seleccionar los clientes que han realizado compras recientemente por una cantidad importante.
- Usar métodos no supervisados para encontrar grupos que reflejen perfiles interesantes para abordar con la campaña.
- Construir un modelo predictivo para determinar los clientes con más posibilidad de responder a la campaña.

Evaluación de parámetros

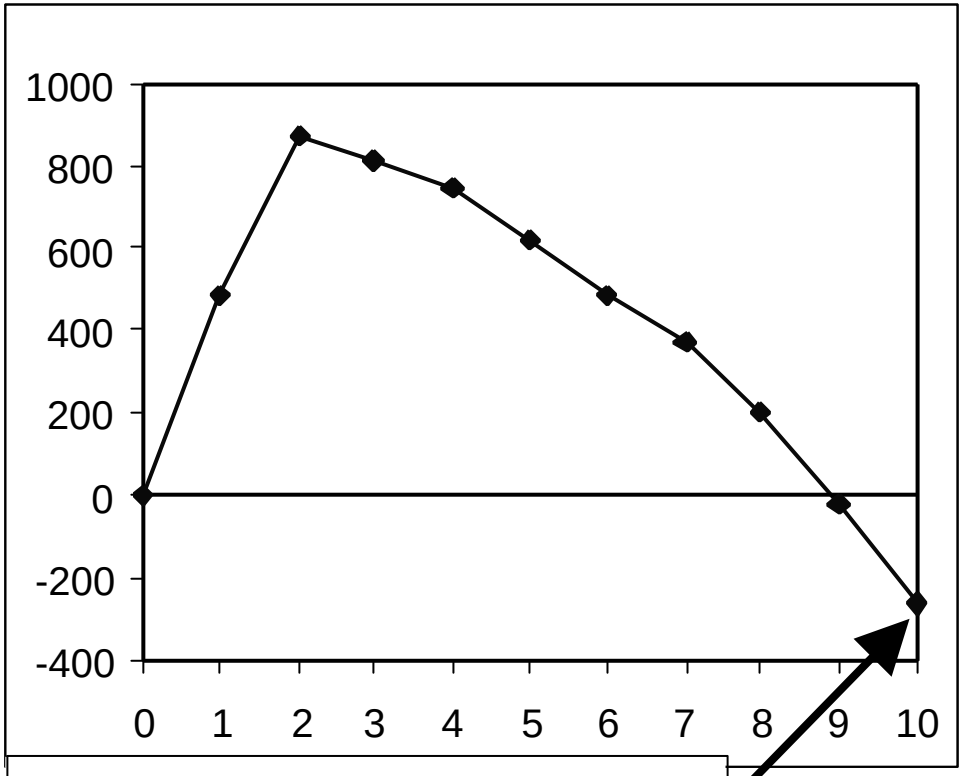


Selección de modelo



Hablar el lenguaje de los usuarios

Utilizando el coste de envío y estimando un beneficio medio por respuesta, podemos cambiar el gráfico de ganancia por uno de **beneficio**.



La estimación indica que el punto máximo de beneficio se obtendría contactando sólo a los mejores 20.000 clientes.

¡Aunque pudiéramos hacer un envío a cada cliente el resultado sería deficitario!

Boosting

•Una innovación incorporada en See5 es el adaptive boosting. La idea es generar varios clasificadores (ya sean árboles de decisión o conjunto de reglas en vez de uno sólo).

•Cuando un nuevo caso debe ser clasificado cada clasificador vota por su clase predicha y los votos se cuentan para determinar la clase final.

Boosting

- ¿Cómo genero varios clasificadores con un único conjunto de entrenamiento?
  - El clasificador normalmente suele cometer errores en algunos ejemplos del conjunto de entrenamiento; supongamos que el primer árbol de decisión, por ejemplo, falle en 7 ejemplos.
  - Cuando el segundo clasificador se construya pondrá mayor atención en clasificar estos 7 casos correctamente. Como consecuencia de esto el segundo clasificador será generalmente diferente del primero. Este cometerá además errores en algunos casos, que serán tratados con mayor atención por el tercer clasificador.
  - Este proceso continúa hasta un número predeterminado de iteraciones, a no ser que lleguemos a un árbol extremadamente ineficiente.

Boosting

- El resultado será algo como esto:

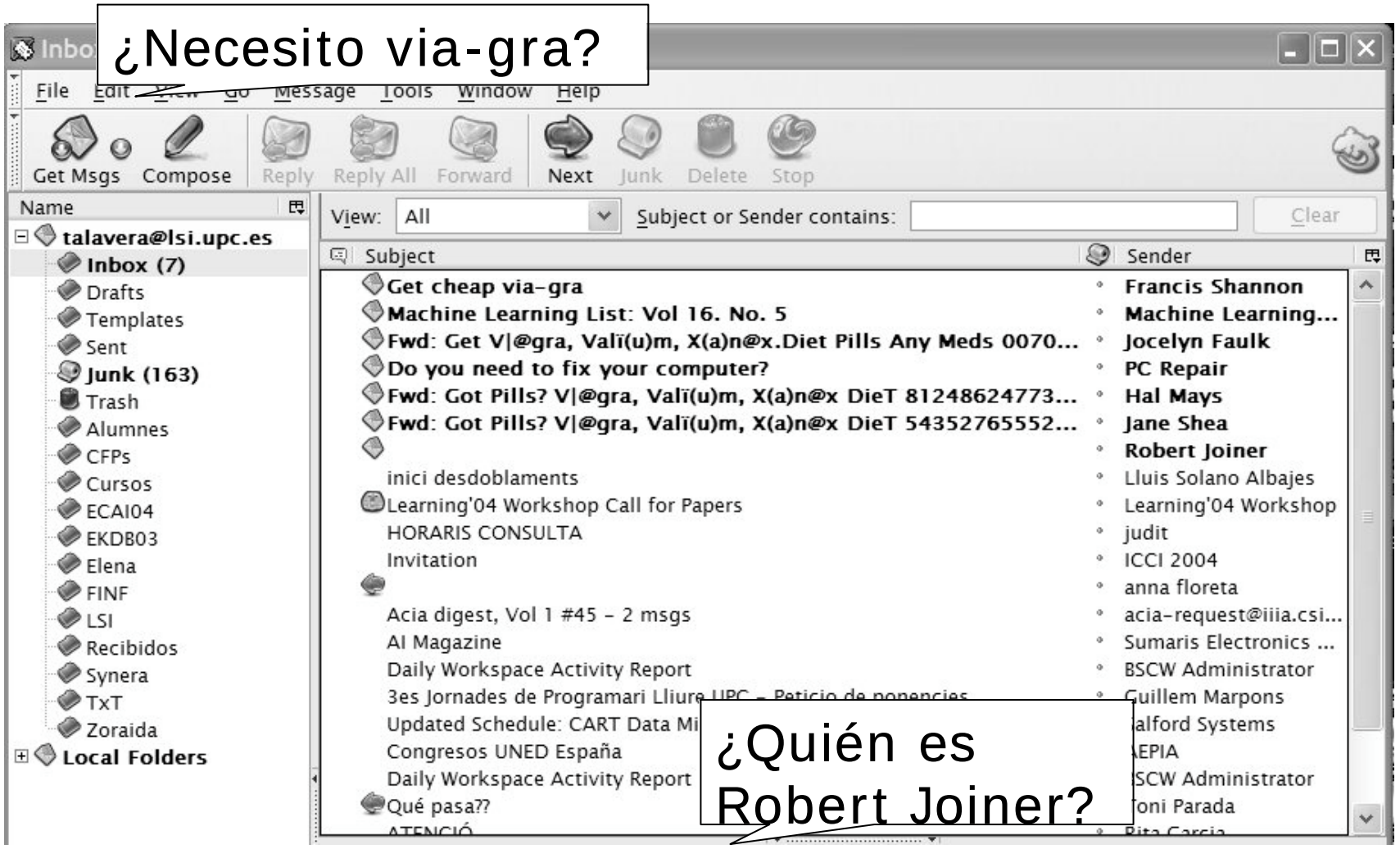
```
Trial          Decision Tree
-------    ------------------
Size   Errors

   0      14    4( 0.4%) -> Coincide con el árbol generado sin boost
   1       7   52( 5.2%)
   2      11    9( 0.9%)
   3      15   21( 2.1%)
   4       7   12( 1.2%)
   5      10    7( 0.7%)
   6       8    8( 0.8%)
   7      13   13( 1.3%)
   8      12   12( 1.2%)
   9      16   54( 5.4%)
boost     2( 0.2%)   << ->  Resultado de la votación para todos los ejemplos.
                            He mejorado el error de 0.4% a 0.2%.
```

Atributos winnowing

- Es una técnica útil cuando hay muchos atributos y no todos ellos son relevantes.
- Se trata de preseleccionar antes del entrenamiento aquellos que a priori pueden ser más relevantes, y tenemos sospechas de que algunos de los atributos no lo son (por ejemplo nº de identificación, nº de dni,...)
- Se debe aplicar cuando hay muchos ejemplos, del orden de 10000.

Algunos aspectos para diseñar un filtro de spam

El dilema via-gra/Joiner



Detección de spam

Un usuario de correo electrónico recibe enormes cantidades de correo no deseado (spam, junk) por haber hecho pública en internet su dirección a merced de robots sin piedad.

El usuario desearía poder identificar rápida y automáticamente los mensajes con más probabilidad de ser spam sin tener que definir reglas de filtrado.

A ver, esos datos...

Lo que necesito...

Id_mensaje	sex	Viagra	laboratori	UNED	Spam
173423	SI	SI	NO	NO	SI
183555	NO	NO	NO	SI	NO
186632	NO	NO	SI	NO	NO

Lo que tengo...

Gone forever are the headaches, hassles and high costs of obtaining the pharmacy products you want and need. When you need them fast: ? V|@Gra ? XAN@x ' S.o.ma < Pnter/m/in > Vali.u.m $ A.t|v@n

Anexamos archivos con la información del I Congreso Internacional de Estilos de Aprendizaje y del IX Congreso Internacional de Informática > Educativa que realizaremos en julio de 2004 en la UNED Madrid España

`m, v:ir, d|pe:x, |, Am:bi3n, S0naT.a

tmana iniciem els a classes de laboratori. o que fara classe la t es el 11, 21 i 31. na farà el 12, 22 i 32, i t. Si un dia es festa no orn. Si us plau aviseu s de la distribució.

Escriben raro ¿no?

Los nombres se escriben incluyendo caracteres aleatorios o especiales para confundir a los filtros: via-gra, V|@gra.

```
Gone forever are the headaches,
hassles and high costs of
obtaining the pharmacy products
you want and need. When you need
them fast: ? V|@Gra ? XAN@x '
S.o.ma < Pnter/m/in > Vali.u.m $
A.t|v@n
```

Necesito un método más flexible para detectar el spam.

Elegir una representación

Hay que convertir la información textual en una forma adecuada para el análisis.

Lo más habitual es usar cada palabra como un atributo distinto (*bag of words*).

Se pueden representar de varias maneras: binaria (aparece o no), número de apariciones,

Hay alternativas menos triviales: n-grams (grupos de palabras), información lingüística, ontologías.

Se genera un número muy elevado de columnas.

Seleccionar / ponderar atributos

Una práctica habitual es ponderar la frecuencia de aparición de cada palabra con su importancia.

Una forma muy popular es el método TFIDF (Term Frequency, Inverse Document Frequency).

Debido a la gran cantidad de atributos que se generan (miles) puede resultar conveniente realizar una selección.

Es normal aplicar métodos muy simples: palabras que se dan con poca frecuencia o alguna medida de correlación con la clase.

Preparación de datos específica

En problemas de text mining, existen métodos específicos de preparación de datos.

Dos prácticas comunes son

Stemming: las palabras con la misma raíz se consideran el mismo atributo.

Stop lists: listas de palabras que no se incluyen en la representación por ser muy frecuentes y no proporconar información. Ej: artículos, preposiciones,...

Selección del método

Los datos no son estáticos, sino que van llegando nuevos mensajes continuamente y las palabras también van cambiando.

El concepto de spam/no spam puede ir variando con el tiempo por lo que necesitamos un método que sea capaz de aprender de forma incremental.

En este caso elegimos el algoritmo Naive Bayes por ser muy eficiente y adaptarse de forma natural al aprendizaje incremental.

Para cada nuevo mensaje, calcularemos la probabilidad de que sea spam y lo marcaremos si supera un umbral.

Ajustes específicos para el problema

Cada vez que llega un nuevo mensaje, se obtiene una representación y se seleccionan las 15 palabras más relevantes.

La relevancia se calcula midiendo la desviación respecto al valor 0.5 de la probabilidad de la palabra entre el spam.

Con estas 15 palabras se aplica Naive Bayes para obtener la probailidad de que sea spam.

Las palabras que se encuentran en los mensajes normales se cuentan dos veces (ver evaluación).

Evaluación

En este problema lo importante no es únicamente la precisión total, sino la distinción entre falsos positivos y negativos.

Un falso positivo puede ser mucho más crítico que un falso negativo porque podemos perder un mensaje importante.

Doblando las frecuencias de las palabras que aparecen en el no spam se sesgan las probabilidades para tender a no detectar falsos positivos.

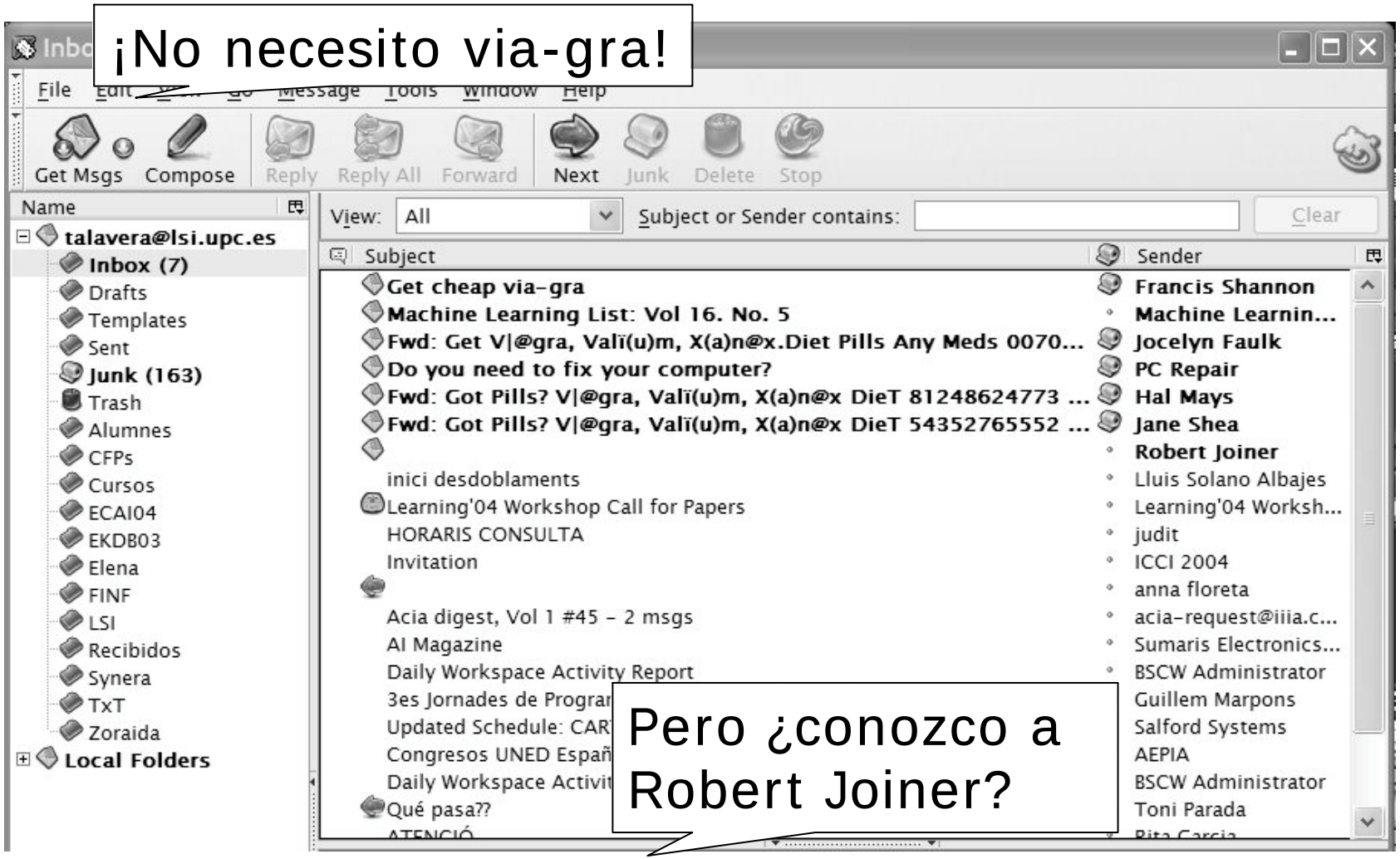
Otras cuestiones

Adicionalmente, puede ser necesario resolver otras cuestiones específicas para el método concreto que usamos aplicado a este problema específico.

A menudo estas cuestiones deben resolverse encontrando la respuesta de forma experimental.

Ej: ¿Qué probabilidad de spam asignamos a palabras que no hemos visto para calcular su relevancia?

¿Dilema solucionado?



El mensaje de Robert Joiner

Why not buy V-I-A-G-R-A - No Prescription Needed !!

Costs over 50% less than Viagra®

http://www.8Eg.34edmnr5.com/gp/default.asp?ID=bw

We also have these medications in highly discounted generic form:

Ambien, Xanax, Phentermine, Lipitor, Nexium, Paxil, and Vioxx.

Physician Consultation: FREE! Fast, FREE delivery

EZ online form

También ofrece viagra (bueno V-I-A-G-R-A).

¿Por qué le cuesta de detectar?

Algo impide que mire los términos adecuados.

El truco de Robert Joiner

```
rhetorician dielectric terpsichore
insight meridian chokeberry borealis
whatever tx constantinople brutal kink
harmon banjo scotsman substitute vat yang
sound voluntary decomposition chalmers
honoree amethystine bellini you've buff
monotreme avoidance chugging debussy
dragonfly moduli waxwork chimique
draftsperson unanimity diamond mckesson
corrosion annual alden augmentation
timothy polytope headache boo conferred
coupon bezel borden contemplate moreland

compensatory oberlin scarf infrequent
liquidus lobule coriolanus newsstand
farmhouse r cheater mathematician
delicate ballroom celesta bergland
pedestal aesthetic uplift ego coverall
transposition chattanooga dynamic delhi
rood genre
```

Al final del mensaje incluye una lista aleatorias de palabras que no se encuentran con frecuencia en mensajes spam.

Lo siento Robert...

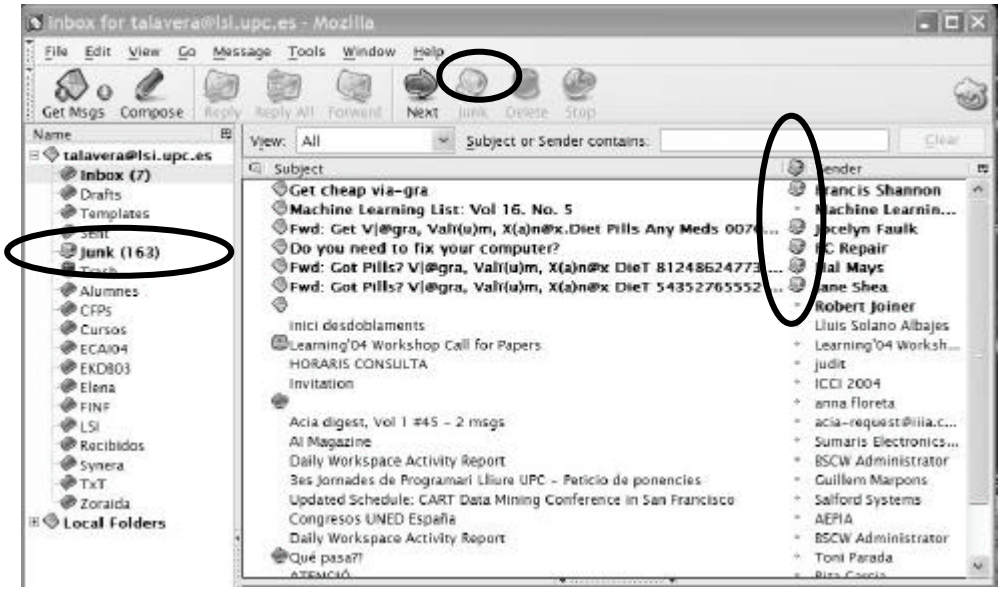
Es posible indicarle al programa de correo que un mensaje es o no spam para que Naive Bayes modifique las probabilidades.

Dependiendo del grado de entrenamiento del sistema, puedo tener que indicárselo varias veces. Con el tiempo, el rendimiento mejora.

El sistema satisface las necesidades del usuario, se adapta a nuevas tácticas de los spammers y evita tener que hacer largas listas de filtros.

Aplicación

No parece muy complejo de implementar.



La herramienta de correo electrónico de Mozilla ya lo incorpora, creo que me lo voy a ahorrar ☺

(no es publicidad, es gratis)