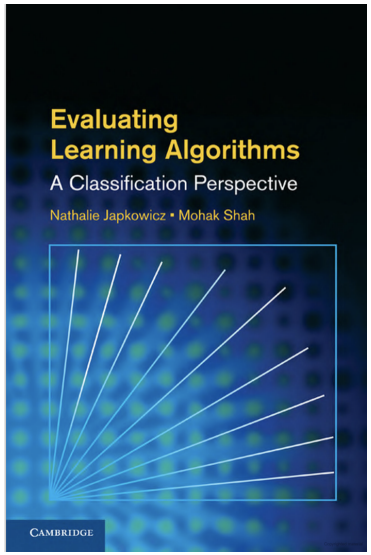
Evaluación

Eduardo Morales y Hugo Jair Escalante

Evaluación

- Una parte importante de este material está tomado de: Machine Learning Evaluation: A Classification Perspective. Curso tutorial de Nathalie Lapkowicz, International Conference on Machine Learning 2011.
- Para más información ver: Evaluating Learning Algorithms: A Classification Perspective Nathalie Japkowicz & Mohak Shah Cambridge University Press, 2011

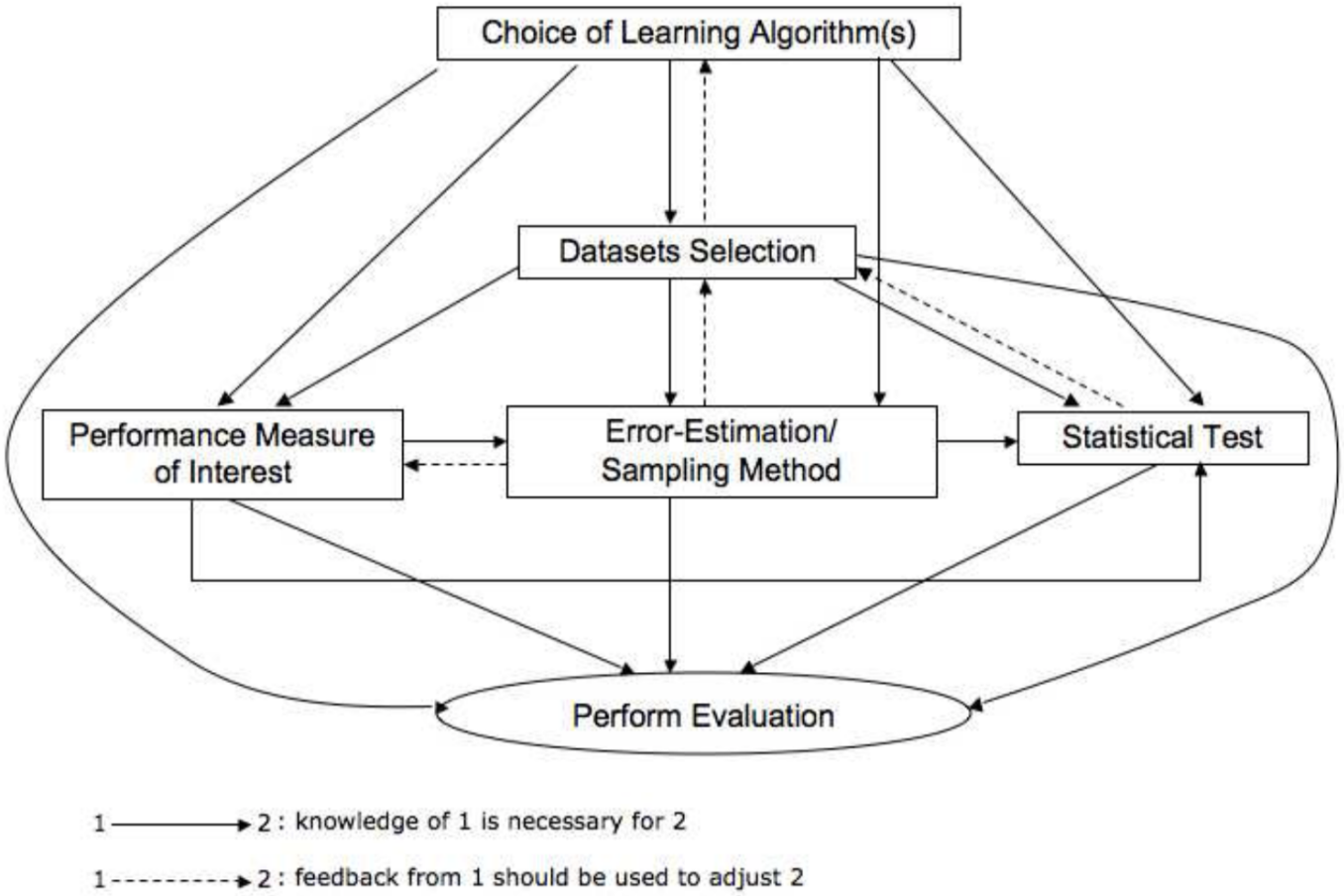


Evaluación

Otroas fuentes *altamente* recomendadas:

- T.G. Dietterich. **Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms** *Neural Computation*, Vol. 10:1895–1924, 1998.
- J. Demsar. **Statistical Comparisons of Classifiers over Multiple Data sets.** *Journal of Machine Learning Research*, Vol. 7:1–30, 2006.
- S. García, F. Herrera. **An Extension to "Statistical Comparisons of Classifiers over Multiple Data sets" for all Pairwise Comparisons.** *Journal of Machine Learning Research*, Vol. 9:2677–2694, 2008.

Evaluación



Introducción

- Uno de los aspectos más importantes dentro de cualquier sistema de aprendizaje es el poder evaluar su desempeño
- La evaluación nos da evidencia para anticipar el correcto funcionamiento del sistema
- Una evaluación sistemática es imprescindible para publicar resultados y avanzar el estado del arte

Introducción

- Diferentes métodos hacen diferentes suposiciones, tienen sesgos y características
- Con tantas variantes de algoritmos de aprendizaje es crítico evaluar objetivamente su desempeño
- Tal evaluación también es imprescindible para seleccionar el mejor modelo (optimización de parámetros)

Evaluación

La evaluación abarca varios aspectos

- ¿Qué medida usar para evaluar un algoritmo?
- ¿Qué prueba estadística usar?
- ¿Qué muestreo de datos?

Evaluación

Cuando corremos clasificadores podemos usar varias medidas de evaluación y lo que queremos saber cuál es el mejor clasificador (e.g., en UCI Breast Cancer)

Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	71.7	.4534	.44	.16	.53	.44	.48	.7	48.11
C4.5	75.5	.4324	.27	.04	.74	.27	.4	.59	34.28
3NN	72.4	.5101	.32	.1	.56	.32	.41	.63	43.37
Ripp	71	.4494	.37	.14	.52	.37	.43	.6	22.34
SVM	69.6	.5515	.33	.15	.48	.33	.39	.59	54.89
Bagg	67.8	.4518	.17	.1	.4	.17	.23	.63	11.30
Boost	70.3	.4329	.42	.18	.5	.42	.46	.7	34.48
RanF	69.23	.47	.33	.15	.48	.33	.39	.63	20.78

Evaluación

Podemos “rankear” los clasificadores y también preguntarnos cuál es el mejor

Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	3	5	1	7	3	1	1	1	2
C4.5	1	1	7	1	1	7	5	7	5
3NN	2	7	6	2	2	6	4	3	3
Ripp	4	3	3	4	4	3	3	6	6
SVM	6	8	4	5	5	4	6	7	1
Bagg	8	4	8	2	8	8	8	3	8
Boost	5	2	2	8	7	2	2	1	4
RanF	7	6	4	5	5	4	7	3	7

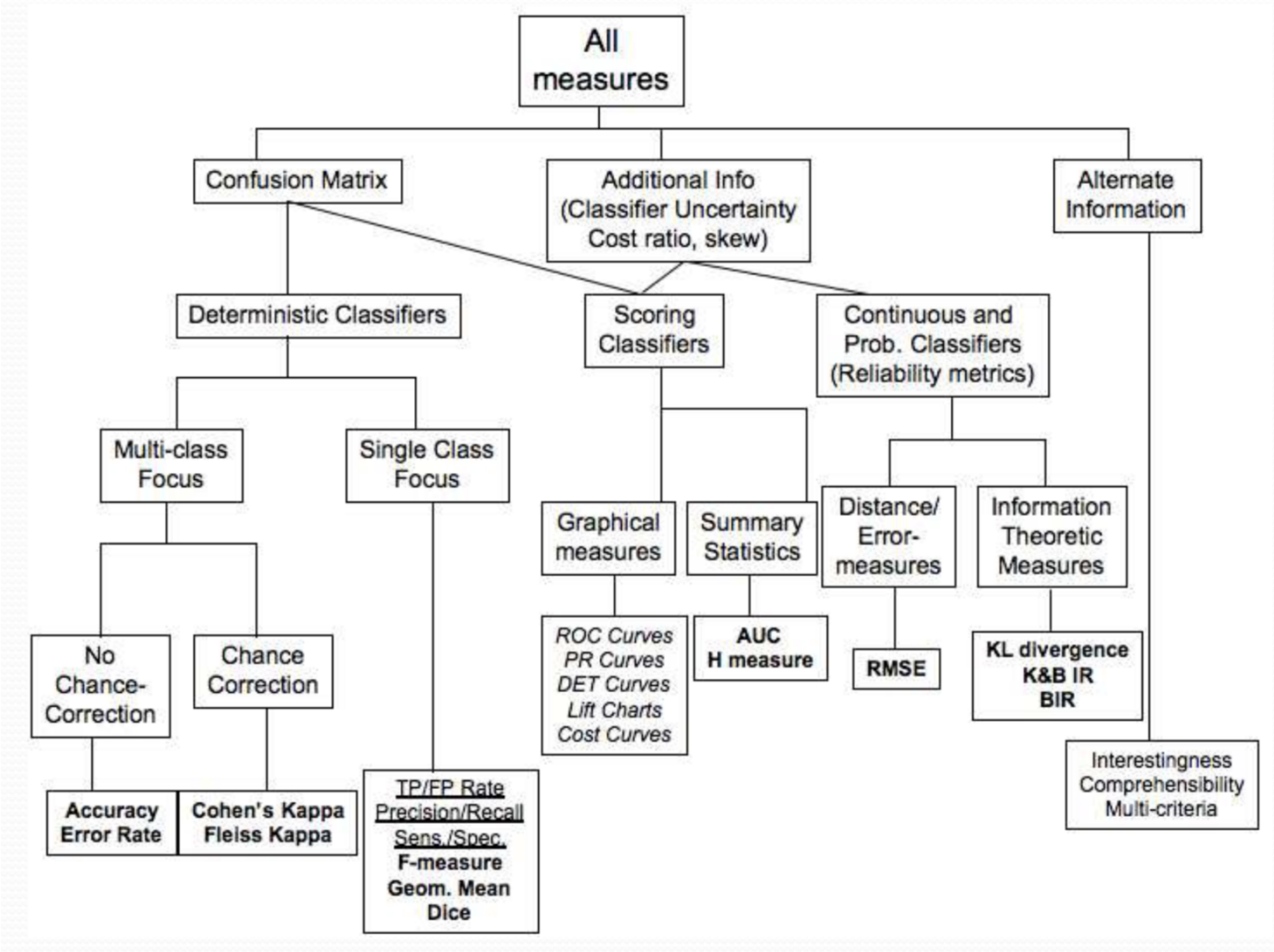
Evaluación

- De entrada las medidas se fijan en diferentes aspectos, algunas aplican para una sola clase y otras son globales (para todas las clases)
- Medidas locales, como precisión y recuerdo, tienden a ser complementarias, no pueden ganar en una sin perder en la otra y miden diferentes aspectos
- Las globales pueden no estar de acuerdo pero es más difícil identificar las peculiaridades ya que son composiciones

Evaluación

- Si estamos trabajando en una aplicación en particular, y queremos medir ciertos aspectos, en general, no es problema decidir qué función(es) usar
- Si lo que queremos es comparar nuestro algoritmo con otros, entonces es un poco más complicado porque para algunas medidas nos puede ir bien pero para otras mal
- Si usamos pocas medidas, nos pueden tachar de crear sesgos en los resultados
- Si usamos muchas, pueden no quedar claras las ventajas de nuestro algoritmo
- En general, se reportan los resultados que reportaron con los que nos queremos comparar, pero de nuevo puede no ser adecuado

Medidas



Medidas de Evaluación

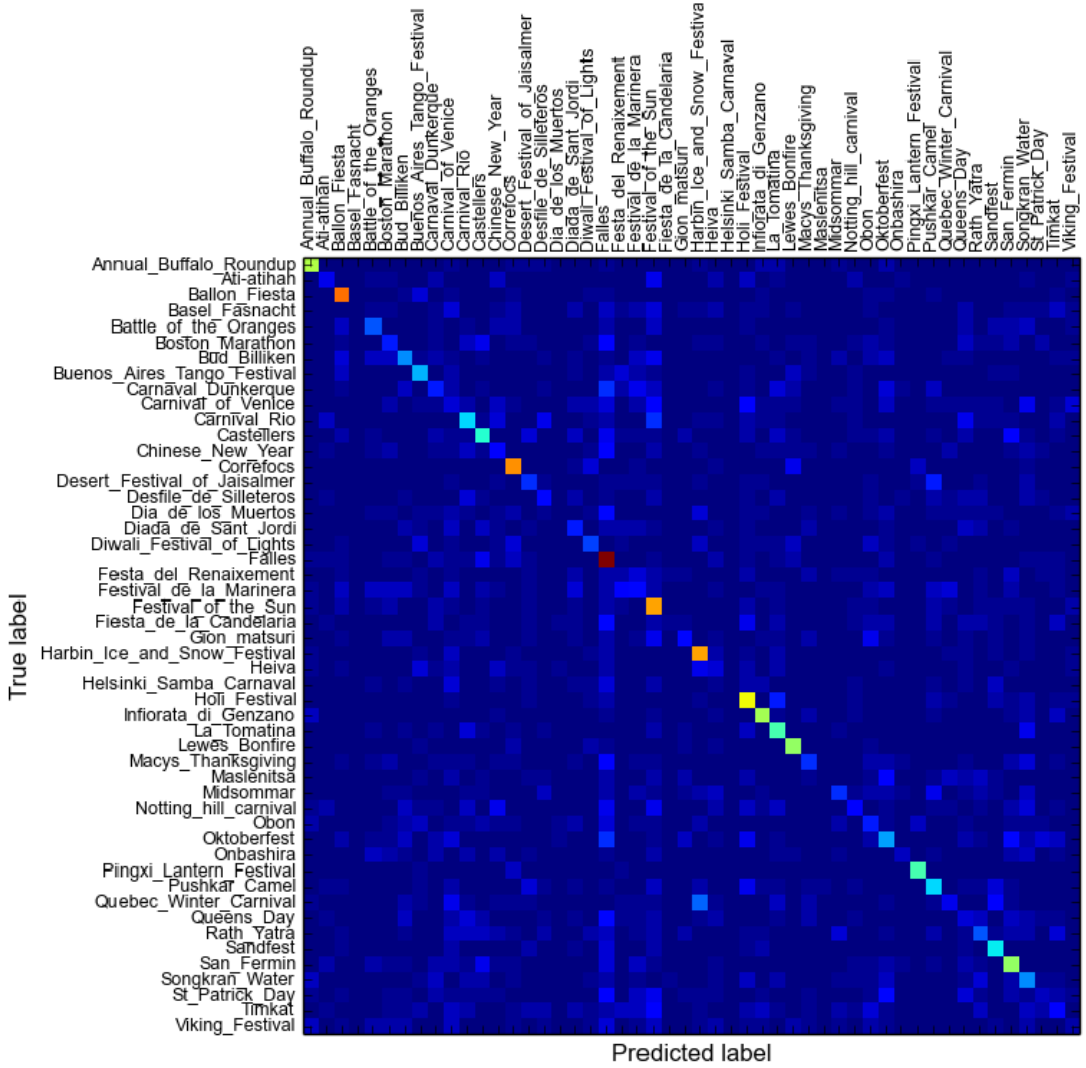
- Normalmente los datos se reportan en lo que se conocen como matrices de confusión (*confusion matrix*) que reportan los verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN)

Matriz de Confusión para clase binaria

	$y_i^T = 1$	$y_i^T = -1$
$\hat{y}_i^T = 1$	***TP***	***FP***
$\hat{y}_i^T = -1$	***FN***	***TN***

- **TP:** Verdaderos positivos
- **FP:** Falsos positivos
- **TN:** Verdaderos negativos
- **FN:** Falsos negativos

Matriz de Confusión para muchas clases



Medidas

Si $P = TP + FN$ y $N = TN + FP$

- **Accuracy:** $= \frac{(TP+TN)}{(P+N)}$
- **Precision:** $Prec(\oplus) = \frac{TP}{(TP+FP)}, Prec(\ominus) = \frac{TN}{(TN+FN)}$
- **Recall/TP rate:** $Rec(\oplus) = \frac{TP}{TP+FN}, Rec(\ominus) = \frac{TN}{TN+FP}$
- **FP Rate:** $= \frac{FP}{N}$
- **Sensitivity (Recall):** $= \frac{TP}{P} = \frac{TP}{TP+FN}$
- **Specificity:** $= \frac{TN}{N} = \frac{TN}{TN+FP}$
- **F-Measure:** $= \frac{2 \times Prec \times Rec}{Prec+Rec}$ (usualmente $\beta = 1$)

Medidas

Con más clases:

- **Macro-promedio.** Se calcula la medida f_1 para cada una de las clases del problema, y se promedian los resultados. Mismo peso a todas las clases
- **Micro-promedio.** Calcula TP, FP, TN, FN para todas las categorías y se calcula la medida f_1. Mismo peso a todos las instancias.

Costo de Clasificación

- A veces el costo de una mala clasificación es más importante para una clase que para la(s) otra(s) - i.e., no es simétrico
- Por ejemplo, diagnosticar una enfermedad mortal
- A veces se tienen pocos datos de una clase, por ejemplo, si un solo día al mes una vaca es fértil, un clasificador podría predecir que las vacas nunca son fértiles con un 97% de éxito (nada malo para muchos dominios) con valores diferentes de cero fuera de la diagonal de la matriz de confusión
- Otra posibilidad es darle más peso o importancia a ciertos datos (algunos sistemas lo permiten) o repetir muchas veces ese dato (para reflejar su peso)

Algunos problemas con Accuracy

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

True class →	Pos	Neg
Yes	400	300
No	100	200
	P=500	N=500

- Los dos nos dan un accuracy del 60%
- Pero uno tiene un fuerte/débil reconocimiento en los positivos y fuerte/débil en los negativos

Problemas con Precision/Recall

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

True class →	Pos	Neg
Yes	200	100
No	300	0
	P=500	N=100

- Los dos tienen el mismo valor de precision (66.66%) y de recuerdo (40%) - Nota: los conjuntos de datos son diferentes
- Aunque tienen el mismo PRR, tienen muy diferentes NRR (aquí *Accuracy* no tiene problema para encontrar esta diferencia)

Clases Desbalanceadas

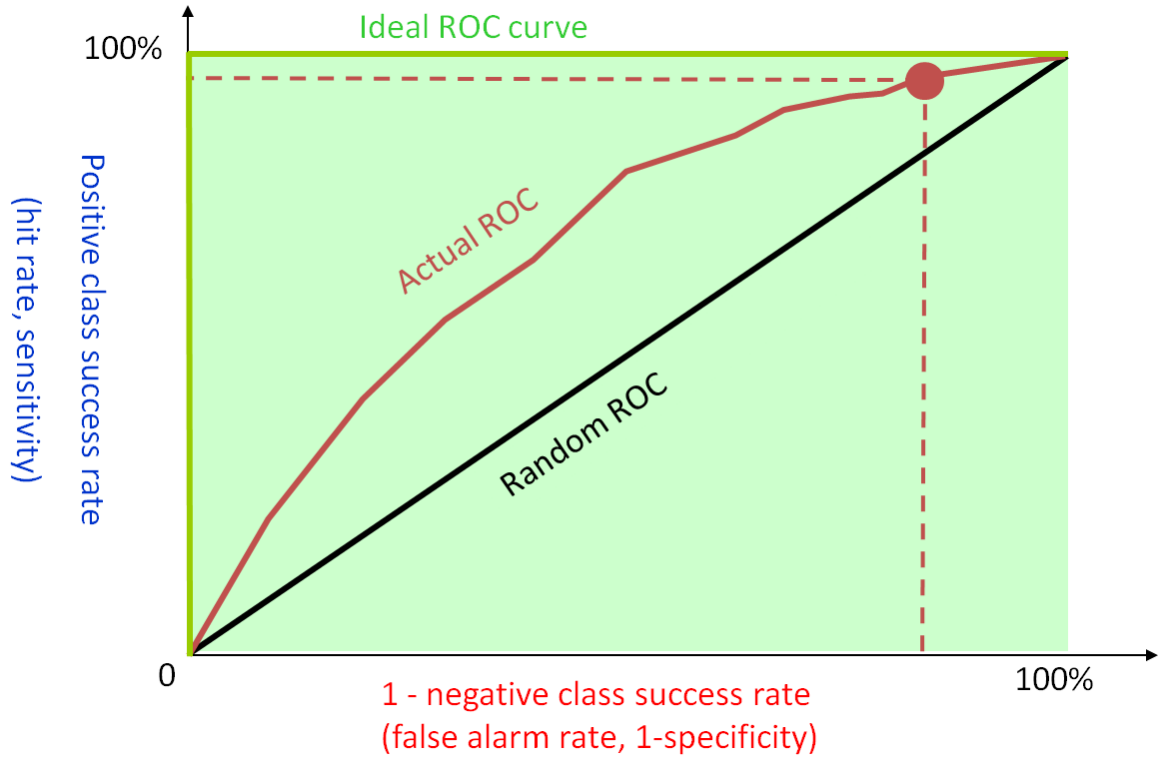
- Una medida común para evaluar salidas categóricas en clases desbalanceadas es el Balanced Error Rate (BAE)

$$BER(\hat{(}f)) = \frac{E_- + E_+}{2}$$

donde E_-/E_+ es la tasa de error en instancias de la clase negativa/positiva respectivamente

Curvas ROC y AUC

- Para atenuar algunos de estos problemas se han propuesto otras medidas de evaluación
- La curva ROC (Receiving Operator Characteristic): Para un umbral dado sobre $\hat{f}(\mathbf{x})$ se obtiene un punto de la curva ROC:



Curvas ROC

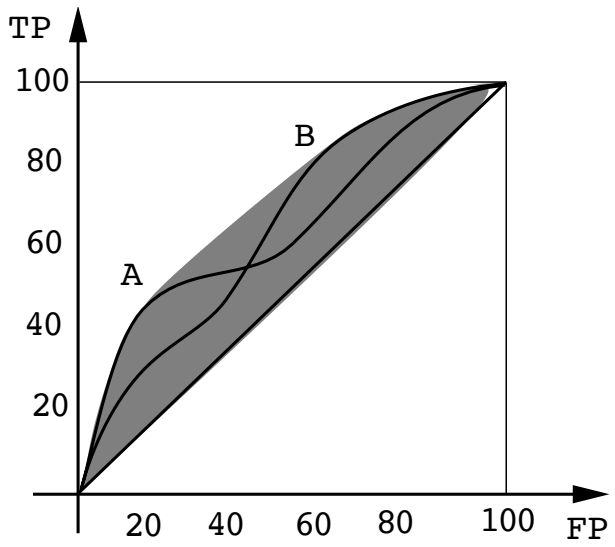


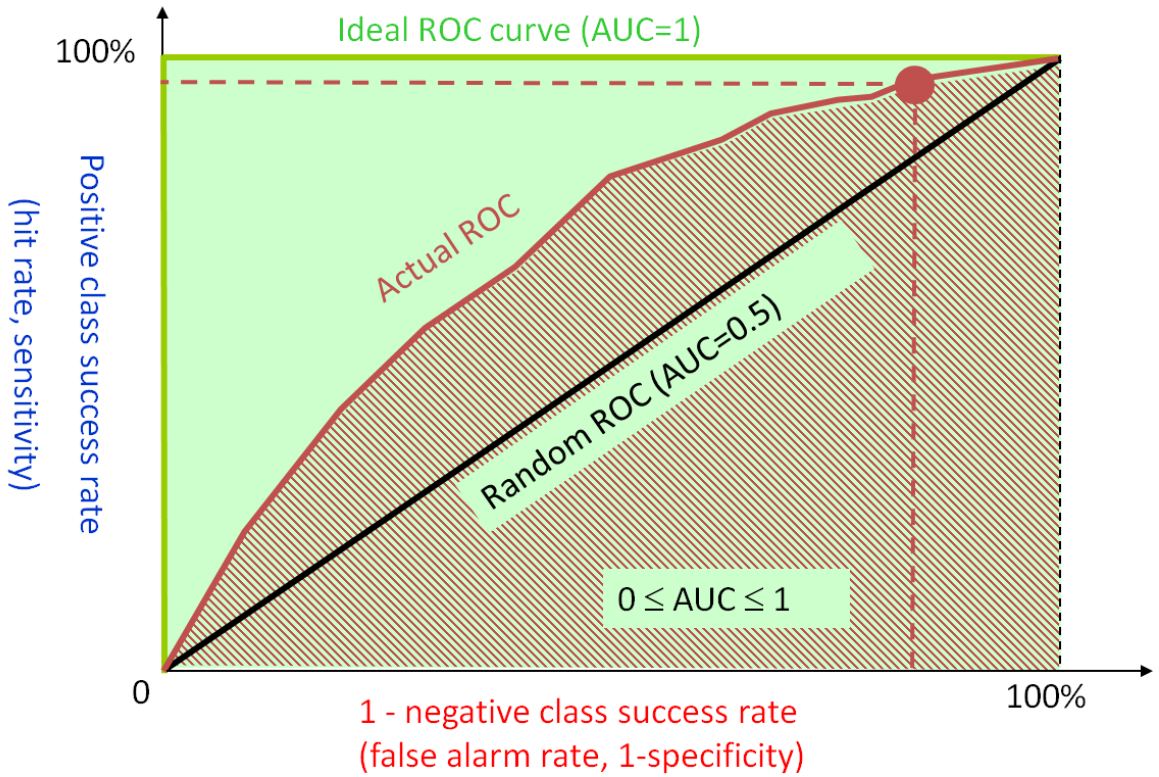
Figure: Curva ROC para dos sistemas de aprendizaje.

Curvas ROC

- Un ejemplo de una tabla ROC típica de dos algoritmos de aprendizaje (A y B) se muestra en la figura 1
- La zona sombreada es lo que se conoce como *convex hull* y se puede obtener cualquier punto de la zona sombreada combinando los clasificadores
- Dentro de la curva ROC, el (0,0) representa siempre negativo, el (1,1) siempre positivo, el (0,1) el clasificador ideal y el (1,0) el clasificador que tiene todo mal
- Los puntos arriba de la diagonal (0,0) - (1,1) representan clasificadores que se comportan mejor que uno aleatorio
- La diagonal (0,1) - (1,0) representan clasificadores que se comportan igual con cualquiera de las dos clases

Curvas ROC y AUC

- A menudo es complicado/subjetivo comparar curvas, puede un solo número resumir una curva?
- AUC: Área bajo la curva



Curvas ROC y AUC

- Muchas veces se usa en lugar de Accuracy
- Las curvas se pueden cruzar
- Las distribuciones de costo de mala clasificación pueden ser diferentes para diferentes algoritmos y la curva puede mostrar más peso para un clasificador que para otro (podemos estar comparando peras con manzanas)
- Existen otras propuestas: H-Measure, curvas de costo, ...

Otras medidas basadas en curvas

Existen otras medidas para evaluar la proporción de falsos positivos y falsos negativos

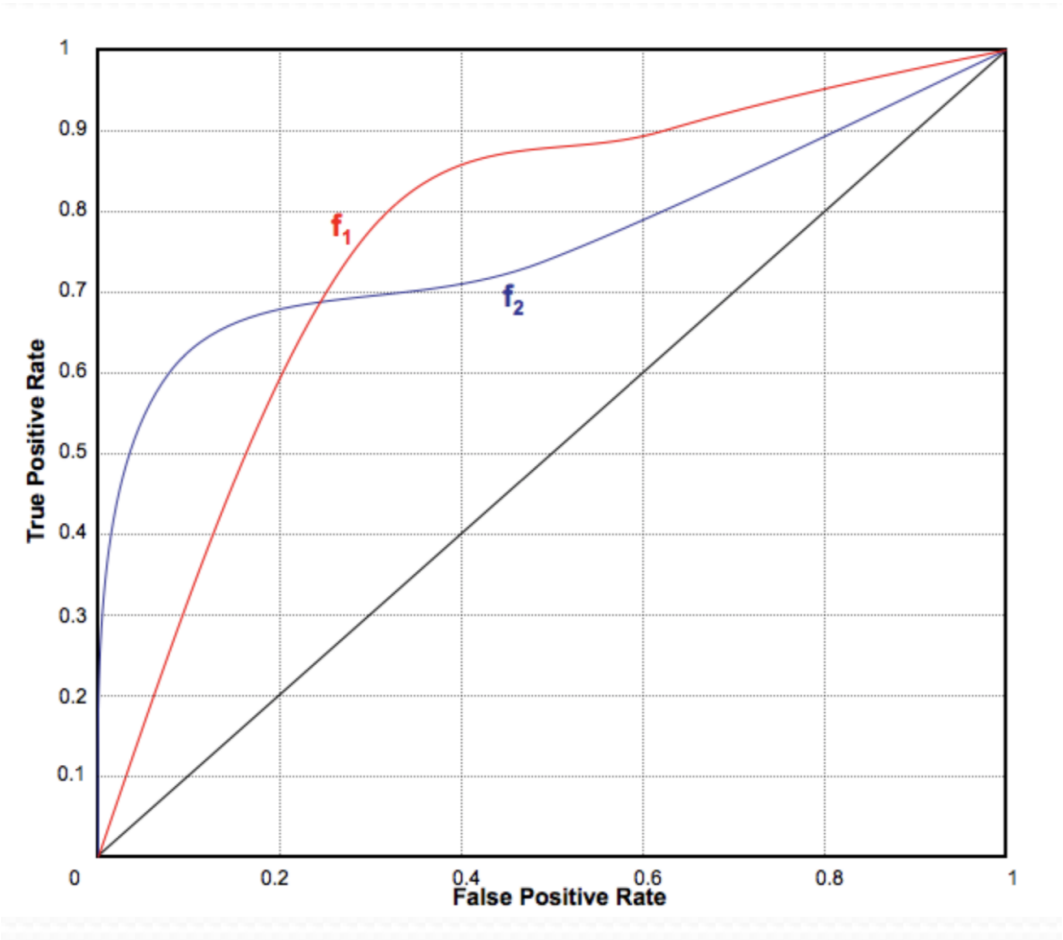
Nombre	Ejes	Explicación
lift chart	TP vs. tamaño subconjunto	TP $\frac{TP+FP}{TP+FP+TN+FN} \times 100\%$
ROC curve	razón TP vs. razón FP	$\frac{TP}{TP+FN} \times 100\%$ $\frac{FP}{FP+TN} \times 100\%$
recall-precision	recall vs. precision	razón TP $\frac{TP}{TP+FP} \times 100\%$

Otras curvas

- Una posibilidad es graficar FPT (*false positive rate*) en el eje X, TPR (*true positive rate*) en el eje Y (curva ROC) y por ejemplo, la frecuencua de positivos (TP/(TP+TN)) en el eje Z
- Estas curvas ROC 3-D nos pueden dar idea de qué tanto depende alguna medida de desempeño con respecto a la distribución de las clases
- Otra medida de evaluación es *Weighted relative accuracy* o WRAcc que se puede comportar mejor que medidas como precisión, *accuracy*, y *F-measure*

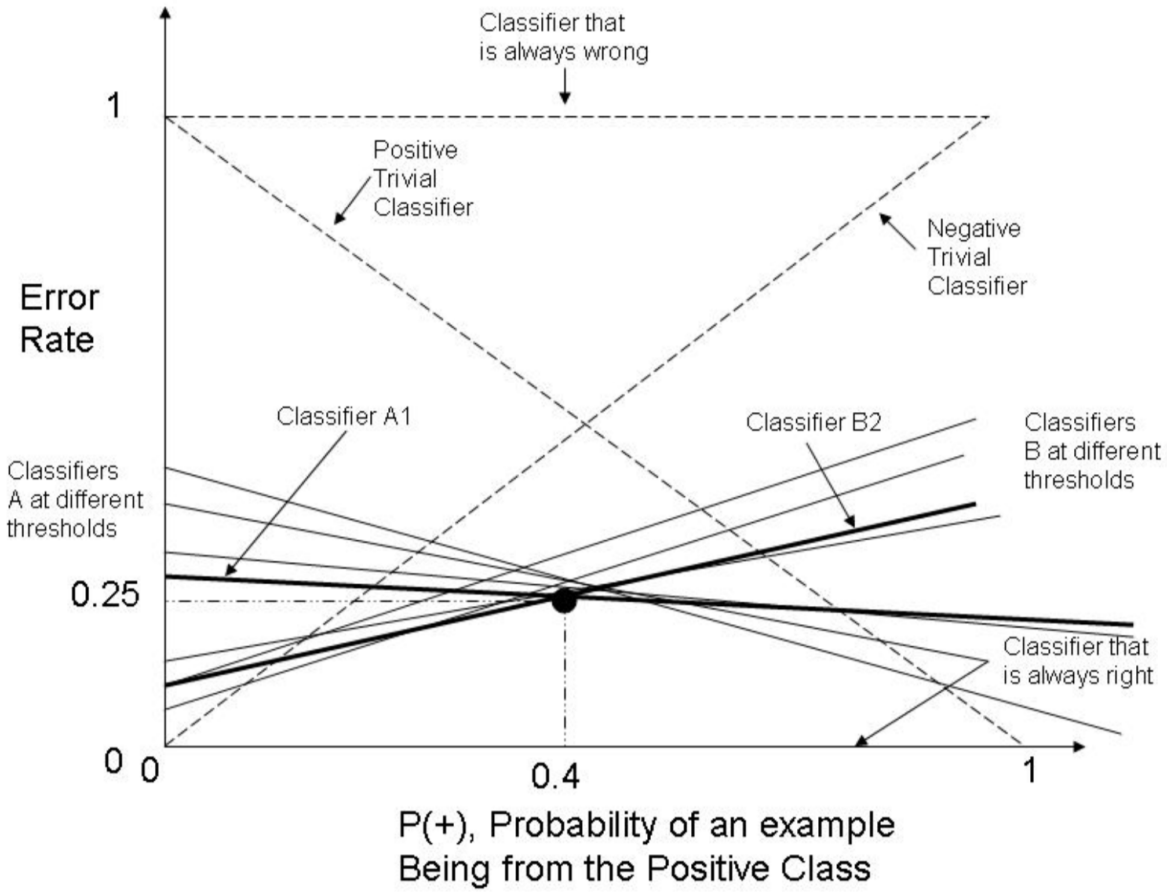
$$\text{WRAcc(IF Cond Then Clase)} = p(\text{Cond})(p(\text{Clase}|\text{Cond}) - p(\text{Clase}))$$

Curvas de Costo



Las curvas ROC, cuando se cruzan, sólo nos dicen que a veces un clasificador en mejor que otro

Curvas de Costo



Las Curvas de Costo nos dicen para qué probabilidades de clase un clasificador es preferible que otro

La medida Kappa de Cohen

- Accuracy no considera que la clasificación correcta puede ser por casualidad
- La medida Kappa tratar de corregir este problema:

$$\kappa = \frac{(P_0 - P_e^C)}{(1 - P_e^C)}$$

donde

- $P_0 =$ la propabilidad de coincidencia entre las clases dadas por el clasificador y por el proceso
- $P_e^C =$ la coincidencia por casualidad en las clases y se define como la suma de la proporción de ejemplos asignados a la clase por la proporción de las etiquetas verdaderas en la base de datos

La medida Kappa de Cohen: Ejemplo

Predicted -> Actual	A	B	C	Total
A	60	50	10	120
B	10	100	40	150
C	30	10	90	130
Total	100	160	140	

- Accuracy $= P_0 = (60 + 100 + 90)/400 = 62.5\%$
- $P_e^C = 100/400 \times 120/400 + 160/400 \times 150/400 \times 140/400 \times 130/400 = 0.33875$
- En este ejemplo, Accuracy es muy optimista

Salidas Continuas

- Root Mean Square Error (RMSE):

$$RMSE(f) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2}$$

  donde m es el número de ejemplos de entrenamiento
- Mean Absolute Error (MAE):

$$MAE(f) = \frac{1}{m}\sum_{i=1}^{m}|f(x_i) - y_i|$$

RMSE: Ejemplo

ID	$f(x_i)$	y_i	$(f(x_i) - y_i)^2$
1	.95	1	.0025
2	.6	0	.36
3	.8	1	.04
4	.75	0	.5625
5	.9	1	.01

$$RMSE(f) = \sqrt{\frac{1}{5}(0.0025 + 0.36 + 0.04 + 0.5625 + 0.1)} = 0.4416$$

Information Score

$$IS(x) = I(P(y|f) \geq P(y)) \times (-log(P(y)) + log(P(y|f)) + I(P(y|f) < P(y)) \times (-log(1 - P(y)) + log(1 - P(y|f))$$

$$IS_{prom} = \frac{1}{m} \sum_{i=1}^{m} IS(x_i)$$

donde $P(y)$ se obtiene de los datos, $P(y|f)$ es la salida de un clasificador probabilista (f), $I(\cdot)$ es la función indicatriz (*indicator*)

x	$P(y_i|f)$	y_i	IS(x)
1	.95	1	0.66
2	.6	0	0
3	.8	1	.42
4	.75	0	.32
5	.9	1	.59

$P(y = 1) = \frac{3}{5} = 0.6;$
$P(y = 0) = \frac{2}{5} = 0.4;$
$I(x_1) = 1 \times (-log(0.6) + log(0.95)) + 0 \times (-log(0.4) + log(0.5)) = 0.66;$
$I_{prom} = \frac{1}{5}(0.66 + 0.42 + 0.32 + 0.59) = 0.40$

Evaluación de Hipótesis

- Uno de los aspectos importantes dentro de cualquier sistema aprendizaje es el poder evaluar su desempeño
- Cuando se tienen pocos datos se tienen los siguientes problemas:
  - Sesgo en la estimación
  - Varianza en la estimación

Sesgo en la estimación

- La precisión observada en la muestra no es un buen estimador de la precisión sobre futuras instancias
- El estimador será **optimista**, más aún cuando se tiene un espacio de hipótesis grande y hay un **sobreajuste** de los datos
- *Es por esto que probamos con datos que no usamos para entrenar*

Varianza en la estimación

- Aún cuando la precisión de la hipótesis se mide con un conjunto de prueba independiente del conjunto de entrenamiento, la precisión medida puede variar de la precisión verdadera y esto depende de los ejemplos de prueba utilizados
- *Mientras más pequeña es la muestra, más grande es la varianza esperada*

Estimación de la Precisión de una Hipótesis

- Normalmente queremos estimar la precisión con la que nuestra hipótesis va a clasificar ejemplos nuevos y queremos saber el error probable de esta estimación
- Si suponemos una distribución de probabilidad de los ejemplos D (la cual no nos dice nada acerca de su clase), la tarea de aprendizaje es aprender un concepto o función meta f, tal que si recibe una instancia x tomada con distribución D nos regrese su clasificación

Estimación de la Precisión de una Hipótesis

Lo que queremos saber es:

1. Dada una hipótesis h y n ejemplos seleccionados aleatoriamente con distribución D, ¿cuál es el mejor estimador de la precisión de h sobre instancias futuras tomadas de esa distribución?
2. ¿Cuál es el error probable de esa estimación de precisión?

Error de Muestra y Error Verdadero

Antes tenemos que distinguir entre dos nociones:

- El *error de muestra*
- El *error verdadero*

Error de Muestra y Error Verdadero

- El *error de muestra* ($error_M(h)$) de una hipótesis h con respecto a la función meta f y la muestra de datos M, dada por la siguiente probabilidad:

$$error_M(h) = \frac{1}{n} \sum_{x \in M} \delta(f(x), h(x))$$

donde n es el número de ejemplos en M y $\delta(f(x), h(x)) = 1$ si $f(x) \neq h(x)$ o 0 de otra forma

Error de Muestra y Error Verdadero

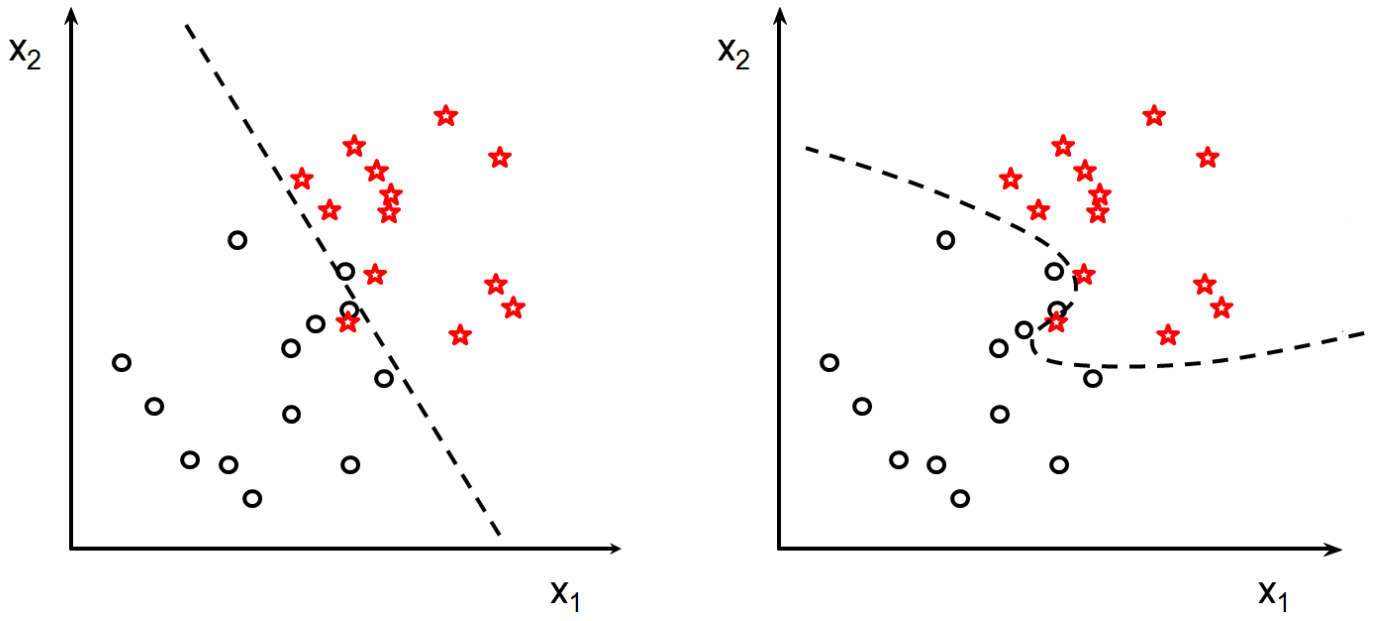
- El *error verdadero* ($error_D(h)$) de una hipótesis h con respecto a la función meta f y distribución D, dado por la probabilidad de que h clasifique mal una instancia seleccionado aleatoriamente de acuerdo a D:

$$error_D(h) = Pr_{x \in D}[f(x) \neq h(x)]$$

donde $Pr_{x \in D}$ denota que la probabilidad se hace sobre la distribución de las instancias D.

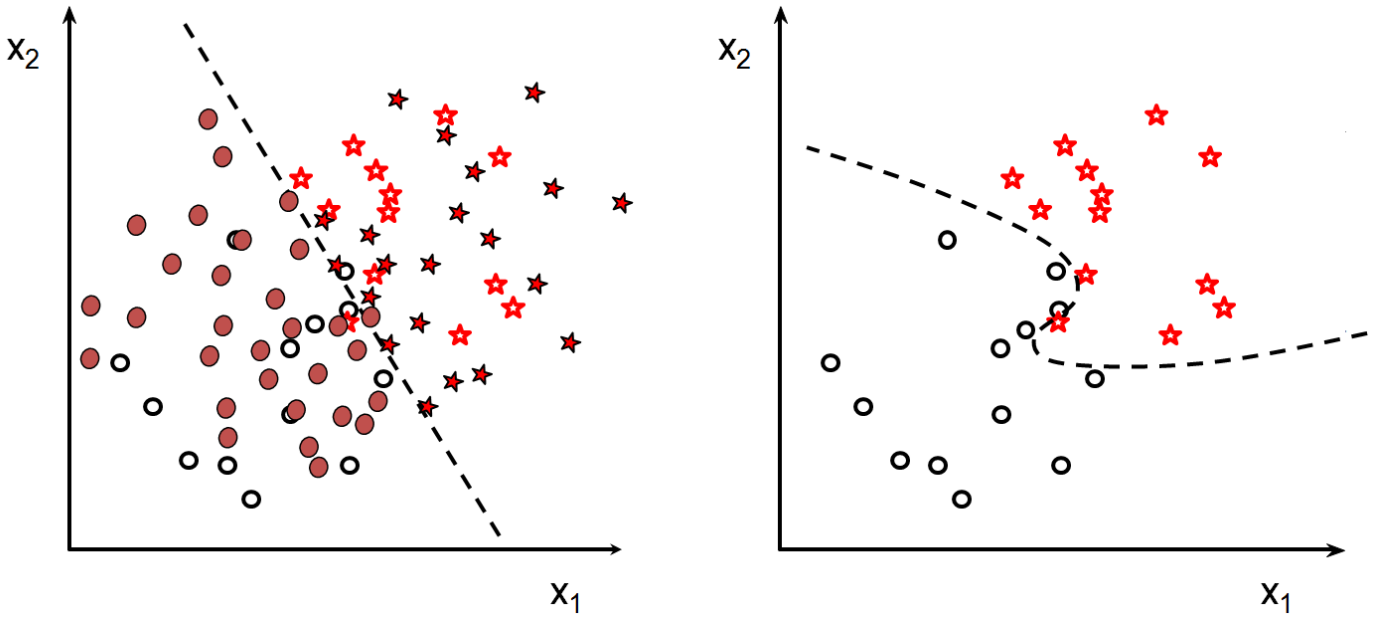
Error de Muestra y Error Verdadero

El error de entrenamiento no es un buen estimado del error de prueba:



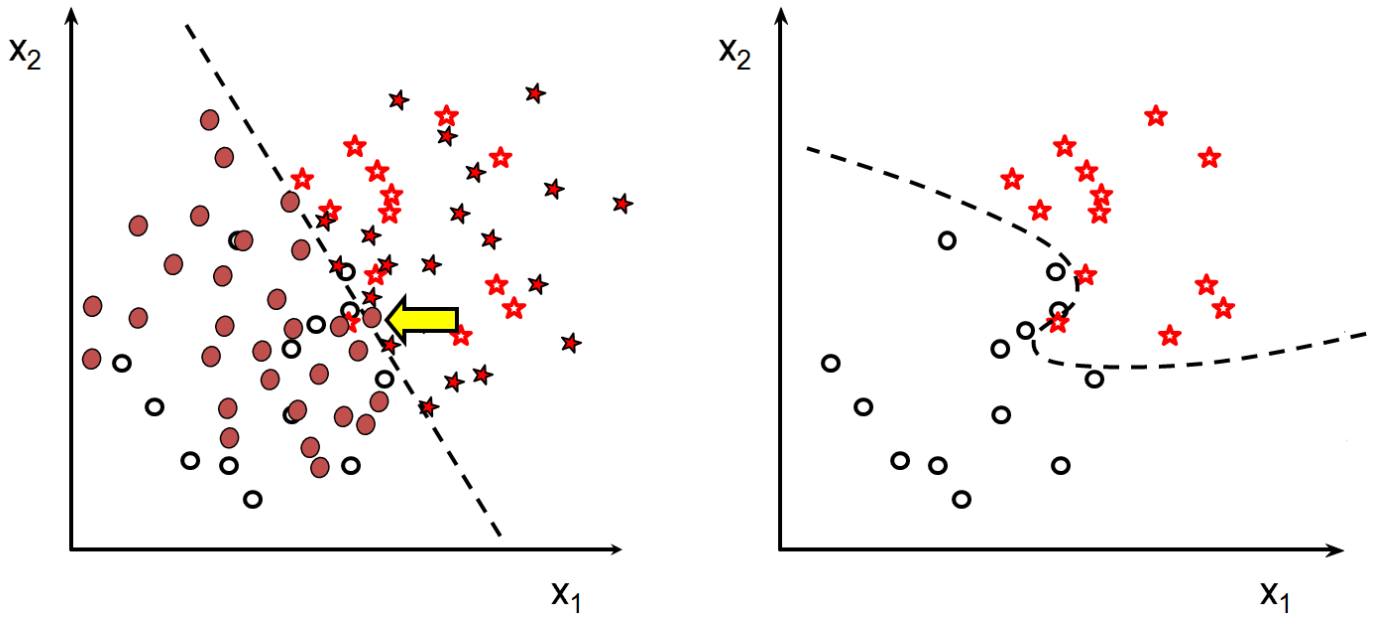
Error de Muestra y Error Verdadero

El error de entrenamiento no es un buen estimado del error de prueba:



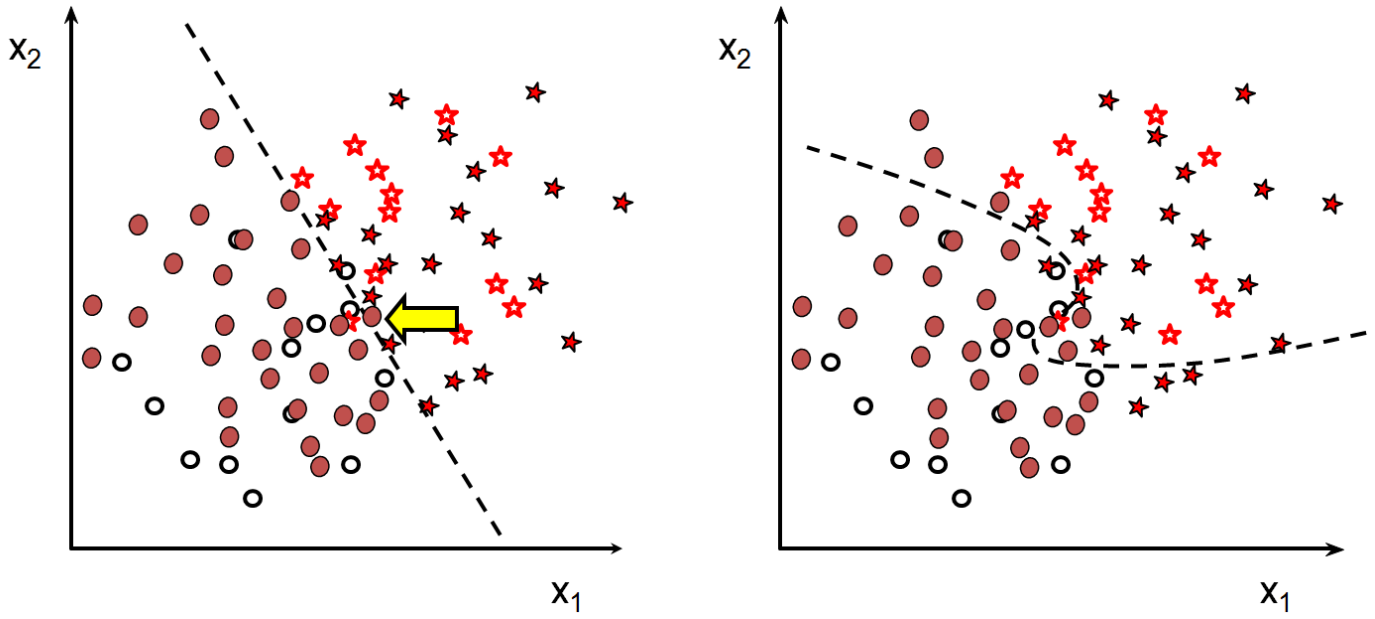
Error de Muestra y Error Verdadero

El error de entrenamiento no es un buen estimado del error de prueba:



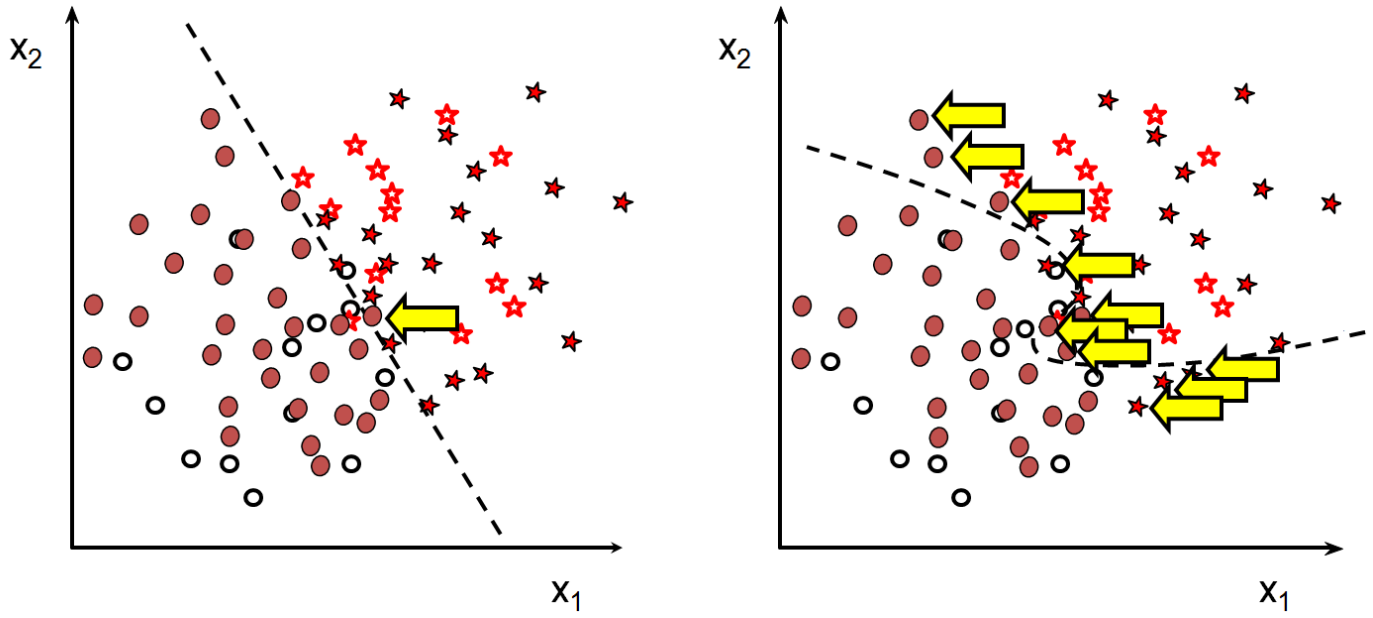
Error de Muestra y Error Verdadero

El error de entrenamiento no es un buen estimado del error de prueba:



Error de Muestra y Error Verdadero

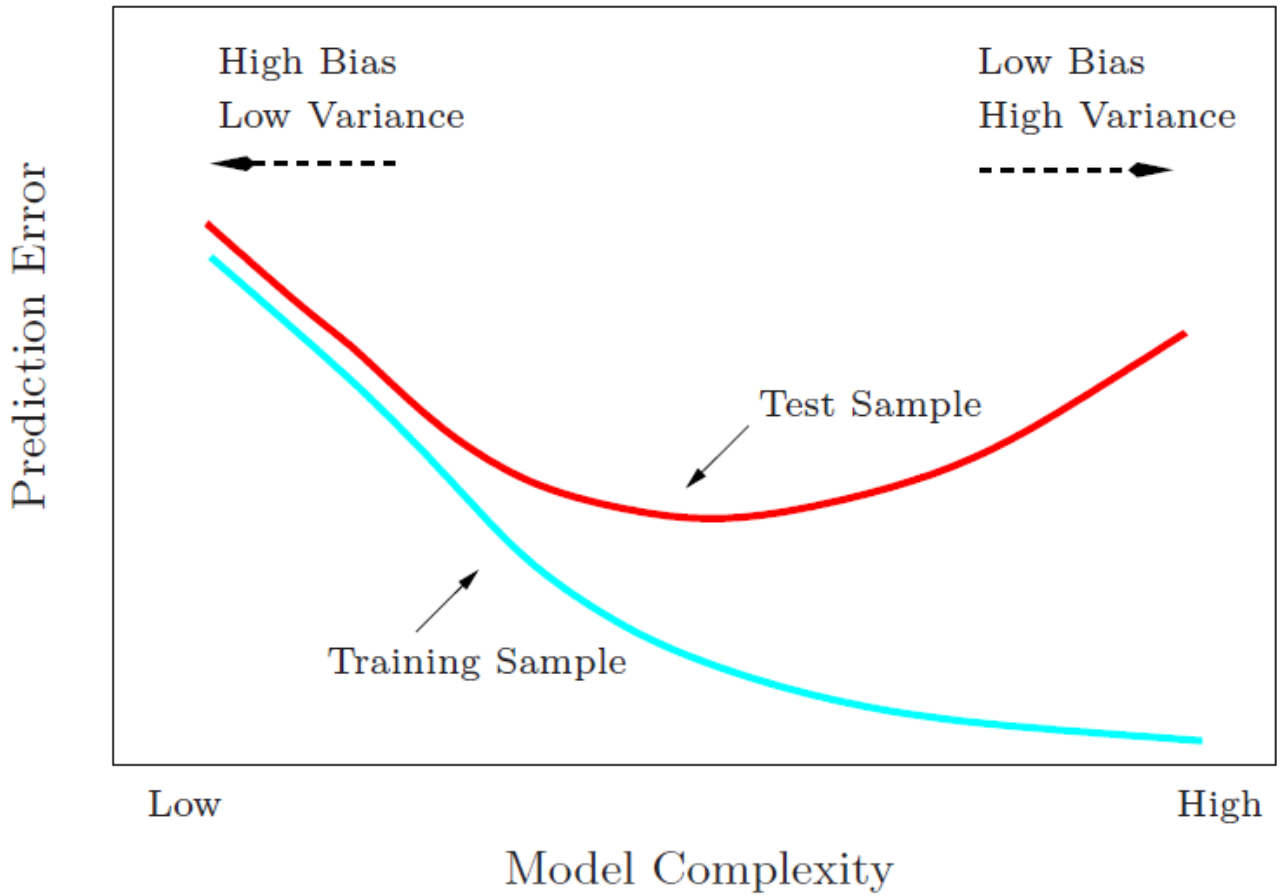
El error de entrenamiento no es un buen estimado del error de prueba:



Dilema Sesgo - Varianza

- Sesgo: Qué tanto se desvía el promedio del modelo de la media verdadera
- Varianza: Qué tanto varían las predicciones para distintos conjuntos de datos

Dilema Sesgo - Varianza



Error de Muestra y Error Verdadero

- Dado que se trabaja con una muestra M de datos, la pregunta es entonces, ¿qué tan buen estimador es $error_M(h)$ de $error_D(h)$?
- Esto depende de los datos
- La clave para responder esto es tomar en cuenta que cuando medimos el error de muestra estamos realizando un experimento con un resultado aleatorio
- Si repetimos este proceso muchas veces, cada vez con una muestra aleatoria M_i de tamaño n, esperariamos observar diferentes valores de $error_{M_i(h)}$, dependiendo de las diferencias aleatorias en las muestras

Distribución Binomial

- Si realizamos k experimentos y hacemos un histograma de los resultados, este se acerca a la forma de una distribución Binomial

$$P(r) = \frac{n!}{r!(n-r)!} p^r (1-p)^{n-r} \tag{1}$$

- La distribución Binomial describe, para cada posible valor de r (de 0 a n), la probabilidad de observar exactamente r resultados "positivos" (en este caso estamos interesados en los errores) dada una muestra aleatoria de n eventos independientes cuya probabilidad real de ocurrencia es p

Distribución Binomial

- Estimar p es equivalente a estimar $error_D(h)$.
- El número r corresponde al número de malas clasificaciones o errores observados sobre n eventos aleatorios y r/n corresponde al $error_M(h)$

Distribución Binomial

El esquema donde aplica la distribución Binomial es:

- Un experimento cuyo resultado puede ser descrito por una variable aleatoria con dos posibles valores (e.g., $Y =$error o $Y =$no error)
- La probabilidad de $Y =$ error, está dado por p y es independiente de cualquier otro experimento
- Dada una secuencia de experimentos independientes, la probabilidad de que la variable aleatoria tome r resultados $Y =$error está dado por la ecuación 1

Distribución Binomial

- El valor esperado de una variable aleatoria ($E(Y) = \sum_{i=1}^{n} y_i Pr(Y = y_i)$) gobernada por una distribución Binomial es:

$$E[Y] = np$$

su varianza ($Var[Y] = E[(Y - E[Y])^2]$) y desviación estandar ($\sigma_Y = \sqrt{Var(Y)}$) están dados por:

$$Var[Y] = np(1-p)$$

$$\sigma_Y = \sqrt{np(1-p)}$$

- Tenemos entonces:

$$error_M(h) = \frac{r}{n}$$

$$error_D(h) = p$$

donde: $n =$ número de ejemplos en M, $r =$ número de ejemplos mal clasificadas por h y $p =$ probabilidad de una mala clasificación de un ejemplo obtenido de D

Distribución Binomial

- El $error_M(h)$ es un *estimador* de $error_D(h)$. El sesgo de la estimación está dado por: $E[Y] - p$
- Si el sesgo es cero se dice que es un estimador sin sesgo (i.e., si en promedio nos da la probabilidad real)
- Como para la distribución Binomial el valor esperado de r es np, y como n es una constante, entonces, el valor esperado de r/n es p
- Para dar una estimación no sesgada la hipótesis y la muestra M deben de escogerse independientemente
- Otro parámetro importante dentro del estimador es su varianza. La varianza de la distribución Binomial es: $np(1-p)$
- En general, no conocemos p pero la podemos estimar con r/n

Ejemplo

- Tenemos 10 errores (r) en una muestra de 40 ejemplos (n)
- El estimado del error es $r/n = 0.25$
- La varianza de esta estimación está dada por la varianza r, ya que n es constante
- Como r sigue una distribución binomial, su varianza está dada por $np(1-p)$
- No conocemos p pero la podemos estimar con r/n, lo cual nos da una estimación de la varianza de r dada por: $40 \cdot 0.25(1-0.25) = 7.5$ lo cual nos da una desviación estandar de: $\sqrt{7.5} = 2.738$
- Esto quiere decir que la desviación estandar del error, o sea de r/n es $2.738/40 = 0.068$.

Desviación Estándar

- Dados r errores en una muestra de n pruebas independientes, la desviación estandar está dada por:

$$\sigma_{error_M(h)} = \frac{\sigma_r}{n} = \sqrt{\frac{p(1-p)}{n}}$$

- Que se puede aproximar substituyendo a p por $r/n = error_M(h)$:

$$\sigma_{error_M(h)} \approx \sqrt{\frac{error_M(h)(1-error_M(h))}{n}}$$

Intervalos de Confianza

- La forma común de describir la incertidumbre asociada a un estimado es darle un intervalo en donde se cree que puede caer el valor real junto con la probabildad de que caiga en ese intervalo
- Un $N\%$ intervalo de confianza para algún parámetro p es un intervalo que se espera que contenga a p con probabilidad $N\%$
- Para encontrar un intervalo de confianza necesitamos encontrar un intervalo centrado alrededor de la media del $error_M(h)$ que sea lo suficientemente ancho como para contener $N\%$ del total de la probabilidad bajo esta distribución
- O sea un intervalo alrededor de $error_M(h)$ en el cual el $error_D(h)$ cae el $N\%$ del tiempo

Intervalos de Confianza

- Para esto, podemos usar el hecho de que para valores grandes de muestras, la distribución Binomial se acerca a una Gaussiana
- Los intervalos de confianza para Gaussianas los podemos obtener de tablas estadísticas
- Con media μ y desviación estandar σ se caera en el intervalo un *N*% del tiempo expresado como: $\mu \pm z_N \sigma$, donde z_N (obtenida de tablas, e.g., ver 1) nos da la distancia alrededor de μ en desviaciones estandar
- Por lo que:

$$error_M(h) \pm z_N \sqrt{\frac{error_M(h)(1 - error_M(h))}{n}}$$

Intervalos de Confianza

- En el ejemplo anterior, si buscamos un intervalo de confianza del 95%, $z_N = 1.96$, por lo que tendriamos un error del $0.25 \pm 1.96\sqrt{\frac{0.25(1-0.25)}{40}} = 0.25 \pm 0.068$
- Esto se basa en dos aproximaciones:
  - Aproximamos $error_D(h)$ con $error_M(h)$
  - Aproximamos la distribución Binomial con una Gaussiana
- Y en las siguientes suposiciones:
  - Valores de hipótesis discretas
  - La muestra M se selecciona aleatoriamente usando la misma distribución de probabilidad con la que se obtendran nuevos datos
  - Los datos son independientes de la hipótesis que se está probando

Intervalos de Confianza

- Una regla de dedo común en estadística es que las dos aproximaciones de arriba son bastante buenas mientras $n \geq 30$ o cuando $np(1-p) \geq 5$
- Para valores menores es recomendable usar una tabla de distribuciones Binomiales

Tabla: Tabla de valores de z_N.

Nivel de Confianza (N%)	50%	68%	80%	90%	95%	98%	99%
Constante z_N	0.67	1.00	1.28	1.64	1.96	2.33	2.58

Inntervalos de Confianza

- Los valores de confianza dados son para los dos lados, a veces queremos saber sólo el de un lado (probabilidad del error a lo más X)
- Como la distribución Normal es simétrica alrededor de su media, un intervalo de $100(1-\alpha)$% de los dos lados es igual a $100(1-\alpha/2)$% de un solo lado

Inntervalos de Confianza

En general, el procedimiento para derivar intervalos de confianza es:

- Identificar el parámetro a estimar (e.g., $error_D(h)$)
- Definir un estimador no sesgado (e.g., $error_M(h)$)
- Determinar la distribución de probabilidad que gobierna al estimador incluyendo su media y varianza
- Determinar los intervalos de confianza

Teorema del Límite Central

- Considere un conjunto de variables independientes aleatoriamente distribuidas $Y_1, \ldots, Y_n$, gobernadas por una media μ y una varianza σ^2
- Cuando $n \to \infty$, la distribución que gobierna a la media se acerca a una distribución Normal, independientemente de las distribuciones que gobiernan a las Y_is
- Esto es importante porque cada vez que definamos un estimador que sea la media de una muestra, la distribución que gobierna el estimador se puede aproximar a una distribución Normal

Pruebas de Significancia Estadística

- Las métricas que vimos nos dan evaluaciones sobre los clasificadores
- Lo que queremos saber es si los resultados observados se pueden atribuir a las características de los clasificadores o se pueden observar al azar
- El propósito de las pruebas de significancia estadística es para ayudar a obtener evidencia de que los resultados de las métricas son representativos del comportamiento general de los clasificadores

Prueba de Hipótesis

- Establecer la hipótesis nula que generalmente es lo opuesto a lo que queremos probar (e.g., los clasificadores A y B tienen el mismo desempeño)
- Después seleccionamos una prueba estadística adecuada y los estadísticos a usar para negar la hipótesis
- Normalmente también seleccionados una región crítica donde debe de caer la estadística (e.g., intervalos de confianza)
- Calculamos la estadística, vemos si está en la región crítica, y si es así, rechazamos la hipótesis nula (si no no quiere decir que la aceptamos)
- El rechazar la hipótesis nula nos da confianza en creer que las observaciones no fueron por causalidad

Prueba de Hipótesis

- La prueba de hipótesis no es una prueba de que nuestras observaciones sean válidas
- Las pruebas estadísticas pueden ser paramétricas (normalmnte hacen suposiciones fuertes sobre los datos) y no paramétricas (no hacen supiosiciones pero sus resultados son menos poderosos)
- No importa cuan pequeña se la diferencia, siempre se puede encontrar una diferencia significativa con suficientes datos

Prueba de Hipótesis

- Algunos investigadores inclusive han cuestionado la utilidad de las pruebas estadísticas que pueden: (i) sobrevalorar los resultados y (ii) limitar la búsqueda de nuevas ideas por el exceso de confianza en los resultados
- También es importante conocer más de las pruebas y sus limitaciones

Comparación entre algoritmos

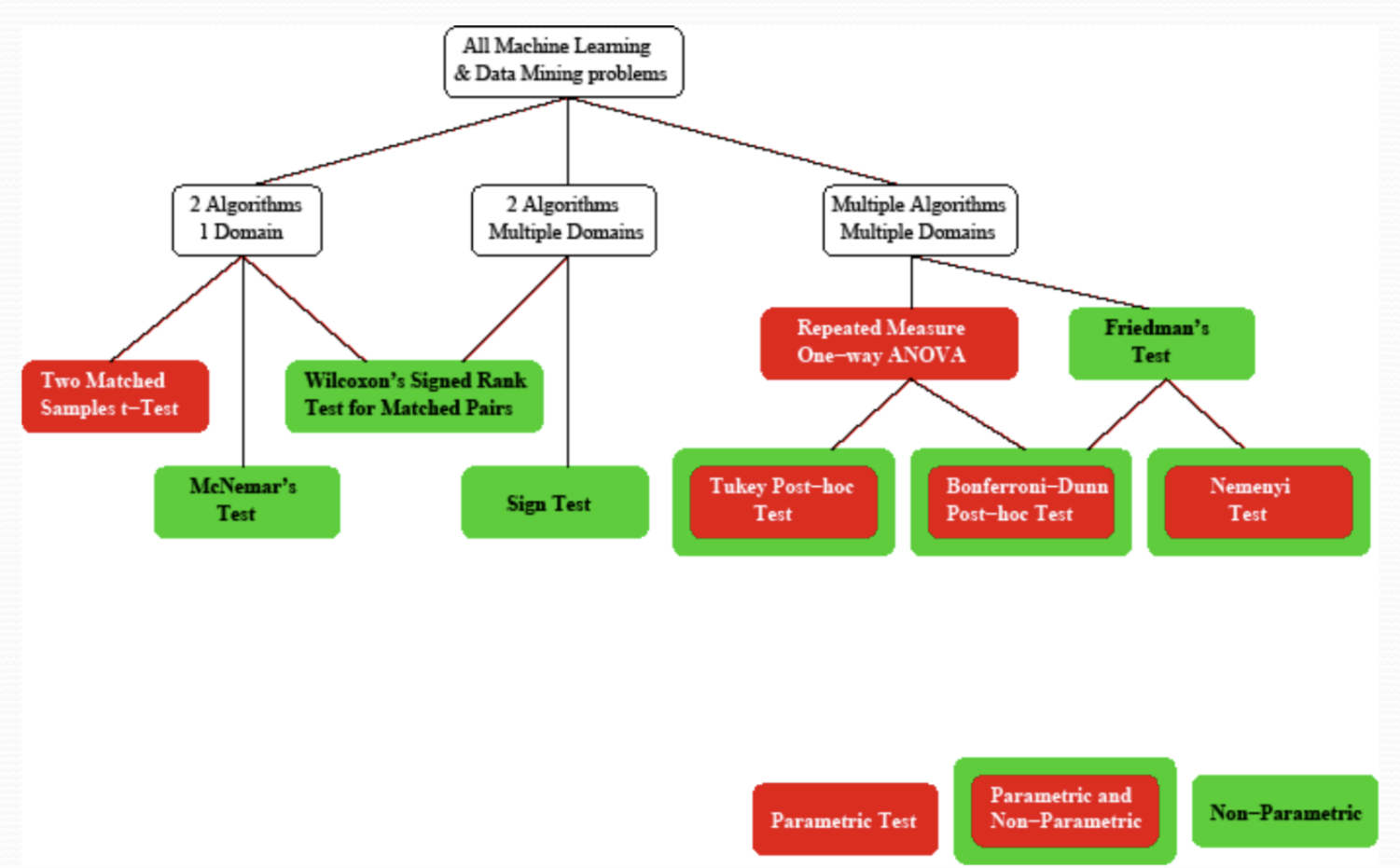
La comparación entre dos algoritmos de aprendizaje ha sido controversial dentro del área por todos los factores que pueden influir en esta comparación:

- Conjuntos de entrenamiento y prueba
- Algoritmos no determinísticos
- Variaciones por parte del usuario
- Calidad y cantidad de los datos
- Conocimiento previo

Prueba de Hipótesis

- Para aplicar una prueba estadística hay que considerar el problema que se quiere atacar y qué tanta información tenemos de la distribución de los datos
- Normalmente se aplican para comparar:
  - Dos algoritmos en un solo dominio
  - Dos algoritmos en muchos dominios
  - Múltiples algoritmos en múltiples dominios

Pruebas Estadísticas



Dos Algoritmos Un Dominio

- La t-test (paramétrica)
- La prueba McNewar (no paramétrica)
- The Sign Test (normalmente usada para muchos dominios, pero puede usarse para uno solo)

Diferencia de Error entre dos Hipótesis

- Si tenemos dos hipótesis h_1 y h_2 obtenidas de muestras M_1 y M_2 con n_1 y n_2 datos cada una, supongamos que queramos evaluar su diferencia entre los errores verdaderos

$$d = error_D(h_1) - error_D(h_2)$$

- Definimos un estimador:

$$\hat{d} = error_{M_1}(h_1) - error_{M_2}(h_2)$$

- Si distribución que gobierna a cada $error_{M_i}(h_i)$ es Normal, entonces $\hat{d}$ es Normal porque la diferencia entre dos distribuciones Normales es Normal

Diferencia de Error entre dos Hipótesis

- La media es d y la varianza de dos distribuciones Normales es su suma, por lo que:

$$\sigma_d^2 \approx \frac{error_{M_1}(h_1)(1 - error_{M_1}(h_1))}{n_1} + \frac{error_{M_2}(h_2)(1 - error_{M_2}(h_2))}{n_2}$$

- Por lo que los itervalos de confianza serían:

$$\hat{\delta} \pm z_N \sqrt{\frac{error_{M_1}(h_1)(1 - error_{M_1}(h_1))}{n_1} + \frac{error_{M_2}(h_2)(1 - error_{M_2}(h_2))}{n_2}}$$

- Si se prueban las hipótesis sobre los mismos datos la varianza tiende a ser menor

Comparación entre algoritmos

- Lo que queremos estimar es cuál algoritmo es en promedio mejor que el otro
- O sea cuál es el valor esperado de la diferencia de errores:

$$E_{M \subset D}[error_D(SA_1(M)) - error_D(SA_2(M))]$$

donde SA_i es el i-ésimo sistema de aprendizaje

- Como tenemos sólo una muestra limitada de datos (D_0) una forma obvia de comparar dos sistemas es dividir los datos en datos de entrenamiento M_0 y datos de prueba T_0 y comparar sus resultados en los de prueba:

$$error_{T_0}(SA_1(M_0)) - error_{T_0}(SA_2(M_0))$$

- Aqui usamos $error_{T_0}(h)$ para estimar $error_D(h)$ y sólo medimos la diferencia en un solo conjunto de entrenamiento

Comparación entre algoritmos

- Para obtener una mejor medida, podemos particionar repetidamente el conjunto D_0 en conjuntos disjuntos de entrenamiento y prueba y calcular la media de los errores del conjunto de prueba para todos los experimentos
- Esto es lo que conocemos como *k-fold cross validation* que usamos cuando tenemos al menos 30 ejemplos de entrenamiento

$$\overline{\delta} = \frac{1}{k} \sum_{i=1}^{k} \delta_i$$

donde $\delta_i = error_{T_i}(h_1) - error_{T_i}(h_2)$

Comparación entre algoritmos

- La media ($\overline{\delta}$) se puede tomar como un estimador de:

$$E_{M \subset D_0}[error_D(SA_1(M)) - error_D(SA_2(M))]$$

donde M representa una muestra aleatoria de tamaño $\frac{k-1}{k}|D_0|$

- Los intervalos de confianza se pueden estimar como:

$$\overline{\delta} \pm t_{N,k-1} s_{\overline{\delta}}$$

donde $t_{N,k-1}$ es parecido a z_N, y $s_{\overline{\delta}}$ es un estimador de la desviación estandar de la distribución que gobierna a $\overline{\delta}$ dado por:

$$s_{\overline{\delta}} = \sqrt{\frac{1}{k(k-1)} \sum_{i=1}^{k} (\delta_i - \overline{\delta})^2}$$

Comparación entre algoritmos

- El primer subíndice de $t_{N,k-1}$ no da el nivel de confianza, el segundo es el número de *grados de libertad* (número de eventos aleatorios independientes)
- Cuando $k \to \infty$, $t_{N,k-1} \to z_N$
- Cuando las pruebas se hacen sobre muestras iguales, se llaman: *paired tests* o *paired t tests* (en este caso, por la distribución *t* que gobierna las pruebas).
- Note que las muestras con que probamos a los 2 algoritmos son idénticas, a estas pruebas se les llama *apareadas*
- Puebas apareadas producen intervalos de confianza más ajustados porque las diferencias en errores se deben a los algoritmos y no a las diferencias de las muestras que se dan cuando no usamos muestras idénticas para los algoritmos

Comparación entre algoritmos

- Esta es la prueba t apareada, en la tabla 2 se muestran los valores para $t_{N,v}$

	Confidence level			
	90%	95%	98%	99%
v=2	2.92	4.30	6.96	9.92
v=5	2.02	2.57	3.36	4.03
v=10	1.81	2.23	2.76	3.17
v=20	1.72	2.09	2.53	2.84
v=30	1.70	2.04	2.46	2.75
v=120	1.66	1.98	2.36	2.62
$v = \infty$	1.64	1.96	2.33	2.58

Tabla: Valores de $t_{N,v}$

Proceso de Evaluación de Significancia con t-test

1. Obtener el valor t, como la razón $\frac{dif\_entre\_medias}{variabilidad\_en\_experimentos} = \frac{\hat{\delta}}{s_{\hat{\delta}}}$
2. Calcular los grados de libertad ($DF = N - 1$)
3. Elegir el nivel de α (o nivel de riesgo), generalmente $\alpha = 0.05$ (5 veces de 100 se encuentra una diferencia significativa entre las medias aún cuando no la hay, i.e., la diferencia resultante fue producto de la suerte)
4. Verificar en la tabla el valor crítico de t. Si el valor observado es mayor que el valor crítico, entonces se rechaza la hipótesis nula. Si es menor no se puede rechazar la hipótesis nula
5. Si la tabla no tiene el número de grados de libertad, se usa el siguiente número menor al real (para 32 usar 30)

Ejemplo

- Suponga que se realizó una prueba de 10-FCV con dos clasificadores y queremos saber si la diferencia entre sus promedios es significativa

Prueba	ALG-1	ALG-2
1	88	85
2	85	80
3	93	87
4	87	82
5	89	85
6	85	82
7	87	83
8	84	79
9	86	80
10	88	86

Tabla: Valores de Precisión para los Algoritmos ALG-1 y ALG-2

Ejemplo

Para estos datos tenemos los siguientes cálculos:

- La diferencia de error medio, $\hat{\delta} = 4.30$.
- La variabilidad entre experimentos, $S_{\hat{\delta}} = 0.42$
- El valor de t calculado es, $t = 10.17$
- El valor crítico encontrado en la tabla para 95% de confianza con 9 grados de libertad es aprox. de $t_{N,v} = 2.3$
- Como el valor observado es mayor que el criítico, entonces se rechaza la hipótesis nula
- La hipótesis nula dice que no hay diferencia entre las medias
- Por tanto, al rechazar la hipótesis nula, se concluye que sí hay una diferencia significativa entre las medias

La prueba t pareada

- La prueba *t* aplica cuando se quiere estimar la media de un conjunto de variables aleatorias independientes cada una de ellas con distribución Normal
- Dadas dos muestras (i.e., resultados de dos algoritmos aplicados a los mismos datos con las mismas particiones) queremos saber si existe una diferencia significativa
- Suponemos que la diferencia es cero (hipótesis nula) y queremos ver si podemos rechazar esta hipótesis
- Nos fijamos en las medias y desviaciones estándares

Suposiciones de la prueba t

- La prueba t supone que las distribuciones son Gaussianas o que se tienen al menos 30 datos
- Las muestras deben de ser representativas de la población
- Las dos muestras tienen la misma varianza (esto no siempre se cumple, e.g., la varianza que muestra un algoritmo de árboles decisión no necesariamente es igual a la varianza de un naïve Bayes)
- En este caso es mejor usar la prueba de McNemar

Prueba de McNemar

- Es una prueba no paramétrica que se usa como alternativa para la prueba t
- La prueba McNemar χ^2 está definida como:

$$\chi^2_{MC} = \frac{(|c_{01} - c_{10}| - 1)^2}{c_{01} + c_{10}}$$

donde:

  - c_{01}: El número de ejemplos mal clasificados por f_1, pero correctamente clasificados por f_2
  - c_{01}: El número de ejemplos mal clasificados por f_2, pero correctamente clasificados por f_1

- Si $c_{01} + c_{10} \geq 20$ entonces la χ^2_{MC} se parece a la prueba χ^2. Si χ^2_{MC} supera a la prueba $\chi^2_{1,1-\alpha}$ se rechaza la hipótesis nula
- Si $c_{01} + c_{10} < 20$ no se puede usar y se debe de usar la *sign test*

Dos clasificadores en múltiples dominios

- No hay una prueba paramétrica clara ya que es difícil establecer la suposición de distribución normal y la prueba t es susceptible de outliers, lo cual puede pasar con pruebas en varios dominios
- Las alternativas no paramétricas son:
  - The Sign Test
  - Wilcoxon's signed-Rank test

The Sign Test

- Se puede usar para comparar dos clasificadores en un solo domunio (usando cada resultado de cada pliegue como muestra) o para comparar dos clasificadores en múltiples dominios
- Contamos cuántas veces el clasificador *A* le gana al clasificador *B*
- La hipótesis nula (los dos clasificadores se desempeñan igual) es válida si el número en que se gana sigue una distribución binomial
- Un clasificador es mejor que otro is gana en por lo menos w_α bases de datos, donde w_α es el valor crítica para la sign test con nivel de significancia α
- Como se supone una distribución binomial se busca en las tablas

The Sign Test: Ejemplo

Dataset	NB	SVM	Adaboost	Rand Forest
Anneal	96.43	99.44	83.63	99.55
Audiology	73.42	81.34	46.46	79.15
Balance Scale	72.30	91.51	72.31	80.97
Breast Cancer	71.70	66.16	70.28	69.99
Contact Lenses	71.67	71.67	71.67	71.67
Pima Diabetes	74.36	77.08	74.35	74.88
Glass	70.63	62.21	44.91	79.87
Hepatitis	83.21	80.63	82.54	84.58
Hypothyroid	98.22	93.58	93.21	99.39
Tic-Tac-Toe	69.62	99.90	72.54	93.94

- NB vs. SVM: $n_{NB} = 4.5, n_{SVM} = 5.5, w_{0.05} = 8$ por lo que no se puede rechazar la hipótesis nula con $\alpha = 0.05$ (1 cola)
- Adaboost vs. Random Forest: $n_{Ada} = 1, n_{RF} = 8.5$ en este caso, se rechaza la hipótesis nula (RF $>_s$ Ada)

Wilkoxon's Signed-Ranked Test

- Dos clasificadores, múltiples bases de datos, no paramétrica y más poderosa que la sign test
- Calcula la diferencia en desempeño de cada clasificador
- Ordena las diferencias absolutas y asignales signos (positivo/negativo)
- Calcula la suma de positivos y negativos (w_{s1}, w_{s2})
- $T_{Wilcox} = min(w_{s1}, w_{w2})$
- Compara el valor critico V_α. Si $V_\alpha \geq T_{Wilcox}$ rechaza la hipótesis nula con nivel de confianza α

Wilkoxon's Signed-Ranked Test

Data	NB	SVM	NB-SVM	|NB-SVM|	Ranks	± Ranks
1	.9643	.9944	-0.0301	0.0301	3	-3
2	.7342	.8134	-0.0792	0.0792	6	-6
3	.7230	.9151	-0.1921	0.1921	8	-8
4	.7170	.6616	+0.0554	0.0554	5	+5
5	.7167	.7167	0	0	Remove	Remove
6	.7436	.7708	-0.0272	0.0272	2	-2
7	.7063	.6221	+0.0842	0.0842	7	+7
8	.8321	.8063	+0.0258	0.0258	1	+1
9	.9822	.9358	+0.0464	0.0464	4	+4
10	.6962	.9990	-0.3028	0.3028	9	-9

- $w_{s1} = 17, w_{s2} = 28 \Rightarrow T_{Wilcox} = min(17, 28) = 17$
- Para $n = 10 - 1$ grados de libertad y $\alpha = 0.005$, $V = 8$ (1 lado), por lo que no rechazamos la hipótesis

Friedman's Test

- Mútiples clasificadores, múltiples bases de datos, no paramétrica
- Todos los algoritmos se rankean en cada dominio por separado (empates se cuentan como los números intermedios del ranking)
- Se suman los lugares para cada clasificador j (R_j)
- La prueba de Friedman es:

$$\chi_F^2 = \left(\frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2 \right) - 3n(k+1)$$

donde n es el número de dominios y k el de clasificadores

Friedman Test: Ejemplo

Domain	Classifier fA	Classifier fB	Classifier fC
1	85.83	75.86	84.19
2	85.91	73.18	85.90
3	86.12	69.08	83.83
4	85.82	74.05	85.11
5	86.28	74.71	86.38
6	86.42	65.90	81.20
7	85.91	76.25	86.38
8	86.10	75.10	86.75
9	85.95	70.50	88.03
19	86.12	73.95	87.18

Domain	Classifier fA	Classifier fB	Classifier fC
1	1	3	2
2	1.5	3	1.5
3	1	3	2
4	1	3	2
5	2	3	1
6	1	3	2
7	2	3	1
8	2	3	1
9	2	3	1
10	2	3	1
$R_{.j}$	15.5	30	14.5

- $\chi_F^2 = \left(\frac{12}{10 \times 3 \times (3+1)} \sum_{j=1}^{3} R_j^2 \right) - 310(3+1) = 15 - 05$
- Para $\alpha = 0.05$ y una prueba de 2-colas, el valor crítico es 7.8. Como $\chi_F^2 > 7.8$ se rechaza la hipótesis nula

La prueba Nemenyi

- Se usa para decir en dónde está la diferencia
- Si R_{ij} es el lugar del clasificador i en la base de datos j, se calcula el lugar promedio de cada clasificador en todas las bases de datos:

$$\overline{R_j} = \frac{1}{n} \sum_{i=1}^{n} R_{ij}$$

- La estadísitca entre el clasificador A y B es:

$$q_{AB} = \frac{\overline{R_A} - \overline{R_B}}{\sqrt{\frac{k(k+1)}{6n}}}$$

donde n es el número de dominios y k el de clasificadores

La prueba Nemenyi: Ejemplo

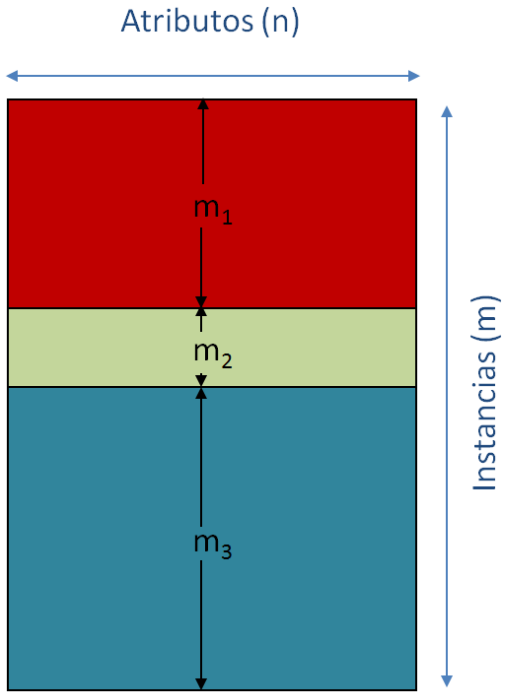
- Del ejemplo anterior tenemos que: $\overline{R_A} = 15.5, \overline{R_B} = 30$ y $\overline{R_C} = 14.5$
- Reemplazando en:

$$q_{XY} = \frac{\overline{R_X} - \overline{R_Y}}{\sqrt{\frac{k(k+1)}{6n}}}$$

- $q_{AB} = -32.22, q_{AC} = 2.22$ y $q_{BC} = 34.44$
- $q_\alpha = 2.55$ con $\alpha = 0.05$, por lo que se rechaza la hipótesis nula para A-B y B-C, pero no para A-C

Muestreo

- Para probar un algoritmo de aprendizaje, normalmente se dividen los datos, en datos de entrenamiento y en datos de prueba
- También se pueden dividir en 3 subconjuntos:
  - **Entrenamiento.** Construcción del clasificador
  - **Validación.** Optimización de parámetros (a veces)
  - **Prueba** Evaluación del clasificador



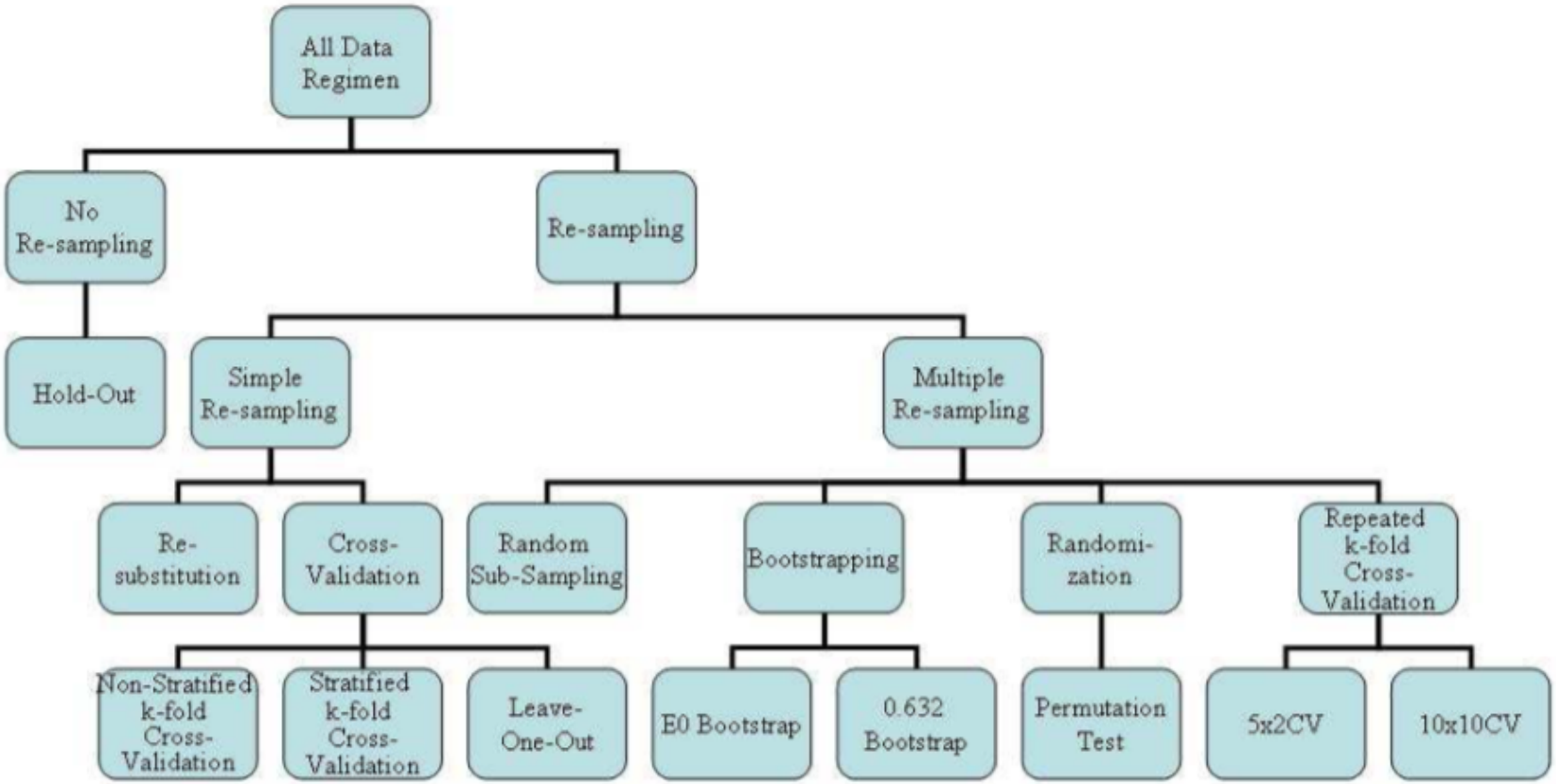
Muestreo

- Lo más común para evitar selecciones de datos sesgadas es repetir el proceso varias veces con diferentes muestras aleatorias
- Esto se podría evitar si tenemos acceso a todos los datos o a una muestra suficientemente grande
- Como normalmente no es el caso, se re-utilizan para estimar de forma más confiable los errores en los clasificadores
- El muestreo o remuestreo se divide en dos:
  - Simple: Cada dato se usa para prueba una sola vez
  - Múltiple: Se puede usar un dato más de una vez

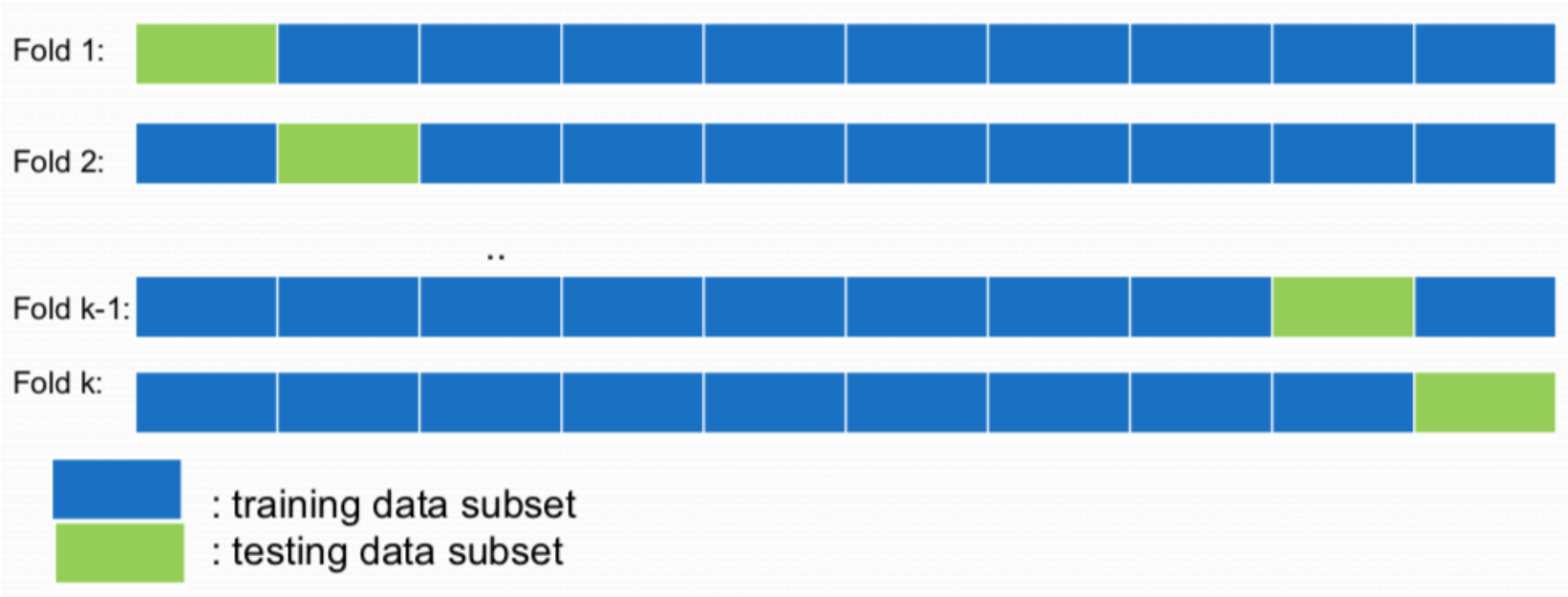
Muestreo: Algunos peligros

- El muestreo se hace seguido de una prueba estadística, pero las pruebas suponen que los datos son independientes
- Esto no se cumple si reutilizamos los mismos datos, por lo que las pruebas estadísticas dejan de ser válidas

Métodos de Muestreo



k-Fold Cross Validation



- La validación cruzada de k pliegues es la más utilizada
- Los datos se dividen aleatoriamente en k subconjuntos, se entrena con $k-1$ subconjuntos y se prueba con el k restante, y se repite para todas las k

k-FCV

- Se deberían de hacer particiones disjuntas del mismo tamaño (de por lo menos 30 datos), usar los datos de las $k-1$ particiones para entrenar y probar las hipótesis con cada partición k. Se realizan k pruebas y se toma la media de los resultados
- A este proceso se le conoce como validación cruzada (*cross validation*), y normalmente se hace para $k = 10$ (*10-fold cross validation*)

Variantes de k-FCV

- Al proceso de cuidar que las clases esten adecuadamente representadas tanto en el conjunto de prueba como en el entrenamiento se llama *estratificación*
- Validación cruzada estratificada: Se usa cuando las clases están desbalanceadas y se busca que esta distribución de clases se mantenga en los datos
- Leave-One-Out CV: En este caso $k = n$ (donde n es el número de datos) y se usa normalmente cuando se tienen pocos datos
- Sus principales desventajas es que no se puede hacer estratificación y que es computacionalmente caro

k-FCV

- Aunque las muestras de prueba son independientes, las de entrenamiento tienen traslapes lo que afecta el sesgo en las estimaciones de error
- LOOCV produce una alta varianza pero existe poco sesgo al entrenar prácticamente con todos los datos

k-FCV

- Para obtener mejores estimaciones muchas veces se repite el proceso varias veces
- En general se han propuesto: 5×2 CV y 10×10 CV
- Esto es repetir 2-fold CV 5 veces (o 10-fold CV 10 veces)
- Otros proponen sustituir la prueba t and final de 5×2 CV por la prueba F

Bootstrapping

- Supone que se tiene una muestra representativa y crea datos muestreando aleatoreamente *con reemplazo* de los datos disponibles
- Esto es lo que se usa en ensambles de clasificadores (i.e., *Bagging*)
- También se usa cuando se tienen pocos datos y existen dos opciones ϵ_0 y ϵ_{632}

Bootstrapping

- El problema es que las muestras se traslapan y por lo tanto dejan de ser independientes
- Este método también se le conoce como *boostrap*
- El muestraer n veces un conjunto de n datos con reemplazo y contruir un conjunto nuevo de tamaño n, hace que algunos elementos esten repetidos y otros no aparezcan
- La probabilidad de que no aparezca un dato es:

$$\left(1 - \frac{1}{n}\right)^n \approx e^{-1} = 0.368$$

- Por lo que el conjunto de entrenamiento tendrá en promedio 63.2% de los datos y el de prueba 36.8%

Bootstrapping

- Con datos D de tamaño m se crean k muestras aleatorias S_i de tamaño m con reemplazo (k es normlamente ≥ 200)
- Se entrena con las k y se prueba con ejemplos de D que no están en B_i
- $\epsilon_0 =$ promedio de las k pruebas (tiende a ser pesimista porque se entrena con 63.2% de los datos)

Otras evaluaciones de hipótesis

- Para compensar esto, se utiliza: $\epsilon_{632} = 0.632\epsilon_0 + 0.368 err(f)$, donde $err(f)$ es el error promedio en todos los datos (el cual tiende a ser optimista)
- El proceso se repite varias veces y se promedio el error. Este método se utiliza sobre todo cuando tenemos muestras pequeñas.
- Cuando los resultados que nos arrojan los algoritmos son probabilísticos (un vector de probabilidades de pertenecer a una clase), se utilizan otros esquemas como *quadratic loss function* o *information loss function*

Bootstrapping

- Bootstrap da mejores estimaciones que validación cruzada cuando se tienen pocos datos
- ϵ_0 es un buen estimador cuando el error real es alto
- ϵ_{632} es un buen estimador con pocos datos y con el error verdadero pequeño
- Bootstrapping no funciona con clasificadores que se ven afectados cuando hay ejemplos duplicados (e.g., k-NN, FOIL)

No Free Lunch Theorem

- El teorema de “no free lunch” de Wolpert dice que si un algoritmo es mejor que otro en una clase de problemas, lo contrario se va a cumplir en otra clase de problemas (i.e., no existe un clasificador mejor que todos en todas las bases de datos)
- Se ha mostrado que clasificadores sencillos muchas veces son superiores que los más sofisticados en ciertos dominios
- El mostrar superioridad de un algoritmo sobre otro(s) no quiere decir que sea mejor en todos los casos

Pruebas y Repositorios

- Existen una gran cantidad de datos que se pueden usar (repositorios reales, datos artificiales, tomados de Internet)
- Muchos sistemas contienen y realizan muchas de estas evaluaciones (e.g., Weka, R)

Comentarios finales

- En problemas reales, no es posible estimar exactamente el error de generalización
- El error de entrenamiento siempre será optimista
- Estrategias de validación nos dan idea de cómo será el error de generalización
- Intentos por mejorar el desempeño de un clasificador en datos de entrenamiento, mediante el incremento de la complejidad del modelo puede llevarnos a sobre-ajustar los datos: *el error de entrenamiento es engañoso!*