

Prácticas de Aprendizaje Automático

Práctica 3

Ajuste de Modelos Lineales

Pablo Mesejo y Manuel Cobo

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial



UNIVERSIDAD
DE GRANADA



Ajuste de Modelos Lineales

- Ajuste y selección del mejor modelo lineal, y estimación del error E_{out} del modelo final
- **Casuística próxima a la realidad:** te llega un problema y... ¿cómo lo resuelves?
 - **Análisis** del Problema, **Preprocesado** de los Datos, Formulación de **Hipótesis**, **Entrenamiento**, **Validación**, y **Discusión** de Resultados

Ajuste de Modelos Lineales



– Problema de clasificación

<https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>

45211 ejemplos, 16 atributos, y salida binaria (última columna del fichero).

– Problema de regresión

<https://archive.ics.uci.edu/ml/datasets/YearPredictionMSD>

515345 ejemplos, 90 atributos y salida a predecir (primera columna del fichero).

1. Analizar y comprender el problema (1 pto)

- ¿Qué es X?
- ¿Qué es Y?
- ¿En qué consiste el problema que tengo que resolver ($f: X \rightarrow Y$)?

¡Ojo! Hay una **delgada línea que separa la visualización/exploración de los datos del data snooping/data dredging**.

- a) Está bien visualizar los datos (de entrenamiento) si es para aprender sobre el problema. Está mal visualizarlos si es para guiar nuestra elección del modelo concreto a usar.
- b) Nuestro propósito no es tanto establecer una hipótesis como comprender mejor nuestro problema.

1. Analizar y comprender el problema (1 pto)

— Una de las claves principales es **no usar los datos de test para absolutamente nada**.

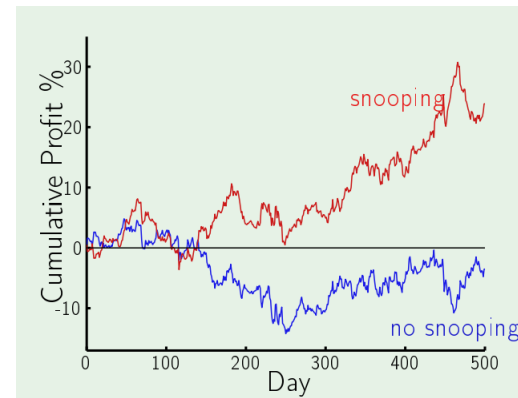
- Ejemplo 1:

- Alguien calcula la media y desviación típica para realizar la estandarización de los datos utilizando todos los ejemplos (incluyendo el test, que no deberíamos conocer de nada).

- » Primero habría que dividir los datos, luego escalar los datos de entrenamiento y, finalmente, escalar los datos de test con los factores de escalado empleados en el entrenamiento.

- Ejemplo 2:

- nuestros datos, al visualizarlos, tenemos la impresión de que son linealmente separables y, por tanto, decidimos usar PLA. Muy probablemente esa premisa sea demasiado fuerte, esté demasiado apegada a nuestros datos, y sea desaconsejable asumirla.



<https://home.work.caltech.edu/slides/slides17.pdf>

1. Analizar y comprender el problema (1 pto)

- Entonces... ¿qué clase de análisis se puede hacer?
 - Descripción de las variables empleadas (¿qué miden?)
 - Histogramas con el porcentaje de ejemplos de cada clase en entrenamiento
 - Para verificar la existencia o no de desbalanceo en los datos
 - Tablas mostrando un resumen de las variables continuas
 - media, desviación típica, mínimo, máximo, porcentaje de valores faltantes, percentiles,...
 - Y categóricas
 - número de ejemplos para cada categoría, rango de los valores, porcentaje de valores faltantes, ...
 - Matrices de correlación de variables continuas (Pearson, Spearman)
 - etc.
- A nivel de visualización y análisis de los datos, **debéis centraros en cuestiones exploratorias y descriptivas siempre enfocadas a comprender mejor el problema.**
- **Debéis dejar muy claro en la memoria qué ha aportado la visualización de variables.**

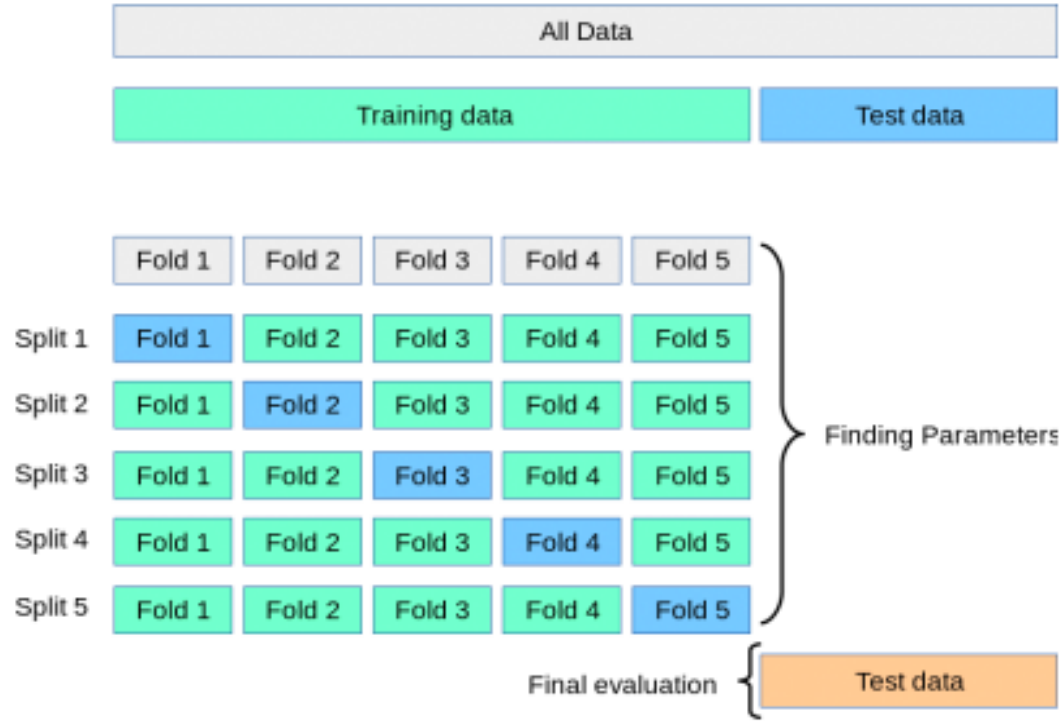
2. Identificar los **conjuntos de hipótesis a usar** (0.5 pts)

- sabemos que vamos a usar modelos lineales, pero ¿qué modelos concretos planteamos usar? ¿Por qué? → **Identificar qué modelos vamos a emplear!**
- Solamente se pide emplear **modelos lineales** (regresión lineal, regresión logística y perceptrón+pocket), junto con las **transformaciones en los datos**, técnicas de **regularización** y **preprocesado** que consideréis más conveniente
 - Si alguien quiere probar a mayores SVM (con kernel no lineal), MLP, RF. ¡Perfecto! Que compare con los modelos lineales y justifique su uso. ¡Pero hay que usar modelos lineales!

3. Definición de los conjuntos de **training, validación y test** (1.5 pts)

- Si la base de datos ya define conjuntos de training y test → unirlos y definir conjuntos de training y test propios.
 - Más adelante, se podría comparar los resultados obtenidos con los nuevos conjuntos de datos y con los conjuntos ya definidos y proporcionados con la base de datos
- ¿Uso de cross-validation? ¿Por qué?

5-fold cross-validation



4. Preprocesado de los datos (2 ptos): manipulaciones sobre los datos iniciales para obtener el conjunto de entrenamiento.

No hay reglas universales. Cada problema y característica requiere un procesamiento diferente.
Aquí aportamos algunas referencias generales.

- eliminar datos sin variabilidad (no son discriminantes)
- eliminar datos extremos/atípicos (*outliers*)
- ¿Qué hacemos con los datos faltantes (*missing data imputation*)?
- reducción de dimensionalidad (p.ej. PCA)
 - El objetivo debe ser reducir el número de variables (es decir, la complejidad de tu problema) mientras se mantiene la información relevante de los datos originales.
- transformaciones en los datos
 - Como ya habéis visto, ciertas transformaciones en los datos pueden permitir que un problema no linealmente separable se convierta en linealmente separable.

4. **Preprocesado de los datos (2 ptos):** manipulaciones sobre los datos iniciales para obtener el conjunto de entrenamiento.

No hay reglas universales. Cada problema y característica requiere un procesado diferente. Aquí aportamos algunas referencias generales.

- escalado de variables
 - Empleada para que todas las variables estén en un rango de valores similar
 - Objetivo:
 - acelerar la optimización/entrenamiento (véase explicación de Andrew Ng en [Normalizing Inputs \(C2W1L09\)](#))
 - hay técnicas especialmente afectadas por la escala de las variables (KNN, K-Means, SVM,...) y otras no (DTs, RF,...)
- codificación de datos
 - a) Si son binarias o numéricas → las podéis dejar como están (a no ser que tengáis algún motivo de peso para codificarlas/transformarlas de otro modo)
 - ¿Tiene sentido normalizar las variables binarias?
 - b) Si son variables categóricas:
 - Nominales → one-hot encoding (rojo – 1 0 0, verde – 0 1 0, azul – 0 0 1)
 - Ordinales → codificación entera (bajo – 1, medio – 2, alto – 3) o one-hot encoding
 - Cíclicas (días, meses, p.ej.) → cosine/sine transformation

5. Métrica de error (1 pto) Discutir su idoneidad para el problema

- a) MSE, MAE, Accuracy,...
- b) No confundir métrica de error (para medir el rendimiento del modelo) y función de pérdida (para optimizar el modelo).
- c) No confundir métricas de error para problemas de clasificación y regresión

6. Discutir todos los parámetros y el tipo de regularización a usar (2 ptos)

- a) Discutir la idoneidad de los valores de los parámetros de la técnica de ajuste. No podéis emplear los valores por defecto para los métodos incluidos en Scikit-learn sin saber qué hacen.
 - learning rate, tamaño de minibatch, criterio de parada, etc.
- b) L2 regularization /weight decay /ridge regression vs L1 regularization
 - En problemas de alta dimensionalidad, podría tener sentido usar L1, porque la solución que proporciona es dispersa (aplica selección de características de modo implícito)
 - Si sabemos que todas las variables deben ser tenidas en cuenta, seguramente mejor L2

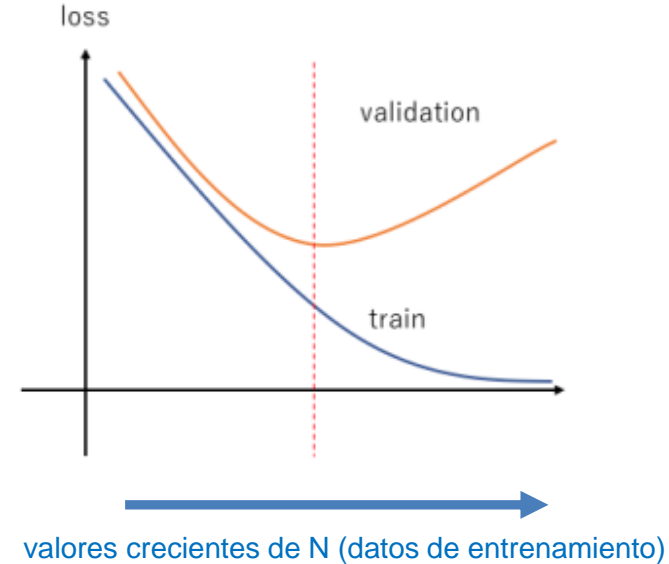
7. Estimación de hiperparámetros y selección de la mejor hipótesis (2 pts)

- Entrenamiento (E_{in}) y Test (E_{test}). Estimación de E_{out}
- Posibilidades de análisis:
 - a) Comparar el error obtenido a través de la selección de modelos (validación cruzada, E_{cv}) con el E_{test} de la mejor hipótesis
 - b) Comparar el E_{test} y/o E_{out} obtenido con distintos porcentajes de training y test
 - c) Emplear baselines con los que comparar.
 - i. Por ejemplo, si tengo un 3% de E_{test} no sé si es mucho, porque a lo mejor un estimador *naive* (la media en regresión, o un clasificador aleatorio en clasificación) ya me da un 4% de error.

8. Curvas de aprendizaje (1 pto)

Una vez hemos entrenado y testeado nuestro modelo hacemos lo siguiente:

- 1) Del conjunto de entrenamiento separamos un conjunto para entrenamiento, puramente dicho, y otro para validación.
- 2) Escogemos, dentro del conjunto de entrenamiento, un conjunto de datos pequeño, entrenamos, guardamos ese error, le pasamos el conjunto de validación, y también guardamos el error de validación correspondiente.
- 3) Así sucesivamente hasta que **hemos entrenado X veces con X conjuntos de entrenamiento progresivamente más grandes, y hemos validado X veces** (siempre con el mismo conjunto).
- 4) Visualizamos y analizamos ambas curvas.



Nota: no confundir estas curvas con las de *early-stopping*, empleadas generalmente en redes neuronales para evitar el sobreentrenamiento, y en donde se emplea un conjunto de validación simultáneamente con el de entrenamiento.

9. Caso real en donde no se distingue entre training y test (es decir, no hay conjunto de test definido) (1 pto)

- a) ¿Cómo escoger el mejor modelo y qué error E_{out} decimos que tiene?
- b) Posibilidades de análisis:
 - a) Analizar el optimismo de un conjunto de test pequeño vs grande, y empleando pocos datos de entrenamiento
 - b) Analizar el compromiso que representa el uso de validación cruzada
 - c) Analizar qué pasaría si validásemos y entrenásemos con los mismos datos. ¿Nuestra estimación del error sería optimista o pesimista?

Consejos generales

Presentad correctamente y describir con claridad el trabajo realizado:

- qué problema se aborda,
- detalles del proceso de validación cruzada,
- selección de hiperparámetros,
- regularización,
- valoración de los resultados finales con cada técnica, etc.

Más detalles importantes:

- la memoria no es el relato de ejecución del código,
- no debéis emitir meras opiniones o presentar decisiones sin justificación.
- evitad también **confundir conceptos básicos**
 - como podría ser considerar SGD como un modelo de aprendizaje automático, o confundir función de pérdida con métricas de evaluación, entre otros.
- no dudéis en **incluir todas las gráficas que consideréis pertinentes, y comentadlas en el cuerpo del texto** (dado que las gráficas no se comentan solas, y es necesario presentarlas y analizarlas).

Entrega

.zip = Código (.py) + Informe (.pdf)

o

.ipynb = Código, informe y resultados integrados en un Colab notebook

1 fichero para regresión
1 fichero para clasificación

Subid la entrega a PRADO, a la actividad creada para ello.

Fecha de entrega: 29 de Mayo

Prácticas de Aprendizaje Automático

Práctica 3

Ajuste de Modelos Lineales

Pablo Mesejo y Manuel Cobo

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial



UNIVERSIDAD
DE GRANADA

