



Adattamento Eccessivo nel machine learning

Docente: Tommaso Muraca



Cos'è l'Adattamento Eccessivo

Un **problema comune** nell'apprendimento automatico è l'**adattamento eccessivo** : l'apprendimento di una funzione che spiega **perfettamente i dati di addestramento** da cui il modello ha appreso, ma **non si generalizza bene ai dati di test invisibili**.

L'**overfitting** si verifica quando un modello impara dai **dati di addestramento** al punto che **si adatta troppo** e inizia a **rilevare idiosincrasie** che **non sono rappresentative dei modelli** nel mondo reale. Ciò diventa particolarmente problematico quando si rende il **modello sempre più complesso**.

L'**underfitting** è il problema **opposto**: il modello **non si adatta abbastanza** e quindi **non è abbastanza complesso** da catturare la tendenza sottostante nei dati.

Dobbiamo quindi sempre stabilire un **Compromesso tra bias-varianza**:

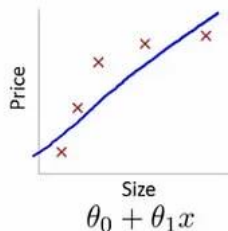
- Il **bias** è la **quantità di errore** introdotta approssimando i fenomeni del mondo reale con un modello semplificato;
- La **varianza** indica **quanto cambia l'errore** di test del modello in base alla variazione dei dati di addestramento. **Riflette la sensibilità del modello alle idiosincrasie** del set di dati su cui è stato addestrato.

Man mano che un modello aumenta di complessità e **diventa più sinuoso** (flessibile), la sua **distorsione diminuisce** (fa un buon lavoro nel spiegare i dati di addestramento), ma la **varianza aumenta** (non si generalizza altrettanto). In definitiva, per avere un **buon modello**, ne è necessario uno con **bassa distorsione e bassa varianza**.

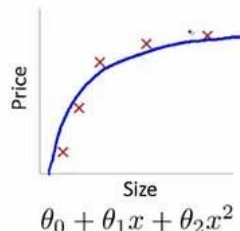
Cos'è l'Adattamento Eccessivo

Quindi la cosa più importante che dobbiamo **valutare è il rendimento del modello sui dati di test**. Nel **machine learning** infatti il nostro **intento non è** creare un **modello accurato al 100%** nei dati di **training**, **ma** un modello in grado di fare le **previsioni più accurate possibili**.

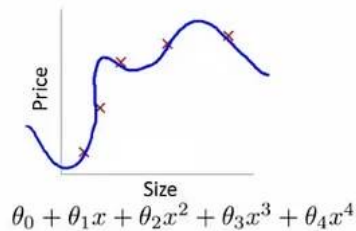
Guardando i **grafici** ci accorgiamo quindi che quello di **sinistra** potrebbe avere **troppi falsi positivi**, quello di **destra** invece **prevederebbe** solo i casi in cui le **caratteristiche fossero identiche ai casi di training**, il grafico **centrale** invece sarebbe il **giusto compromesso tra bias e varianza**.



High bias
(underfit)



"Just right"



High variance
(overfit)

Cos'è l'Adattamento Eccessivo

Per **evitare l'overfitting** abbiamo quindi due modi fondamentali:

1. **Utilizzare più dati di allenamento** . Più ne abbiamo, più difficile sarà adattare eccessivamente i dati imparando troppo da ogni singolo esempio di formazione.
2. **Utilizzare la regolarizzazione** . Aggiungere una penalità nella funzione di perdita per la costruzione di un modello che assegna troppo potere esplicativo a una qualsiasi caratteristica o consente di prendere in considerazione troppe caratteristiche.

$$Cost = \frac{\sum_1^n ((\beta_1 x_i + \beta_0) - y_i)^2}{2 * n} + \lambda \sum_{i=0}^1 \beta_i^2$$

La **prima parte** della somma di cui sopra è la nostra **normale funzione di costo**. La **seconda parte** è un termine di **regolarizzazione** che aggiunge una **penalità per coefficienti beta elevati** che danno troppo potere esplicativo a qualsiasi caratteristica specifica. Con questi due elementi in atto, la **funzione di costo ora si trova in equilibrio tra due priorità**: spiegare i **dati di addestramento** ed **evitare** che tale spiegazione diventi **eccessivamente specifica**.

Il **coefficiente lambda** del termine di regolarizzazione nella funzione di costo è un **iperparametro**: un'impostazione generale del modello che può essere aumentata o diminuita (ovvero ottimizzata) per **migliorare le prestazioni**. Un **valore lambda più elevato penalizzerà più duramente** i grandi coefficienti beta che **potrebbero portare a un potenziale overfitting**. Per decidere il **miglior valore di lambda**, dobbiamo **utilizzare un metodo chiamato convalida incrociata o k-fold validation** che prevede di fornire una **parte dei dati di addestramento durante l'addestramento** e quindi vedere **quanto bene il tuo modello spiega** la parte trattenuta. Parleremo in maniera più approfondita di questa tecnica più avanti.

Introduzione alla Classificazione

Docente: Tommaso Muraca



Classificazione: predire un'etichetta




Questa email è spam oppure no? Quella persona ripagherà il prestito? Gli utenti faranno clic sull'annuncio? Chi è quella persona nella tua foto su Facebook?




Questi sono alcuni dei **problemi che potremmo risolvere con la Classificazione**: essa ci permette di prevedere un'etichetta target discreta Y .

La **classificazione** ha quindi lo **scopo di assegnare una classe** alle nuove osservazioni a cui molto probabilmente **appartengono**, sulla base di un **modello di classificazione** costruito a partire da **dati di addestramento etichettati**.

L'**accuratezza** delle classificazioni dipenderà dall'**efficacia dell'algoritmo scelto**, da **come lo applichiamo** e dalla **quantità di dati di addestramento** utili di cui disponiamo.

Supervised Learning: Classification

training set	Observation #	Input image (X)	Label (Y)
	1		"dog"
	2		"cat"
	3		"dog"

	N		"dog"
test set	1		???
	2		???

Classificazione: Regressione Logistica

La **regressione logistica** è uno dei metodi di **classificazione** : il modello restituisce la **probabilità** che una **variabile target categoriale Y** appartenga a una **determinata classe**.

Un buon **esempio** di classificazione è **determinare se una richiesta di prestito è fraudolenta**.

In definitiva, la **banca vuole sapere** se deve **concedere o meno** un **prestito al mutuatario** e ha una certa **tolleranza per il rischio** che la **richiesta** sia effettivamente **fraudolenta**. In questo caso, l'**obiettivo della regressione logistica** è **calcolare la probabilità** (tra 0% e 100%) che la **richiesta sia fraudolenta**. Con queste probabilità, possiamo **impostare** una **soglia** al di sopra della quale siamo **disposti a concedere un prestito** al mutuatario e **al di sotto** della quale **neghiamo** la richiesta di prestito **o segnaliamo la richiesta per un'ulteriore revisione**.

Sebbene la regressione logistica venga **spesso utilizzata per la classificazione binaria** in cui sono presenti **due classi**, tenete presente che **la classificazione può essere eseguita con qualsiasi numero di categorie** (ad esempio quando si assegna a cifre scritte a mano un'etichetta compresa tra 0 e 9 o si utilizza il riconoscimento facciale per rilevare quali amici si trovano in una foto su Facebook).

Possiamo usare semplicemente i minimi quadrati ordinari?

No. Se addestrassimo un **modello di regressione lineare** su una serie di esempi in cui **$Y = 0$ o 1** , potremmo finire per **prevedere alcune probabilità inferiori a 0 o superiori a 1**, il che **non ha senso**.

Utilizzeremo invece un **modello di regressione logistica** (o modello logit) **progettato per assegnare una probabilità** compresa tra 0% e 100% **che Y appartenga a una determinata classe**.

Classificazione: Regressione Logistica

Il **modello logit** è una **modifica della regressione lineare** che assicura di produrre una **probabilità** compresa tra 0 e 1 applicando la **funzione sigmoide** che, quando rappresentata **graficamente**, assomiglia alla caratteristica **curva a forma di S**.

$$S(x) = \frac{1}{1 + e^{-x}}.$$

Ricordiamo la **forma originale del nostro modello di regressione lineare semplice**, che ora **chiameremo $g(x)$** poiché lo utilizzeremo all'interno di una funzione composta:

$$g(X) = \beta_0 + \beta_1 x + \epsilon$$

Per **risolvere il problema** di ottenere output del modello **inferiori a 0 o superiori a 1**, definiremo una **nuova funzione $F(g(x))$** che **trasforma $g(x)$ schiacciando l'output della regressione lineare** su un valore nell'intervallo $[0,1]$.

Quindi **inseriamo $g(x)$ nella funzione sigmoide sopra**, ottenendo una **funzione della nostra funzione originale** che restituisce una probabilità compresa tra 0 e 1:

$$P(Y = 1) = F(g(x)) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Qui abbiamo **isolato p** , la **probabilità che $Y=1$** , sul **lato sinistro dell'equazione**. Se vogliamo risolvere **per ottenere $\beta_0 + \beta_1 x + \epsilon$ pulito** sul **lato destro** in modo da poter **interpretare direttamente i coefficienti beta** che impareremo, ci ritroveremo invece **con il rapporto log-odds , o logit , su il lato sinistro** – da qui il nome “modello logit”:

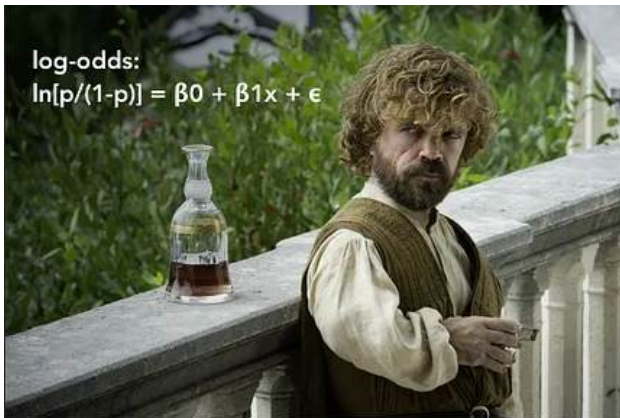
$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x + \epsilon$$

Classificazione: Regressione Logistica

Il **rapporto log-odd** è semplicemente il **logaritmo naturale del rapporto odd**, $p/(1-p)$, che poteva emergere qualche anno fa nelle conversazioni quotidiane dei fan di Game of Thrones:

"Ehi, quali pensi siano le **probabilità che Tyrion Lannister muoia in questa stagione di Game of Thrones?**"

"Hmm. Conoscendo la serie **è sicuramente 2 volte più probabile** che accada. **Quote 2 a 1**. Certo, potrebbe sembrare troppo importante per essere ucciso, ma abbiamo visto tutti cosa hanno fatto a Ned Stark..."



log-odds:
 $\ln[p/(1-p)] = \beta_0 + \beta_1 x + \epsilon$

<— IS HE GONNA DIE?

$p = P(\text{Tyrion dies}) = 2/3$

$1-p = P(\text{Tyrion doesn't die}) = 1/3$

odds ratio: $p/(1-p) = 2.0$
"He's gonna die. 2-to-1 odds"

log-odds ratio: $\ln[p/(1-p)] = 0.693$
"He's gonna die. .693 log-odds"

Machine Learning for Humans 🤖🧠

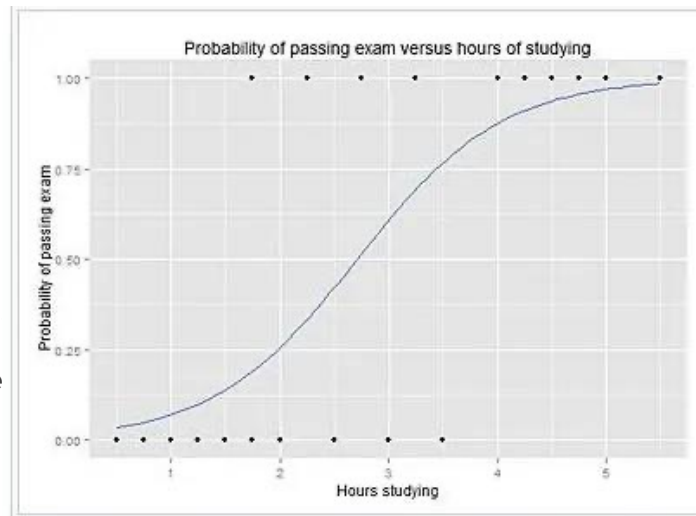
Classificazione: Regressione Logistica

L'output del modello di regressione logistica visto prima assomiglia ad una curva a S che mostra $P(Y=1)$ in base al valore di X ----->

Per **prevedere l'etichetta Y** (spam/non spam, cancro/non cancro, frode/non frode, ecc.) è necessario **impostare** un limite di probabilità, o **soglia**, per un **risultato positivo**. Ad esempio: "Se il nostro **modello ritiene** che la **probabilità che questa email sia spam è superiore al 70%**, **etichettala come spam**, altrimenti non etichettarla."

La **soglia** dipende dalla **nostra tolleranza ai falsi positivi** rispetto ai falsi negativi. Se stiamo **diagnosticando un cancro**, la **tolleranza sarebbe molto bassa** per i falsi negativi, perché anche se c'è una **possibilità molto piccola che il paziente abbia il cancro**, **dovresti eseguire ulteriori test per esserne sicuro**. Quindi imposteremmo una **soglia molto bassa per un risultato positivo**.

Nel caso di richieste di **prestito fraudolente**, d'altro canto, la **tolleranza per i falsi positivi potrebbe essere maggiore**, in particolare per i **prestiti più piccoli**, poiché **un'ulteriore verifica è costosa e un piccolo prestito potrebbe non valere i costi operativi aggiuntivi** e gli attriti per richieste di prestito non fraudolente.



Classificazione: Regressione Logistica

Come nel caso della **regressione lineare**, anche **nella regressione logistica** utilizziamo la **discesa del gradiente** per **apprendere i parametri beta** che **minimizzano la perdita**.

Nella regressione logistica, la **funzione di costo** è **fondamentalmente una misura di quanto spesso hai previsto 1 quando la risposta vera era 0, o viceversa**. Di **seguito** è riportata una **funzione di costo regolarizzato proprio come quella che abbiamo esaminato per la regressione lineare**.

$$Cost = \frac{\sum_1^n y^i \log(h_\beta(x^i)) + (1 - y^i) \log(1 - h_\beta(x^i))}{2 * n} + \lambda \sum_{i=1}^2 \beta_i^2$$

Lo so che è complessa **un'equazione lunga come questa**, ma basta **suddividerla in più parti** e pensare a cosa sta succedendo concettualmente in ciascuna parte, allora i dettagli inizieranno ad avere senso.

Il **primo pezzo è la perdita di dati**, cioè quanta **discrepanza c'è tra le previsioni del modello e la realtà**. La **seconda parte è la perdita di regolarizzazione**, ovvero **quanto penalizziamo il modello** per avere parametri di grandi dimensioni che appesantiscono notevolmente determinate funzionalità.

Minimizzando questa funzione di costo con la **discesa del gradiente** otterremo esattamente la **funzione di sopra**. Abbiamo quindi **creato un modello di regressione logistica per effettuare previsioni sulle classi nel modo più accurato possibile**.

Andiamo ora a vedere qualche esempio pratico.