Tecniche di insieme e classificazione con dati sbilanciati

Introduzione alla classificazione con dati sbilanciati

Nicola Torelli

2024

University of Trieste

Metodi di insieme e classificazione

Bagging

Random Forests

Boosting

Apprendimento con dati sbilanciati

Metodi di insieme e classificazione

Ancora su metodi di insieme

- I metodi di insieme si rivelano utili in particolare con tecniche come gli alberi di classificazione. Come si è detto essi possono essere instabili e producono alberi troppo complessi e classificatori deboli.
- Ricordiamo i principali metodi di insieme già visti nel caso della regressione:
  - **Bagging** *(Breiman, 1996)*: Adatta molti classificatori (anche alberi di grandi dimensioni) a versioni ricampionate bootstrap dei dati di addestramento si combinano le classificazioni ottenute per **voto di maggioranza**
  - **Boosting** *(Freund & Shapire, 1996)* e sue varianti: Adatta molti classificatori (spesso alberi molto piccoli) a versioni riponderate dei dati di addestramento. Si classifica in base voto di maggioranza pesato
  - **Random Forests** *(Breiman 1999)*: versione più elaborata del bagging per combinare alberi di regressione/classificazione.
- Salvo che per le foreste casuali, l'idea di combinare più previsioni o classificazioni può quindi essere utilizzata per qualsiasi tecnica (ad esempio, classificazione logistica, NN, ecc.) e non è limitata agli alberi

Combinazione di classificatori e voto di maggioranza

- L'idea è di combinare l'output di diversi classificatori per ogni punto dati (y, $\boldsymbol{x}$): questo aiuta quando i classificatori hanno punti di forza complementari.
- Supponiamo che siano disponibili i dati nel training set su p covariate $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$ e che la variabile risposta sia y
- Siano $\hat{y}_1 = f_1(\boldsymbol{x}), \ldots, \hat{y}_M = f_M(\boldsymbol{x})$, M diverse classificazioni ottenute per lo stesso dato x.
- Per ogni classe k abbiamo una previsione $f_m^k(x, t)$ uguale a 0 o 1. Quindi

$$f_{comb}(x) = \frac{1}{M} \sum_{m=1}^{M} f_m^k(\boldsymbol{x})$$

  per ogni classe k,

$f_{comb}^k(x, t)$ è quindi la proporzione dei voti per la classe k. Prevediamo la classe con il maggior numero di voti **(voto di maggioranza)**

Bagging

Bagging

- Bagging (bootstrap aggregation): consiste nel prendere molti set di training dalla popolazione, stimare un modello di classificazione separato per ogni training set e combinare le classificazioni risultanti.
- Possiamo a tal fine
  - prelevare B campioni dallo stesso set di dati di addestramento tramite il bootstrap
  - usare il b-esimo set di addestramento bootstrap per ottenere la classificazione e
  - utilizzare i voti di maggioranza andando a vedere quale è il valore k che per ciascun data è più frequente.

- Il bagging può ridurre drasticamente la varianza di una procedura instabile (come ad esempio gli alberi), portando a una classificazione più accurata.
- i confini decisionali sono meno spigolosi, tuttavia di solito si riduce la varianza, ma può aumentare leggermente il bias.

Out-of-bag

- L'utilizzo di campioni bootstrap di osservazioni consente l'uso dell'out-of-bag.
- In ogni campione di bootstrap, alcuni dei dati del training set originale restano esclusi.
- In media, ogni campione bootstrap utilizza circa i due terzi delle osservazioni
- Il restante terzo non è utilizzato per costruire ad esempio l'albero e sono le osservazioni out-of-bag (letteralmente fuori dal sacco) (OOB).
- Per ogni classificatore $\hat{f}^b(\boldsymbol{x})$ i dati del training set che non sono nel campione possono essere usati come test set.
- E' quindi possibile valutare l'errata classificazione su questi dati al di fuori del campione utilizzato per l'adattamento (out-of-bag), evitando così la convalida incrociata o riducendo la necssità di un test set separato.

Random Forests

Foreste casuali

- Un popolare affinamento del bagging, in particolare, è stato originariamente sviluppato nel contesto degli alberi di classificazione
- Ad ogni suddivisione dell'albero, viene estratto un campione casuale di covariate m, e solo queste sono considerate per la divisione.
- Tipicamente $m = \sqrt{p}$ o $\log_2 p$, dove p è il numero totale di covariate
- Ogni albero viene adattato su un campione bootstrap, e l'accuratezza viene monitorata tramite le osservazioni lasciate fuori dal campione bootstrap (OOB)
- Le foreste casuali cercano di migliorare il bagging "diminuendo" ulteriormente la correlazione fra alberi ottenuti su diversi campioni bootstrap".

Importanza delle covariate

Il difetto dei meodi di insieme è che sono di difficile interpretazione. Le foreste casuali tuttavia consentono almeno di ottenere una graduatoria per importanza delle variabili in un problema di classificazione.

- Per ogni albero cresciuto in una foresta casuale, si calcola il numero di voti per la classe corretta nei dati out-of-bag.
- Si esegue poi la permutazione casuale dei valori di un predittore (diciamo la variabile X_k) nei dati OOB e quindi si controlla il numero di voti per la classe corretta.
- Sottrarre il numero di voti per la classe corretta nei dati con variabile X_k permutata dal numero di voti per la classe corretta nei dati OOB originali.
- La media di questo numero su tutti gli alberi della foresta è il punteggio di importanza grezzo per la variabile X_k. Il punteggio viene normalizzato poi con la deviazione standard.
- Le variabili con valori elevati per questo punteggio sono classificate come più importanti. Se la costruzione di un modello senza i valori originali del predittore (cioè permutati casualmente) fornisce una previsione peggiore, significa che la variabile è importante.

Boosting

Il Boosting

- Il boosting è stato progettato, inizialmente, esclusivamente per problemi di classificazione binaria
- Idea: simile al boosting, ma invece che considerare campioni bootstrap, ogni successivo campione è tratto dalla popolazione con probabilità disuguali. A ogni iterazione, si da più peso alle osservazioni che in precedenza erano state classificate in modo errato, per costringere il classificatore, nei passaggi successivi, a cercare di classificare correttamente tali osservazioni "difficili". Invenzione di Freund e Schapire (1997)

La versione originale ADAboost

- In particolare
  - Si inizia utilizzando i dati originali con pesi di campionamento tutti uguali $p_i = 1/n$
  - All'iterazione t, si modificano le probabilità correnti $p_1^t, p_2^t, \ldots, p_n^t$, si addestra un classificatore (anche molto semplice) e si calcola e_t , il tasso di errore del classificatore sul campione originale. Sia $\beta_t = e_t/(1 - e_t)$
  - Per quei punti che sono classificati correttamente (e solo per essi), si diminuiscono le probabilità di estrazione ponendole pari a $p_i^t = p_i^t \beta_t$ (se necessario si normalizzano così che sommino a 1)
  - Si fa questo passaggio per molte (diciamo 1000) iterazioni.

Proprietà del boosting

- Al termine la classificazione finale avviene con voto di maggioranza ponderata, con pesi $\alpha_t = log(1/\beta_t)$ (più peso quindi ai classificatori con meno errori).
- Il boosting da spesso performance migliori del bagging
- Se applicata agli alberi, la ponderazione limita ulteriormente la correlazione fra essi e si concentra sulle regioni nello spazio delle covariate mancate dal classificatore nei passi precedenti.
- Poichè funziona molto bene anche con classificatori molto semplici e non molto efficaci (basta che classifichino meglio di una classificazione a caso) spesso si usano alberi molto elementari anche con una sola suddivisione (stump)

Varianti del boosting

- La principale variante di ADAboost, nel caso di modelli di regressione è il gradient boosting (o suoi ulteriori affinamenti come XGBoost - eXtreme Gradient Boosting)
- Nel caso dei problemi di classificazione, e in particolare degli alberi, la tecnica va modificata per tenere conto della differente funzione di perdita e della diversa natura dei residui conseguenti all'utilizzo di un classificatore debole. L'algoritmo specifico è detto "TreeBoost"
- Come detto il boosting (o sue varianti) può essere utilizzato anche con gli stumps (alberi con due soli nodi). Tuttavia risulta spesso più efficiente porre il numero di nodi pari a un valore fra 3 e 5.
- Il principale parametro da fissare è il numero di iterazioni (se troppo alto potrebbe generare overfitting), le diverse versioni del boosting possono richiedere di fissare altri parametri per evitare sovraadattamento.
- Versioni ulteriori del boosting prevedono l'uso di tecniche di regolarizzaione o di usare sottocampioni bootstrap per costruire ad ogni passo il classificatore
- il pacchetto `R` denominato `gbm` *generalized gradient boosting* contiene le suddette varianti. Esistono tuttavia altri pacchetti come `BRT` o ulteriori estensioni nel pacchetto `xgboost`

Apprendimento con dati sbilanciati

Classificazione con set di dati sbilanciati

- Il problema dello sbilanciamento dei dati emerge nei problemi di classificazione supervisionata e qui citeremo solo il caso (più rilevante) di classificazione binaria
- È un problema che si verifica quando una delle due classi che rappresenta la variabile di output (di solito è anche la classe di principale interesse) è **rara** e quindi è molto meno rappresentata dell'altra classe nel dataset
- È una situazione che si incontra in molte applicazioni reali:
  - in molti problemi di rilevamento delle frodi, il numero di frodi osservate è (fortunatamente) un evento raro
  - quando si prevede l'insolvenza di un'impresa, l'evento di fallimento in un dato anno non è molto frequente
  - in medicina, molte malattie specifiche nella popolazione hanno solitamente una frequenza molto bassa
  - ci sono molti esempi in cui i clienti sono molto fedeli e osservare il tasso di abbandono dei clienti è raro

Grado di sbilanciamento

- Il grado di sbilanciamento per la variabile di risposta può essere misurato da IR che è definito come il rapporto tra il numero di casi della classe prevalente diviso per il numero di dati nella classe rara. Dire che IR è 100 significa che per 1 punto dati nella classe rara (positiva) ci sono 100 casi della classe prevalente (negativa).
- In realtà qualsiasi set di dati presenta un certo grado di squilibrio, ma la gravità del problema per un problema di classificazione a due classi emerge solitamente quando l'IR è almeno maggiore di 10
  - IR maggiore di 100 definisce un forte sbilanciamento
  - IR maggiore di 1000 è un set di dati estremamente distorto (sbilanciamento molto forte)
- Negli ultimi due casi è senz'altro necessario affrontare il problema di sbilanciamento prima della costruzione di qualsiasi regola di classificazione o per definire l'algoritmo di apprendimento automatico
- La dimensione del data set è un altro elemento importante per definire la gravità del problema

Perché gli algoritmi di classificazione standard possono fallire con dati sbilanciati?

- Classificatori standard come la regressione logistica, le Support Vector Machines (SVM), gli alberi di classificazione o altri algoritmi sono adatti per training set bilanciati.
- Di fronte a scenari sbilanciati, queste tecniche di classificazione spesso forniscono risultati non ottimali, funzionano infatti molto bene per i dati del gruppo maggioritario, e tendono a funzionare mediocremente sui dati del gruppo di minoranza
  - Il processo di apprendimento di un classificatore è spesso guidato da metriche di performance globali come l'accuratezza delle previsioni che inducono una distorsione verso la classe maggioritaria
  - I rari dati della classe minoritaria vengono visti dal modello di classificazione come rumore .
- Negli ultimi due decenni sono stati sviluppati molti approcci statistici e di apprendimento automatico per far fronte alla classificazione dei dati sbilanciata, la maggior parte dei quali si è basata su tre strategie: (i) tecniche di campionamento, (ii) apprendimento sensibile ai costi e (iii) possibili modifiche dell'algoritmo di apprendimento

Metriche per misurare le prestazioni di classificatori in problemi a due classi

- La qualità di un algoritmo di apprendimento viene valutata osservando le sue prestazioni sui dati di test.
- Le misure di performance più semplici si basano sul confronto tra le previsioni del classificatore ei valori reali (matrice di confusione).

		previsto		
		positivo	negativo	totale
Effettivo	positivo	TP	FN	POS
	negativo	FP	TN	NEG
	totale	PredPOS	PredNEG	N

- La misura più semplice, l'accuratezza (ACC) è definita come

$$ACC = \frac{TP + TN}{N}$$

- si noti che l'ACC **non** può essere utilizzato come misura di performance in set di dati sbilanciati: un classificatore banale che classifica sempre i nuovi casi nella classe maggioritaria avrà un'accuratezza pari alla proporzione della classe maggioritaria nel campione
- Se abbiamo lo 0.1% del campione di casi con un tipo di tumore raro, una strategia (non brillante) che prevede sempre non sei malato otterrà

Altre metriche e loro utilizzo per dati sbilanciati

- Tasso di veri positivi (recall o sensibilità) $= \frac{TP}{POS}$
- Tasso di veri negativi (specificità) $= \frac{TN}{NEG}$
- Valore predittivo positivo (precision) $= \frac{TP}{predPOS}$
- $F1 = (1+\beta^2)\frac{\text{precisione}\cdot\text{richiamo}}{(\beta^2\text{precisione})+\text{richiamo}}$
- K di Cohen $K = \frac{(TP+TN)-EXPACC}{1-EXPACC}$ dove EXPACC indica i dati sulla diagonale attesi nella matrice di confusione in ipotesi di indipendenza
- Tali misure dipendono dalla soglia per quei metodi (regressione logistica, alberi) che stimano un punteggio (probabilità) che può essere a volte arbitraria
- AUC which does not depend on a given threshold is a most appropriate measure for comparing performances with imbalanced dataset (ROC curve, and AUC, can be very unstable when the test dataset is small and imbalanced).

Approcci alla classificazione con dati sbilanciati

- Tecniche di pretrattamento dei dati (ricampionamento e generazione di dati sintetici)

Eseguite prima di costruire il modello di apprendimento per ottenere dati di input bilanciati nella costruzione del classificatore

- Apprendimento sensibile ai costi di errata classificazione
- Modifiche specifiche degli algoritmi di classificazione per l'apprendimento con dati sbilanciati

Tecniche di ricampionamento: sottocampionamento

- Questa strategia appartiene alla prima categoria di rimedi: le tecniche di pretrattamento
- L'obiettivo è quello di ottenere un campione bilanciato (diciamo che una delle classi avrà non meno del 30% dei casi, o meglio ancora le due classi sono di dimesansione equivalente) sia per il training che per il test set.
- Sottocampionamento:

questo metodo riduce la classe maggioritaria. Consiste nel selezionare casualmente un sottoinsieme di osservazioni (senza sostituzione) dalla classe di maggioranza per rendere bilanciato il set di dati. Questo metodo può essere utilizzato con grande successo quando il set di dati è davvero enorme. Può essere migliorato aggiungendo una strategia per selezionare i dati più rilevanti per la classificazione.

Tecniche di (ri)campionamento: sovracampionamento

- Elimina i danni della distribuzione distorta dela variabile risposta moltiplicando nuovi campioni di classi di minoranza. I dati della classe rara vengono ricampionati (con sostituzione) per bilanciare il campione (si noti che può indurre overfitting)
- Il sovracampionamento può essere ottenuto anche generando nuove dati sintetici a partire da quelli disponibili.
  - Esistono molti metodi specificamente progettati per generare nuovi dati sintetici (clonazione dei dati) per la classe di minoranza. Ne descriveremo brevemente due:
    - SMOTE (Chawla et al, 2002)
    - ROSE (Menardi & Torelli, 2014)
- Metodi ibridi: sono una combinazione del metodo del sovracampionamento e del metodo del sottocampionamento.

ROSE: Random OverSampling Examples

- ROSE ha lo scopo di sovracampionare la classe rara creando nuovi punti dati (sintetici) che siano il più simili possibile a quelli esistenti (reali)
- ROSE suggerisce anche di sottocampionare la classe maggioritaria (possibilmente clonando i dati con la stessa strategia)
- ROSE sceglie un punto casuale dall'insieme raro e quindi viene generato un nuovo punto nel suo intorno secondo una funzione del kernel (equivale a mettere una distribuzione gaussiana multivariata centrata sul punto e selezionare un nuovo punto da quella gaussiana)
- Il metodo di clonazione è formalmente basato su una stima della densità del kernel della distribuzione delle variabili predittive all'interno della classe rara (o talvolta anche della prevalente). Ciò risulta essere equivalente all'ottenimento di uno schema di ricampionamento bootstrap **lisciato**

ROSE: Random OverSampling Examples

- Esiste un pacchetto in R denominato *ROSE*. Può anche essere richiamato direttamente all'interno del pacchetto *caret*. Recentemente una versione di Rose è stata resa disponibile anche all'interno di *Scikit-learn* in Python
- *ROSE* può anche essere usato per sottocampionare la classe prevalente.
- A volte è utile una combinazione di undersampling e oversampling (possibilmente clonando anche dati dalla classe prevalente)
- Di solito il set di test viene lasciato invariato. Ma gli autori di ROSE suggeriscono che per una stima più stabile di alcune delle metriche (come l'AUC) anche per il set di test è possibile clonare alcuni nuovi dati

SMOTE

- SMOTE è un metodo popolare (il più noto) per la clonazione dei dati.
- Genera nuovi dati nella classe rara per ottenere tanti casi quanti nella classe prevalente.
- La generazione di nuovi casi avviene selezionando prima a caso un caso dalla classe rara e considerando i punti K che sono più vicini a questo punto.
- Vengono generati nuovi punti selezionando una posizione casuale lungo la linea che collega il punto selezionato a uno dei punti K vicini
- Ci sono molte varianti di SMOTE
- In R SMOTE è una funzione disponibile all'interno del pacchetto R *DMwR* o nel pacchetto `smotefamily`. Può anche essere richiamato direttamente all'interno del pacchetto *caret*. SMOTE e le sue varianti sono disponibili in Python Scikit-learn

Alcuni problemi pratici

Nelle applicazioni reali bisogna scegliere:

1. quando lo squilibrio è un problema. A volte si può considerare di bilanciare il training set anche quando IR non è maggiore di 4-5. Con piccoli set di dati potrebbe comunque essere vantaggioso. Mentre in caso di grandi (o enormi set di dati) strategie semplici/ingenue come il sottocampionamento della classe prevalente possono dare buoni risultati
2. quale metodo è più appropriato. Questa è una questione di gusti. Di solito, si suggerisce di considerare strategie alternative e confrontare le prestazioni. Ogni metodo ha i suoi meriti e quale sia il più appropriato dipende dalle caratteristiche specifiche dei dati
3. la natura dei predittori è importante.

- SMOTE e ROSE funzionano meglio quando la maggior parte dei predittori è quantitativa. Con i predittori categoriali può essere più appropriato un semplice sottocampionamento o sovracampionamento.
- quando si applica ROSE è necessaria una maggiore attenzione nel caso di variabili miste (ad esempio variabili con inflazione di zeri) o variabili limitate inferiormente (o superiormente)

Altri rimedi

1. Apprendimento sensibile ai costi

In molti casi i due errori non hanno la stessa importanza. Ai due errori FP e FN può essere associato un costo diverso e all'errore più rilevante per il problema specifico viene assegnato un valore più alto. La funzione di perdita da minimizzare per l'algoritmo cambierà di conseguenza e potrebbe concentrarsi sulla previsione accurata della classe di minoranza

2. Modifica degli algoritmi standard

Uno degli esempi più notevoli è la modifica delle procedure di boosting/bagging per tenere conto dello squilibrio dei dati. Si noti quindi che spesso il semplice uso di metodi di insieme può alleviare il problema dello squilibrio dei dati