Introduzione al Data Mining

e applicazioni al dominio del Contact Management

Parte V: combinazione di più modelli

Andrea Brunello

Università degli Studi di Udine

In collaborazione con dott. Enrico Marzano, CIO Gap srl
progetto Active Contact System

Contenuti

Parte V:

- combinazione di più modelli:
  - Bagging;
  - Randomization;
  - Boosting.

Combinazione di più modelli

- Spesso il combinare più modelli porta ad un significativo aumento delle performance di classificazione;
- il prezzo da pagare riguarda la difficile interpretabilità del “supermodello” ottenuto;
- come combinare il risultato di più modelli?
  - nel caso di output qualitativo, es. votazione
  - nel caso di output quantitativo, es. media

Bagging

Idea che sfrutta l'instabilità del metodo di costruzione del modello (es. alberi di decisione):

- supponiamo di avere un insieme di dataset, tutti della stessa dimensione, rappresentativi per il problema che si vuole affrontare;
- costruiamo un albero di decisione a partire da ciascuno di essi;
- in generale, ciascun albero sarà diverso, e classificherà in maniera corretta alcune istanze;
- si combinano i risultati per fornire l'output finale.

Risultati tendenzialmente migliori rispetto all'uso di un unico classificatore.

$\rightsquigarrow$ riduce la *variance*, errore dovuto al particolare training set utilizzato; aiuta ad evitare l'*overfitting*

Bagging (2)

Problema: ottenere molti dataset di training distinti può essere difficile. La soluzione è una “approssimazione” dell’idea:

- partiamo da un unico insieme di n esemplari;
- generiamo multinsiemi di cardinalità n, selezionando casualmente con reinserimento;
- in generale è possibile costruire un qualsiasi modello su essi, il cui processo di training sia instabile;
- Bagging $\approx$ Bootstrap Aggregating.

Randomization

Il bagging costruisce diverse varianti di un modello introducendo casualità nel processo di generazione dell'input.

- è tuttavia applicabile solo a metodologie di training instabili.

Idea $\rightarrow$ introdurre casualità nel processo di generazione del modello:

- es. negli alberi di ricerca, ad ogni nodo scegliere il miglior attributo su cui effettuare lo split da un sottoinsieme casuale degli attributi.

Importante determinare il giusto *trade-off* fra varietà dei modelli generati ed accuratezza del singolo.

Randomization (2)

Due implementazioni largamente utilizzate:

- **Random Forest**: stessa procedura del bagging, con la differenza che per la generazione di ciascun albero viene utilizzato l'approccio randomization.
- **Rotation Forest**: miglioramento dell'approccio *Random Forest*, utilizza *random subspaces* (applicazione di randomization allo instance-based learning), *PCA* e *bagging*.

Boosting

Si propone di migliorare la tecnica di bagging:

- nel bagging i singoli modelli componenti vengono generati separatamente;
- nel boosting la costruzione di un nuovo modello è influenzata dai precedenti;
  - intuitivamente si cerca di generare un insieme di modelli complementari
- inoltre, nel boosting in sede di voto si dà maggior peso al risultato fornito dai modelli ritenuti più "affidabili".

Boosting (2)

Implementazione ampiamente usata: **AdaBoost** (Adaptive Boosting).

Sperimentalmente si osserva che:

- boosting tende a produrre classificatori più performanti rispetto a bagging;
- a differenza che nel bagging, c'è però un rischio di fallimento su casi reali, tendenzialmente indice di *overfitting*.

Riferimenti

I. H. Witten, E. Frank, M. A. Hall: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 3rd edition, 2011.

G. James, D. Witten, T. Hastie, R. Tibshirani: *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.