Apprendimento Automatico (Feature Selection e Kernel Learning)

Fabio Aiolli
www.math.unipd.it/~aiolli

Sito web del corso
www.math.unipd.it/~aiolli/corsi/1516/aa/aa.html

Servono tutti gli attributi?

- Gli attributi (o variabili) dovrebbero essere utilizzati solo se veramente utili (rilevanti) per la classificazione/predizione
  - Meno attributi implicano modelli (p.e. alberi di decisione) di classificazione-predizione più compatti e che hanno bisogno di un numero minore di esempi di apprendimento per ottenere buoni risultati (minor varianza)
  - Modelli che usano pochi attributi sono più semplici da comprendere per un umano e più facilmente rappresentabili

Due approcci

- Feature Selection
  - Si seleziona un sottoinsieme degli attributi tra quelli originali.
- Feature Extraction
  - Si derivano nuovi attributi (features) da quelli originali. Per esempio, nuove fetaures sono ottenute come combinazioni di attributi.

Meriti dei due approcci

- ✓ Feature Selection

  Rimozione di features non rilevanti o ridondanti. Migliore interpretabilità del modello predittivo. Preferibili per applicazioni dove l'interpretabilità è più importante dell'accuratezza.

- ✓ Feature Extraction

  Tipicamente più potenti generando features più discriminative. Accuratezza migliore. Preferibili per applicazioni dove l'accuratezza è più importante dell'interpretabilità.

Feature Selection

- ✓ Filters methods

  Considera caratteristiche generali del training set. Feature selection è un pre-processamento dei dati indipendente dall'algoritmo di predizione

- ✓ Wrapper methods

  La selezione delle feature viene fatta sulla base della loro capacità predittive, tipicamente su un insieme hold-out

- ✓ Embedded methods

  Una via di mezzo. La selezione delle feature è inserita nella ottimizzazione (training) del modello predittivo

Feature Extraction

- ✓ Il metodo più importante si chiama Principal Component Analysis (PCA) e consiste nella estrazione di un insieme di features non correlate linearmente (componenti principali)
- ✓ Il numero di componenti principali è solitamente molto inferiore al numero di features originali

Applicazioni

- ✓ Computational biology. Pochi esempi (decine di campioni) e moltissime features (migliaia di geni)
- ✓ Face recognition. Quali sono le features di una faccia più importanti per il riconoscimento?
- ✓ Health studies. Generalmente il calcolo delle feature ha un costo alto!
- ✓ Financial engineering and risk management. Moltissimi fattori coinvolti nell'evoluzione. Ridurre il numero di fattori a quelli più importanti riduce la complessità e migliora la interpretabilità dei risultati.
- ✓ Text classification. Ogni termine è associato ad una features. Riducendo il numero di features si riduce la complessità computazionale degli algoritmi di apprendimento.

Kernel Learning

IDEA: Apprendere la funzione (o la matrice) kernel

1. Metodi parametrici per il kernel learning
2. Transductive feature extraction con kernel non lineari
3. Spectral Kernel Learning
4. Multiple Kernel Learning

Metodi parametrici per il KL

Ottimizza i parametri di una funzione kernel parametrica (per esempio RBF, Poly)

$$k(x, z) = e^{-(x-z)^t M(x-z)}$$

Scelte particolari:

$$M = \beta_0 I$$

$$M = diag(\beta_1, \dots, \beta_m)$$

Transductive feature extraction con kernel non lineari

Effettua una feature extraction implicitamente nel feature space.

Kernel Principal Component Analysis (KPCA) è l'esempio più popolare di questo approccio.

Si calcolano *implicitamente* le proiezioni sugli autovettori (direzioni principali)

Problema: è un approccio solo trasduttivo! Necessario usare tecniche *out-sample* per approssimare i valori del kernel su nuovi esempi

Spectral Kernel Learning

La matrice kernel (essendo definita positiva) può essere scritta nel modo seguente:

$$K = \sum_{s=1}^{n} \lambda_s \, u_s \cdot {u_s}^t$$

λ_s autovalori di K

u_s autovettori di K

Utilizzando una trasformazione $\tau(\lambda)$ implicitamente agiamo nello spazio delle feature. Infatti , il kernel modificato può essere ottenuto con $X = U \, \Lambda^{1/2}$.

L'idea quindi è quella di ottimizzare lo spettro della matrice kernel. Anche questo è un approccio trasduttivo, percui valgono le problematiche del caso precedente!

Multiple Kernel Learning

In questo caso si tratta di combinare linearmente kernel diversi definiti a priori e apprendere la combinazione:

$$K = \sum_{s=1}^{n} \mu_s \, K_s$$

- ✓ Fixed methods o heuristic methods: una semplice regola (magari euristica) viene utilizzata per il calcolo dei coefficienti (semplice media dei kernel, basati sulla accuratezza dei singoli kernel, ecc.)
- ✓ Optimization methods: inglobano i coefficienti come variabile da apprendere nel problema di ottimizzazione (per esempio la SVM)

Per approfondire…

Veronica Bolon-Canedo,
Michele Donini,
Fabio Aiolli

“Feature and kernel learning”

23th European Symposium on Artificial Neural Networks, ESANN 2015, Bruges, Belgium, April 22-24, 2015