(F94) Metodi statistici per l'apprendimento

Cross-validazione

Docente: *Nicolò Cesa-Bianchi* versione 7 marzo 2017

Scelta del parametro tramite validation set. Come abbiamo visto, gli algoritmi di apprendimento hanno spesso dei parametri (il valore di k in k-NN o il numero di nodi nei classificatori ad albero) e la scelta corretta del parametro evita che si verifichino underfitting e overfitting. Per determinare il valore corretto del parametro non possiamo però basarci sul test set, che —come abbiamo detto— può solo essere utilizzato per valutare l'accuratezza del predittore prodotto dall'algoritmo, non per determinare il predittore stesso. Il modo corretto di procedere è quello di isolare una porzione di training set (possibilmente scelta a caso) che chiameremo *validation set*. Per scegliere il valore del parametro possiamo allora eseguire più volte, con valori del parametro diversi, l'algoritmo di apprendimento sul training set a cui è stato sottratto il validation set. L'insieme di predittori risultante viene poi testato sul validation set per individuare quello con *validation error* minore. A questo punto eseguiamo nuovamente l'algoritmo di apprendimento sull'intero training set (compreso di validation set) utilizzando il valore del parametro che ha prodotto il predittore con validation error minimo. Il predittore risultante viene infine testato sul test set per determinarne l'accuratezza.

Valutazione di un algoritmo tramite cross-validazione esterna. Mentre il test error permette di valutare l'accuratezza di un predittore, la cross-validazione esterna è una tecnica che permette di valutare l'accuratezza di un algoritmo di apprendimento stimando l'accuratezza media dei predittori prodotti dall'algoritmo. Supponiamo che $S \equiv \big\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\big\}$ siano tutti i dati in nostro possesso (usiamo la notazione insiemistica ricordando però che S è tecnicamente un multinsieme). Partizioniamo S in N sottoinsiemi $D_1, \ldots, D_N$ ciascuno di cardinalità m/N (per semplicità, supponiamo che m sia divisibile per N, il caso estremo $N = m$ fornisce una stima dell'errore chiamata **leave-one-out**). Denotiamo con $S^{(k)}$ l'insieme S a cui abbiamo tolto tutti gli elementi di D_k.

Per esempio, se dividiamo $S = \big\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{20}, y_{20})\big\}$ in $N = 4$ sottoinsiemi

$$
\begin{aligned}
D_1 &= \big\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_5, y_5)\big\} \\
D_2 &= \big\{(\boldsymbol{x}_6, y_6), \ldots, (\boldsymbol{x}_{10}, y_{10})\big\} \\
D_3 &= \big\{(\boldsymbol{x}_{11}, y_{11}), \ldots, (\boldsymbol{x}_{15}, y_{15})\big\} \\
D_4 &= \big\{(\boldsymbol{x}_{16}, y_{16}), \ldots, (\boldsymbol{x}_{20}, y_{20})\big\}
\end{aligned}
$$

allora $S^{(2)} = \big\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_5, y_5), (\boldsymbol{x}_{11}, y_{11}), \ldots, (\boldsymbol{x}_{20}, y_{20})\big\}$.

Per stimare le prestazioni tipiche di un algoritmo A eseguiamo l'algoritmo su ciascun $S^{(k)}$, $k = 1, \ldots, N$. Siano $h^{(1)}, \ldots, h^{(N)}$ i predittori così prodotti. La stima dell'accuratezza di A mediante

cross-validazione di grado N è allora formulata come:

$$\frac{1}{N}\sum_{k=1}^{N}\widetilde{\mathrm{er}}_k\big(h^{(k)}\big)$$

dove

$$\widetilde{\mathrm{er}}_k\big(h^{(k)}\big) = \frac{N}{m}\sum_{(\boldsymbol{x},y)\in D_k}\ell\big(y, h^{(k)}(\boldsymbol{x})\big)$$

è l'errore di $h^{(k)}$ —rispetto ad una data funzione di perdita ℓ— misurato su D_k, ovvero sulla parte di dati che non è stata utilizzata per il training di $h^{(k)}$.

Scelta del parametro tramite cross-validazione interna. La cross-validazione può anche essere utilizzata come alternativa più sofisticata al validation set per stimare il valore ottimo del parametro col quale eseguire un dato algoritmo di apprendimento. In questo caso parliamo di cross-validazione interna in quanto agiamo soltanto sul training set. Il procedimento di cross-validazione interna sul training set è equivalente a quello di cross-validazione esterna sull'intero insieme di dati. Ovvero, il training set viene suddiviso in N blocchi di uguale grandezza e l'algoritmo A viene eseguito N volte con un valore fissato i del parametro utilizzando ciascun blocco a turno come validation set ed i rimanenti $N-1$ blocchi come training set. Mediando il validation error sugli N blocchi otteniamo l'errore di cross-validazione $\mathrm{er}_i^{\mathrm{cv}}$ per l'algoritmo A eseguito col parametro fissato al valore i. Questo processo viene ripetuto più volte con diversi valori del parametro finché non si trova un valore i^* che corrisponde approssimativamente a $\operatorname{argmin}_i \mathrm{er}_i^{\mathrm{cv}}$. A questo punto l'algoritmo A viene nuovamente eseguito sull'intero training set col parametro fissato a i^* allo scopo di produrre un predittore che sperabilmente esibirà un basso errore sul test set.

Chiaramente, i meccanismi di cross-validazione interna ed esterna possono essere combinati. Ovvero, possiamo applicare la cross-validazione esterna per stimare la prestazione tipica di un algoritmo eseguito con parametro ottimizzato, dove l'ottimizzazione del parametro è realizzata tramite cross-validazione interna.