Riconoscimento e Recupero dell'Informazione per Bioinformatica

LAB. 8 – PRTools (2)

Pietro Lovato

Corso di Laurea in Bioinformatica
Dip. di Informatica – Università di Verona
A.A. 2016/2017

Ripasso: validazione dei classificatori

- Obiettivo: valutare la capacità di **generalizzazione** di un classificatore

- Capacità di classificare correttamente anche oggetti sconosciuti non utilizzati per il training

Ripasso: validazione dei classificatori

- Testing set: insieme che contiene oggetti del problema, per cui si conosce la classe vera, ma non utilizzati per costruire il classificatore

- In molti problemi non abbiamo a disposizione training e testing set, ma un unico dataset etichettato.

Ripasso: validazione dei classificatori

In pratica:

- Si suddivide l'insieme a disposizione in due parti:
  - Una parte per l'addestramento del classificatore
  - Una parte per testare
    - Si contano gli errori che il classificatore commette sul **testing set**

Creazione di training e testing set in PRTools

```
>> [A,B,IA,IB] = gendat(D,N)
```

Input:

- D: dataset di input (N.B.: è il dataset completo, si occuperà PRTools di suddividerlo in train/test)
- N: se N > 1, numero di oggetti da selezionare per costruire A; se 0<N<1, percentuale di oggetti da selezionare per costruire A

Creazione di training e testing set in PRTools

```
>> [A,B,IA,IB] = gendat(D,N)
```

Output:

- `A,B`: dataset di output (si possono usare uno come train e l'altro come test)
- `IA,IB`: indici degli oggetti di `D` che sono stati selezionati in `A` e in `B` rispettivamente

Validazione: matrice di confusione

- Una possibile validazione si può fare controllando l'errore di classificazione
- Oppure, costruendo una matrice di confusione C: indica come un classificatore si comporta rispetto alle classi
- C(i,j) = numero di elementi della classe i classificati come elementi della classe j

7 oggetti della classe 1 sono stati correttamente classificati

3 oggetti della classe 2 sono stati erroneamente classificati come classe 1

	Estimated label	
True label	**1**	**2**
1	7	1
2	3	9

Matrice di confusione in PRTools

```
>> confmat(T)
```

- T: mapping applicato al dataset di test
- Ricordare la pipeline di classificazione in PRTools:

```
>> W = D_train * ldc; %ldc classificatore
>> T = D_test * W;
```

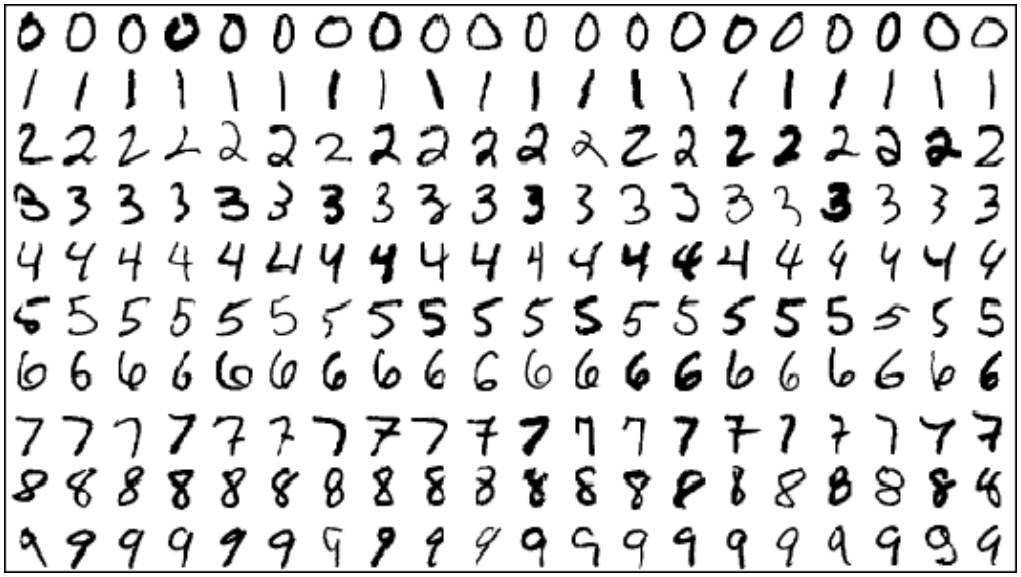
Esercizio 1

- Controllare che prtools sia installato correttamente (vedi slide lezione precedente)

- Caricare in memoria il dataset mfeat_fou:

```
>> prdatasets
>> A = mfeat_fou
```

- Task: classificazione di cifre scritte a mano



Esercizio 1

- Suddividere il dataset in due partizioni casuali usando il comando `gendat` (ogni partizione deve contenere il 50% degli oggetti totali)
  - Un dataset usato per il train, l'altro per il test

- Addestrare il classificatore ldc sul training set e valutare la matrice di confusione sul test set
  - Quali sono le classi che si confondono di più?

Cross-validazione in PRTools

Ulteriore tecnica di validazione in PRTools: comando `prcrossval`

- PRTools permette di validare un classificatore con un'unica routine:

```
>> [E,STD] = prcrossval(A,CLASSF,NFOLDS,NREP)
```

Cross-validazione in PRTools

```
>> [E,STD] = prcrossval(A,CLASSF,NFOLDS,NREP)
```

Input:

- `A`: dataset di input (N.B.: è il dataset completo, si occuperà PRTools di suddividerlo in train/test)
- `CLASSF`: classificatore non addestrato (es. `svc`)
- `NFOLDS`: in quante partizioni suddividere il dataset
- `NREP`: quante volte ripetere la suddivisione casuale

Cross-validazione in PRTools

```
>> [E,STD] = prcrossval(A,CLASSF,NFOLDS,NREP)
```

Output:

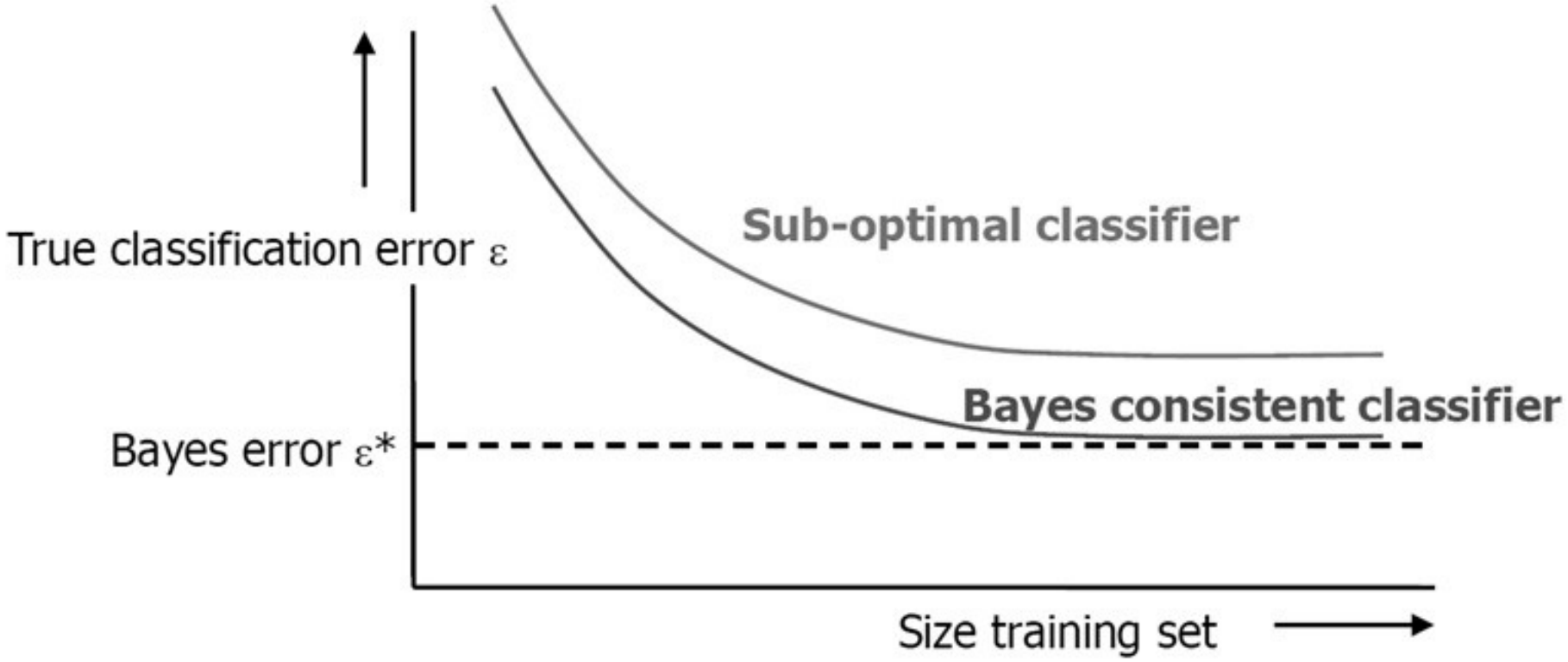
- E: errore medio di classificazione sul testing set
- STD: deviazione standard delle varie ripetizioni

Esercizio 2

- Effettuare un test di crossvalidazione sul dataset mfeat_fou
  - Classificatore da usare: ldc
  - Protocollo di test: 10-fold crossvalidation
  - Provare con 5 ripetizioni
  - Ripetere l'esercizio usando 2-fold crossvalidation
    - In quale dei due casi la deviazione standard sui risultati è più alta? E l'errore?
    - Quale dei due protocolli sembra più veloce da eseguire?

Learning curve

- Variazione dell'errore al variare della dimensione del training set



Learning curve in PRTools

```
>> E = cleval(A,CLASSF,TRAINSIZES,NREPS)
>> plote(E)
```

- A: input dataset
- CLASSF: classificatore (o insieme di classificatori) non addestrati da valutare
- TRAINSIZES: vettore che specifica il numero di oggetti di train da considerare
- NREPS: numero di ripetizioni casuali

Esercizio 3

- Seguire il tutorial sulle learning curve qui: http://www.37steps.com/prtools/examples/learning-curves/

Esercizio 3

- Costruire le learning curve sul dataset mfeat_fou nel seguente modo:
- Classificatori: `{ldc, knnc}`
- Numero di oggetti da considerare:

  `[100, 200, 300, 400, 500, 600, 700]`
- Provare con 2 ripetizioni
- Quale classificatore sembra essere più appropriato?