C4.5 Algorithms for Machine Learning

C4.5 Algorithms for Machine Learning

- **Apprendimento di alberi decisionali**
- c4.5 [Qui93b,Qui96] **Evoluzione di ID3**, altro sistema del medesimo autore, J.R. Quinlan
- Ispirato ad uno dei primi sistemi di questo genere, **CLS** (Concept Learning Systems) di E.B. Hunt
- Lo sviluppo del software è cessato, in favore di C5.0, ma **rimane uno dei punti di riferimento nella propria classe di algoritmi**. Ultima release: 8.0
- Algoritmo scritto in C per Unix: **disponibile su:** http://www.rulequest.com/Personal/c4.5r8.tar.gz
- **Tutorial su c4.5:** http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html

Costruzione dell'albero: algoritmo base

Algoritmo base, usato in CLS

- **T**: insieme degli esempi (training set)
- **$\{C_1, C_2, \ldots, C_k\}$**: insieme delle classi;
- Considera l'insieme T:
  - **if** T contiene uno o più esempi, tutti appartenenti alla medesima classe **then**
    - singola foglia con etichetta la classe
  - **if** T contiene casi che appartengono a più classi **then**
    - partizionamento di T in piu' sottoinsiemi secondo un test su un attributo: nodo associato al test, con un sottoalbero per ogni possibile risultato del test stesso
    - richiama l'algoritmo su ogni ramo/sottoinsieme

Condizioni di terminazione

In realtà, C4.5 si ferma se:

- **T contiene uno o più esempi, tutti appartenenti alla medesima classe**
  → singola foglia con etichetta la classe
- **T non contiene nessun esempio** (insieme vuoto)
  → singola foglia con etichetta la classe piu' frequente nell'insieme padre
- **Nessun test riesce a generare almeno due insiemi con 2*MINOBJ esempi**
  → singola foglia con etichetta la classe più frequente (alcuni esempi saranno mal classificati)
- Altre condizioni...

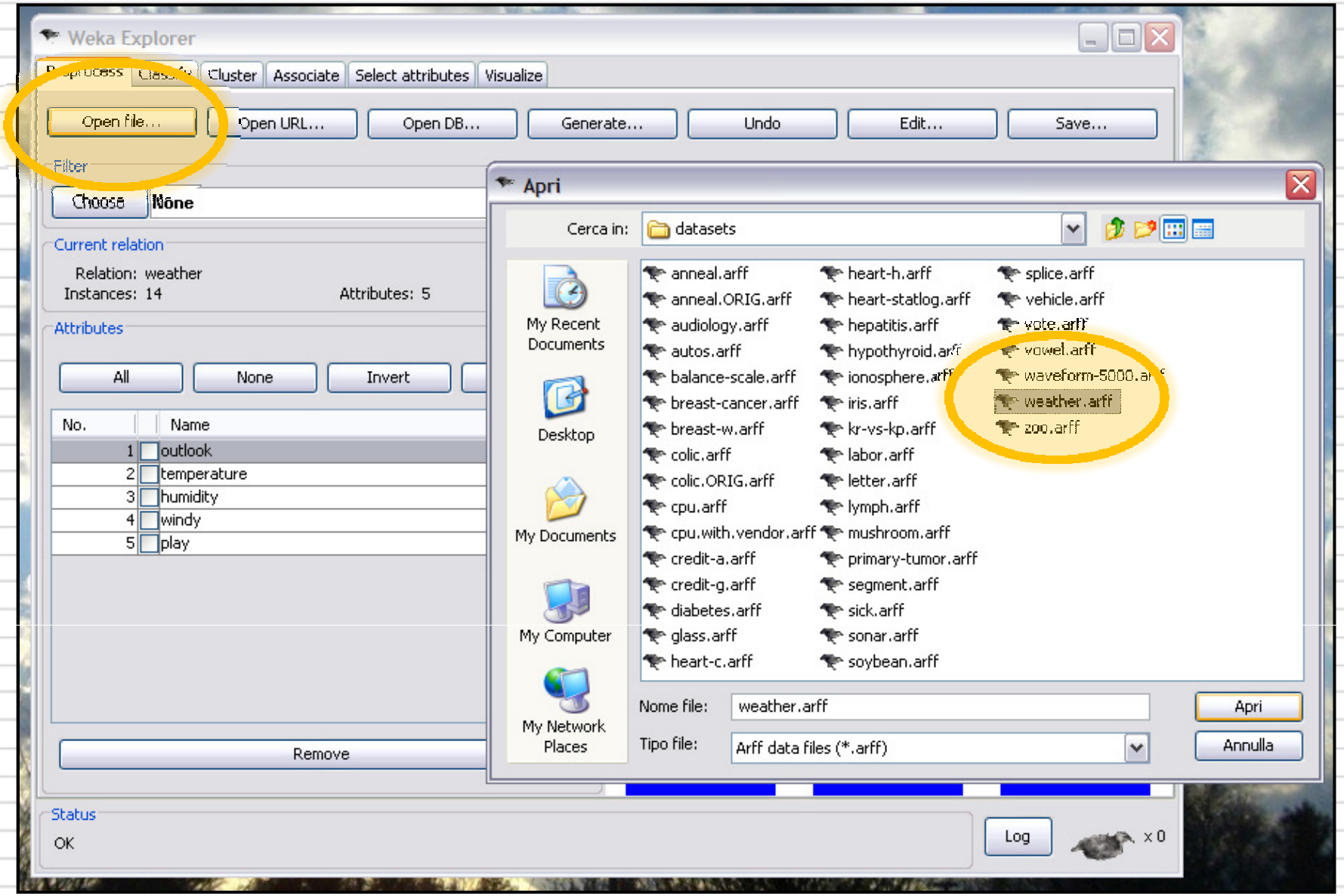
Esempio: golf

- **Istanze**: giornate libere
- **Concetti**: giornata adatta a giocare a golf e giornata non adatta a giocare a golf
- Descrizione degli esempi: **linguaggio attributo-valore**
  - `outlook`, con valori {sunny,overcast,rain}
  - `temperature`, con valori numerici
  - `humidity`, con valori numerici
  - `windy`, con valori {true, false}

All'opera!

1. Effettuate il login con l'utente temporaneo lab2_XX
2. C4.5 è implementato nella suite di strumenti per data mining "WEKA", installata sulle macchine
3. Avviate WEKA
4. Dal menu "Applications" avviate "Explorer"
5. Dal sito del corso, scaricate l'archivio con i datasets ed estraetelo
6. Dalla pagina "preprocess" di WEKA, aprire il file "weather.arff"

All'opera!


Weka Explorer
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Current relation
Relation: weather
Instances: 14
Attributes: 5
Attributes
All
None
Invert
No.
Name
1 outlook
2 temperature
3 humidity
4 windy
5 play
Remove
Status
OK
Log
x 0
Apri
Cerca in:
datasets
My Recent Documents
Desktop
My Documents
My Computer
My Network Places
anneal.arff
anneal.ORIG.arff
audiology.arff
autos.arff
balance-scale.arff
breast-cancer.arff
breast-w.arff
colic.arff
colic.ORIG.arff
cpu.arff
cpu.with.vendor.arff
credit-a.arff
credit-g.arff
diabetes.arff
glass.arff
heart-c.arff
heart-h.arff
heart-statlog.arff
hepatitis.arff
hypothyroid.arff
ionosphere.arff
iris.arff
kr-vs-kp.arff
labor.arff
letter.arff
lymph.arff
mushroom.arff
primary-tumor.arff
segment.arff
sick.arff
sonar.arff
soybean.arff
splice.arff
vehicle.arff
vote.arff
vowel.arff
waveform-5000.ar
weather.arff
zoo.arff
Nome file:
weather.arff
Tipo file:
Arff data files (*.arff)
Apri
Annulla

Cosa c'è in un file .arff?

1. Il nome della "relazione" (con il significato che ha in algebra relazionale)

```
@relation weather
```

2. Gli attributi (l'ultimo è la classe – di default)

```
@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
```

3. I dati (questi dovremmo conoscerli...)

```
@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
...
```

Cosa c'è in un file .arff?

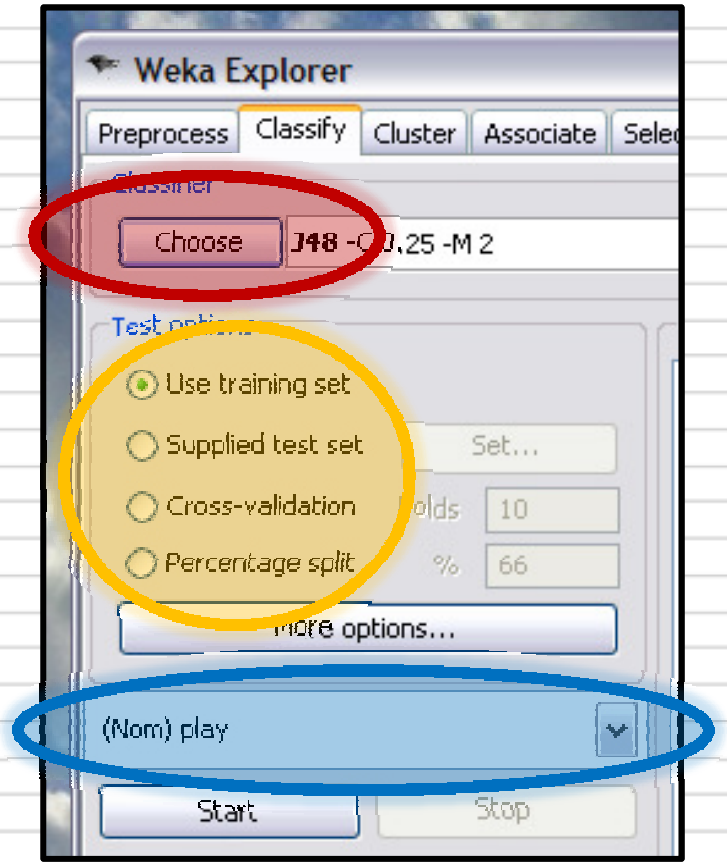
No	Outlook	Temp (°F)	Humid (%)	Windy	Class
D1	sunny	75	70	T	Play
D2	sunny	80	90	T	Don't Play
D3	sunny	85	85	F	Don't Play
D4	sunny	72	95	F	Don't Play
D5	sunny	69	70	F	Play
D6	overcast	72	90	T	Play
D7	overcast	83	78	F	Play
D8	overcast	64	65	T	Play
D9	overcast	81	75	F	Play
D10	rain	71	80	T	Don't Play
D11	rain	65	70	T	Don't Play
D12	rain	75	80	F	Play
D13	rain	68	80	F	Play
D14	rain	70	96	F	Play

Histograms

1. Il riquadro in basso a destra mostra mediante un istogramma la distribuzione delle classi a seconda dell'attributo selezionato
   - `Attributi discreti: distribuzione per ogni valore`
   - `Attributi continui: dominio diviso in "bins", distribuzione per ogni bin`
2. Le associazioni classe-colore si possono inferire selezionando come attributo la classe stessa
3. "Visualize all" visualizza gli istogrammi per tutti gli attributi

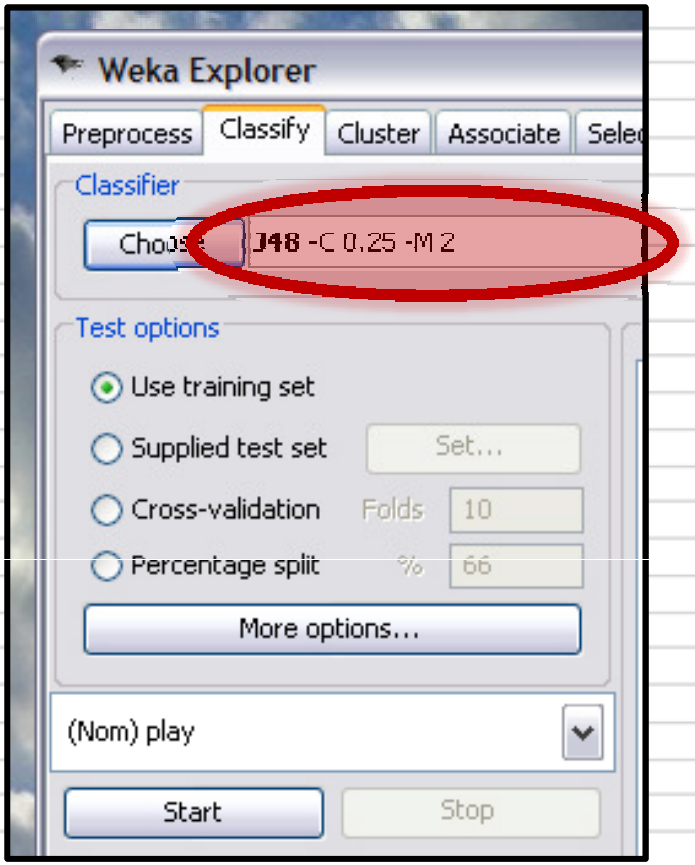
Classificazione

1. Nella pagina “Classify” possiamo scegliere:
   - Che classificatore utilizzare
   - Come effettuare il testing
   - Quale attributo considerare come classe
2. Come classificatore utilizzare *J4.8*
3. Effettuare il testing sul *training set*
4. Come classe scegliere *“play”*



Classificazione

1. Facendo click sul nome del classificatore si può accedere ad altre opzioni:
   - **`binarySplits`**: use binary splits on nominal attributes
   - **`confidenceFactor`**: the confidence factor used for pruning (smaller values incur more pruning).
   - **`minNumObj`**: the minimum number of instances per leaf.
   - **`saveInstanceData`**: save the training data for visualization.
   - **`Unpruned`**: no pruning is performed.
2. Provate ad avviare la classificazione



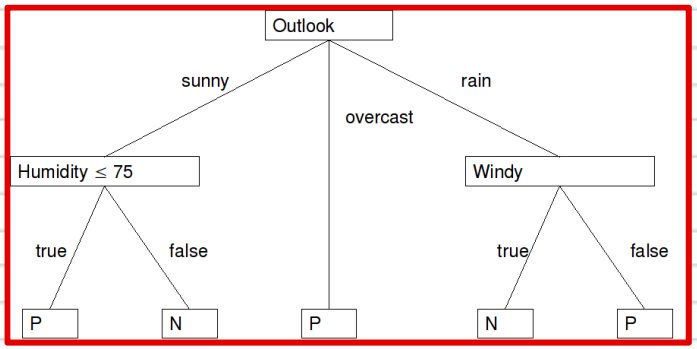
Output (1)

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    evaluate on training data
```

Output (2)

```
J48 pruned tree
------------------

outlook = sunny
|   humidity <= 75: yes (2.0)
|   humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)
```

Questo è l'albero decisionale che conosciamo...



Che si può visualizzare!
Click destro sulla lista dei risultati, poi “vizualize tree”

Output (3)

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          14
  100      %
Incorrectly Classified Instances         0
  0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0       %
Root relative squared error              0       %
Total Number of Instances               14
```

... e questi sono I risultati della valutazione del classificatore sul training set: nessun errore è commesso

Output (4)

```
=== Confusion Matrix ===

 a b   <-- classified as
 9 0 | a = yes
 0 5 | b = no
```

"Falsi negativi"

"Falsi positivi"

$$\text{Recall} = \frac{\text{True Positive}}{\#\,\text{positive instances}} = \frac{TP}{FN + TP}$$

$$\text{Precision} = \frac{\text{True positive}}{\#\,\text{classified positive}} = \frac{TP}{TP + FP}$$

Output (5)

1. Nella pagina “Visualize” potete vedere dei grafici di dispersione che mostrano come si dispongono le istanze rispetto ai valori di coppie di attributi
2. Fate click su un grafico
   - Potete selezionare che attributo mettere sull’asse delle X e quale sull’asse delle Y
   - Introdurre del rumore (jitter) serve a “separare” istanze con i medesini valori per una coppia di attributi
   - Con “select instance” potete selezionare solo alcuni punti
3. **ESERCIZIO:** Visivamente, quale attributo produce il miglior partizionamento delle istanze?

Attributi incerti

1. Fate un backup del file “weather.arff”
2. Poi provate a sostituire

   `overcast,72,90,TRUE,yes` ---> `?,72,90,TRUE,yes`
3. Notate che l’albero appreso è (più o meno) quello visto a lezione
4. Come varia la matrice di confusione?
5. **ESERCIZIO:** Classificazione di nuove istanze
   - Fate una copia di weather.arff
   - Eliminate tutte le istanze, introducetene una artificiale
   - Fornite il nuovo file come “test set” (“supplied test set...”, a sinistra)
   - Come vengono classificate le nuove istanze?
   - Come si comporta il classificatore con attributi sconosciuti?

Error Based Pruning (1)

- c4.5 utilizza una tecnica chiamata error-based pruning
- Prima di salvarlo, l'algoritmo tenta di semplificare l'albero
- Per ogni foglia C4.5 stima una probabilità di classificazione erronea **su esempi sconosciuti** basandosi su alcune considerazioni statistiche
- Per ogni nodo la probabilità di errore è la somma di quella dei sottonodi/foglie

Ad ogni nodo:

- Se la stima di errore può essere diminuita sostituendo quel nodo con una foglia, C4.5 lo fa
- Se la stima di errore può essere diminuita sostituendo quel nodo con il suo ramo più “pesante”, C4.5 lo fa

Error Based Pruning (2)

1. Fate due copie del file “labor.arff”
2. Nella prima copia mantenete le prime 40 istanze (training set)
3. Nella seconda copia mantenete le restanti 17 istanze (test set)
4. Provate ad apprende un albero **con e senza error based pruning**
   - Come sono i due alberi?
   - Che prestazioni hanno sul training set?
   - E sul test set?

Ricordate che obiettivo del pruning è migliorare le performance su casi sconosciuti!

ESERCIZI

Provate a dare un'occhiata ad alcuni data sets...

1. Nell'archivio che avete scaricato ce ne sono alcuni
   - Mushrooms.arff: distinguere funghi commestibili ("edible") e velenosi ("poisonous").
   - Iris.arff: distinguere varietà del fiore iris (pare sia un esempio classico in machine learning...)
   - Sick.arff: distinguere pazienti malati e sani in base all'esito di alcune analisi
2. Alcuni repository sono disponibili on-line:
   - http://www.cs.waikato.ac.nz/ml/weka/index_datasets.html
   - http://www.ics.uci.edu/~mlearn/MLSummary.html