Lezione 4

Decision Trees

Giovedì, 25 gennaio 2007

Giuseppe Manco

Riferimenti:
Chapter 3, Mitchell
Section 5.2 Hand, Mannila, Smith
Section 7.3 Han, Kamber
Chapter 6 Witten, Frank

Outline

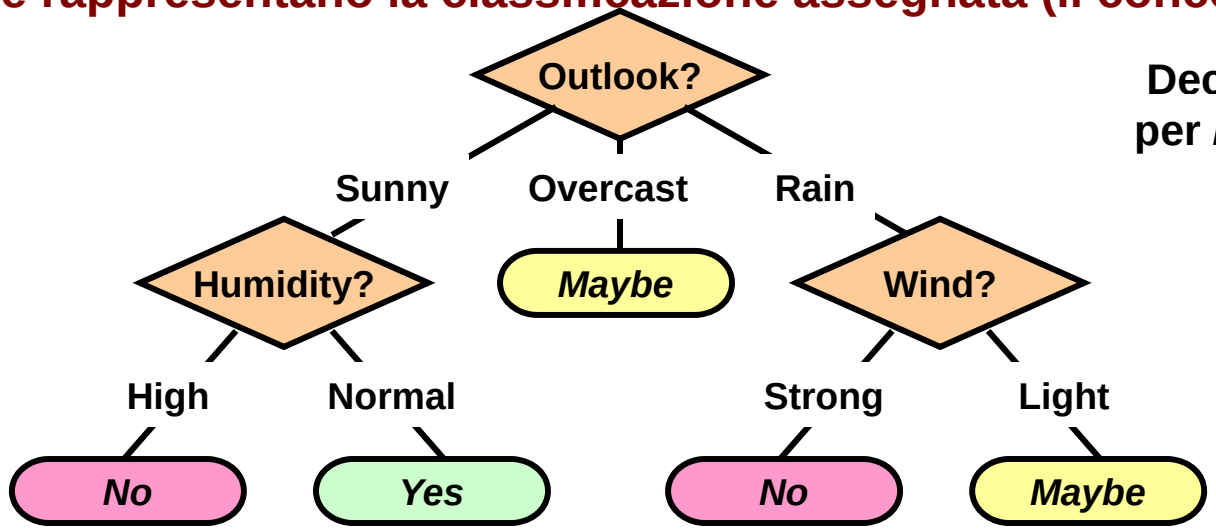
- **Alberi di decisione**
  - **Esempi**
  - **Modelli: quando usarli**
- **Entropia e Information Gain**
- ***L'algoritmo C45***
  - **Top-down induction of decision trees**
    - **Calcolo della riduzione di entropia (information gain)**
    - **Utilizzo dell'information Gain nella costruzione dell'albero**
  - **Spazio di ricerca, Inductive bias in *ID3/C45***
- **Metodi alternativi**
  - **L'algoritmo CART**
  - **L'algoritmo CHAID**
  - **Confronto tra Decision Tree Classifiers**

L’esempio classico: play tennis?

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

Alberi di decisione

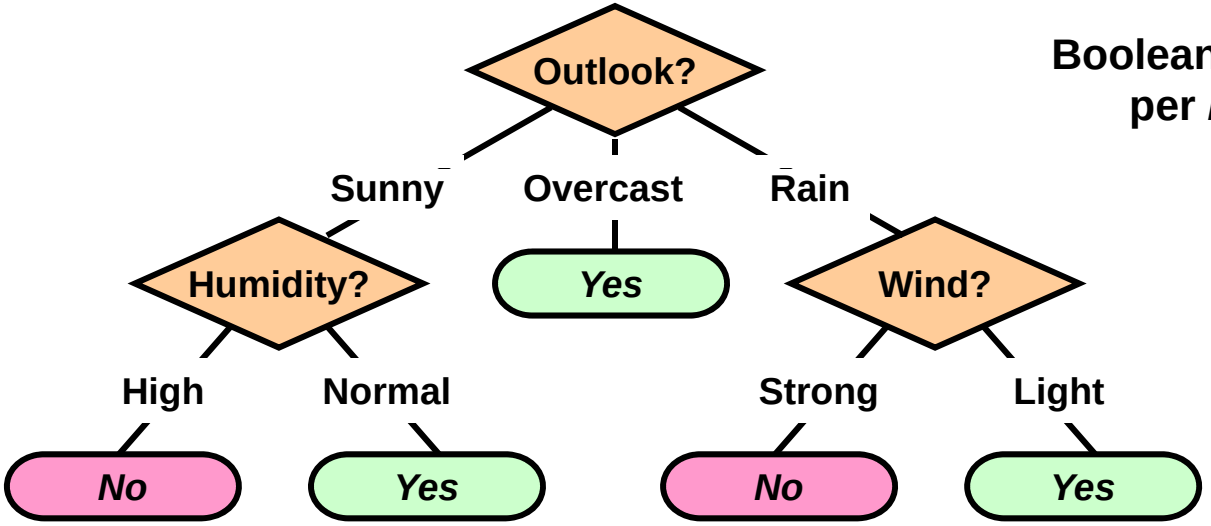
- **Classificatori**
  - Le istanze (esempi non etichettati) sono rappresentati come vettori di attributi ("features")
- **I nodi interni sono test per i valori di attributi**
  - Tipicamente: test di eguqglianza (Esempio: "Wind = ?")
  - diseguaglianza, ogni altro test possibile
- **I rami (cammini) rappresentano valori di attributi**
  - Corrispondenza uno-ad-uno (esempio: "Wind = Strong", "Wind = Light")
- **Le foglie rappresentano la classificazione assegnata (il concetto appreso)**



Decision Tree per *PlayTennis*

Boolean Decision Trees

- **Funzioni booleane**
  - **Potere espressivo: l'insieme universo (ovvero, possono esprimere qualsiasi funzione booleana)**
  - **D: Perché?**
    - **R: Possono essere riscritti sotto forma di regole in Forma Normale Disgiuntiva (DNF)**
    - **Esempio: (*Sunny* $\wedge$ *Normal-Humidity*) $\vee$ *Overcast* $\vee$ (*Rain* $\wedge$ *Light-Wind*)**



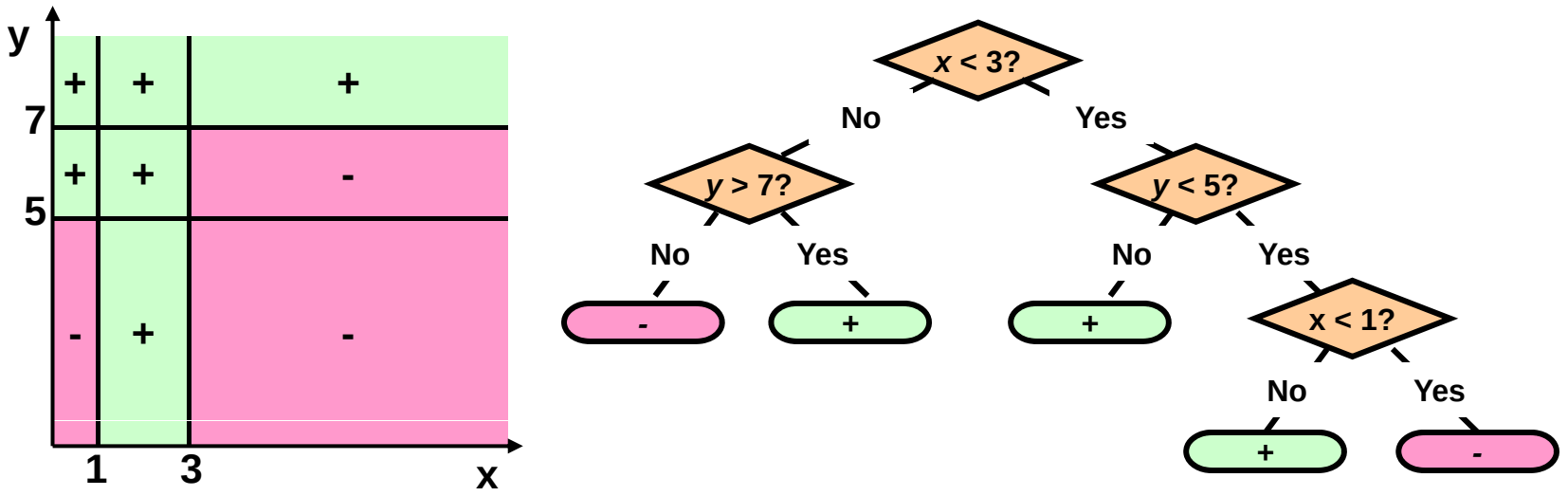
Boolean Decision Tree per *PlayTennis*

Quando possono essere usati gli alberi di decisione

- La funzione target è discreta
- Sono necessarie ipotesi disgiuntive
- L'insieme di training contiene rumore
- Esempi
  - Diagnosi medica
  - Risk analysis
    - Credito, prestiti
    - Assicurazioni
    - Frodi

Alberi di decisione e decision boundaries

- **Le istanze sono di solito rappresentate utilizzando attributi discreti**
  - **Valori tipici: Nominale/categorico ({red, yellow, green})**
  - **Valori numerici**
    - **Discretizzazione**
    - **Utilizzo di thresholds per I nodi di split**
- **In pratica, lo spazio delle istanze si divide in rettangoli paralleli agli assi**

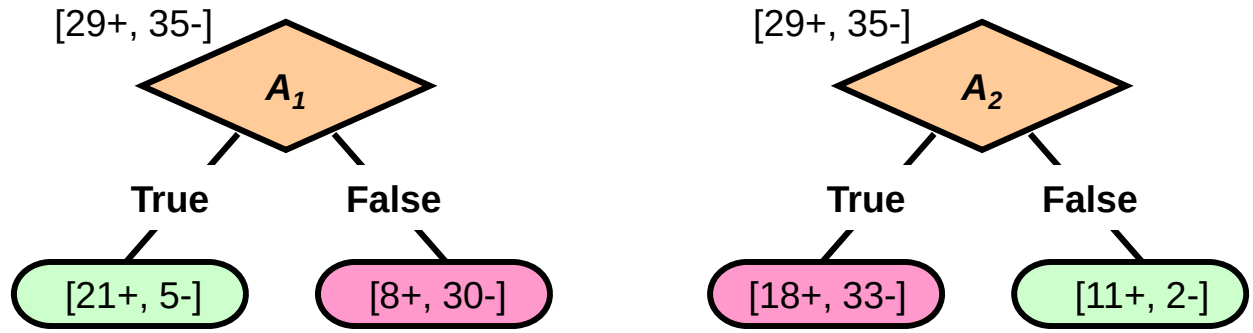


Decision Tree Learning: Top-Down Induction

- **Algorithm *Build-DT* (*D*, *Attributi*)**

  IF tutti gli esempi hanno la stessa etichetta
  THEN RETURN (nodo foglia con etichetta)
  ELSE
  IF *Attributi* = Ø
  THEN RETURN (foglia con etichetta di maggioranza)
  ELSE
  scegli il migliore attributo *A* come radice
  FOR EACH valore *v* di *A*
  Crea una diramazione dalla radice con la condizione *A* = *v*
  IF {*x* ∈ *D*: *x.A* = *v*} = Ø
  THEN RETURN (foglia con etichetta di maggioranza)
  ELSE *Build-DT* ({*x* ∈ *D*: *x.A* = *v*}, *Attributi* ~ {A})

- **Quale attributo è il migliore?**



La scelta del “migliore” attributo

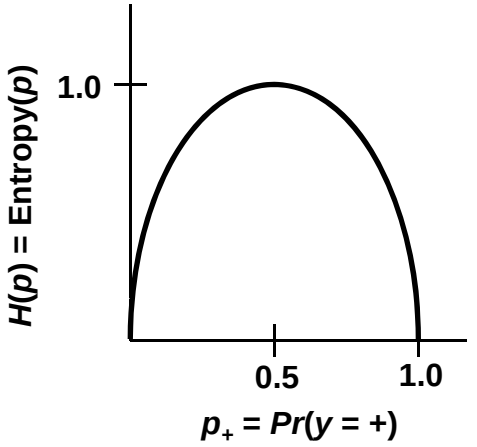
- **Obiettivo**
  - Costruzione di un albero che sia il più compatto possibile (Occam’s Razor)
  - Sotto il vincolo della consistenza con le etichette del training set
- **Ostacoli**
  - Trovare l’ipotesi consistente *minimale* (= il decision tree minimale) è NP-hard
  - Algoritmo ricorsivo (*Build-DT*)
    - Strategia divide-et-impera
    - Una strategia euristica greedy
    - Non garantisce l’ottimalità: può convergere ad un minimo locale
- **Decisione principale: l’attributo da scegliere**
  - Desiderata: attributi che splittano gli esempi in insiemi che sono relativamente “puri”
    - Che devono portare più rapidamente possibile ad un nodo foglia

Criteri per trovare il migliore split

- **Information gain (ID3 – C4.5)**
  - **Entropia, un concetto basato sulla teoria dell'informazione**
    - **Misura l'impurità di uno split**
    - **Seleziona l'attributo che massimizza la riduzione di entropia**
- **Gini index (CART)**
  - **Seleziona l'attributo che minimizza la varianza**
- **Statistica del $\chi 2$ su tabelle di contingenza (CHAID)**
  - **Misura la correlazione tra un attributo e l'etichetta di classe**
  - **Seleziona l'attributo con la massima correlazione**

Entropia: Nozione intuitiva

- **Una misura dell'incertezza**
  - La quantità
    - Purezza: quanto un insieme di istanze è prossimo alla situazione "ideale" (una sola etichetta)
    - Impurità (disordine): quanto siamo vicini all'incertezza totale (tutte le etichette distinte)
  - La misura: Entropia
    - Direttamente proporzionale a impurità, incertezza, irregolarità, sorpresa
    - Inversamente proporzionale alla purezza, certezza, regolarità, ridondanza
- **Esempio**
  - Si assuma $\mathbf{H} = \{0, 1\}$, distribuita in accordo a $Pr(y)$
    - consideriamo (almeno 2) etichette discrete
    - Per etichette continue: entropia differenziale
  - Entropia ottimale per y: uno dei due casi
    - $Pr(y = 0) = 1$, $Pr(y = 1) = 0$
    - $Pr(y = 1) = 1$, $Pr(y = 0) = 0$
  - Qual'è la distribuzione di probabilità meno pura?
    - $Pr(y = 0) = 0.5$, $Pr(y = 1) = 0.5$
    - Corrisponde alla massima incertezza
    - Una funzione concava



*Claude Shannon

Born: 30 April 1916

Died: 23 February 2001

"Il padre dell'information theory"

Claude Shannon, who has died aged 84, perhaps more than anyone laid the groundwork for today's digital revolution. His exposition of information theory, stating that all information could be represented mathematically as a succession of noughts and ones, facilitated the digital manipulation of data without which today's information society would be unthinkable.

Shannon's master's thesis, obtained in 1940 at MIT, demonstrated that problem solving could be achieved by manipulating the symbols 0 and 1 in a process that could be carried out automatically with electrical circuitry. That dissertation has been hailed as one of the most significant master's theses of the 20th century. Eight years later, Shannon published another landmark paper, *A Mathematical Theory of Communication*, generally taken as his most important scientific contribution.

Shannon applied the same radical approach to cryptography research, in which he later became a consultant to the US government.

Many of Shannon's pioneering insights were developed before they could be applied in practical form. He was truly a remarkable man, yet unknown to most of the world.

Entropia: Definizione Information-theoretic

- **Componenti**
  - ***D*: un insieme di esempu $\{\langle x_1, c(x_1)\rangle, \langle x_2, c(x_2)\rangle, \ldots, \langle x_m, c(x_m)\rangle\}$**
  - **$p_+ = Pr(c(x) = +)$, $p_- = Pr(c(x) = -)$**
- **Definizione**
  - ***H* è definita su una funzione di probabilità *p***
  - ***D* contiene esempi in cui la frequenza delle etichette + e – denota p_+ e p_-**
  - **L'entropia di *D* su c è:**

$$H(D) \equiv -p_+ \log_b(p_+) - p_- \log_b(p_-)$$

- **Qual'è l'unità di misura di H?**
  - **Dipende dalla base b del logaritmo (bits per $b = 2$, naturali for $b = e$, etc.)**
  - **Un singolo bit è richiesto per codificare ogni esempio nel caso peggiore ($p_+ = 0.5$)**
  - **Se c'è meno incertezza (ad esempio, $p_+ = 0.8$), possiamo utilizzare meno di un bit per ciascun esempio**

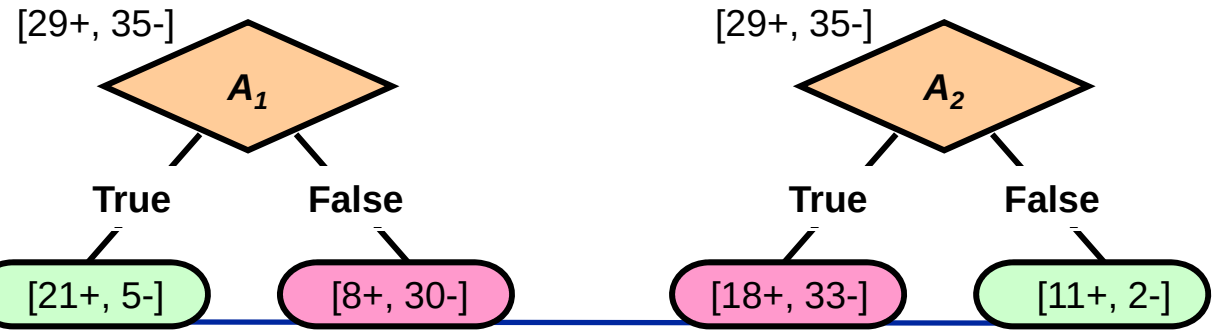
Information Gain

- **Partizionamento sui valori degli attributi**
  - Una partizione di *D* è una collezione di insiemi disgiunti la cui unione è *D*
  - Obiettivo: misurare quanto l'incertezza diminuisce se utilizziamo un attributo A come criterio di split
- **Definizione**
  - L'information gain of *D* relativamente all'attributo A è la riduzione di entropia dovuta allo splitting su A:

$$Gain(D, A) \equiv -H(D) - \sum_{v \in values(A)} \left[\frac{|D_v|}{|D|} \bullet H(D_v) \right]$$

dove D_v è $\{x \in D: x.A = v\}$

- **Qual'è l'attributo migliore?**



Un esempio illustrativo

- **Training set per *PlayTennis***

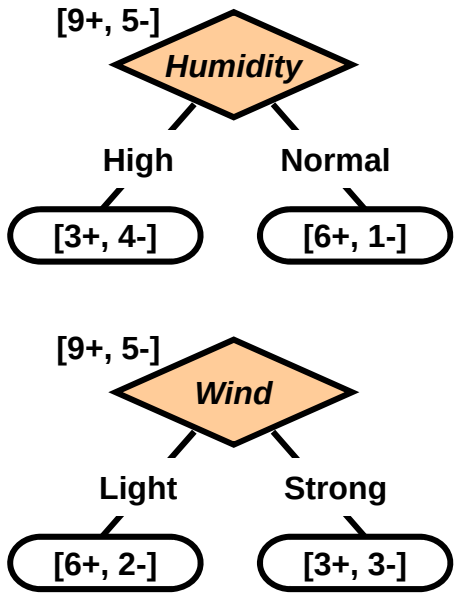
Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

- ***ID3*** ≡ ***Build-DT*** **utilizzando** ***Gain***(•)
- **Come viene costruito un albero con *ID3*?**

Decision Tree per *PlayTennis* con *ID3* [1]

- **Selezioniamo l'attributo radice**

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No



- **Distribuzione "a priori" (non condizionata) : 9+, 5-**
  - *H*(*D*) = -(9/14) lg (9/14) - (5/14) lg (5/14) *bits* = 0.94 bits
  - *H*(*D*, *Humidity* = High) = -(3/7) lg (3/7) - (4/7) lg (4/7) = 0.985 bits
  - *H*(*D*, *Humidity* = Normal) = -(6/7) lg (6/7) - (1/7) lg (1/7) = 0.592 bits
  - *Gain*(*D*, *Humidity*) = 0.94 - (7/14) * 0.985 + (7/14) * 0.592 = 0.151 bits
  - Analogamente, *Gain* (*D*, *Wind*) = 0.94 - (8/14) * 0.811 + (6/14) * 1.0 = 0.048 bits

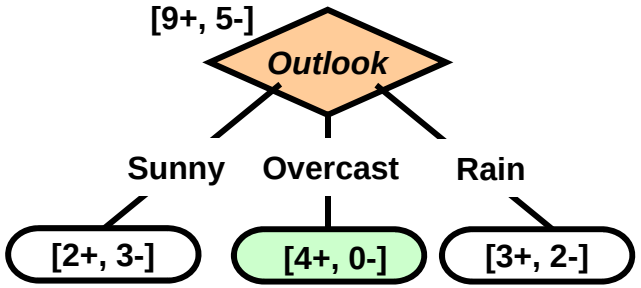
$$Gain(D,A) \equiv -H(D) - \sum_{v \in values(A)} \left[\frac{|D_v|}{|D|} \bullet H(D_v) \right]$$

Decision Tree per *PlayTennis* con *ID3* [2]

- **Selezione del nodo radice**

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

  - *Gain*(*D*, *Humidity*) = 0.151 bits
  - *Gain*(*D*, *Wind*) = 0.048 bits
  - *Gain*(*D*, *Temperature*) = 0.029 bits
  - *Gain*(*D*, *Outlook*) = 0.246 bits



- **Selezione del prossimo nodo (la radice del sottoalbero)**
  - Continua fino a quando ogni esempio è incluso nei cammini o la purezza è del 100%
  - Che significa purezza = 100%?
  - Possiamo avere *Gain*(*D*, *A*) < 0?

Decision Tree per *PlayTennis con ID3* [3]

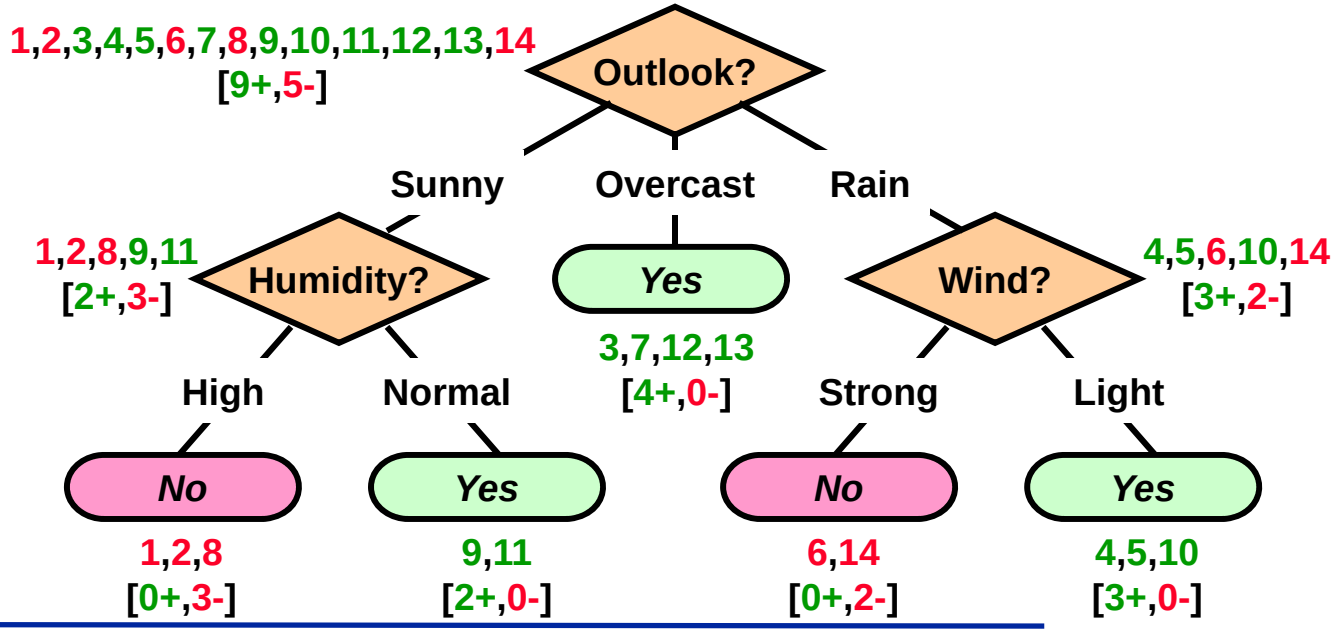
- **Selezione del prossimo attributo (la radice del sottoalbero)**

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis?*
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

  - convenzione: lg (0/a) = 0
  - $Gain(D_{Sunny}, Humidity) = 0.97 - (3/5) * 0 - (2/5) * 0 = \underline{0.97 \text{ bits}}$
  - $Gain(D_{Sunny}, Wind) = 0.97 - (2/5) * 1 - (3/5) * 0.92 = 0.02$ bits
  - $Gain(D_{Sunny}, Temperature) = 0.57$ bits
- **Induzione top-down**
  - Per gli attributi discreti, termina in $O(n)$ splits
  - Effettua al più un passo sui dati ad ogni livello (perché?)

Decision Tree per *PlayTennis* con *ID3* [4]

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No



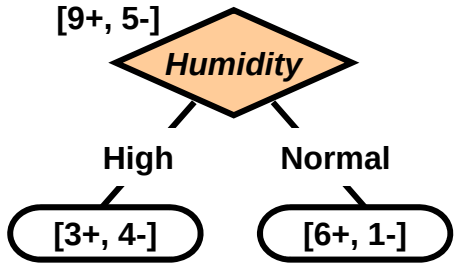
CHAID: idea di base

- **Componenti**
  - ***D*: un insieme di esempi {<x_1, $c(x_1)$>, <x_2, $c(x_2)$>, …, < x_m, $c(x_m)$>}**
  - **$p_+ = Pr(c(x) = +)$, $p_- = Pr(c(x) = -)$**
  - ***H(.): funzione di valutazione degli attributi***
- **Definizione**
  - **H è definita sul test del Chi-Quadro**
  - **Per ogni attributo X, si calcola la tabella di contingenza rispetto alla classe C**
    - **Calcoliamo il χ^2-value e il corrispondente p-value p_X**
  - **Si seleziona l'attributo che ha il più piccolo p-value e lo si confronta con un α_{split}-value**
    - Se $p_X < \alpha_{split}$, allora si utilizza X come attributo di split
    - Se $p_X > \alpha_{split}$, allora non c'è un attributo associato al nodo
      - Il nodo è una foglia e l'etichetta è quella di maggioranza
- **Cosa rappresenta H?**
  - **La misura della correlazione di c con l'attributo esaminato**

CHAID: esempio (α_{split} = 0.02)

- **Attributo radice:**

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis?*
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No



Humidity/PlayTennis	Yes	No	Totale
High	3	4	7
Normal	6	1	7
Totale	9	5	14

χ^2=2.5, d.f.=1 $p_{Humidity}$ = 0.0588

Wind/PlayTennis	Yes	No	Totale
Light	6	2	8
Strong	3	3	6
Totale	9	5	14

χ^2=0.9, d.f.=1 p_{Wind} = 0.2590

Outlook/PlayTennis	Yes	No	Totale
Sunny	2	3	5
Rain	4	0	4
Overcast	3	2	5
Totale	9	5	14

χ^2=3.5467, d.f.=2 $p_{Outlook}$ = 0.0849

Temperature/PlayTennis	Yes	No	Totale
Hot	2	2	4
Mild	4	2	6
Cool	3	1	4
Totale	9	5	14

χ^2=0.5704, d.f.=2 $p_{Outlook}$ = 0.3759

CHAID: raffinamenti

- **Se un attributo ha più di 2 valori, possiamo provare a raggruppare i valori**
  - **Il raggruppamento tende a mettere insieme valori omogenei rispetto alla classe**
    - **Situazione identica alla discretizzazione**
  - **Procedura:**
    1. **Se un attributo X ha più di 2 valori, trova la coppia di valori meno significativa (con p-value più alto) rispetto alla classe C**
       - Se $p > \alpha_{merge}$, allora raggruppa la coppia, e vai a 1
       - Se $p < \alpha_{merge}$, stop
    2. **Il p-value corrispondente agli attributi X modificati va aggiustato**
       - Per dargli una significatività statistica
       - c=numero di valori originari
       - r=numero di valori ottenuti

$$B_{free} = \sum_{i=0}^{r-1} (-1)^i \frac{(r-i)^c}{r!(r-i)!}$$

Esempio ($\alpha_{split} = 0.05$, $\alpha_{merge} = 0.05$)

Outlook/PlayTennis	Yes	No	Totale
Sunny	2	3	5
Rain	4	0	4
Overcast	3	2	5
Totale	9	5	14

χ^2=3.5467, d.f.=2 $p_{Outlook}$ = 0.0849

Outlook/PlayTennis	Yes	No	Totale
Sunny	2	3	5
Overcast	3	2	5
Totale	95	5	10

χ^2=4, d.f.=1 $p_{Outlook}$ =0.5164

Outlook/PlayTennis	Yes	No	Totale
Sunny	2	3	5
Rain	4	0	4
Totale	6	3	9

χ^2=3.6, d.f.=1 $p_{Outlook}$ =0.0348

Outlook/PlayTennis	Yes	No	Totale
Rain	4	0	4
Overcast	3	2	5
Totale	7	2	9

χ^2=2.0571, d.f.=1 $p_{Outlook}$ = 0.0997

Esempio ($\alpha_{split} = 0.05$, $\alpha_{merge} = 0.05$) [cont.]

Outlook/PlayTennis	Yes	No	Totale
Sunny	2	3	5
Rain	4	0	4
Overcast	3	2	5
Totale	9	5	14

χ^2=3.5467, d.f.=2 $p_{Outlook}$ = 0.0849

Outlook/PlayTennis	Yes	No	Totale
Sunny,Rain	6	3	9
Overcast	3	2	5
Totale	9	5	14

χ^2=0.0622, d.f.=1 $p_{Outlook}$ = 1.5503

Outlook/PlayTennis	Yes	No	Totale
Sunny	2	3	5
Rain,Overcast	7	2	9
Totale	9	5	14

χ^2=1.9980, d.f.=1 $p_{Outlook}$ =0.1039

Outlook/PlayTennis	Yes	No	Totale
Sunny,Overcast	5	5	10
Rain	4	0	4
Totale	9	5	14

χ^2=3.1111, d.f.=1 $p_{Outlook}$ =0.0477

CART – Classification And Regression Tree

- **Sviluppato nel periodo 1974-1984 da 4 statistici**
  - **Leo Breiman (Berkeley), Jerry Friedman (Stanford), Charles Stone (Berkeley), Richard Olshen (Stanford)**
- **Permette stime accurate quando i dati contengono rumore**

Gini index (Cart)

- **Componenti**
  - ***D*: un insieme di esempi {<x_1, $c(x_1)$>, <x_2, $c(x_2)$>, …, < x_m, $c(x_m)$>}**
  - **$p_+ = Pr(c(x) = +)$, $p_- = Pr(c(x) = -)$**
  - ***H(.): funzione di valutazione degli attributi***
- **Definizione**
  - ***H* è definita sulla funzione di probabilità *p***
  - ***D* contiene esempi in cui le frequenze delle etichette + e - sono p_+ and p_-**
  - **L'indice di gini su *D* relativo a *c* è:**

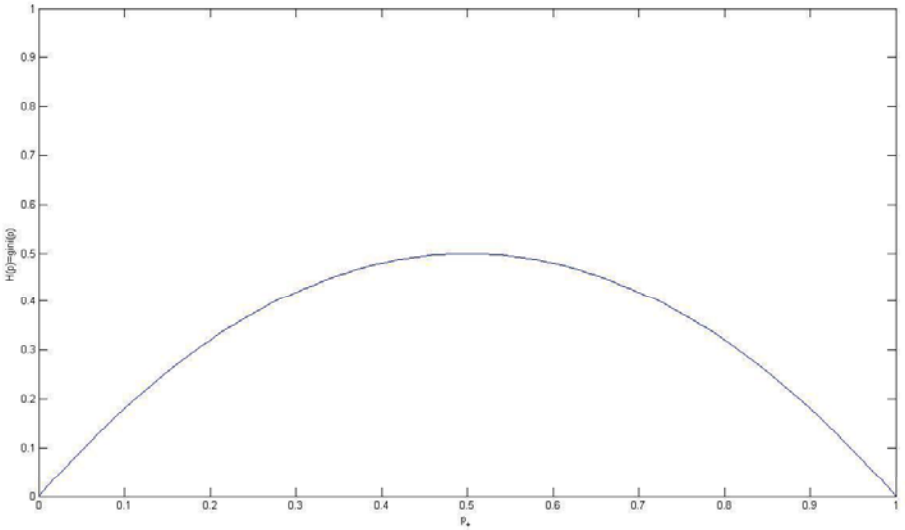
$$H(D) \equiv 1 - p_+^2 - p_-^2$$

- **Cosa rappresenta H?**
  - **La varianza di D**
  - Se il *D* è partizionato in D_1, D_2 allora

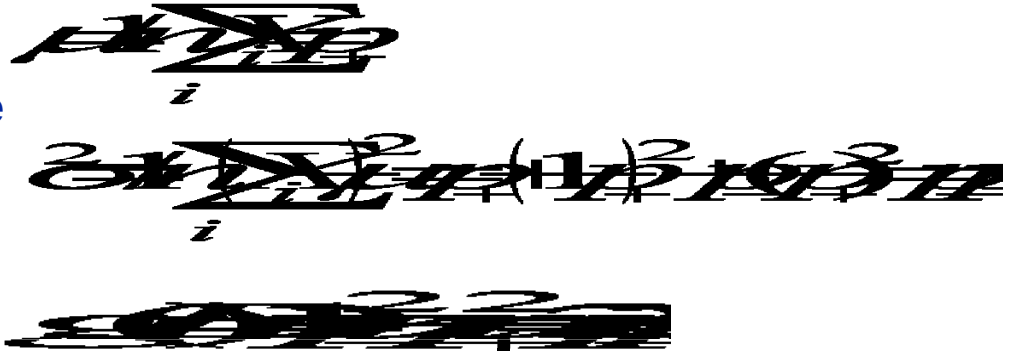
$$gini_{split}(D_1, D_2) = gini(D_1) \times \frac{|D_1|}{|D|} + gini(D_2) \times \frac{|D_2|}{|D|}$$

Gini index

- **Osservando che $p_+ = 1 - p_-$**



- *L'indice gini è massimo ai bordi e minimo al centro*
- **Interpretazione**
  - **In termini di varianza: ad ogni istanza è associata una variabile casuale X_i che può assumere valore 0/1 (in base alla classe)**
  - **L'indice di gini indica il tasso d'errore al nodo N se l'etichetta è selezionata casualmente dagli esempi di N**



Gini Index: generalizzazione

- **Se un dataset D contiene esempi da n classi, gini index, gini(D) è definito come**

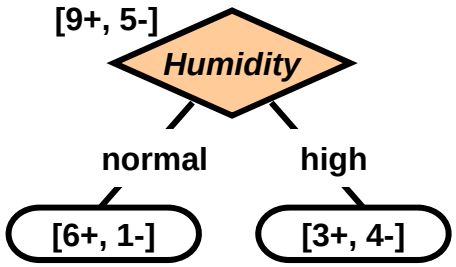
dove p_j è la frequenza relativa della classe j in D.

- **gini(D) è minimizzata se le classi di D sono molto sbilanciate.**
- **Split multipli:**

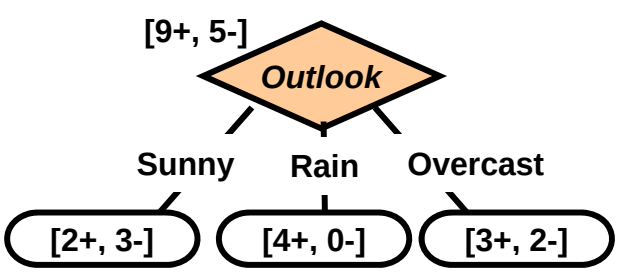
$$gini_{split}(D_1,...,D_k) = gini(D_1)\times\frac{|D_1|}{|D|}+...+gini(D_k)\times\frac{|D_k|}{|D|}$$

Gini index - Esempio

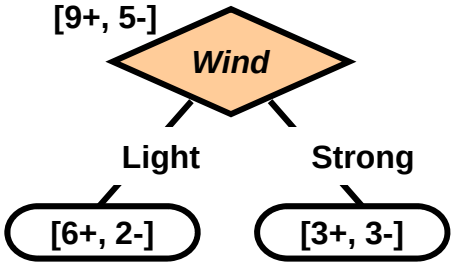
Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No



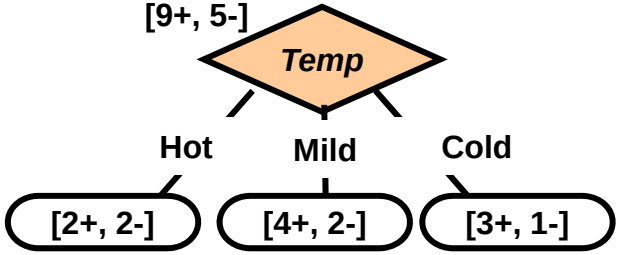
Gini(Hum)=0.0918



Gini(Outlook)=0.1163



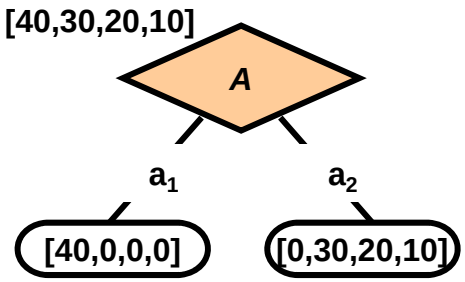
Gini(Hum)=0.0306

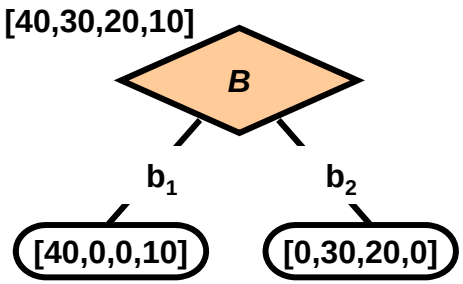


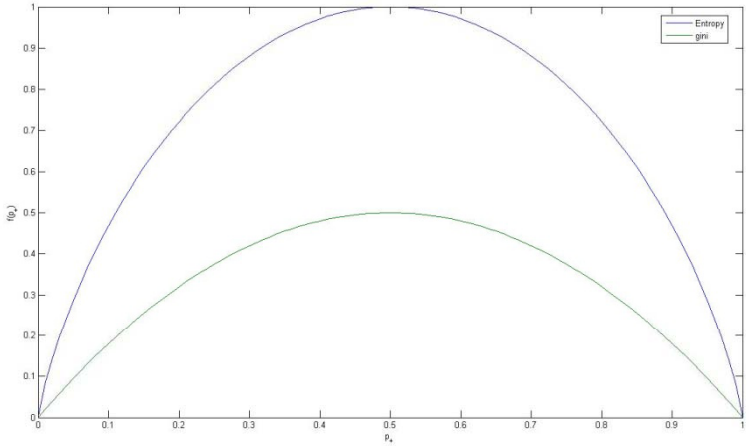
Gini(Temp)=0.0187

Entropia vs. Gini

- **Gini tende ad isolare la classe più grande da tutte le altre**
- **L'entropia tende a trovare gruppi di classi comunque bilanciate**



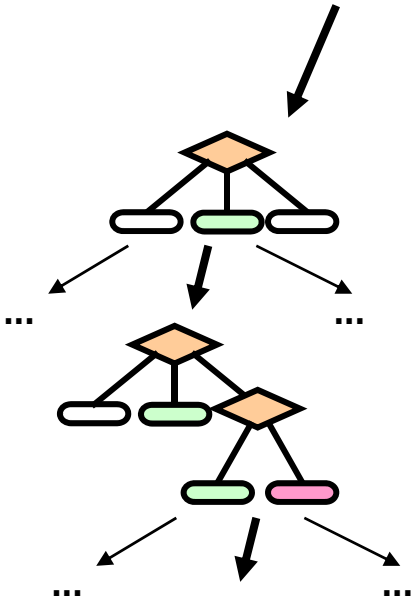




Esercizio: studiare la differenza di comportamento rispetto al test del Chi quadro

Lo spazio delle ipotesi nei *Decision Trees*

- **Problema di ricerca**
  - Lo spazio dei decision trees può rappresentare tutte le possibili funzioni discrete
    - Pros: espressività; flessibilità
    - Cons: complessità computazionale; alberi troppo grandi
  - Obiettivo: trovare il miglior decision tree (albero minimale consistente)
  - Ostacolo: trovare quest'albero è NP-hard
  - Tradeoff
    - Utilizzo di euristiche
    - Algoritmo greedy
    - *Strategia* hill-climbing senza backtracking
- **Statistical Learning**
  - Le decisioni basate su descrittori statistici
  - I dati possono essere ri-analizzati
  - Robusto al rumore



Inductive Bias in *ID3*

- **Heuristic : Search : Inductive Bias : Inductive Generalization**
  - ***H* è l'insieme di tutti i sottoinsiemi di *X***
  - **⇒ Obiettivo? Non proprio…**
    - **Preferenza per gli alberi piccoli**
    - **Preferenza per gli alberi con un information gain alto nei nodi vicini alla radice**
    - ***Gain*(•): un'euristica che cattura il bias induttivo di *ID3***
  - **Bias in *ID3***
    - **La preferenza è codificata nella funzione H**
- **Preferenza per gli alberi piccoli**
  - **Occam's Razor bias: le ipotesi più semplici sono quelle che spiegano le osservazioni**

Estensioni

- **Assunzioni nell'algoritmo precedente**
  - **Output discreto**
    - **Valori reali in output sono possibili**
    - **Regression trees [Breiman *et al*, 1984]**
  - **Input discreto**
    - **Metodi di discretizzazione**
    - ***Disuguaglianze* invec che uguaglianze sui nodi**
- **Scalabilità**
  - **Critica in database mining su very large databases (VLDB)**
  - **Good news: esistono algoritmi efficienti per processare molte istanze**
  - **Bad news: molto inefficienti su molti *attributi***
- **Tolleranza**
  - **Dati con rumore (rumore di classificazione ≡ etichette non corrette; rumore di attributo ≡ dati inaccurati o imprecisi)**
  - **Valori mancanti**

Sommario

- **Decision Trees (DTs)**
  - **Possono essere booleani ($c(x) \in \{+, -\}$) o variare su classi multiple**
- **Algoritmo *Build-DT*: Induzione Top Down**
  - **Calcolo del migliore attributo su cui splittare**
  - **Partizionamento ricorsivo**
- **Entropia, Information Gain, gini, Chi-quadro**
  - **Obiettivo: misurare il livello di incertezza che si ha nello splittare su un attributo *A***
- **Build-DT come algoritmo di ricerca di uno spazio di ipotesi**