

ΕΜΠ ΔΠΜΣ

Εφαρμοσμένες Μαθηματικές Επιστήμες  
Αλγόριθμοι Εξόρυξης Πληροφορίας

Διάλεξη 05:

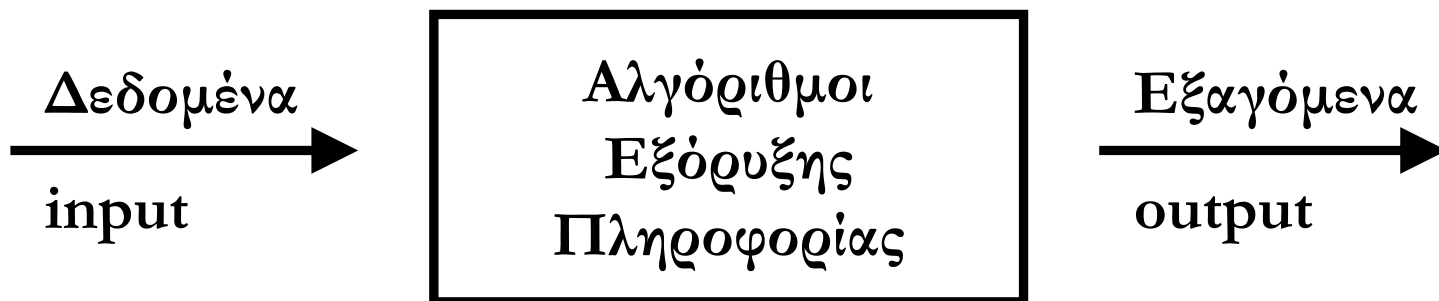
Αλγόριθμοι εκμάθησης

Μέρος Α

Δένδρα & Κανόνες



# Αλγόριθμοι



- Μέχρι τώρα μελετήθηκε το σύνολο των συνοδευτικών – προ και μετά – της μάθησης διαδικασιών
- Η κατανόησή τους επιτρέπει τη διείσδυση στον πυρήνα της εξόρυξης πληροφορίας, τους αλγορίθμους μάθησης



# Πρώτα τα βασιικά

- Πολύ συχνά, απλοί κανόνες επιτυγχάνουν μη αναμενόμενα υψηλή ακρίβεια
  - Αίτιο η απλοϊκή δομή που συχνά υποβόσκει πίσω από πραγματικά σύνολα δεδομένων
  - Κατά συνέπεια ένα και μόνο χαρακτηριστικό είναι συχνά ικανό να διακρίνει αποτελεσματικά την τάξη ενός παραδείγματος
- Σε κάθε περίπτωση, συνίσταται αρχικά ο πειραματισμός με τις πλέον απλές των μεθόδων
- Για το λόγο αυτό, παραθέτονται αρχικά οι κυριότερες αυτών



# Επαγωγή απλοϊκών κανόνων

- Συλλογισμός: Δημιουργία κανόνων ελέγχου ενός και μόνο χαρακτηριστικού
- 1R: αλγόριθμος εκμάθησης δένδρου απόφασης ενός επιπέδου
  - Δημιουργία κανόνων ελέγχου ενός ξεχωριστού χαρακτηριστικού
- Βασική έκδοση (*ονομαστικά χαρακτηριστικά*)
  - Ένας κλάδος για κάθε τιμή
  - Κάθε κλάδος εκχωρεί την τάξη με τη μεγαλύτερη συχνότητα
  - Τιμή σφάλματος: ποσοστό υποδειγμάτων που δεν ανήκουν στην πλειοψηφούσα τάξη του αντίστοιχου κλάδου
  - Επιλογή χαρακτηριστικού με την μικρότερη τιμή σφάλματος



# Ψευδοκώδικας 1R

For each attribute,

For each value of the attribute, make a rule as follows:

count how often each class appears

find the most frequent class

make the rule assign that class to this attribute-value

Calculate the error rate of the rules

Choose the rules with the smallest error rate

- Οι άγνωστες τιμές (missing values) θεωρούνται ως ξεχωριστή τιμή του χαρακτηριστικού
- Αριθμητικά χαρακτηριστικά: Διακριτοποίηση!
  - Με ελαχιστοποίηση του συνολικού σφάλματος
  - Αποφυγή υπερπροσαρμογής (overfitting) μέσω ενίσχυσης προτεραιότητας ελαχίστου αριθμού υποδειγμάτων πλειοψηφούσας τάξης ανά διάστημα



# 1R για το πρόβλημα καιρού

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Attribute	Rules	Errors	Total errors
Outlook	Sunny → No	2/5	4/14
	Overcast → Yes	0/4	
	Rainy → Yes	2/5	
Temp	Hot → No*	2/4	5/14
	Mild → Yes	2/6	
	Cool → Yes	1/4	
Humidity	High → No	3/7	4/14
	Normal → Yes	1/7	
Windy	False → Yes	2/8	5/14
	True → No*	3/6	

\* ισοπαλία



# Αριθμητικά χαρακτηριστικά με αποφυγή υπερπροσαρμογής

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86		
Rainy	75	80		
...	...	...		

Attribute	Rules	Errors	Total errors
Outlook	Sunny → No	2/5	4/14
	Overcast → Yes	0/4	
	Rainy → Yes	2/5	
Temperature	$\leq 77.5 \rightarrow$ Yes	3/10	5/14
	$> 77.5 \rightarrow$ No*	2/4	
Humidity	$\leq 82.5 \rightarrow$ Yes	1/7	3/14
	$> 82.5$ and $\leq 95.5 \rightarrow$ No	2/6	
	$> 95.5 \rightarrow$ Yes	0/1	
Windy	False → Yes	2/8	5/14
	True → No*	3/6	



# Σύνοψη περι 1R

- Η μέθοδος δημοσιεύτηκε το 1993 από τον Holte
  - Πειραματική αποτίμηση της μεθόδου σε 16 κλασσικά σύνολα δεδομένων (με χρήση CV)
  - Ο βέλτιστος ελάχιστος αριθμός υποδειγμάτων προέκυψε πειραματικά ίσος με 6
  - Η απόδοση των κανόνων της 1R ήταν σχεδόν ίση με εκείνη περισσότερο πολύπλοκων δένδρων απόφασης
- Η προσέγγιση 'Πρώτα τα βασικά' αποδίδει!
  - Αρχικά, προσδιορισμός τυπικής απόδοσης με χρήση απλοϊκών τεχνικών
  - Στη συνέχεια, πειραματισμός με περισσότερο εξεζητημένες μεθόδους εκμάθησης & σύγκριση της απόδοσης τους με την προηγούμενα καθορισμένη τυπική τιμή



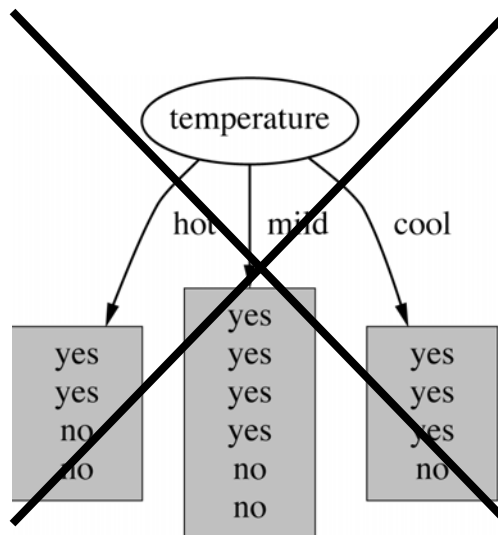
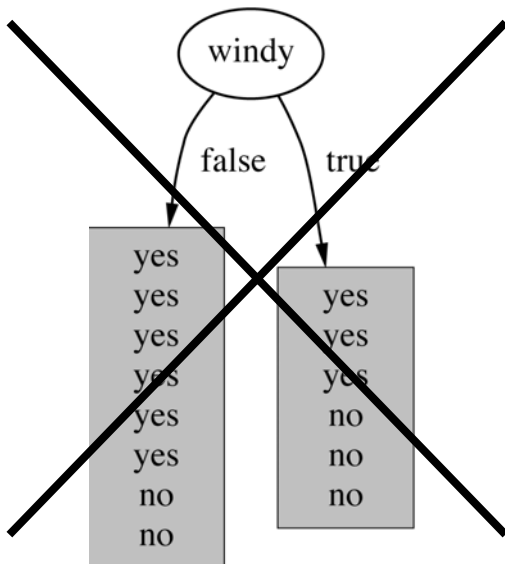
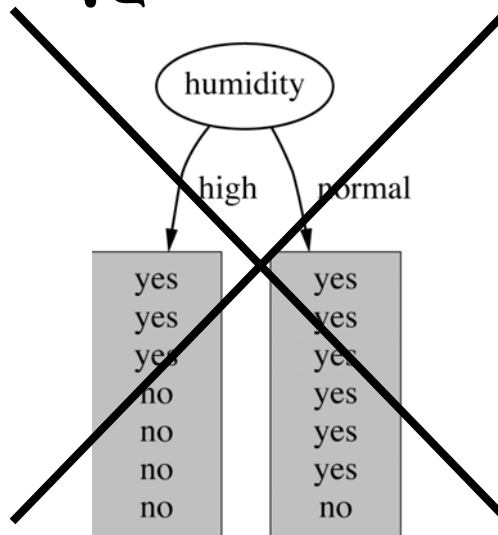
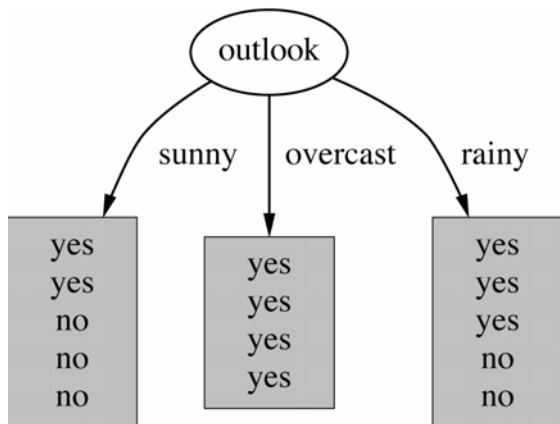


# Κατασκευή δένδρων απόφασης

- Στρατηγική: από πάνω προς τα κάτω (top down)  
Επαναληπτική υλοποίηση της μεθόδου *διαίρει & βασίλευε* (*divide-and-conquer*)
  - Επιλογή χαρακτηριστικού αρχικού κόμβου (ρίζας)  
Δημιουργία κλάδων για κάθε πιθανή τιμή του χαρακτηριστικού
  - Διάσπαση υποδειγμάτων σε υποσύνολα  
Ένα για κάθε κλάδο που εκτείνεται από τη ρίζα
  - Επανάληψη των παραπάνω για κάθε κλάδο με χρήση μόνο του υποσυνόλου των υποδειγμάτων κάθε κλάδου
- Ολοκλήρωση όταν όλα τα υποδείγματα ανήκουν στην ίδια τάξη



# Επιλογή χαρακτηριστικού





# Κριτήριο επιλογής χαρακτηριστικού

- Ποιο χαρακτηριστικό θα επιλεγεί?
  - Στόχος: δημιουργία μικρότερου δυνατού δένδρου
  - Μέθοδος: επιλογή χαρακτηριστικού που δημιουργεί τα περισσότερα ομοιογενή υποσύνολα κλάδων
- *Σύνηθες κριτήριο ανομοιογένειας: κέρδος πληροφορίας (διάλεξη 03)*
  - Το κέρδος πληροφορίας αυξάνεται με την αύξηση της μέσης ομοιογένειας των υποσυνόλων
- Μέθοδος: επιλογή χαρακτηριστικού που δίδει μέγιστο κέρδος πληροφορίας



# Παράδειγμα: χαρακτηριστικό *Outlook*

- *Outlook* = *Sunny* :  
 $\text{info}([2,3]) = \text{entropy}(2/5,3/5) = -2/5 \log(2/5) - 3/5 \log(3/5) = 0.971 \text{ bits}$
- *Outlook* = *Overcast* :  
 $\text{info}([4,0]) = \text{entropy}(1,0) = -1 \log(1) - 0 \log(0) = 0 \text{ bits}$       **Πρόβλημα: δεν ορίζεται!**
- *Outlook* = *Rainy* :  
 $\text{info}([3,2]) = \text{entropy}(3/5,2/5) = -3/5 \log(3/5) - 2/5 \log(2/5) = 0.971 \text{ bits}$
- Κέρδος πληροφορίας με την εισαγωγή κόμβου για το χαρακτηριστικό:  
 $\text{info}([3,2],[4,0],[3,2]) = (5/14) \times 0.971 + (4/14) \times 0 + (5/14) \times 0.971 = 0.693 \text{ bits}$



# Υπολογισμός κέρδους πληροφορίας

- Κέρδος πληροφορίας:

(απαιτούμενος για την περιγραφή των υποδειγμάτων)

όγκος πληροφορίας πριν τη διάσπαση –

– όγκος πληροφορίας μετά τη διάσπαση

$$\begin{aligned} \text{gain}(\textit{Outlook}) &= \text{info}([9,5]) - \text{info}([2,3],[4,0],[3,2]) \\ &= 0.940 - 0.693 \\ &= 0.247 \text{ bits} \end{aligned}$$

- Κέρδος πληροφορίας ανά χαρακτηριστικό στα δεδομένα καιρού:

$$\text{gain}(\textit{Outlook}) = 0.247 \text{ bits}$$

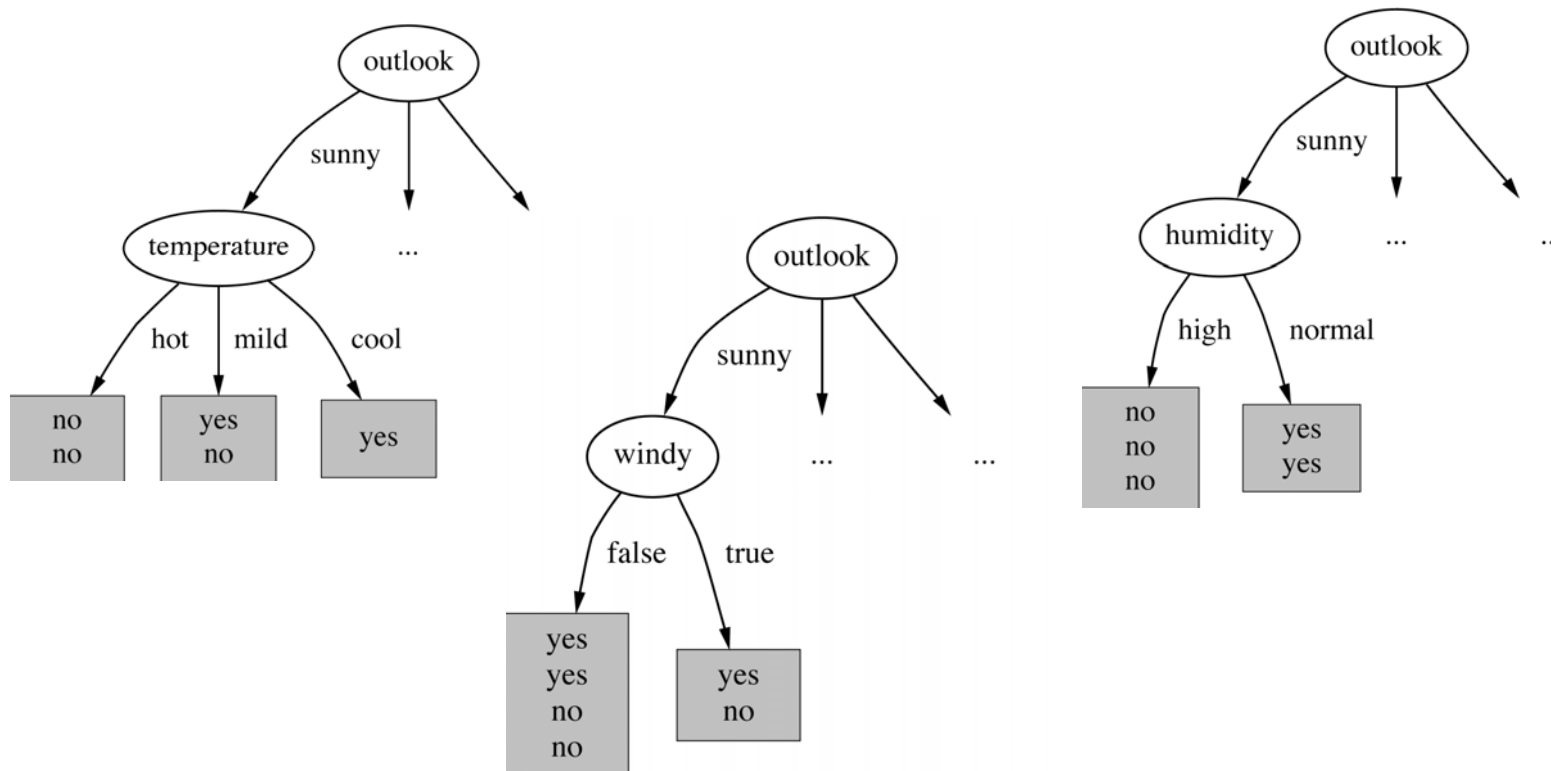
$$\text{gain}(\textit{Temperature}) = 0.029 \text{ bits}$$

$$\text{gain}(\textit{Humidity}) = 0.152 \text{ bits}$$

$$\text{gain}(\textit{Windy}) = 0.048 \text{ bits}$$



# Περαιτέρω διάσπαση...



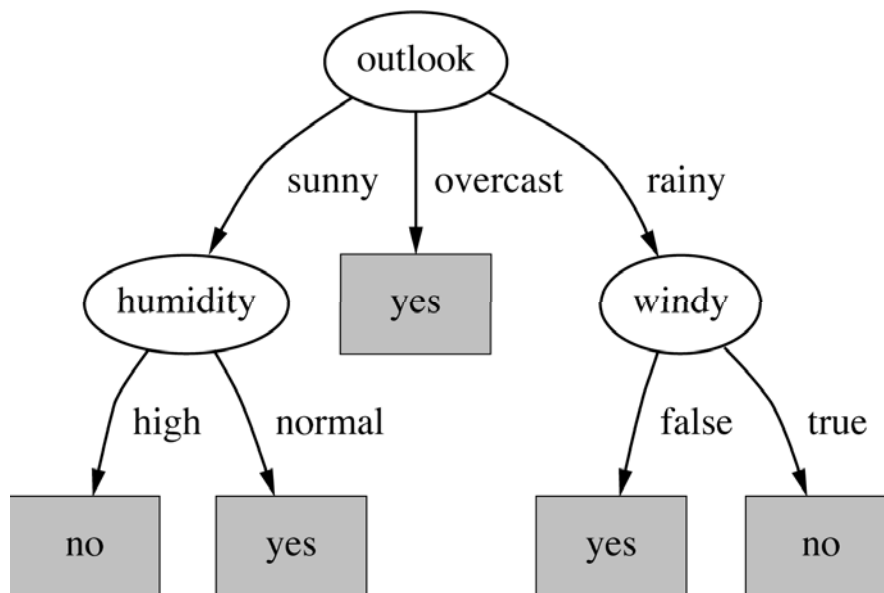
$$\text{gain}(\text{Temperature}) = 0.571 \text{ bits}$$

$$\text{gain}(\text{Humidity}) = 0.971 \text{ bits}$$

$$\text{gain}(\text{Windy}) = 0.020 \text{ bits}$$



# Τελική μορφή δένδρου απόφασης



- Σημείωση: η απόλυτη ομοιογένεια των υποδειγμάτων όλων των φύλλων δεν είναι αναγκαία
  - Συχνά όμοια υποδείγματα ανήκουν σε διαφορετική τάξη
  - ⇒ η διάσπαση διακόπτεται όταν τα δεδομένα δεν δύναται να διασπασθούν περαιτέρω



# Χαρακτηριστικά με μεγάλο αριθμό τιμών

- Πρόβλημα: χαρακτηριστικά με μεγάλο αριθμό τιμών (ακραίο παράδειγμα: κωδικός ID)
- Η ομοιογένεια των υποσυνόλων είναι περισσότερο πιθανή για μεγάλο αριθμό τιμών
  - Το κριτήριο κέρδους πληροφορίας μεροληπτεί υπέρ της επιλογής χαρακτηριστικών με μεγάλο αριθμό τιμών
  - Πιθανή συνέπεια η υπερπροσαρμογή (επιλογή υποβέλτιστου χαρακτηριστικού)





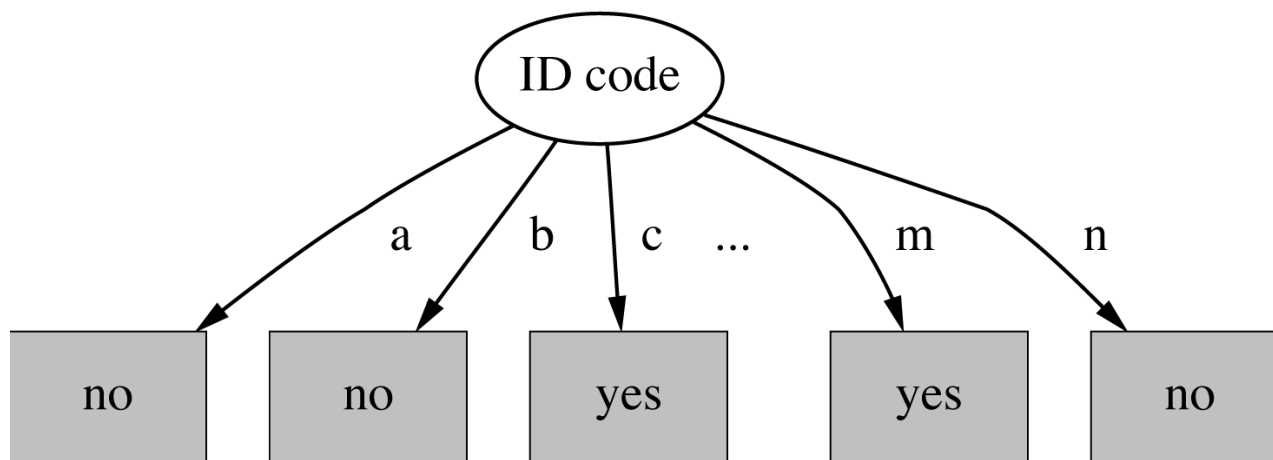
# Παράδειγμα: Δεδομένα καιρού με κωδικό *ID*



ID code	Outlook	Temp.	Humidity	Windy	Play
A	Sunny	Hot	High	False	No
B	Sunny	Hot	High	True	No
C	Overcast	Hot	High	False	Yes
D	Rainy	Mild	High	False	Yes
E	Rainy	Cool	Normal	False	Yes
F	Rainy	Cool	Normal	True	No
G	Overcast	Cool	Normal	True	Yes
H	Sunny	Mild	High	False	No
I	Sunny	Cool	Normal	False	Yes
J	Rainy	Mild	Normal	False	Yes
K	Sunny	Mild	Normal	True	Yes
L	Overcast	Mild	High	True	Yes
M	Overcast	Hot	Normal	False	Yes
N	Rainy	Mild	High	True	No



# Τμήμα δένδρου για το χαρακτηριστικό κωδικού *ID*



- Εντροπία διαχωρισμού:

$$\text{info}(\text{"ID code"}) = \text{info}([0,1]) + \text{info}([0,1]) + \dots + \text{info}([0,1]) = 0 \text{ bits}$$

- Το κέρδος πληροφορίας μεγιστοποιείται για το χαρακτηριστικό του κωδικού ID (0.940 bits)
- Άλλο κριτήριο;



# Λόγος κέρδους (gain ratio)

- Λόγος κέρδους: τροποποίηση του κέρδους πληροφορίας, αντιμετωπίζει το πρόβλημα μεροληψίας
- Υπολογίζει τον αριθμό & το μέγεθος του υποσυνόλου των υποδειγμάτων κάθε κλάδου για την επιλογή χαρακτηριστικού
  - Διορθώνει το κέρδος πληροφορίας λαμβάνοντας υπ' όψιν την *εγγενή πληροφορία (intrinsic information)* της διάσπασης
- Εγγενής πληροφορία: εντροπία της κατανομής των υποδειγμάτων σε κλάδους
  - Απαιτούμενος όγκος πληροφορίας για την περιγραφή του κλάδου στον οποίο ανήκει το υπόδειγμα (πολλοί κλάδοι → μεγάλος όγκος)



# Υπολογισμός λόγου κέρδους

- Παράδειγμα: εγγενής πληροφορία κωδικού ID

$$\text{info}([1,1,\dots,1]) = 14 \times (-1/14 \times \log 1/14) = 3.807 \text{ bits}$$

- Ορισμός λόγου κέρδους:

$$\text{gain\_ratio}(\text{"Attribute"}) = \frac{\text{gain}(\text{"Attribute"})}{\text{intrinsic\_info}(\text{"Attribute"})}$$

– Αντιστρόφως ανάλογο της εγγενής πληροφορίας

- Παράδειγμα:

$$\text{gain\_ratio}(\text{"ID\_code"}) = \frac{0.940 \text{ bits}}{3.807 \text{ bits}} = 0.246$$



# Λόγοι κέρδους για τα δεδομένα καιρού

Outlook		Temperature	
Info:	0.693	Info:	0.911
Gain: 0.940-0.693	0.247	Gain: 0.940-0.911	0.029
Split info: $\text{info}([5,4,5])$	1.577	Split info: $\text{info}([4,6,4])$	1.362
Gain ratio: 0.247/1.577	0.156	Gain ratio: 0.029/1.362	0.021
Humidity		Windy	
Info:	0.788	Info:	0.892
Gain: 0.940-0.788	0.152	Gain: 0.940-0.892	0.048
Split info: $\text{info}([7,7])$	1.000	Split info: $\text{info}([8,6])$	0.985
Gain ratio: 0.152/1	0.152	Gain ratio: 0.048/0.985	0.049



# Περισσότερα περί λόγου κέρδους

- Το χαρακτηριστικό “Outlook” εξακολουθεί να υπερτερεί των άλλων τριών
- Ωστόσο, ο “κωδικός ID” φέρει το μέγιστο λόγο κέρδους
  - Τυπική διόρθωση: *ad hoc* έλεγχος για την αποφυγή διαχωρισμού με βάση το συγκεκριμένο χαρακτηριστικό
- Πρόβλημα κριτηρίου: πιθανή υπερβολική αντιστάθμιση
  - Πιθανή επιλογή χαρακτηριστικού λόγω μικρής εγγενούς πληροφορίας και μόνο
  - Τυπική διόρθωση: έλεγχος των χαρακτηριστικών με κέρδος πληροφορίας μεγαλύτερο του μέσου και μόνο αυτών



# Σύνοψη περί δένδρων απόφασης



- Επαγωγή δένδρων απόφασης από πάνω προς τα κάτω: αλγόριθμος ID3, Ross Quinlan
  - Ο λόγος κέρδους αποτελεί τροποποίηση του βασικού αυτού αλγορίθμου
  - αλγόριθμος C4.5: χειρίζεται επίσης αριθμητικά χαρακτηριστικά, απύσες τιμές, θορυβώδη δεδομένα (@weka: J48)
- Υπάρχει πλήθος κριτηρίων επιλογής χαρακτηριστικού
  - Ωστόσο παρουσιάζουν μικρή διαφοροποίηση ως προς το αποτέλεσμα



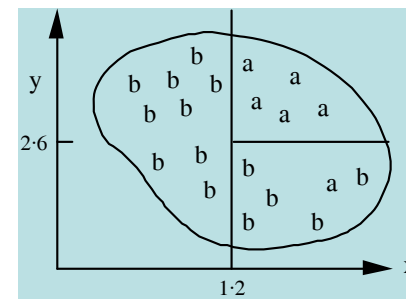
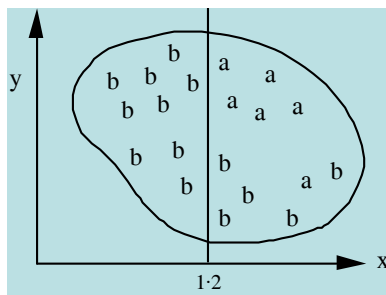
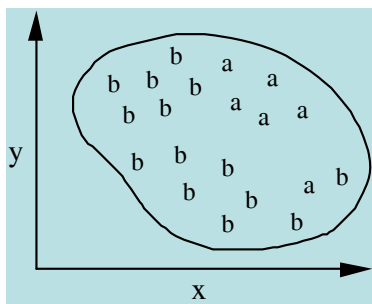
# Αλγόριθμοι κάλυψης

- Μετατροπή δένδρου απόφασης σε σύνολο κανόνων
  - Άμεση, ωστόσο το σύνολο κανόνων προκύπτει υπερβολικά σύνθετο
  - Περισσότερο αποδοτικές μέθοδοι μετατροπής αργιτά πολύπλοκες
- Εναλλακτικά: απευθείας δημιουργία συνόλου κανόνων
  - Για κάθε τάξη, εύρεση συνόλου κανόνων που καλύπτουν το σύνολο των υποδειγμάτων (αποκλείοντας υποδείγματα που δεν ανήκουν στην τάξη)
- Η προσέγγιση καλείται αλγόριθμος κάλυψης:
  - Σε κάθε βήμα, αναζητείται κανόνας που 'καλύπτει' κάποια από τα υποδείγματα





# Παράδειγμα: Δημιουργία κανόνα



If true  
then class = a

If  $x > 1.2$   
then class = a

If  $x > 1.2$  and  $y > 2.6$   
then class = a

- Πιθανά σύνολα κανόνων για την τάξη “b”:

If  $x \leq 1.2$  then class = b

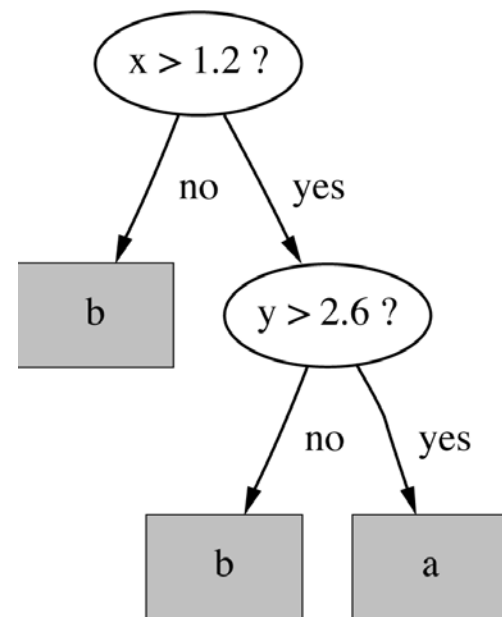
If  $x > 1.2$  and  $y \leq 2.6$  then class = b

- Δυνατή η περαιτέρω προσθήκη κανόνων, ως τη δημιουργία ‘τέλειου’ συνόλου



# Κανόνες ή δένδρα;

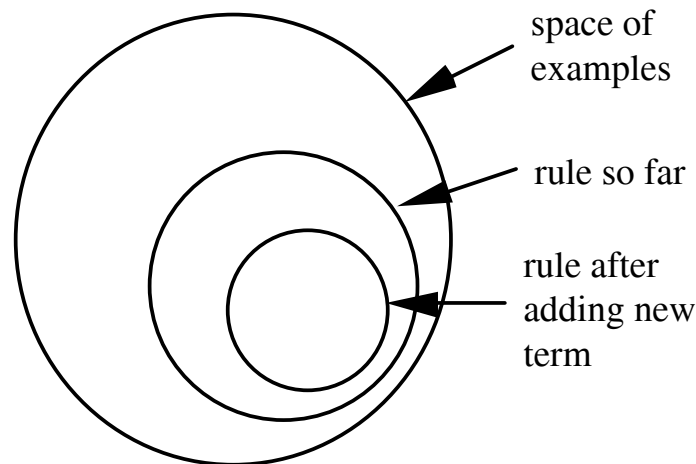
- Αντίστοιχο δένδρο απόφασης:  
(δίδει τα ίδια ακριβώς αποτελέσματα)
  - σύνολα κανόνων: μπορούν να είναι περισσότερο διαυγή,
  - δένδρα κανόνων: συχνά πάσχουν λόγω επαναλαμβανόμενων υποδένδρων
- Επίσης, σε περίπτωση πολλαπλών τάξεων
  - αλγόριθμος κάλυψης: μελετά μία τάξη κάθε φορά,
  - εκμάθηση με δένδρα απόφασης: λαμβάνει υπ' όψιν όλες τις τάξεις





# Απλοϊκός αλγόριθμος κάλυψης

- Δημιουργία κανόνα με την προσθήκη ελέγχων που μεγιστοποιούν την ακρίβεια του κανόνα
- Αντίστοιχο πρόβλημα με την επιλογή χαρακτηριστικού προς διάσπαση στα δένδρα ελέγχου:
  - Ωστόσο, κατά τη δημιουργία δένδρου μεγιστοποιείται η ολική ομοιογένεια
- Κάθε νέος έλεγχος μειώνει την κάλυψη του κανόνα





# Επιλογή ελέγχου

- Στόχος: μεγιστοποίηση ακρίβειας
  - $t$  συνολικός αριθμός υποδειγμάτων που καλύπτονται από τον κανόνα
  - $p$  υποδείγματα της τάξης που εκχωρεί ο κανόνας (σωστές προβλέψεις)
  - $t - p$  αριθμός σφαλμάτων κανόνα

⇒ Επιλογή κριτηρίων που μεγιστοποιούν το λόγο  $p/t$
- Ολοκλήρωση διαδικασίας όταν  $p/t = 1$  ή το σύνολο των υποδειγμάτων δεν μπορεί να διασπαστεί περαιτέρω



# Παράδειγμα φακών επαφής

- Αναζήτηση κανόνα της μορφής:

**IF ?**

**then recommendation = hard**

- Πιθανοί κανόνες:

**Age = Young** 2/8

**Age = Pre-presbyopic** 1/8

**Age = Presbyopic** 1/8

**Spectacle prescription = Myope** 3/12

**Spectacle prescription = Hypermetrope** 1/12

**Astigmatism = no** 0/12

**Astigmatism = yes** 4/12

**Tear production rate = Reduced** 0/12

**Tear production rate = Normal** 4/12



# Παράδειγμα φακών επαφής

## Προσθήκη κανόνα

- Προσθήκη βέλτιστου κανόνα:

**If astigmatism = yes  
then recommendation = hard**

- Ο κανόνας καλύπτει τα υποδείγματα:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None



# Παράδειγμα φακών επαφής

## Περαιτέρω προσθήκη

- Ο κανόνας ως τώρα: `If astigmatism = yes`  
`and ?`  
`then recommendation = hard`
- Πιθανοί περαιτέρω έλεγχοι:

<code>Age = Young</code>	<code>2 / 4</code>
<code>Age = Pre-presbyopic</code>	<code>1 / 4</code>
<code>Age = Presbyopic</code>	<code>1 / 4</code>
<code>Spectacle prescription = Myope</code>	<code>3 / 6</code>
<code>Spectacle prescription = Hypermetrope</code>	<code>1 / 6</code>
<code>Tear production rate = Reduced</code>	<code>0 / 6</code>
<code>Tear production rate = Normal</code>	<code>4 / 6</code>



# Παράδειγμα φακών επαφής Τροποποιημένος κανόνας

- Προσθήκη βέλτιστου ελέγχου:

**If astigmatism = yes**

**and tear production rate = normal**

**then recommendation = hard**

- Ο κανόνας καλύπτει πλέον και τα ακόλουθα υποδείγματα:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Normal	None





# Παράδειγμα φακών επαφής

## Περαιτέρω προσθήκη

- Ως τώρα:

**If astigmatism = yes**

**and tear production rate = normal**

**and ?**

**then recommendation = hard**

- Πιθανοί έλεγχοι:

**Age = Young** 2/2

**Age = Pre-presbyopic** 1/2

**Age = Presbyopic** 1/2

**Spectacle prescription = Myope** 3/3

**Spectacle prescription =** 1/3

**Hypermetrope**

- Ισοπαλία μεταξύ 1<sup>ου</sup> & 4<sup>ου</sup> ελέγχου
  - Επιλογή ελέγχου με μεγαλύτερη κάλυψη



# Παράδειγμα φακών επαφής

## Τελικό αποτέλεσμα

- Τελική μορφή κανόνα:

**If astigmatism = yes  
and tear production rate = normal  
and spectacle prescription =  
myope  
then recommendation = hard**

- Όμοια προκύπτει ο κανόνας για σύσταση “hard lenses” (δημιουργήθηκε από παραδείγματα που δεν καλύπτονται από τον πρώτο κανόνα):

**If age = young and astigmatism =  
yes  
and tear production rate =  
normal  
then recommendation = hard**

- Επανάληψη της μεθόδου για τις άλλες δύο τάξεις



# Ψευδοκώδικας PRISM

For each class C

Initialize E to the instance set

While E contains instances in class C

    Create a rule R with an empty left-hand side that predicts class C

    Until R is perfect (or there are no more attributes to use) do

        For each attribute A not mentioned in R, and each value v,

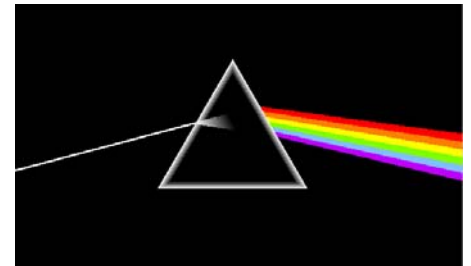
            Consider adding the condition  $A = v$  to the left-hand side of R

            Select A and v to maximize the accuracy  $p/t$

            (break ties by choosing the condition with the largest p)

        Add  $A = v$  to R

    Remove the instances covered by R from E





# Κανόνες ή λίστες απόφασης;

- Ο αλγόριθμος PRISM (χωρίς τον εξωτερικό βρόγχο) παράγει λίστα κανόνων απόφασης για μία τάξη
  - Οι επόμενοι κανόνες σχεδιάζονται για υποδείγματα που δεν καλύπτονται από προηγούμενους κανόνες
  - Η σειρά δεν έχει σημασία, καθώς όλοι οι κανόνες προβλέπουν την ίδια τάξη
- Ο εξωτερικός βρόγχος λαμβάνει υπ 'όψιν κάθε τάξη χωριστά
  - Δεν υποδηλώνεται εξάρτηση από τη διαδοχή
- Προβλήματα: επικαλυπτόμενοι κανόνες, αναγκαίος ορισμός προιαθορισμένου κανόνα



# Διαχώρισε & βασίλευε (separate and conquer)

- Αλγόριθμοι όπως ο PRISM (για το χειρισμό μίας τάξης) υλοποιούν τη μέθοδο ‘διαχώρισε & βασίλευε’ (*separate-and-conquer*):
  - Αρχικά, προσδιορισμός ωφέλιμου κανόνα
  - Στη συνέχεια, διαχωρισμός των υποδειγμάτων που καλύπτει ο κανόνας από τα υπόλοιπα
  - Τέλος, ‘κατάκτηση’ των υποδειγμάτων που απομένουν
- Διαφοροποίηση από τις μεθόδους ‘διαίρει & βασίλευε’ (*divide-and-conquer*) :
  - Το υποσύνολο που καλύπτεται από τον κανόνα δεν χρειάζεται να εξερευνηθεί στη συνέχεια



# Μετά τα βασικά, τι;

- Ως τώρα μελετήθηκαν απλοϊκές δομές αλγορίθμων μάθησης
  - Κανόνες
  - Δένδρα απόφασης
  - Αλγόριθμοι κάλυψης
- Ωστόσο, η χρήση ενός αλγορίθμου σε ευρύ πεδίο ρεαλιστικών εφαρμογών προϋποθέτει
  - Χειρισμό αριθμητικών χαρακτηριστικών
  - Χειρισμό αγνώστων τιμών
  - Ανθεκτικότητα στην ύπαρξη θορύβου
  - Δυνατότητα εξόρυξης σύνθετων δομικών περιγραφών
- Οι απλοϊκές δομές χρειάζεται να επεκταθούν ώστε να ικανοποιούν αυτές τις προδιαγραφές
  - Επόμενο βήμα αποτελεί η θεώρηση των δομών αυτών σε πλήρες εύρος της πολυπλοκότητάς τους



# Δένδρα απόφασης

- Επέκταση του αλγορίθμου ID3:
  - Για αριθμητικά χαρακτηριστικά: *άμεση*
  - Για ευαίσθητο χειρισμό αγνώστων τιμών: *σύνθετη*
  - Για σταθερότητα με θορυβώδη δεδομένα:  
*απαιτείται μηχανισμός κλαδέματος*
- C4.5 (Ross Quinlan)
  - Ο πλέον γνωστός και ίσως ευρύτερα χρησιμοποιούμενος αλγόριθμος εκμάθησης
  - Εμπορικός διάδοχος: C5.0



# Αριθμητικά χαρακτηριστικά

- Τυπική μέθοδος: δυαδικοί διαχωρισμοί
  - Για παράδειγμα,  $temp < 45$
- Σε αντίθεση με τα ονομαστικά χαρακτηριστικά, κάθε αριθμητικό χαρακτηριστικό έχει πολλά πιθανά σημεία διαχωρισμού
- Άμεση επέκταση της διαδικασίας:
  - Υπολογισμός κέρδους πληροφορίας (ή άλλου κριτηρίου) για κάθε πιθανό σημείο διαχωρισμού
  - Επιλογή βέλτιστου σημείου διαχωρισμού
  - Το κέρδος πληροφορίας για το βέλτιστο αυτό σημείο είναι το κέρδος πληροφορίας για το χαρακτηριστικό
- Μεγαλύτερες απαιτήσεις σε υπολογιστικό κόστος





# Κλάδεμα

- Στόχος: πρόληψη υπερπροσαρμογής δένδρου σε θόρυβο των δεδομένων
- Μέθοδος: 'κλάδεμα' δένδρου απόφασης
- Δύο στρατηγικές:
  - *Μετακλάδεμα (postpruning)*  
αποκοπή περιττών τμημάτων ενός ήδη ολοκληρωμένου δένδρου
  - *Προκλάδεμα (prepruning)*  
διακοπή ανάπτυξης ενός κλαδιού
- Στην πράξη προτιμάται το μετακλάδεμα
  - Το προκλάδεμα είναι υπολογιστικά φθηνότερο, ωστόσο συχνά προκαλεί πρόωρη διακοπή



# Προκλάδεμα (prepruning)

- Βασίζεται σε έλεγχο στατιστικής σημαντικότητας
  - Διακοπή ανάπτυξης ενός δένδρου όταν δεν υπάρχει στατιστικά σημαντική συσχέτιση ανάμεσα σε οποιοδήποτε χαρακτηριστικό και την τάξη ενός συγκεκριμένου φύλλου
- Πλέον συνήθης έλεγχος:  $X^2$
- Ο αλγόριθμος ID3 χρησιμοποιεί τον έλεγχο  $X^2$  σε προσθήκη του κριτηρίου κέρδους πληροφορίας
  - Μόνο τα στατιστικά σημαντικά χαρακτηριστικά δύναται να επιλεγούν από τη διαδικασία του κέρδους πληροφορίας



# Πρώιμη διακοπή

	a	b	class
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0



- Το προκλάδεμα μπορεί να διακόψει την ανάπτυξη σε ένα μη ώριμο στάδιο: *πρώιμη διακοπή (early stopping)*
- Κλασσικό παράδειγμα: Ισοτιμία προβλήματος XOR
  - Κανένα χαρακτηριστικό δεν εμφανίζει σημαντική συσχέτιση με την τάξη
  - Η δομή είναι ορατή μόνο μετά την περάτωση του δένδρου
  - Το προκλάδεμα διακόπτει εξ 'αρχής την ανάπτυξη του δένδρου
- Ωστόσο: τα προβλήματα XOR δεν συναντώνται συχνά
- Επίσης: το προκλάδεμα πολύ ταχύτερο από το μετακλάδεμα



# Μετακλάδεμα (postpruning)

- Αρχικά, κατασιευή πλήρους δένδρου
- Στη συνέχεια, κλάδεμα
  - Το πλήρες δένδρο φανερώνει όλες τις αλληλεπιδράσεις μεταξύ των χαρακτηριστικών
- Πρόβλημα: κάποια υποδένδρα πιθανώς οφείλονται σε τυχαίες επιρροές
- Δύο λειτουργίες κλαδέματος:
  - Αντικατάσταση υποδένδρου
  - Ανύψωση υποδένδρου
- Στρατηγικές:
  - Υπολογισμός κόστους
  - Έλεγχος σημαντικότητας
  - Αρχή MDL



# Παράδειγμα εργασιακών διαπραγματεύσεων

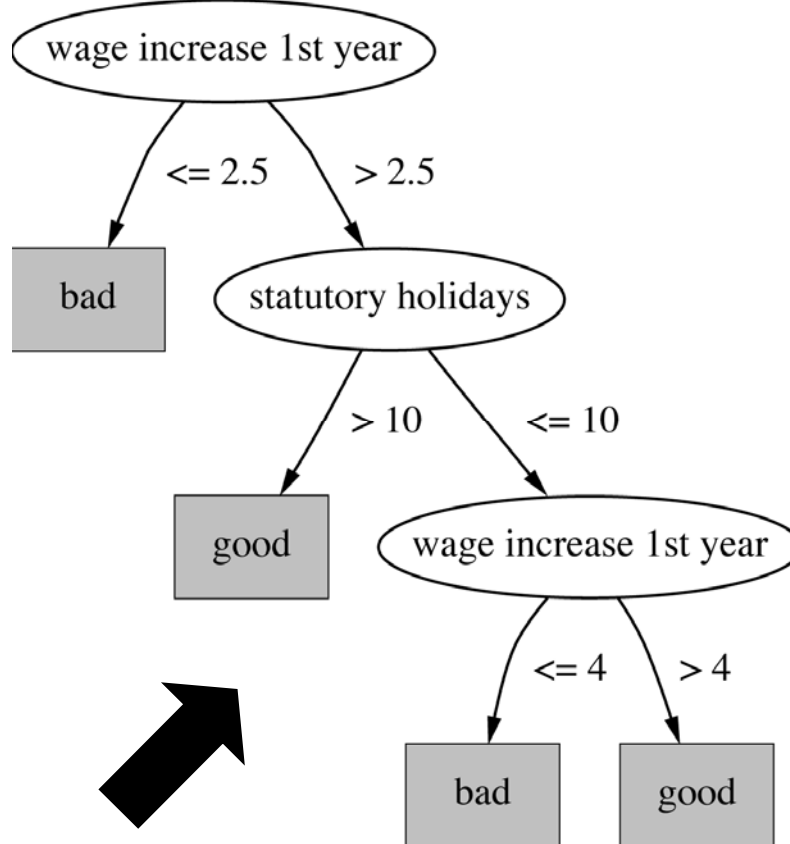
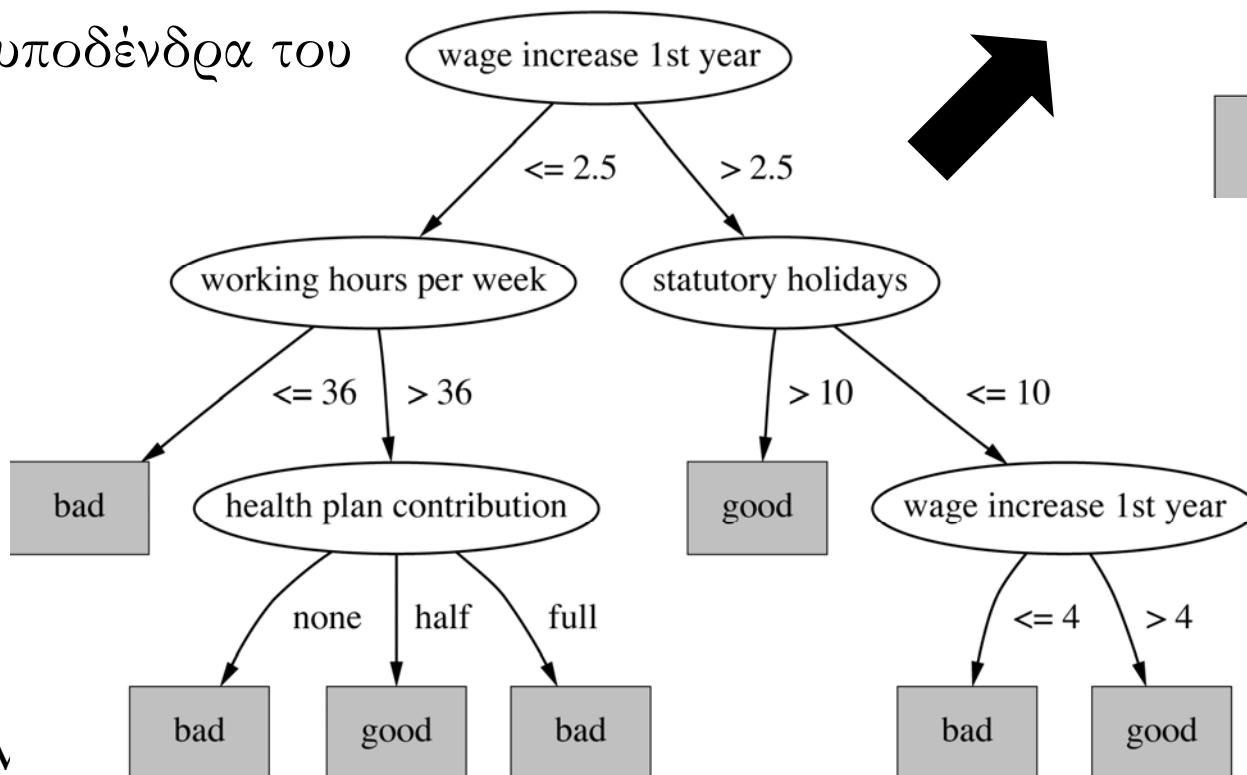


Attribute	Type	1	2	3	...	40
Duration	(Number of years)	1	2	3		2
Wage increase first year	Percentage	2%	4%	4.3%		4.5
Wage increase second year	Percentage	?	5%	4.4%		4.0
Wage increase third year	Percentage	?	?	?		?
Cost of living adjustment	{none,tcf,tc}	none	tcf	?		none
Working hours per week	(Number of hours)	28	35	38		40
Pension	{none,ret-allw, empl-cntr}	none	?	?		?
Standby pay	Percentage	?	13%	?		?
Shift-work supplement	Percentage	?	5%	4%		4
Education allowance	{yes,no}	yes	?	?		?
Statutory holidays	(Number of days)	11	15	12		12
Vacation	{below-avg,avg,gen}	avg	gen	gen		avg
Long-term disability assistance	{yes,no}	no	?	?		yes
Dental plan contribution	{none,half,full}	none	?	full		full
Bereavement assistance	{yes,no}	no	?	?		yes
Health plan contribution	{none,half,full}	none	?	full		half
Acceptability of contract	{good,bad}	bad	good	good		good



# Αντικατάσταση υποδένδρου

- Προσέγγιση *Bottom-up*
- Θεώρηση (υπο)δένδρου προς αντικατάσταση μόνο εάν δεν αντικαθίσταται κάποιο από τα υποδένδρα του

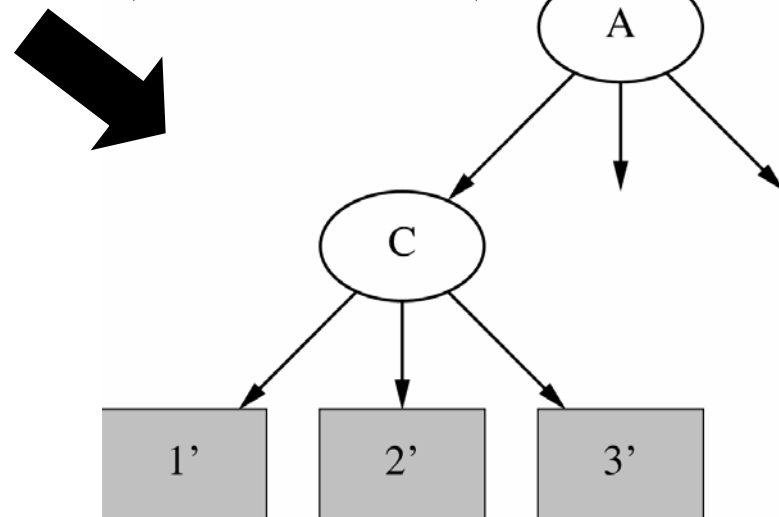
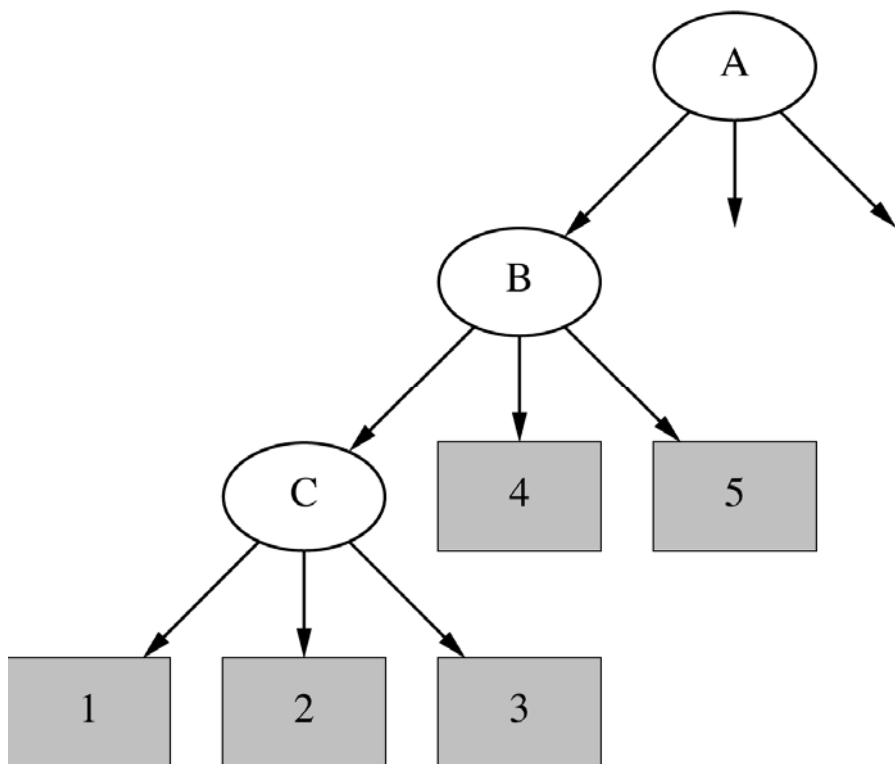




# Ανύψωση υποδένδρου

- Αποκοπή φύλλου
- Αναδιανομή υποδειγμάτων
- Περισσότερο αργή της αντικατάστασης

*(αξίζει τον κόπο;)*





# Κλάδεμα με υπολογισμό σφάλματος

- Συνθήκη: Το κλάδεμα πραγματοποιείται μόνο όταν μειώνεται το εκτιμώμενο σφάλμα
- Το σφάλμα επί των δεδομένων εκπαίδευσης δεν αποτελεί αξιόπιστη εκτίμηση  
(δεν θα υποδείκνυε ποτέ κλάδεμα)
- Χρήση συνόλου παρακράτησης για κλάδεμα  
(κλάδεμα μειωμένου σφάλματος, reduced-error pruning)
  - Μειονέκτημα: το δένδρο δημιουργείται από λιγότερα δεδομένα





# Μέθοδος αλγορίθμου C4.5

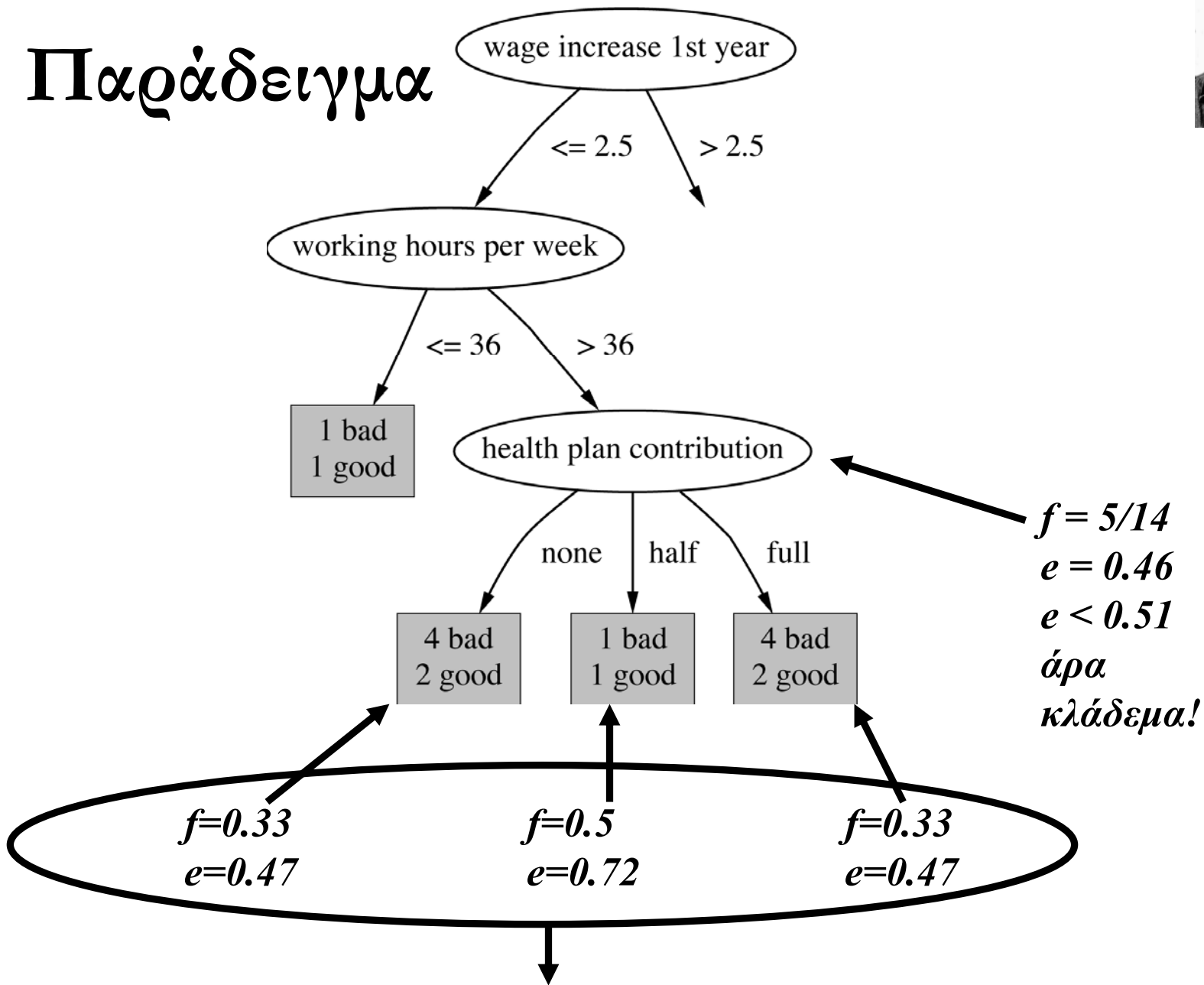
- Μέθοδος αλγορίθμου C4.5
  - Εύρεση διαστήματος εμπιστοσύνης από τα δεδομένα εκπαίδευσης
  - Ευρετική επιλογή ορίου για κλάδεμα
  - Τυπική μέθοδος βασισμένη στη διαδοχή Bernoulli
- Η εκτίμηση σφάλματος του υποδένδρου αποτελεί σταθμισμένο άθροισμα των εκτιμήσεων σφάλματος όλων των φύλλων του
- Εκτίμηση σφάλματος κόμβου:

$$e = \left( f + \frac{z^2}{2N} + z \sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) / \left( 1 + \frac{z^2}{N} \right)$$

- Όπου  $f$  το σφάλμα στα δεδομένα εκπαίδευσης &  $N$  ο αριθμός των υποδειγμάτων που καλύπτονται από το φύλλο
- Αν  $c = 25\%$ , τότε  $z = 0.69$  (κανονική κατανομή)



# Παράδειγμα



Στάθμιση με λόγο 6:2:6 δίνει 0.51



# Πολυπλοκότητα επαγωγής δένδρου

- Έστω
  - $m$  χαρακτηριστικά
  - $n$  υποδείγματα εκπαίδευσης
  - Βάθος δένδρου  $O(\log n)$  (παραδοχή)
- Υπολογιστικό κόστος κατασκευής δένδρου:  $O(m n \log n)$
- Αντικατάσταση υποδένδρου  $O(n)$
- Ανύψωση υποδένδρου  $O(n (\log n)^2)$ 
  - Μέσο κόστος πιθανής ανακατανομής υποδειγμάτων σε κάθε κόμβο, μεταξύ της ρίζας και του φύλλου:  $O(\log n)$
- Ολικό κόστος:  $O(m n \log n) + O(n (\log n)^2)$



# Μετατροπή δένδρων σε κανόνες

- Αρχικά & απλοϊκά: ένας κανόνας για κάθε φύλλο
- Ακολούθως: κλάδεμα των συνθηκών που μειώνουν το εκτιμώμενο σφάλμα των κανόνων
  - Συνήθως με εξαντλητική αναζήτηση (άλλες μέθοδοι επίσης εφικτές)
  - Συχνά παράγονται διπλότυποι κανόνες, απαιτείται τελικός έλεγχος επί αυτού
- Στη συνέχεια
  - Εύρεση όλων των κανόνων για κάθε τάξη
  - Επιλογή υποσυνόλου με οδηγό την αρχή MDL
- Ταξινόμηση των υποσυνόλων για αποφυγή αντικρουόμενων υποδείξεων
- Τέλος: αποκοπή κλάδων (με εξαντλητική αναζήτηση) σε περίπτωση που μειώνεται το συνολικό σφάλμα επί των δεδομένων εκπαίδευσης
- @weka: PART



# Περισσότερα περί του αλγορίθμου C4.5

- Αργός για μεγάλα σύνολα δεδομένων με θόρυβο
- Η εμπορική έκδοση C5.0 χρησιμοποιεί μία διαφορετική τεχνική
  - Αρχικά ταχύτερη και ελαφρά πιο ακριβής
  - Δεν περιγράφεται στην ανοιχτή βιβλιογραφία
- Παράμετροι του C4.5
  - Επίπεδο εμπιστοσύνης (default 25%):  
χαμηλότερες τιμές επιφέρουν ειτενέστερο κλάδεμα
  - Ελάχιστος αριθμός υποδειγμάτων ανά φύλλο (default 2)



# Σύνοψη περί δένδρων απόφασης

## *TDIDT: Top-Down Induction of Decision Trees*

- Η πλέον εκτενώς μελετημένη μέθοδος μηχανικής μάθησης που χρησιμοποιεί για εξόρυξη πληροφορίας
- Διαφορετικά κριτήρια για επιλογή χαρακτηριστικού ή ελέγχου σπάνια επιφέρουν σημαντική διαφορά
- Διαφορετικές μέθοδοι κλαδέματος κυρίως μεταβάλλουν το μέγεθος του δένδρου που προκύπτει
- Ο αλγόριθμος C4.5 κατασκευάζει δένδρα απόφασης ελέγχου *μίας μεταβλητής (univariate)*
- Κάποια άλλα συστήματα TDIDT μπορούν να κατασκευάσουν δένδρα *πολλών μεταβλητών (multivariate, για παράδειγμα CART)*



# Κανόνες ταξινόμησης (classification rules)

- Συνήθης διαδικασία: *separate-and-conquer*
- Συνήθεις διαφοροποιήσεις:
  - Μέθοδος αναζήτησης (για παράδειγμα εξαντλητική, beam search, ...)
  - Κριτήρια επιλογής (για παράδειγμα ακρίβεια, ...)
  - Μέθοδος κλαδέματος (για παράδειγμα MDL, σύνολο παρακρατήσης, ...)
  - Κριτήριο διακοπής (για παράδειγμα ελάχιστη ακρίβεια)
  - Επεξεργασία μετά την εκπαίδευση
- Επίσης: Λίστα απόφασης ή ένα υποσύνολο κανόνων για κάθε τάξη;



# Κριτήρια επιλογής

- Βασικός αλγόριθμος κάλυψης:
  - Συνεχής προσθήκη συνθηκών σε κανόνα για τη βελτίωση της ακριβειάς του
  - Προσθήκη της συνθήκης που επιτυγχάνει τη μέγιστη αύξηση της ακριβειας
- Κριτήριο 1:  $p/t$ 
  - $t$  συνολικά υποδείγματα που καλύπτονται από τον κανόνα
  - $p$  αριθμός αυτών που είναι θετικά (ανήκουν στην πλειοψηφούσα τάξη)
  - Παράγει κανόνες που δεν καλύπτουν αρνητικά υποδείγματα
  - Επίσης δύναται να παράγει κανόνες με πολύ μικρή κάλυψη (ειδικές περιπτώσεις ή θόρυβος;)
- Κριτήριο 2: κέρδος πληροφορίας  $p (\log(p/t) - \log(P/T))$ 
  - $P$  και  $T$  ο αριθμός των θετικών και του συνόλου πριν τη προσθήκη της νέας συνθήκης
  - Δίνει έμφαση περισσότερο στα θετικά παρά στα αρνητικά υποδείγματα
- Τα κριτήρια αλληλεπιδρούν με τον επιλεγμένο μηχανισμό κλαδέματος





# Άγνωστες τιμές & αριθμητικά χαρακτηριστικά

- Συνήθης μεταχείριση των άγνωστων τιμών:  
*σε κάθε έλεγχο αποτυγχάνουν (fail)*
  - Ο αλγόριθμος πρέπει είτε να
    - Χρησιμοποιεί άλλους ελέγχους για να διαχωρίσει τα θετικά υποδείγματα
    - Να αφήσει τα συγκεκριμένα υποδείγματα για επόμενο βήμα της διαδικασίας
- Σε κάποιες περιπτώσεις, συνίσταται ο χειρισμός της 'άγνωστης' ως ξεχωριστή τιμή
- Τα αριθμητικά χαρακτηριστικά μεταχειρίζονται όπως και στα δένδρα απόφασης



# Επιλογές κατά το κλάδεμα

- Δύο κύριες στρατηγιές:
  - Αυξητικό (*incremental*) κλάδεμα
  - Ολικό (*global*) κλάδεμα
- Άλλη διαφοροποίηση: κριτήριο κλαδέματος
  - Σφάλμα σε σύνολο παρακράτησης (κλάδεμα μειωμένου σφάλματος, *reduced-error pruning*)
  - Στατιστική σημαντικότητα
  - Αρχή MDL
- Επίσης: μετακλάδεμα ή προκλάδεμα



# Χρήση συνόλου κλαδέματος

- Για λόγους στατιστικής εγκυρότητας, το δένδρο αποτιμάται σε δεδομένα που δεν χρησιμοποιούνται για εκπαίδευση:
  - Αυτό προϋποθέτει ένα *σύνολο ανάπτυξης (growing set)* και ένα *σύνολο κλαδέματος (pruning set)*
- *κλάδεμα μειωμένου σφάλματος:*  
κατασκευή συνόλου κανόνων και μετά περικοπή του
- *Αυξητικό κλάδεμα μειωμένου σφάλματος (incremental reduced-error pruning):* απλοποίηση κανόνα μετά την κατασκευή του
  - Δυνατός ο επαναδιαχωρισμός των δεδομένων μετά το κλάδεμα του κανόνα
- Επίσης επωφελής: διαστρωμάτωση



# Αλγόριθμος αυξητικού κλαδέματος μειωμένου σφάλματος



```
Initialize E to the instance set
Until E is empty do
  Split E into Grow and Prune in the ratio 2:1
  For each class C for which Grow contains an instance
    Use basic covering algorithm to create best perfect rule
    for C
    Calculate w(R): worth of rule on Prune
      and w(R-): worth of rule with final condition
      omitted
    If w(R-) < w(R), prune rule and repeat previous step
  From the rules for the different classes, select the one
  that's worth most (i.e. with largest w(R))
  Print the rule
  Remove the instances covered by rule from E
Continue
```



# Κριτήρια κατά το αυξητικό κλάδεμα μειωμένου σφάλματος

- $[p + (N - n)] / T$ 
  - ( $N$  συνολικός αριθμός αρνητικών υποδειγμάτων)
  - διαισθητικά:
    - $p = 2000$  and  $n = 1000$  vs.  $p = 1000$  and  $n = 1$
- Βαθμός επιτυχίας  $p / t$ 
  - πρόβλημα:  $p = 1$  and  $t = 1$   
ή  $p = 1000$  and  $t = 1001$
- $(p - n) / t$ 
  - Ίδια επίδραση με το βαθμό επιτυχίας, ισούται με  $2p/t - 1$
- Η εύρεση ενός απλού κριτηρίου, συμβατού με τη διαίσθηση μοιάζει δύσκολη
- Χρήση κριτηρίου υπεργεωμετρικής / δυωνυμικής κατανομής;



# Αλγόριθμος INDUCT

Initialize E to the instance set

Until E is empty do

    For each class C for which E contains an instance

        Use basic covering algorithm to create best perfect rule for C

        Calculate  $m(R)$ : significance for rule  
                and  $m(R-)$ : significance for rule with final condition omitted

        If  $m(R-) < m(R)$ , prune rule and repeat previous step

    From the rules for the different classes, select the most significant one (i.e. with smallest  $m(R)$ )

    Print the rule

    Remove the instances covered by rule from E

Continue

- Εκτελεί αυξητικό κλάδεμα



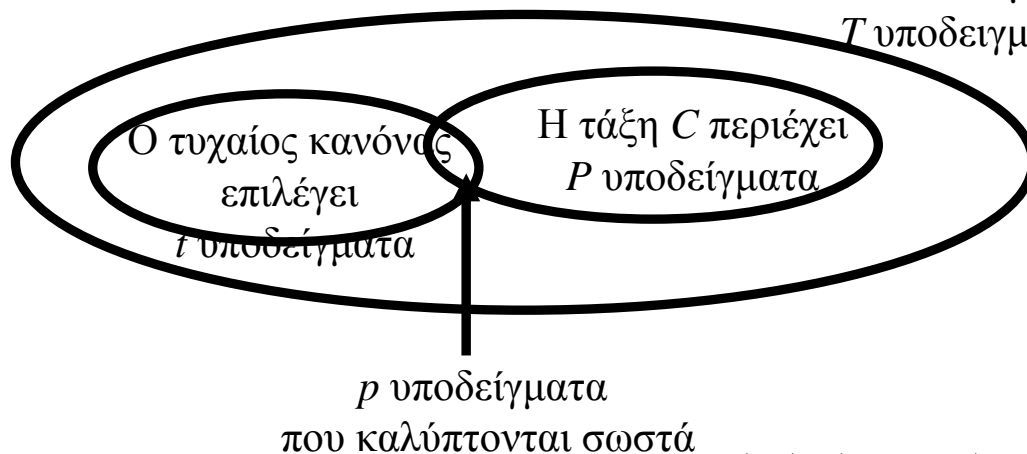
# Υπολογισμός σημαντικότητας

- Μέτρο σημαντικότητας για ένα κανόνα του αλγορίθμου INDUCT:
  - Πιθανότητα ένας τυχαίος κανόνας με ίδια κάλυψη να αποδίδει τουλάχιστον εξίσου
- Για τον τυχαίο κανόνα  $R$  επιλέγονται τυχαίες  $t$  περιπτώσεις από το σύνολο δεδομένων
- Πόσο πιθανό είναι  $p$  από αυτές να ανήκουν στη σωστή τάξη;
- Η πιθανότητα αυτή δίδεται από την υπεργεωμετρική κατανομή



# Υπεργεωμετρική κατανομή

Σύνολο δεδομένων  
 $T$  υποδειγμάτων



$\Pr[t$  τυχαία υποδείγματα,  
ακριβώς  $p$  είναι της τάξης  $C$ ]

$$\frac{\binom{P}{p} \binom{T-P}{t-p}}{\binom{T}{t}}$$

Ποια η πιθανότητα ένας τυχαίος κανόνας να αποδίδει *τουλάχιστον*  $m$  εξίσου?

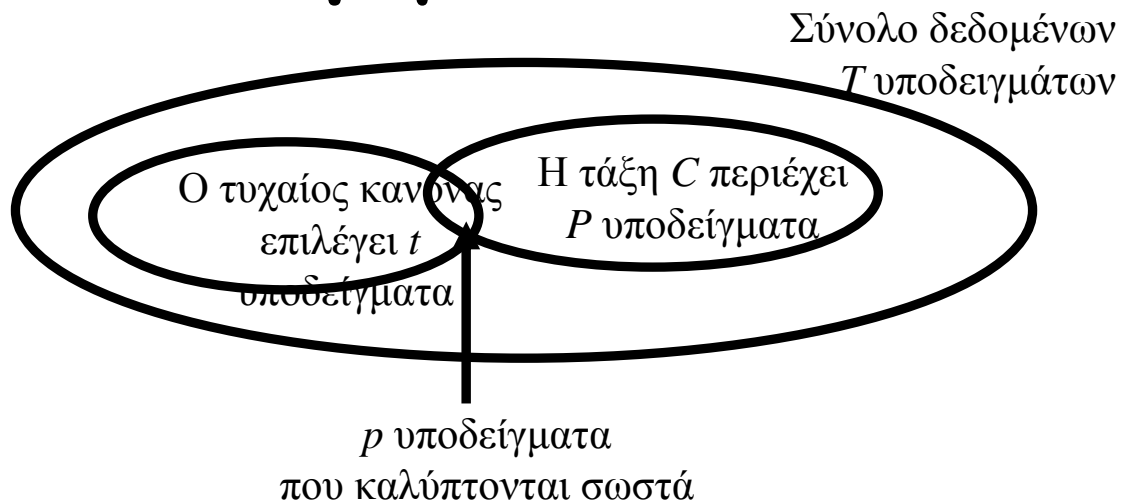
$$m(R) = \sum_{i=p}^{\min(t,P)} \frac{\binom{P}{i} \binom{T-P}{t-i}}{\binom{T}{t}}$$

*Αυτή είναι η στατιστική σημαντικότητα του κανόνα*





# Διωνυμική κατανομή



- Ο υπολογισμός της υπεργεωμετρικής είναι δύσκολος
- Προσέγγιση: δείγμα με επανατοποθέτηση *αντί με επανατοποθέτηση* *αντί χωρίς επανατοποθέτηση*

$$\binom{t}{p} \left(\frac{P}{T}\right)^p \left(1 - \frac{P}{T}\right)^{t-p}$$



# Παραλλαγές

- Δημιουργία κανόνων με κατάταξη
  - Εκκίνηση με τη μικρότερη (σε συχνότητα) τάξη
  - Η μεγαλύτερη τάξη αφήνεται στην κάλυψη του προεπιλεγμένου (default) κανόνα
- Κριτήριο διακοπής
  - Διακοπή παραγωγής κανόνων όταν η ακρίβεια είναι πολύ χαμηλή
- Μαθητής κανόνων (rule learner) RIPPER:
  - Χρησιμοποιεί ως κριτήριο διακοπής την αρχή MDL
  - Περιλαμβάνει βήμα μετα-επεξεργασίας για τη τροποποίηση των κανόνων με βάση την MDL



# Αλγόριθμος PART

- Ολική βελτιστοποίηση μη απαιτούμενη
  - Αντίθετα από C4.5 & RIPPER
- Παράγει απεριόριστη λίστα απόφασης χρησιμοποιώντας την βασική μέθοδο διαχώρισε & βασίλευε
- Κατασκευάζει επιμέρους (*partial*) δένδρο απόφασης για την απόκτηση ενός κανόνα
  - Ένας κανόνας αποκόπτεται μόνο όταν όλα τα συμπεράσματά του είναι γνωστά
  - Πραγματοποιεί βιαστική γενίκευση (*hasty generalization*)
- Χρησιμοποιεί τη διαδικασία του C4.5 για την κατασκευή των δένδρων



# Κατασκευή επιμέρους δένδρου

Expand-subset (S):

Choose test T and use it to split set of examples  
into subsets

Sort subsets into increasing order of average  
entropy

while

    there is a subset X not yet been expanded  
    AND all subsets expanded so far are leaves

    expand-subset(X)

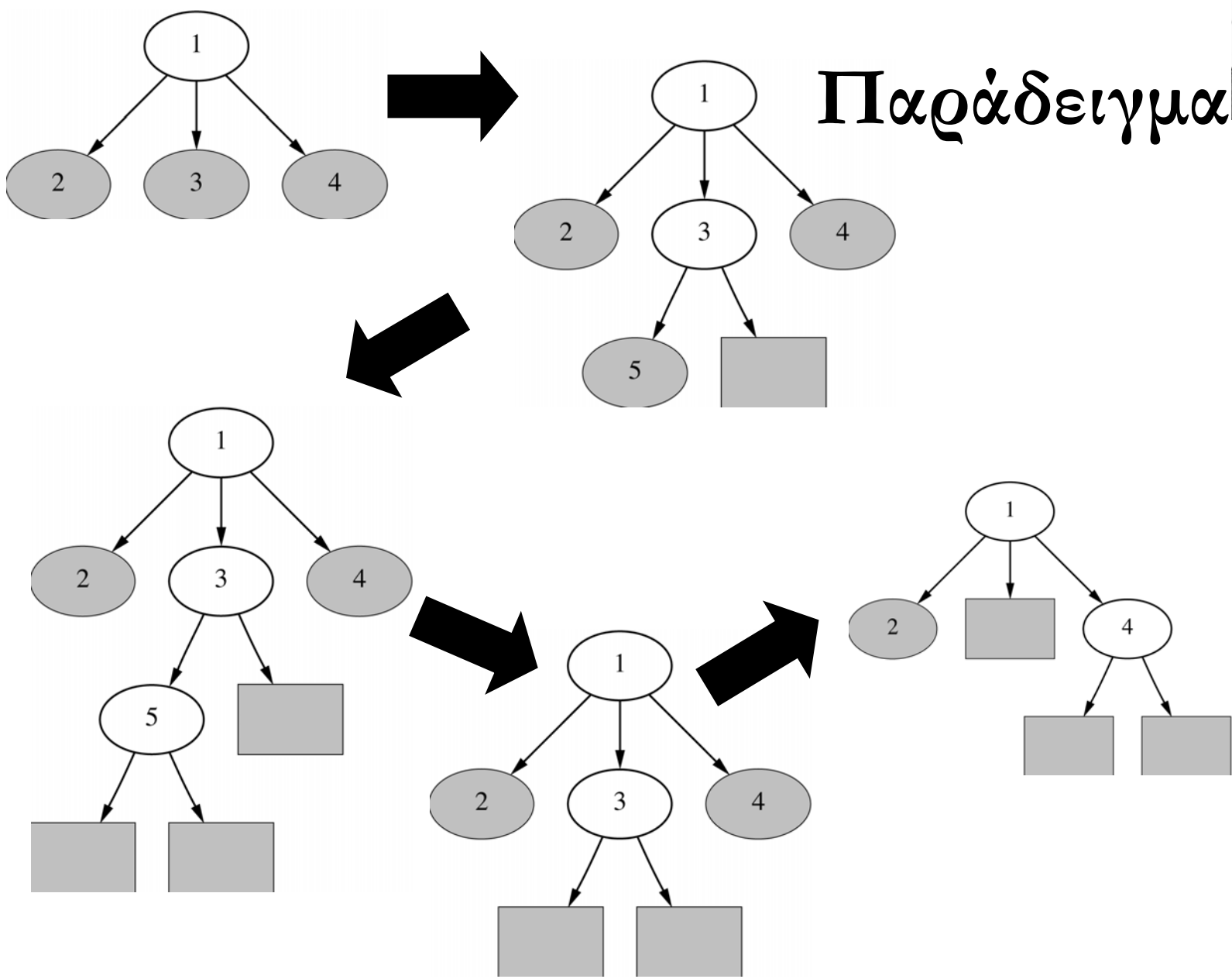
if

    all subsets expanded are leaves  
    AND estimated error for subtree  
         $\geq$  estimated error for node

    undo expansion into subsets and make node a leaf



# Παράδειγμα





# Περί του αλγορίθμου PART

- Μετατροπή φύλλου με μεγαλύτερη κάλυψη σε κανόνα
- Χειρισμός απόντων τιμών όπως ο C4.5
  - Δηλαδή τμηματοποίηση χαρακτηριστικού
- Απαιτούμενος χρόνος δημιουργίας κανόνα:
  - Χείριστο σενάριο: ίδιος με εκείνον για την κατασκευή και το κλάδεμα ενός δένδρου
    - Προκύπτει για δεδομένα με θόρυβο
  - Βέλτιστο σενάριο: ίδιος με εκείνον για την κατασκευή ενός απλού κανόνα
    - Προκύπτει για δεδομένα χωρίς θόρυβο



# Εφαρμογή στο weka



# trees@weka



- *ADTree* Build alternating decision trees
- *DecisionStump* Build one-level decision trees
- *Id3* Basic divide-and-conquer decision tree algorithm
- *J48* C4.5 decision tree learner (implements C4.5 revision 8)
- *LMT* Build logistic model trees
- *M5P* M5¢ model tree learner
- *NBTree* Build a decision tree with Naïve Bayes classifiers at the leaves
- *RandomForest* Construct random forests
- *RandomTree* Construct a tree that considers a given number of random features at each node
- *REPTree* Fast tree learner that uses reduced-error pruning
- *UserClassifier* Allow users to build their own decision tree





# rules@weka



- *ConjunctiveRule* Simple conjunctive rule learner
- *DecisionTable* Build a simple decision table majority classifier
- *JRip* RIPPER algorithm for fast, effective rule induction
- *M5Rules* Obtain rules from model trees built using M5c
- *Nnge* Nearest-neighbor method of generating rules using nonnested generalized exemplars
- *OneR* 1R classifier
- *Part* Obtain rules from partial decision trees built using J4.8
- *Prism* Simple covering algorithm for rules
- *Ridor* Ripple-down rule learner
- *ZeroR* Predict the majority class (if nominal) or the average value (if numeric)



# Τέλος

Επόμενη διάλεξη:  
Αλγόριθμοι εκμάθησης, μέρος Β