ΕΜΠ ΔΠΜΣ

Εφαρμοσμένες Μαθηματικές Επιστήμες

Αλγόριθμοι Εξόρυξης Πληροφορίας

Διάλεξη 07:

Αλγόριθμοι εκμάθησης ΜέροςΓ

Συναρτήσεις & μετα-μαθησιακοί Αλγόριθμοι

Αλγόριθμοι

- Η παρούσα διάλεξη αποτελεί το τρίτο και τελευταίο μέρος της εστίασης στον πυρήνα της διαδικασίας εξόρυξης πληροφορίας
- Θα μελετηθούν γραμμικά μοντέλα, μηχανές διανυσμάτων υποστήριξης, νευρωνικά δίκτυα και μετα-μαθησιακοί αλγόριθμοι

Γραμμικά μοντέλα

- Λειτουργούν εκ φύσεως με αριθμητικά χαρακτηριστικά
- Τυπική τεχνική αριθμητικής πρόβλεψης: *γραμμική παλινδρόμηση (linear regression)*
  - Το αποτέλεσμα είναι γραμμικός συνδυασμός των χαρακτηριστικών

$$x = w_0 + w_1 a_1 + w_2 a_2 + \ldots + w_k a_k$$

- Οι βαρύτητες υπολογίζονται από τα δεδομένα εκπαίδευσης
- Προβλεπόμενη τιμή για το πρώτο υπόδειγμα εκπαίδευσης $\mathbf{a}^{(1)}$

$$w_0 a_0^{(1)} + w_1 a_1^{(1)} + w_2 a_2^{(1)} + \ldots + w_k a_k^{(1)} = \sum_{j=0}^{k} w_j a_j^{(1)}$$

Ελαχιστοποίηση τετραγωνικού σφάλματος

- Επιλογή $k+1$ συντελεστών προς ελαχιστοποίηση του τετραγωνικού σφάλματος στα δεδομένα εκπαίδευσης
- Τετραγωνικό σφάλμα:

$$\sum_{i=1}^{n}\left(x^{(i)} - \sum_{j=0}^{k} w_j a_j^{(i)} \right)^2$$

- Ο υπολογισμός των συντελεστών είναι εφικτός *(και αξιόπιστος)* μόνο όταν ο αριθμός των διαθέσιμων υποδειγμάτων είναι *(κατά πολύ)* μεγαλύτερος του αριθμού των χαρακτηριστικών
- Μειονέκτημα γραμμικών μοντέλων (προφανώς…): **υπόθεση γραμμικότητας**
- Η ελαχιστοποίηση του *απολύτου σφάλματος* είναι περισσότερο απαιτητική

Ταξινόμηση (classification)

- *Οποιαδήποτε* τεχνική παλινδρόμησης μπορεί να χρησιμοποιηθεί, εκτός της αριθμητικής πρόβλεψης, επίσης και για ταξινόμηση
  - Εκπαίδευση: υλοποίηση μίας παλινδρόμησης για κάθε τάξη
    - Έξοδος ίση με 1 για τα υποδείγματα που ανήκουν στην τάξη, 0 για τα υπόλοιπα
  - Πρόβλεψη: επιλογή της τάξης του μοντέλου με τη μεγαλύτερη έξοδο (*τιμή συμμετοχής, membership value*)
- Η γραμμική αυτή μέθοδος καλείται *γραμμική παλινδρόμηση πολλαπλής απόκρισης, multi-response linear regression*

Λογαριθμική παλινδρόμηση (logistic regression)

- Πρόβλημα: παραβίαση κάποιων υποθέσεων κατά την υιοθέτηση γραμμικής παλινδρόμησης σε προβλήματα ταξινόμησης
  - Παράδειγμα: οι τιμές συμμετοχής μπορούν να προκύψουν εκτός του [0,1]
- *Λογαριθμική (logistic)* παλινδρόμηση: εναλλακτική της γραμμικής
  - Κατασκευάζει γραμμικό μοντέλο βασισμένο σε μετασχηματισμένη μεταβλητή-στόχο
  - Εφαρμόζεται κυρίως σε προβλήματα ταξινόμησης
  - Αποπειράται τον απευθείας υπολογισμό της πιθανότητας κάθε τάξης *(μέγιστη πιθανότητα, maximum likelihood)*
  - Γραμμικό μοντέλο πιθανότητας:

$$\log\left(\frac{P}{1-P}\right) = w_0 a_0 + w_1 a_1 + w_2 a_2 + \ldots + w_k a_k$$

Πιθανότητα τάξης

Παλινδρόμηση ανά ζεύγη (pairwise regression)

- Εναλλακτική μέθοδος παλινδρόμησης για ταξινόμηση:
  - Μία συνάρτηση παλινδρόμησης για κάθε *ζεύγος* τάξεων, με χρήση υποδειγμάτων των τάξεων αυτών και μόνο
  - Εκχώρηση εξόδου ίσης με +1 για το ένα μέλος του ζεύγους, –1 για το άλλο
- Η πρόβλεψη πραγματοποιείται μέσω ψηφοφορίας (voting)
  - Η τάξη που συλλέγει τις περισσότερες ψήφους επιλέγεται ως πρόβλεψη
- Περισσότερο ακριβής και υπολογιστικά δαπανηρή

Γενικά περί γραμμικών μοντέλων

- Ακατάλληλα στην περίπτωση που τα δεδομένα παρουσιάζουν μη γραμμικές εξαρτήσεις
  - Ωστόσο μπορούν να αποτελέσουν πρώτη ύλη για την κατασκευή περισσότερο πολύπλοκων σχημάτων (για παράδειγμα, model trees)
- Παράδειγμα: η γραμμική παλινδρόμηση πολλαπλής απόκρισης ορίζει ένα *υπερεπίπεδο (hyperplane)* για κάθε δύο τάξεις:
  - Το παράδειγμα εκχωρείται στην τάξη 1 αντί της τάξης 2 όταν…

$$(w_0^{(1)} - w_0^{(2)})a_0 + (w_1^{(1)} - w_1^{(2)})a_1 + (w_2^{(1)} - w_2^{(2)})a_2 + \ldots + (w_k^{(1)} - w_k^{(2)})a_k > 0$$

Γραμμική παλινδρόμηση με χρήση perceptron

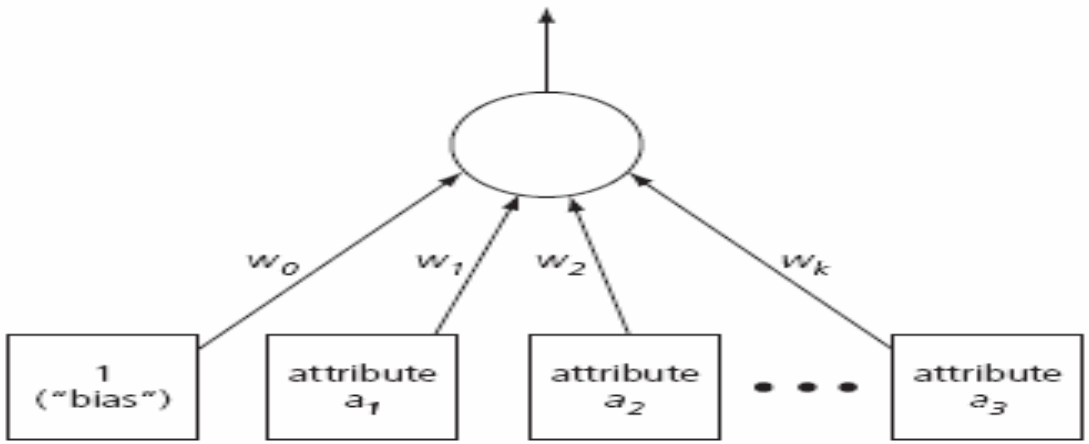
- Εναλλακτική προσέγγιση: εκμάθηση υπερεπιπέδου που διαχωρίζει τα υποδείγματα διαφορετικών τάξεων
  - Εάν τα δεδομένα δύναται να διαχωριστούν τέλεια σε ομάδες με χρήση υπερεπιπέδου, καλούνται ως *γραμμικά διαχωρίσιμα (linearly separable)*
  - Σε αυτή την περίπτωση, εφαρμόζεται ένας απλοϊκός αλγόριθμος για την εύρεση του διαχωριστικού υπερεπιπέδου
  - Εξίσωση υπερεπιπέδου:

$$w_0 + w_1 a_1 + w_2 a_2 + \ldots + w_k a_k = 0$$

    - a_i: τιμές χαρακτηριστικών, w_i: βαρύτητες

Ο αλγόριθμος εκμάθησης του perceptron

```
Set all weights to zero
Until all instances in the training data are classified
   correctly
      For each instance I in the training data
         If I is classified incorrectly by the perceptron
            If I belongs to the first class add it to
               the weight vector
            else subtract it from the weight vector
```



the perceptron

Παράδειγμα

- Υποθέτουμε την ύπαρξη δύο τάξεων
  - Αν άθροισμα>0 → τάξη 1, αλλιώς τάξη 2
- Αν υπόδειγμα *a* της τάξης 1 ταξινομείται λανθασμένα, ο αλγόριθμος αυξάνει την έξοδο του *a* κατά $a_0^2 + a_1^2 + \ldots + a_k^2$
  - Επομένως το υπερεπίπεδο μετακινείται προς τη σωστή για την ταξινόμηση του υποδείγματος *a* κατεύθυνση
- Αποδεικνύεται ότι ο αλγόριθμος συγκλίνει σε πεπερασμένο αριθμό επαναλήψεων εάν τα δεδομένα είναι γραμμικά διαχωρίσιμα
- Το υπερεπίπεδο που προκύπτει καλείται *perceptron* και αποτελεί τον πρόγονο των νευρωνικών δικτύων

Ο αλγόριθμος winnow

- Για προβλήματα ταξινόμησης και σύνολα δεδομένων με δυαδικά χαρακτηριστικά, υπάρχει άλλη μία εναλλακτική στο perceptron: *Winnow*
  - Οι δύο μέθοδοι διαφοροποιούνται ως προς την ενημέρωση των βαρυτήτων
  - Ο αλγόριθμος συνίσταται στην περίπτωση που το σύνολο δεδομένων περιέχει πολλά δυαδικά χαρακτηριστικά χαμηλής συσχέτισης
  - @weka: Winnow

While some instances are misclassified
for every instance *a*
classify *a* using the current weights
if the predicted class is incorrect
if *a* belongs to the first class
for each a_i that is 1, multiply w_i by *a*
(if a_i is 0, leave w_i unchanged)
otherwise
for each a_i that is 1, divide w_i by *a*
(if a_i is 0, leave w_i unchanged)

Επέκταση γραμμικής ταξινόμησης

- Οι γραμμικοί ταξινομητές αδυνατούν να μοντελοποιήσουν μη γραμμικά όρια τάξεων
- Τέχνασμα:
  - Απεικόνιση χαρακτηριστικών σε νέο χώρο που αποτελείται από συνδυασμούς των τιμών των χαρακτηριστικών
  - Για παράδειγμα: το σύνολο των γινομένων *n* παραγόντων που μπορούν να κατασκευαστούν από τα χαρακτηριστικά
- Παράδειγμα με δύο χαρακτηριστικά και $n = 3$:

$$x = w_1 a_1^{\ 3} + w_2 a_1^2 a_2 + w_3 a_1 a_2^2 + w_3 a_2^3$$

- Πολυώνυμα επαρκούς βαθμού μπορούν να προσεγγίσουν μη γραμμικά όρια απόφασης με την απαιτούμενη ακρίβεια

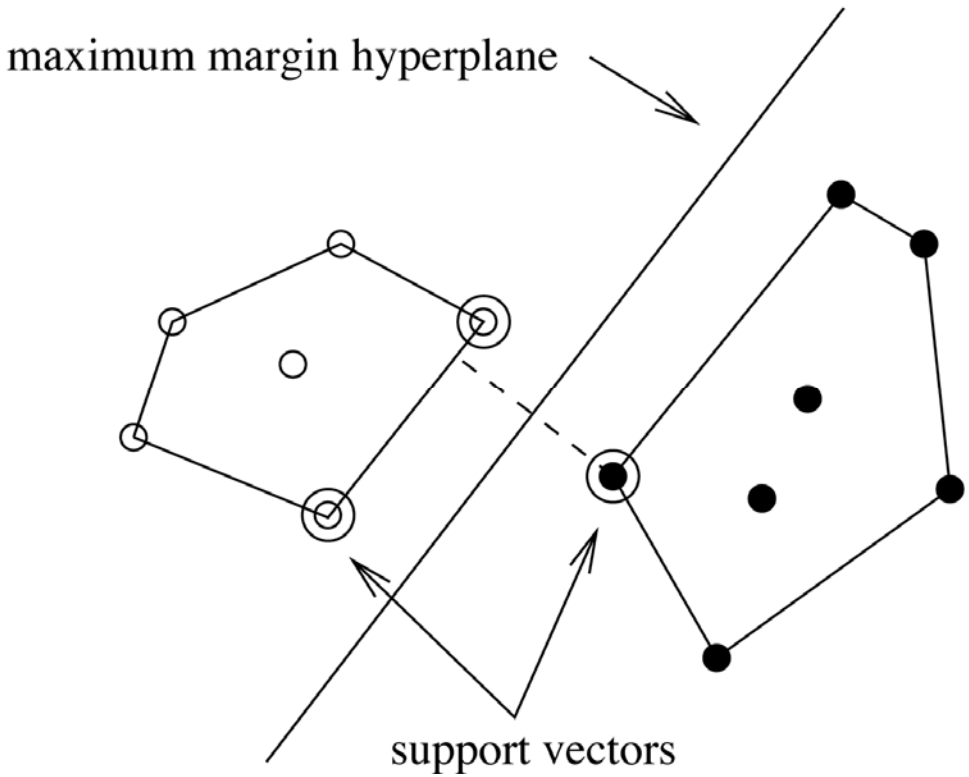
Μειονεκτήματα της προσέγγισης

- 1ο μειονέκτημα: ταχύτητα
  - 10 χαρακτηριστικά, $n = 5 \Rightarrow$ >2000 συντελεστές
  - Χρήση γραμμικής παλινδρόμησης με επιλογή χαρακτηριστικών
  - Ο χρόνος εκτέλεσης είναι ανάλογος του κύβου του αριθμού των χαρακτηριστικών
- 2ο μειονέκτημα: υπερπροσαρμογή
  - Ο αριθμός των συντελεστών είναι μεγάλος σε σχέση με τον αριθμό των υποδειγμάτων εκπαίδευσης
  - Η δυνατότητα γενίκευσης των μοντέλων είναι περιορισμένη

Μηχανές διανυσμάτων υποστήριξης (support vector machines)

- Οι *μηχανές διανυσμάτων υποστήριξης* αποτελούν αλγορίθμους που επιτυγχάνουν εκμάθηση γραμμικών ταξινομητών
- Ανθεκτικοί στην υπερπροσαρμογή, καθώς μαθαίνουν ένα ειδικό γραμμικό όριο απόφασης:
  - Το *υπερεπίπεδο μέγιστου περιθωρίου* (*maximum margin hyperplane)*
- Χαμηλό υπολογιστικό κόστος, ακόμη και στην περίπτωση μη γραμμικότητας
  - Χρήση μαθηματικού τεχνάσματος για την αποφυγή δημιουργίας "ψευδο-χαρακτηριστικών"
  - Ο μη γραμμικός χώρος ανακύπτει με έμμεσο τρόπο

Το υπερεπίπεδο μέγιστου περιθωρίου



Το υπερεπίπεδο μέγιστου περιθωρίου

- Έστω σύνολο δεδομένων δύο γραμμικά διαχωρίσιμων τάξεων
  - Υπάρχει υπερεπίπεδο στο χώρο των υποδειγμάτων που ταξινομεί χωρίς λάθος όλα τα υποδείγματα εκπαίδευσης
- Το υπερεπίπεδο μέγιστου περιθωρίου είναι εκείνο το οποίο επιτυγχάνει το μέγιστο διαχωρισμό μεταξύ των τάξεων
- Ως *κυρτό περίβλημα (convex hull)* ενός συνόλου σημείων ορίζεται το μικρότερο δυνατό κυρτό πολύγωνο που εσωκλείει το σύνολο των σημείων:
  - Το περίγραμμά του αναδεικνύεται όταν ενωθεί κάθε σημείο του συνόλου με κάθε άλλο σημείο με μία ευθεία γραμμή
- Οι δύο τάξεις είναι γραμμικά διαχωρίσιμες → τα κυρτά περιβλήματά τους δεν επικαλύπτονται
- Ως υπερεπίπεδο μέγιστου περιθωρίου ορίζεται εκείνο με τη μέγιστη απόσταση από τα κυρτά περιβλήματα

Διανύσματα υποστήριξης

- Τα υποδείγματα με τη μικρότερη απόσταση από το υπερεπίπεδο μέγιστου περιθωρίου καλούνται *διανύσματα υποστήριξης (support vectors)*
- Το σύνολο των διανυσμάτων υποστήριξης καθορίζει με μοναδικό τρόπο το υπερεπίπεδο μέγιστου περιθωρίου για το πρόβλημα εκμάθησης
  - Επομένως όλα τα υπόλοιπα υποδείγματα μπορούν να διαγραφούν χωρίς να επηρεάσουν τη θέση και τον προσανατολισμό του!
- Κατά συνέπεια, το υπερεπίπεδο μπορεί να γραφεί ως

$$x = w_0 + w_1 a_1 + w_2 a_2$$

$$x = b + \sum_{i \text{ is supp. vector}} \alpha_i y_i \mathbf{a}(i) \bullet \mathbf{a}$$

  - y_i: η τιμή τάξεως του υποδείγματος εκπαίδευσης **a(***i***)**, (*i*: διάνυσμα υποστήριξης)
  - b & a_i: αριθμητικές παράμετροι που καθορίζονται από τον αλγόριθμο εκμάθησης
  - **a**: υπόδειγμα ελέγχου

Εύρεση διανυσμάτων υποστήριξης

$$x = b + \sum_{i \text{ is supp. vector}} \alpha_i y_i \mathbf{a}(i) \bullet \mathbf{a}$$

- Διάνυσμα υποστήριξης: υπόδειγμα εκπαίδευσης για το οποίο $\alpha_i > 0$
- Εύρεση διανυσμάτων υποστήριξης και καθορισμός α_i & b
  Πρόβλημα *βελτιστοποίησης με δευτεροβάθμιους περιορισμούς* (*constrained quadratic optimization)*
  - Υπάρχουν διαθέσιμα εργαλεία για την αντιμετώπιση αυτού ακριβώς του προβλήματος
  - Ωστόσο, εξειδικευμένοι αλγόριθμοι είναι περισσότερο γρήγοροι
  - Παράδειγμα: αλγόριθμος *sequential minimal optimization* του Platt (@weka: SMO)
- Σημείωση: προϋπόθεση διαχωρίσιμων δεδομένων!

Μη γραμμικές Μηχανές Διανυσμάτων Υποστήριξης

- Τα "ψευδο-χαρακτηριστικά" αντιπροσωπεύουν συνδυασμούς χαρακτηριστικών
- Η υπερπροσαρμογή δεν είναι πιθανή, καθώς το υπερεπίπεδο μέγιστου περιθωρίου είναι σταθερό
  - Μετακινείται μόνο κατά την προσθήκη ή αφαίρεση υποδειγμάτων εκπαίδευσης που συνιστούν διανύσματα υποστήριξης
  - Υπάρχει συνήθως μικρός αριθμός διανυσμάτων υποστήριξης συγκριτικά με το μέγεθος του συνόλου εκπαίδευσης
- Ο υπολογιστικός χρόνος όμως ακόμη αποτελεί ζήτημα
  - Σε κάθε υπολογισμό του διανυσματικού γινομένου υπεισέρχονται και τα "ψευδο-χαρακτηριστικά"

Μαθηματικό τέχνασμα

- Αποκλεισμός των "ψευδο-χαρακτηριστικών"
  - Υπολογισμός του διανυσματικού γινομένου πριν την υλοποίηση της μη γραμμικής απεικόνισης
    - Παράδειγμα: αντί του $x = b + \sum_{i \text{ is supp. vector}} \alpha_i y_i \mathbf{a}(i) \bullet \mathbf{a}$

      υπολόγισε το $x = b + \sum_{i \text{ is supp. vector}} \alpha_i y_i (\mathbf{a}(i) \bullet \mathbf{a})^n$
    - Αντιστοιχεί σε απεικόνιση στο χώρο των υποδειγμάτων, επεκτεινόμενο σε όλα τα γινόμενα *n* χαρακτηριστικών
- Η απεικόνιση καλείται *συνάρτηση πυρήνα (kernel function)*
  - Ως τώρα χρησιμοποιήθηκε η πολυωνυμική $\mathbf{(xy)^n}$, ωστόσο υπάρχει πληθώρα άλλων διαθέσιμων προς χρήση συναρτήσεων
    - Εκκίνηση με n=1 (γραμμικό μοντέλο) και βηματική αύξηση αυτού ως την παύση της βελτίωσης του υπολογιζόμενου σφάλματος

Θόρυβος

- Υπόθεση: τα δεδομένα είναι διαχωρίσιμα (στον αρχικό ή σε μετασχηματισμένο χώρο)
- Οι μηχανές διανυσμάτων υποστήριξης μπορούν να εφαρμοστούν επίσης σε δεδομένα με θόρυβο με την εισαγωγή μίας παραμέτρου θορύβου C
- Η παράμετρος C περιορίζει την επιρροή οποιουδήποτε από τα υποδείγματα εκπαίδευσης επί των οριακών περιοχών απόφασης
  - Εισάγεται δηλαδή ο περιορισμός: $0 \leq \alpha_i \leq C$
  - Το πρόβλημα συνεχίζει να αφορά δευτεροβάθμια βελτιστοποίηση
- Ο βέλτιστος προσδιορισμός της παραμέτρου C επιτυγχάνεται πειραματικά

Αραιά δεδομένα

- Οι αλγόριθμοι μηχανών διανυσμάτων υποστήριξης επιταχύνονται στην περίπτωση που τα δεδομένα είναι *αραιά* (*sparse*, δηλαδή πολλές τιμές είναι μηδενικές)
- Ο λόγος; Ο υπολογισμός πολλών διανυσματικών γινομένων
- Αραιά δεδομένα $\Rightarrow$ χρονικά αποδοτικός υπολογισμός των γινομένων
  - Επανάληψη μόνο επί των μη μηδενικών τιμών
- Δύνανται να επεξεργασθούν αραιά σύνολα δεδομένων με δεκάδες χιλιάδων χαρακτηριστικών

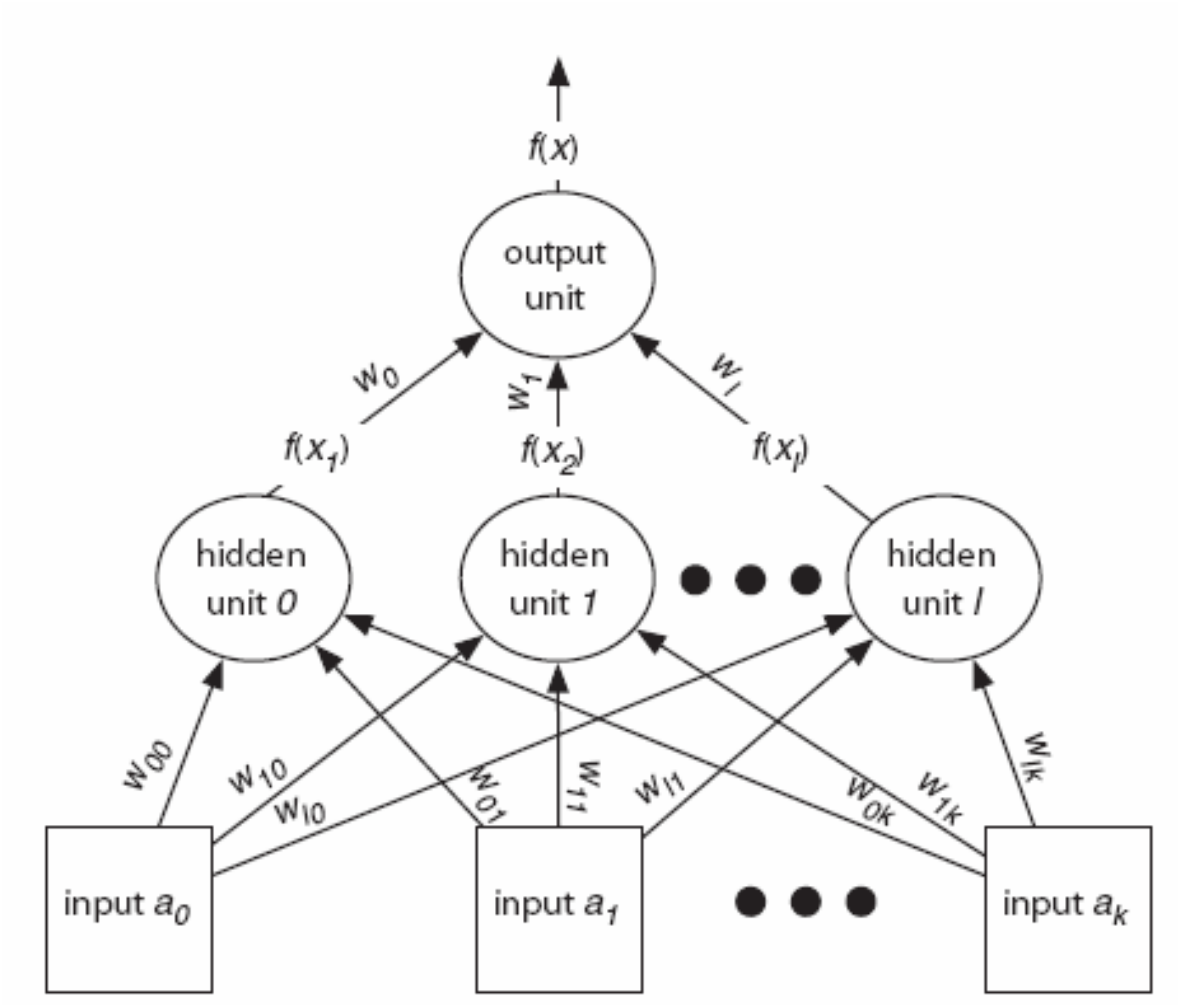
Παλινδρόμηση διανυσμάτων υποστήριξης (support vector regression)

- Η έννοια του υπερεπιπέδου μέγιστου περιθωρίου εφαρμόζεται μονάχα στην ταξινόμηση
- Για αριθμητικά χαρακτηριστικά: (@weka: SMOreg)
  - Βασική ιδέα: εύρεση συνάρτησης που προσεγγίζει τα σημεία εκπαίδευσης μέσω της ελαχιστοποίησης του σφάλματος πρόβλεψης
  - Κρίσιμη διαφοροποίηση: το σύνολο των σημείων με απόκλιση μεγαλύτερη από μία –καθορισμένη από το χρήστη- παράμετρο ε αποβάλλονται
  - Ο κίνδυνος υπερπροσαρμογής περιορίζεται μέσω της ταυτόχρονης προσπάθειας μεγιστοποίησης της ομαλότητας της συνάρτησης
  - Διανύσματα υποστήριξης: όλα εκείνα τα σημεία που βρίσκονται κοντά στην επιφάνεια της συνάρτησης
  - Απαιτείται συμβιβασμός μεταξύ του σφάλματος πρόβλεψης και της ομαλότητας της συνάρτησης
    - Έλεγχος μέσω επιβολής ανώτατου ορίου C στις ανώτατες τιμές των συντελεστών a_i

Πολυεπίπεδα (multilayer) perceptrons

- Σύνδεση πολλών απλών όμοιων προς το perceptron μοντέλων σε μία ιεραρχική δομή: *νευρωνικά δίκτυα (neural networks)*
  - Ικανά να απεικονίσουν μη γραμμικά όρια αποφάσεων
  - Ένα perceptron περιγράφεται συχνά ως ένας τεχνητός νευρώνας
  - Οι νευρώνες του ανθρώπινου εγκεφάλου διασυνδέονται σε μαζική κλίμακα, επιτρέποντας την αποσύνθεση ενός προβλήματος σε υποπροβλήματα που μπορούν να επιλυθούν σε επίπεδο νευρώνα
    - Η παρατήρηση αυτή ενέπνευσε τη δημιουργία δικτύων τεχνητών νευρώνων

Παράδειγμα νευρωνικού δικτύου



Εκμάθηση πολυεπίπεδου perceptron

- Ζητήματα:
  - Εκμάθηση της δομής του δικτύου
  - Εκμάθηση των βαρών σύνδεσης
- *Ανάστροφη μετάδοση (backpropagation)*: ένας συγκριτικά απλοϊκός αλγόριθμος για τον καθορισμό των βαρών ενός δικτύου δεδομένης δομής
- Δομή δικτύου: συνήθως καθορίζεται με πειραματισμούς – ίσως και με την κατάλληλη δόση γνώσης πεδίου
  - Συχνά ένα κρυφό (hidden) επίπεδο είναι επαρκές
  - Ο κατάλληλος αριθμός νευρώνων για αυτό το επίπεδο καθορίζεται μέσω μεγιστοποίησης της εκτιμώμενης ακρίβειας

Ανάστροφη μετάδοση (backpropagation)

- Γνωστά: δεδομένα, δομή δικτύου
- Ζητούμενα: κατάλληλα βάρη για τις συνδέσεις του δικτύου
- *Επικλινής κάθοδος (gradient descent)*
  - Τροποποίηση των βαρών των ενώσεων που οδηγούν στους νευρώνες του κρυφού επιπέδου με βάση την ισχύ της συνεισφοράς κάθε νευρώνα στην τελική πρόβλεψη
  - *Sigmoid function, learning rate, standardization, early stopping, momentum*

@weka

- ***LeastMedSq*** Robust regression using the median rather than the mean
- ***LinearRegression*** Standard linear regression
- ***Logistic*** Build linear logistic regression models
- ***MultilayerPerceptron*** Backpropagation neural network
- ***PaceRegression*** Build linear regression models using Pace regression
- ***RBFNetwork*** Implements a radial basis function network

@weka

- ***SimpleLinearRegression*** Learn a linear regression model based on a single attribute
- ***SimpleLogistic*** Build linear logistic regression models with built-in attribute selection
- ***SMO*** Sequential minimal optimization algorithm for support vector classification
- ***SMOreg*** Sequential minimal optimization algorithm for support vector regression
- ***VotedPerceptron*** Voted perceptron algorithm
- ***Winnow*** Mistake-driven perceptron with multiplicative updates

"Μετα" μαθησιακά σχήματα

- Θεμελιώδης ιδέα: κατασκευή πολλών "εμπειρογνωμόνων" ("experts"), ανάδειξη πλειοψηφούσας γνώμης
  - "εμπειρογνώμονας": ένα μοντέλο που δημιουργήθηκε με τεχνικές μηχανική μάθησης
- Πλεονέκτημα:
  - Συχνά βελτιώνει σημαντικά την προβλεπτική ικανότητα
- Μειονέκτημα:
  - Τα εξαγόμενα είναι πολύ δύσκολο να αναλυθούν
- Σχήματα:
  - *Εμφωλίαση (bagging)*
  - *Ενδυνάμωση (boosting)*
  - *Συσσώρευση (stacking)*
  - *Κώδικες διόρθψσης σφαλμάτων εξόδου (error-correcting output codes)*

Εφαρμόζονται τόσο για ταξινόμηση, όσο και για αριθμητική πρόβλεψη

Εμφωλίαση (bagging)

- Συνδυασμός προβλέψεων μέσω καταμέτρησης ψήφων / εύρεσης μέσου όρου
  - *Bagging* = *bootstrap aggregating*
  - Η απλούστερη μέθοδος!
  - Κάθε μοντέλο λαμβάνει ισοδύναμη βαρύτητα
- "Εξιδανικευμένη" έκδοση:
  - Δειγματοληψία αρκετών συνόλων εκπαίδευσης μεγέθους *n* (αντί του ενός συνόλου μεγέθους *n*)
  - Κατασκευή ενός ταξινομητή για κάθε σύνολο εκπαίδευσης
  - Συνδυασμός των προβλέψεων των ταξινομητών
- Όταν το σχήμα εκμάθησης είναι *ασταθές* (*unstable*), η εμφωλίαση βελτιώνει σχεδόν πάντα την αποδοτικότητα
  - Καθώς διαφορετικά μικρές αλλαγές στα δεδομένα εκπαίδευσης μπορούν να επιφέρουν σημαντικές μεταβολές στο μοντέλο
    - για παράδειγμα, δένδρα απόφασης

Αποσύνθεση προκατάληψης-διακύμανσης

- Για την ανάλυση της επιρροής επί της απόδοσης *καθενός* συνόλου εκπαίδευσης
- Υπόθεση άπειρων ταξινομητών, κατασκευασμένων με βάση διαφορετικά σύνολα εκπαίδευσης μεγέθους *n*
- Για κάθε μαθησιακό σχήμα,
  - *Bias (προκατάληψη)* = αναμενόμενο σφάλμα του μετα-ταξινομητή σε νέα δεδομένα
  - *Variance (διακύμανση)* = αναμενόμενο σφάλμα λόγω του συγκεκριμένου συνόλου εκπαίδευσης που χρησιμοποιήθηκε
- Συνολικό αναμενόμενο σφάλμα: bias + variance

Γενικά περί εμφωλίασης

- Η εμφωλίαση λειτουργεί καθώς μειώνει τη *διακύμανση* μέσω της καταμέτρησης ψήφων / εύρεσης μέσου όρου
  - Σε κάποιες παθολογικές καταστάσεις το συνολικό σφάλμα μπορεί να αυξηθεί
  - Συνήθως, η βελτίωση είναι ανάλογη του αριθμού των ταξινομητών
- Πρόβλημα: στην πράξη, μόνο ένα δεδομένο είναι διαθέσιμο!
- Λύση: παραγωγή νέων συνόλων μεγέθους *n* μέσω δειγματοληψίας *με επανατοποθέτηση*
- Σημαντική βελτίωση ιδιαίτερα στην περίπτωση θορύβου
- Για αριθμητικά χαρακτηριστικά: εύρεση μέσου όρου και όχι σύστασης πλειοψηφίας
- @weka: *Bagging, Decorate, RandomCommittee, Vote*

Ταξινομητές εμφωλίασης

Κατασκευή μοντέλου

```
Let n be the number of instances in the training
data
For each of t iterations:
  Sample n instances from training set
    (with replacement)
  Apply learning algorithm to the sample
  Store resulting model
```

Ταξινόμηση

```
For each of the t models:
  Predict class of instance using model
Return class that is predicted most often
```

Ενδυνάμωση (boosting)

- Υλοποιεί επίσης ψηφοφορία (για ταξινόμηση) / μέσο όρο (για αριθμητική πρόβλεψη) για τον συγκερασμό των εξόδων ξεχωριστών μοντέλων
- Ωστόσο, αποδίδει βαρύτητα στα μοντέλα με βάση την απόδοσή τους
- **Επαναληπτική μέθοδος**: τα νέα μοντέλα επηρεάζονται από την απόδοση εκείνων που κατασκευάστηκαν προηγούμενα
  - Ενθάρρυνση εστίασης των νέων μοντέλων στα λανθασμένα –με βάση τα προηγούμενα μοντέλα– ταξινομημένα υποδείγματα
  - Διαισθητική αιτιολόγηση: τα μοντέλα –ως εμπειρογνώμονες– πρέπει να αλληλοσυμπληρώνονται
- Διάφορες τεχνοτροπίες ενδυνάμωσης
  - Μία ευρέως χρησιμοποιούμενη: *AdaBoostM1* (για ταξινόμηση)

AdaBoost.M1

Κατασκευή μοντέλου

```
Assign equal weight to each training instance
For t iterations:
  Apply learning algorithm to weighted dataset,
    store resulting model
  Compute model's error e on weighted dataset
  If e = 0 or e > 0.5:
    Terminate model generation
  For each instance in dataset:
    If classified correctly by model:
       Multiply instance's weight by e/(1-e)
  Normalize weight of all instances
```

Ταξινόμηση

```
Assign weight = 0 to all classes
For each of the t models (or fewer):
  For the class this model predicts
    add –log e/(1-e) to this class's weight
Return class with highest weight
```

Περισσότερα περί ενδυνάμωσης

- Η ενίσχυση απαιτεί βαρύτητες, ωστόσο
  - Μπορεί να εφαρμοστεί και χωρίς αυτές…
    - Επαναληπτική δειγματοληψία με πιθανότητες εμφάνισης καθορισμένες από βάρη
    - μειονέκτημα: δεν χρησιμοποιούνται όλα τα υποδείγματα
    - πλεονέκτημα: αν σφάλμα> 0.5, εφικτή η επανάληψη της δειγματοληψίας
- Η ιδέα πηγάζει από τη *θεωρία υπολογιστικής μάθησης (computational learning theory)*
- Θεωρητικά:
  - Το σφάλμα εκπαίδευσης φθίνει εκθετικά
- Επίσης:
  - Λειτουργεί ικανοποιητικά εάν οι ταξινομητές βάσης δεν είναι περίπλοκοι και
  - Το σφάλμα τους δεν μεγεθύνεται ταχέως

Περισσότερα περί ενδυνάμωσης

- Συνέχιση της διαδικασίας ενδυνάμωσης μετά το μηδενισμό του σφάλματος εκπαίδευσης;
- Αινιγματική παρατήρηση:
  το σφάλμα γενίκευσης συνεχίζει να μειώνεται!
- Ένας ισχυρός συνδυασμένος ταξινομητής μπορεί να δημιουργηθεί από απλοϊκούς ταξινομητές!
  - Η ενδυνάμωση λειτουργεί με *αδύναμα (weak)* αρχικά μοντέλα
    - Για παράδειγμα, με *decision stumps* (δένδρα ενός επιπέδου)
  - Μοναδική προϋπόθεση: το σφάλμα δεν ξεπερνά το 0.5
- LogitBoost:
  περισσότερο εξεζητημένο σχήμα ενδυνάμωσης, χρησιμοποιεί μέθοδο παλινδρόμησης ως αρχικό μοντέλο
- @weka: *AdaBoostM1, LogitBoost, MultiBoostAB, RacedIncrementalLogitBoost*

Συσσώρευση (stacking)

- Stacking/ stacked generalization
  - Λιγότερο ευρέως χρησιμοποιούμενη΄, έναντι της εμφωλίασης και ενδυνάμωσης
  - Δεν υπάρχει κοινά αποδεκτή μέθοδος υλοποίησης
  - Δεν χρησιμοποιείται για το συνδυασμό μοντέλων του ίδιου τύπου
- Για τον συγκερασμό των προβλέψεων των αρχικών μοντέλων δεν πραγματοποιείται ψηφοφορία, αλλά χρησιμοποιείται μετα-μοντέλο εκμάθησης
  - Αρχικά μοντέλα: *επίπεδο-0*
  - Μετα-μοντέλα: *επίπεδο-1*
  - Οι προβλέψεις των αρχικών μοντέλων αποτελούν τις εισόδους του μετα-μοντέλου
- Τα αρχικά μοντέλα είναι συνήθως διαφορετικά σχήματα εκμάθησης
- Αδύνατη η χρήση των προβλέψεων επί των δεδομένων εκπαίδευσης για τη δημιουργία δεδομένων για το μετα-μοντέλο!
  - Στη θέση τους, απαιτείται η χρήση σχήματος τύπου διασταυρωμένης επικύρωσης
- Δύσκολη η θεωρητική ανάλυση

Περισσότερα περί συσσώρευσης

- Εάν τα αρχικά μοντέλα εξάγουν πιθανότητες, αυτές μπορούν να χρησιμοποιηθούν ως είσοδοι για το μετα-μαθησιακό σχήμα
- Επιλογή αλγορίθμου για χρήση ως μετα-μαθησιακό σχήμα
  - Οποιοδήποτε σχήμα!
  - Προτιμούνται μοντέλα σχετικά σφαιρικά και λεία, καθώς
    - Τα αρχικά μοντέλα υλοποιούν όλη την μάθηση
    - Με αυτό τον τρόπο μειώνεται ο κίνδυνος υπερπροσαρμογής
- Η συσσώρευσης μπορεί να εφαρμοστεί επίσης για αριθμητική πρόβλεψη
- @weka: *Stacking, StackingC, Grading*

Κώδικες διόρθωσης σφαλμάτων εξόδου (error-correcting output codes, ECOC)

- Μετατροπή προβλήματος πολλών τάξεων σε δυαδικά προβλήματα
  - Απλοϊκό σχήμα:
    ένας κώδικας ανά τάξη
- Ιδέα: χρήση *κωδικών διόρθωσης σφαλμάτων (error-correcting codes)*
  - Οι αρχικοί ταξινομητές δίνουν
    1011111, αληθής τάξη = ??
- Χρησιμοποίηση κωδικών λέξεων που εμφανίζουν μεγάλη *απόσταση κατά Hamming (Hamming distance)* μεταξύ κάθε ζεύγους
  - Μπορεί να διορθώσει ως και $(d-1)/2$
    σφάλματα μοναδιαίου bit

class	class vector
a	1000
b	0100
c	0010
d	0001

class	class vector
a	1111111
b	0000111
c	0011001
d	0101010

Περισσότερα περί ECOC

- Δύο κριτήρια:
  - *Διαχωρισμός σειράς*:
    ελάχιστη απόσταση μεταξύ των σειρών
  - *Διαχωρισμός στηλών*:
    ελάχιστη απόσταση μεταξύ των στηλών
    - και των συμπληρωμάτων τους
    - Αίτιο; Εάν οι στήλες είναι πανομοιότυπες, οι αρχικοί ταξινομητές θα οδηγηθούν πιθανά στα ίδια σφάλματα
    - Η διόρθωση σφάλματος αδυνατίζει όταν τα σφάλματα είναι συσχετιζόμενα
- 3 τάξεις $\Rightarrow$ μόνο 2^3 δυνατές στήλες
  - 4 εκ των οποίων είναι συμπληρωματικές
  - Ανέφικτος ο διαχωρισμός σειρών και στηλών
- Η μέθοδος λειτουργεί μόνο με προβλήματα περισσοτέρων των 3 τάξεων

Εξαντλητικοί ECOC

- *Εξαντλητικός (exhaustive)* κώδικας για k τάξεις:
  - Οι στήλες αποτελούν κάθε δυνατή συμβολοσειρά k-στοιχείων…
  - … με εξαίρεση των συμπλήρωμάτων τους και των συμβολοσειρών με μόνο 1/0
  - Κάθε λέξη κώδικα περιέχει $2^{k-1} - 1$ bits

Εξαντλητικός κώδικας, $k = 4$

class	class vector
a	1111111
b	0000111
c	0011001
d	0101010

- Τάξη 1: λέξη κώδικα αποτελούμενη μόνο από άσσους
- Τάξη 2: 2^{k-2} μηδενικά αποτελούμενα από $2^{k-2} - 1$ άσσους
- Τάξη i : εναλλαγές από 2^{k-i} μηδενικά και $2^{k-i} - 1$ άσσους
- Πολλές τάξεις $\Rightarrow$ το κόστος της μεθόδου γίνεται απαγορευτικό
  - Ο αριθμός των στηλών αυξάνει εκθετικά

@weka

- ***AdaBoostM1*** Boost using the AdaBoostM1 method
- ***AdditiveRegression*** Enhance the performance of a regression method by iteratively fitting the residuals
- ***AttributeSelectedClassifier*** Reduce dimensionality of data by attribute selection
- ***Bagging*** Bag a classifier; works for regression too
- ***ClassificationViaRegression*** Perform classification using a regression method
- ***CostSensitiveClassifier*** Make its base classifier cost sensitive
- ***CVParameterSelection*** Perform parameter selection by cross-validation
- ***Decorate*** Build ensembles of classifiers by using specially constructed artificial training examples

@weka

- ***FilteredClassifier*** Run a classifier on filtered data
- ***Grading*** Metalearners whose inputs are base-level predictions that have been marked as correct or incorrect
- ***LogitBoost*** Perform additive logistic regression
- ***MetaCost*** Make a classifier cost-sensitive
- ***MultiBoostAB*** Combine boosting and bagging using the MultiBoosting method
- ***MultiClassClassifier*** Use a two-class classifier for multiclass datasets
- ***MultiScheme*** Use cross-validation to select a classifier from several candidates
- ***OrdinalClassClassifier*** Apply standard classification algorithms to problems with an ordinal class value

@weka

- ***RacedIncrementalLogitBoost*** Batch-based incremental learning by racing logit-boosted committees
- ***RandomCommittee*** Build an ensemble of randomizable base classifiers
- ***RegressionByDiscretization*** Discretize the class attribute and employ a classifier
- ***Stacking*** Combine several classifiers using the stacking method
- ***StackingC*** More efficient version of stacking
- ***ThresholdSelector*** Optimize the F-measure for a probabilistic classifier
- ***Vote*** Combine classifiers using average of probability estimates or numeric predictions

Τέλος

Επόμενη διάλεξη:

ΠαρουσιάσειςΕργασιών