

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ»

ΣΗΜΕΙΩΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ ΚΑΙ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Αναπλ. Καθηγητής Γεώργιος Α. Τσιχριντζής

ΠΕΙΡΑΙΕΥΣ 2005

1. Ιστορική Αναδρομή

Στην ιστορία της έρευνας γύρω από το πρόβλημα της εκμάθησης διακρίνει κανείς τέσσερις περιόδους οι οποίες χαρακτηρίζονται από τέσσερα σημαντικά γεγονότα:

- (i) την κατασκευή των πρώτων μηχανών εκμάθησης
- (ii) την κατασκευή των θεμελίων της θεωρίας
- (iii) την κατασκευή των νευρωνικών δικτύων
- (iv) την κατασκευή δομών εναλλακτικών προς τα νευρωνικά δίκτυα.

Σε διαφορετικές περιόδους, διαφορετικά θέματα θεωρούνταν ως σημαντικά. Συνολικά, αυτή η έρευνα αποτελεί μία πολύπλοκη και (αντιφατική) εικόνα της διερεύνησης του προβλήματος της εκμάθησης.

(i) Το perceptron του Rosenblatt (δεκαετία του '60)

Πριν από περισσότερα από 50 χρόνια, ο F. Rosenblatt πρότεινε το πρώτο μοντέλο μιας μηχανής εκμάθησης, το οποίο ονόμασε perceptron. Αυτό το γεγονός αποτέλεσε και την πραγματική αρχή της μαθηματικής ανάλυσης των διαδικασιών εκμάθησης. Η ιδέα του perceptron δεν ήταν καινούρια. Είχε ήδη συζητηθεί στη νευροφυσιολογική βιβλιογραφία για πολλά χρόνια. Η καινοτομία του Rosenblatt έγκειται στο ότι περιέγραψε το μοντέλο ως ένα πρόγραμμα για υπολογιστές και απέδειξε με απλά πειράματα ότι το μοντέλο μπορεί να γενικευτεί. Το perceptron κατασκευάστηκε για να λύνει προβλήματα αναγνώρισης προτύπου και στην απλούστερη περίπτωση κατασκευάζει ένα κανόνα για το διαχωρισμό δεδομένων σε δύο διαφορετικές κατηγορίες χρησιμοποιώντας δεδομένα παραδείγματα.

(ii) Κατασκευή των θεμελίων της θεωρίας εκμάθησης (δεκαετίες του '60 και του '70).

Αμέσως μετά τη δημοσιοποίηση των πειραμάτων με το perceptron προτάθηκαν και άλλοι τύποι μηχανών εκμάθησης, όπως η madaline του B. Widrow ή οι μήτρες εκμάθησης του K. Steinbuch. Παράλληλα, άρχισε και η κατασκευή εξειδικευμένου υλικού εκμάθησης. Σε αντίθεση με το perceptron, όμως, οι μηχανές αυτές θεωρήθηκαν από την αρχή ως εργαλεία για την επίλυση συγκεκριμένων προβλημάτων παρά ως γενικά μοντέλα του φαινομένου της εκμάθησης. Για την επίλυση συγκεκριμένων προβλημάτων, αναπτύχθηκαν επίσης προγράμματα υπολογιστών, συμπεριλαμβανομένων προγραμμάτων για την κατασκευή λογικών συναρτήσεων διαφόρων τύπων. Αυτά τα προγράμματα δεν επηρέασαν τη μελέτη των γενικών φαινομένων εκμάθησης.

Το επόμενο βήμα στην κατασκευή ενός γενικού τύπου μηχανής εκμάθησης έγινε το 1986 όταν προτάθηκε η τεχνική της λεγόμενης οπισθοδιάδοσης (*backpropagation*) για την ταυτόχρονη εύρεση των βαρών πολλών νευρώνων. Η μέθοδος αποτέλεσε την αρχή μιας νέας εποχής στην ιστορία των μηχανών εκμάθησης.

Ανάμεσα στην κατασκευή του perceptron (1960) και την υλοποίηση της τεχνικής οπισθοδιάδοσης (1986) δεν συνέβη καμιά εκθαμβωτική πρόοδος στο χώρο της εφαρμοσμένης ανάλυσης. Αντίθετα, τα χρόνια αυτά ήταν εξαιρετικά παραγωγικά στο χώρο της στατιστικής θεωρίας εκμάθησης.

(iii) Νευρωνικά δίκτυα (δεκαετία του '80)

Το 1986, διάφοροι ερευνητές πρότειναν ανεξάρτητα μια μέθοδο για την ταυτόχρονη κατασκευή των διανυσματικών συντελεστών όλων των νευρώνων του perceptron με τη μέθοδο της οπισθοδιάδοσης (LeCun, 1986, Rumelhart, Hinton και Williams, 1986). Η ιδέα της μεθόδου είναι εξαιρετικά απλή και βασίζεται στη χρήση μεθόδων βελτιστοποίησης *απότομης καθόδου* (*steepest descent*).

(iv) Επιστροφή στην αφετηρία (δεκαετία του '90)

Την τελευταία τριετία έχει σημειωθεί κάποια αλλαγή όσον αφορά στα νευρωνικά δίκτυα. Περισσότερη προσοχή εστιάζεται τώρα σε δομές εναλλακτικές των νευρωνικών δικτύων, όπως στη μελέτη της μεθόδου των *συναρτήσεων ακτινικής βάσης* (*radial basis functions*) (Powell, 1992 για επισκόπηση). Όπως και τη δεκαετία του '60, τα νευρωνικά δίκτυα ονομάζονται πολυστρωματικά perceptrons. Τα ειδικά και προχωρημένα κεφάλαια της στατιστικής θεωρίας εκμάθησης προσελκύουν τώρα περισσότερους ερευνητές. Πιο συγκεκριμένα, τόσο η αρχή της *ελαχιστοποίησης του δομικού ρίσκου* (*structural risk minimization*) όσο και η αρχή του *ελάχιστου μήκους περιγραφής* (*minimum description length*) αποτελούν δημοφιλή θέματα της ανάλυσης. Επίσης διαδόθηκε η ανάλυση με βάση τη θεωρία μικρού δείγματος, παρά την ασυμπτωτική. Εμφανίζεται έτσι με επιστροφή στη θεμελίωση της στατιστικής θεωρίας εκμάθησης, αλλά και στην κατεύθυνση της σύνθεσης βέλτιστων αλγόριθμων εκμάθησης.

2. Εισαγωγή

Σκοπός του μαθήματος είναι να δοθεί μια συστηματική παρουσίαση των βασικών θεμάτων της Αναγνώρισης Προτύπου, δηλαδή της επιστήμης που προσπαθεί να αναγνωρίσει αυτόματα χρήσιμες κανονικότητες σε θορυβώδη και περίπλοκα περιβάλλοντα. Η ανάπτυξη της επιστήμης αυτής ξεκίνησε από τη δεκαετία του '60 και εξελίχθηκε παράλληλα με την ανάπτυξη άλλων επιστημών, όπως η στατιστική, η θεωρία επικοινωνιών, η θεωρία των αυτομάτων, η θεωρία του αυτομάτου ελέγχου, η επιχειρησιακή έρευνα, η βιολογία, η ψυχολογία, η γλωσσολογία, και η πληροφορική.

Η επιστήμη της αναγνώρισης προτύπου είναι εξαιρετικά ευρεία και καμμία θεωρία από μόνη της δεν καλύπτει πλήρως όλα τα θέματα με τα οποία ασχολείται. Κάθε πεδίο εφαρμογής έχει τα δικά του μοναδικά χαρακτηριστικά που διαμορφώνουν και την κατάλληλη προσέγγιση.

Η πλέον γενική θεωρία είναι η *θεωρία της ταξινόμησης* (*classification theory*), η οποία βασίζεται στην στατιστική θεωρία αποφάσεων και παρέχει τις μαθηματικές διαδικασίες για την αναπαράσταση τους με τη μορφή αφηρημένων διανυσμάτων. Γενικές, διαδικασίες που με επιτυχία κατασκευάζουν διανυσματικές αναπαραστάσεις δεν υπάρχουν. Κάθε πρόβλημα επιδέχεται μία διαδικασία που ταιριάζει με τα χαρακτηριστικά του.

Μέχρι σήμερα, η ικανότητα των υπολογιστών να *αντιλαμβάνονται* (*perceive*) το περιβάλλον τους είναι σχετικά περιορισμένη. Μία ποικιλία αισθητηρίων διατίθεται για τη μετατροπή

φωτός, ήχου, θερμοκρασίας, κλπ. σε ηλεκτρικά σήματα. Όταν το περιβάλλον είναι ελεγχόμενο και τα σήματα έχουν μια απλή ερμηνεία, όπως στα συνήθη τερματικά εισόδου υπολογιστών, τα προβλήματα μηχανικής αντίληψης είναι τετριμμένα. Όταν όμως ο υπολογιστής πρέπει να διαβάσει χειρόγραφους χαρακτήρες ή να αναλύσει βιοϊατρικές φωτογραφίες, τότε το πρόβλημα της αντίληψης είναι πολύ πιο σημαντικό.

Από την άλλη πλευρά, η ευκολία με την οποία ζώα ή και έντομα αντιλαμβάνονται το περιβάλλον τους είναι και ενθαρρυντική και απογοητευτική ταυτόχρονα. Ψυχολογικές και φυσιολογικές μελέτες έχουν αποκαλύψει σημαντικές λεπτομέρειες για τη συμπεριφορά των ζώων, όσον αφορά στην αντίληψη, αλλά ακόμα δεν την κατανοούμε σε βαθμό που να μπορούμε να την επαναλάβουμε.

Η έλλειψη μιας πλήρους θεωρίας αντίληψης δεν εμποδίζει την προσπάθεια επίλυσης απλών προβλημάτων. Πολλά από αυτά αναφέρονται στην ταξινόμηση προτύπου, δηλαδή στην ανάθεση ενός αντικειμένου ή γεγονότος σε μία από διάφορες προκαθορισμένες κλάσεις.

Παράδειγμα:

Ας υποθέσουμε ότι επιθυμούμε την ταξινόμηση ξυλείας με βάση το δέντρο από το οποίο προήλθε και με χρήση ενός οπτικού αισθητηρίου. Η κάμερα «αισθάνεται» εικόνες τις οποίες στέλνει στον *εξαγωγέα χαρακτηριστικών (feature extractor)* ο οποίος ελαττώνει τα δεδομένα σε ένα μικρό αριθμό *χαρακτηριστικών (features)* ή ιδιοτήτων που διακρίνουν τις εικόνες ξυλείας από διαφορετικά δένδρα. Οι τιμές των χαρακτηριστικών δίνονται στη συνέχεια σε ένα *ταξινομητή (classifier)* ο οποίος και τελικά αποφασίζει για τον τύπο της ξυλείας.

Τα διάφορα είδη ξυλείας διαφέρουν ως προς τη φωτεινότητα και επομένως ένα χαρακτηριστικό θα μπορούσε να αποτελεί η φωτεινότητα ποσοτικοποιημένη σε μια μεταβλητή x . Με δειγματοληψία από διάφορα είδη ξυλείας, μπορούμε να καταλήξουμε σε ιστογράμματα όπως το επόμενο:

Επομένως, μπορούμε να ορίσουμε μια *κρίσιμη τιμή* x_0 της μεταβλητής x , έτσι ώστε να θεωρούμε την ξυλεία ως τύπου Β αν το x που μετράμε είναι μεγαλύτερο του x_0 , αλλιώς ως τύπου Α.

Από μόνο του το χαρακτηριστικό x δεν επαρκεί για ακριβή ταξινόμηση της ξυλείας. Ένα άλλο χαρακτηριστικό μπορεί να αποτελέσει η γράμμωση της ξυλείας όπως ποσοτικοποιείται σε μια μεταβλητή με τη μέτρηση του μεγέθους και της συχνότητας από άσπρο σε μαύρο στη εικόνα. Έχουμε τώρα δύο χαρακτηριστικά (x_1 και x_2), δηλαδή ο εξαγωγέας χαρακτηριστικών έχει ελαττώσει μια ολόκληρη εικόνα σε ένα διάνυσμα χαρακτηριστικών $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Ο ταξινομητής πρέπει να χωρίσει το χώρο (επίπεδο) τιμών του \underline{x} σε δύο περιοχές, έτσι ώστε η πρώτη να αντιστοιχεί σε ξυλεία τύπου Α μόνο και η δεύτερη σε τύπου Β μόνο.

Είναι φανερό ότι η σχεδίαση του εξαγωγέα χαρακτηριστικών πρέπει να διευκολύνει τη σχεδίαση του ταξινομητή. Γενικά, ο σχεδιασμός του εξαγωγέα χαρακτηριστικών είναι περισσότερο εξαρτημένος από το συγκεκριμένο πρόβλημα από τη σχεδίαση του ταξινομητή.

Σε ορισμένα προβλήματα αναγνώρισης προτύπου και μηχανικής αντίληψης, δεν επαρκεί η απλή ταξινόμηση, αλλά απαιτείται και περιγραφή των επιμέρους μερών μιας εικόνας και των μεταξύ τους σχέσεων – ένα εξαιρετικά δύσκολο πρόβλημα. Μία προσέγγιση στο πρόβλημα είναι με τη βοήθεια εννοιών, της θεωρίας γλωσσών και την παραγωγή ενός γλωσσολογικού μοντέλου ανάλυσης εικόνας. Αυτή η προσέγγιση αποτελεί και το αντικείμενο της Ανάλυσης Εικόνας.

3. Μπεϋζιανή Θεωρία Αποφάσεων (ΜΘΑ)

Η ΜΘΑ είναι η βασική στατιστική προσέγγιση στο πρόβλημα της ταξινόμησης προτύπου και βασίζεται στην περιγραφή του προβλήματος με πιθανοτικούς όρους. Στην πραγματικότητα αποτελεί μια αυστηρή παρουσίαση διαδικασιών κοινής λογικής. Ας ξαναγυρίσουμε στο παράδειγμα της ταξινόμησης της ξυλείας και ας υποθέσουμε ότι υπάρχουν μόνο δύο διαφορετικά είδη ξυλείας. Θεωρούμε μια τυχαία μεταβλητή ω , η οποία περιγράφει το είδος της ξυλείας και παίρνει τιμές ω_A (για ξυλεία τύπου A) και ω_B (για ξυλεία τύπου B). Η μεταβλητή ω υποδηλώνει την κατάσταση της φύσης (*state of nature*). Ας υποθέσουμε ότι γνωρίζουμε εκ των προτέρων (*prior, a priori*) πιθανότητες $P(\omega_A)$ και $P(\omega_B)$ να είναι η ξυλεία τύπου A ή B, αντίστοιχα. Προφανώς $P(\omega_A), P(\omega_B) \geq 0$ και $P(\omega_A) + P(\omega_B) = 1$.

Ας υποθέσουμε ότι είμαστε αναγκασμένοι να ταξινομήσουμε ένα κομμάτι ξύλου χωρίς να το δούμε. Τότε ο κανόνας ταξινόμησης θα είναι προφανώς: Αποφάσισε ω_A αν $P(\omega_A) > P(\omega_B)$, αλλιώς αποφάσισε ω_B . Αυτός ο κανόνας δίνει πάντα το ίδιο αποτέλεσμα. Αν $P(\omega_A) \gg P(\omega_B)$, τότε συνήθως θα δουλεύει καλά και η πιθανότητα λάθους θα είναι μικρή. Αν $P(\omega_A) = P(\omega_B)$, τότε η απόφαση θα είναι σωστή μόνο τις μισές φορές. Η πιθανότητα λάθους είναι γενικά η ελάχιστη των $P(\omega_A)$ και $P(\omega_B)$ και θα δούμε ότι κανείς κανόνας δεν μπορεί να δώσει μικρότερη πιθανότητα λάθους.

Ας υποθέσουμε τώρα ότι έχουμε μια μέτρηση φωτεινότητας x ως επιπλέον ένδειξη (χαρακτηριστικό) για την ταξινόμηση. Υποθέτουμε ότι η μεταβλητή x είναι μια συνεχής τυχαία μεταβλητή της οποίας η κατανομή εξαρτάται από την κατάσταση της φύσης. Συμβολίζουμε με $p(x|\omega_j)$ τη δεσμευμένη συνάρτηση πυκνότητας πιθανότητας (*conditional probability density function*) της μεταβλητής x δεδομένης της κατάστασης της φύσης ω_j . Θεωρώντας ότι γνωρίζουμε τις πιθανότητες $P(\omega_j)$ και τις συναρτήσεις $p(x|\omega_j)$, ο κανόνας του Bayes δίνει την εκ των υστέρων (*posterior, a posteriori*) πιθανότητα $P(\omega_j | x)$ της κατάστασης ω_j δεδομένης της μέτρησης x :

$$P(\omega_j | x) = \frac{p(x | \omega_j)P(\omega_j)}{\sum_{j=A}^B p(x | \omega_j)P(\omega_j)}$$

Ας υπολογίσουμε την πιθανότητα λάθους δεδομένης της μέτρησης x

$$P(\text{λάθος}|x) = \begin{cases} P(\omega_A | x), & \text{όταν αποφασίζουμε } \omega_B \\ P(\omega_B | x), & \text{όταν αποφασίζουμε } \omega_A \end{cases}$$

Επομένως, η πιθανότητα λάθους $P(\text{λάθος} | x)$ δεδομένης της παρατήρησης x ελαχιστοποιείται αν $P(\omega_A|x) > P(\omega_B|x)$ όταν αποφασίζουμε ω_A και αν $P(\omega_A|x) < P(\omega_B|x)$ όταν αποφασίζουμε ω_B . Ο κανόνας αυτός ελαχιστοποιεί τη μέση πιθανότητα λάθους. Πράγματι:

$$\begin{aligned} P(\text{λάθος}) &= \int P(\text{λάθος}, x) dx \\ &= \int P(\text{λάθος} | x) p(x) dx \end{aligned}$$

και, αφού $p(x) \geq 0$, η πιθανότητα $P(\text{λάθος})$ ελαχιστοποιείται όταν $P(\text{λάθος}|x)$ ελαχιστοποιείται.

Περίληπτικά, ο κανόνας Bayes λέει:

Αποφάσισε ω_A όταν $P(\omega_A|x) > P(\omega_B|x)$, αλλιώς αποφάσισε ω_B .

Σε ισοδύναμη μορφή

Αποφάσισε ω_A όταν $p(x|\omega_A)P(\omega_A) > p(x|\omega_B)P(\omega_B)$, αλλιώς αποφάσισε ω_B .

Υποθέτοντας ότι $p(x|\omega_B) > 0$ για κάθε x , έχουμε τον ισοδύναμο κανόνα:

Αποφάσισε ω_A όταν $\frac{p(x | \omega_A)}{p(x | \omega_B)} > \frac{P(\omega_B)}{P(\omega_A)}$, αλλιώς αποφάσισε ω_B .

Η έκφραση $p(x|\omega_j)$ ονομάζεται *πιθανοφάνεια (likelihood)* του ω_j και το πηλίκο $\frac{p(x | \omega_A)}{p(x | \omega_B)}$ *λόγος πιθανοφανειών (likelihood ratio)*.

Περιληπτικά, για ταξινόμηση σε μια από δύο κλάσεις, ο κανόνας του Bayes οδήγησε στο σχηματισμό του λόγου πιθανοφανειών και τη σύγκρισή του με κάποια τιμή. Ο ίδιος κανόνας μπορεί να εξαχθεί και με την προσέγγιση Neyman – Pearson.

Γενική Μπεϋζιανή Θεωρία Αποφάσεων

Πέραν του παραδείγματος αυτού, έχουμε γενικά να αντιμετωπίσουμε τα εξής:

- περισσότερα από ένα χαρακτηριστικά
- περισσότερες από δύο κλάσεις
- δράσεις πέρα από την ταξινόμηση
- συνάρτηση κόστους άλλη από την μέση πιθανότητα λάθους.

Ας υποθέσουμε ότι διαθέτουμε ένα διάνυσμα χαρακτηριστικών \underline{x} , ένα σύνολο $\Omega = \{\omega_1, \omega_2, \dots, \omega_s\}$ s κλάσεων και ένα σύνολο $A = \{\alpha_1, \alpha_2, \dots, \alpha_a\}$ a δράσεων. Έστω $\lambda(\alpha_i | \omega_j)$ το κόστος της δράσης α_i όταν η κατάσταση της φύσης είναι ω_j . Υποθέτουμε ότι το διάνυσμα χαρακτηριστικών \underline{x} είναι ένα τυχαίο διάνυσμα διαστάσεως d με δεσμευμένη συνάρτηση πυκνότητας πιθανότητας $p(\underline{x} | \omega_j)$. Τέλος $P(\omega_j)$ είναι η εκ των προτέρων πιθανότητα η φύση να βρίσκεται στην κατάσταση ω_j . Η εκ των υστέρων πιθανότητα $P(\omega_j | \underline{x})$ δίνεται από τον κανόνα του Bayes:

$$P(\omega_j | \underline{x}) = \frac{p(\underline{x} | \omega_j) P(\omega_j)}{\sum_{j=1}^s p(\underline{x} | \omega_j) P(\omega_j)}$$

Το αναμενόμενο κόστος όταν παρατηρούμε το διάνυσμα \underline{x} και αναλαμβάνουμε τη δράση α_i είναι:

$$R(\alpha_i | \underline{x}) = \sum_{j=1}^s \lambda(\alpha_i | \omega_j) P(\omega_j | \underline{x}).$$

Η ποσότητα $R(\alpha_i | \underline{x})$ ονομάζεται *δεσμευμένο ρίσκο (conditional risk)*. Ο σκοπός μας είναι να διαλέξουμε εκείνη τη δράση που ελαχιστοποιεί το δεσμευμένο κόστος. Η διαδικασία οδηγεί ακριβώς στον κανόνα του Bayes.

Ένας *κανόνας απόφασης (decision rule)* είναι μία συνάρτηση $\alpha(\underline{x})$ που υποδεικνύει ποια δράση να αναλάβουμε για κάθε δυνατή παρατήρηση \underline{x} . Πιο συγκεκριμένα, η *συνάρτηση απόφασης (decision function)* $\alpha(\underline{x})$ αντιστοιχίζει σε κάθε \underline{x} μια από τις δυνατές δράσεις $\alpha_1, \alpha_2, \dots, \alpha_a$.

Το συνολικό ρίσκο R που αντιστοιχεί σε ένα κανόνα απόφασης είναι:

$$R = \int R(a(\underline{x}) | \underline{x}) p(\underline{x}) d\underline{x}$$

και, αφού $p(\underline{x}) > 0$, ελαχιστοποιείται όταν επιλεγεί ο κανόνας $a(\underline{x})$ ώστε να ελαχιστοποιείται το δεσμευμένο ρίσκο $R(a(\underline{x}) | \underline{x})$. Έχουμε επομένως τον κανόνα:

υπολόγισε $R(a_i | \underline{x}) = \sum_{j=1}^s \lambda(a_i | \omega_j) P(\omega_j | \underline{x})$, $i=1, 2, \dots, a$ και διάλεξε τη δράση εκείνη που δίνει το μικρότερο δεσμευμένο ρίσκο.

Όταν δύο δράσεις δίνουν το ίδιο ακριβώς δεσμευμένο ρίσκο, δεν έχει σημασία ποια από τις δύο θα επιλεγεί – η επιλογή μεταξύ των δύο γίνεται στην τύχη.

4. Ταξινόμηση σε μια από δύο κλάσεις

Ας εξετάσουμε την ειδική περίπτωση της γενικής μπεϋζιανής θεωρίας όπου $\Omega = \{\omega_1, \omega_2\}$ και $A = \{a_1, a_2\}$. Πιο συγκεκριμένα, η δράση a_j αντιστοιχεί στο να αποφασίσουμε ότι ω_j είναι η κατάσταση της φύσης. Ορίζουμε $\lambda_{ij} = \lambda(a_i | \omega_j)$ ως το κόστος της απόφασης a_i όταν η πραγματική κατάσταση της φύσης είναι ω_j . Το δεσμευμένο ρίσκο είναι:

$$R(a_1 | \underline{x}) = \lambda_{11} P(\omega_1 | \underline{x}) + \lambda_{12} P(\omega_2 | \underline{x})$$

$$R(a_2 | \underline{x}) = \lambda_{21} P(\omega_1 | \underline{x}) + \lambda_{22} P(\omega_2 | \underline{x})$$

Από τη γενική θεωρία, ο βέλτιστος κανόνας υποδεικνύει να αποφασίσουμε ω_1 αν $R(a_1 | \underline{x}) < R(a_2 | \underline{x})$. Ισοδύναμα, αποφασίζουμε ω_1 αν

$$(\lambda_{21} - \lambda_{11}) P(\omega_1 | \underline{x}) > (\lambda_{12} - \lambda_{22}) P(\omega_2 | \underline{x}).$$

Συνήθως, το κόστος ενός λάθους είναι μεγαλύτερο του κόστους μιας σωστής απόφασης, δηλαδή $\lambda_{21} > \lambda_{11}$ και $\lambda_{12} > \lambda_{22}$. Με βάση τον νόμο του Bayes, έχουμε τον ισοδύναμο κανόνα

$$(\lambda_{21} - \lambda_{11}) p(\underline{x} | \omega_1) P(\omega_1) > (\lambda_{12} - \lambda_{22}) p(\underline{x} | \omega_2) P(\omega_2).$$

ή

$$\frac{p(\underline{x} | \omega_1)}{p(\underline{x} | \omega_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \cdot \frac{P(\omega_2)}{P(\omega_1)}$$

που είναι της μορφής της σύγκρισης του λόγου πιθανοφανειών με μια τιμή.

Ταξινόμηση ελάχιστης πιθανότητας λάθους

Μια καλή επιλογή (πχ., σε σχεδιασμό τηλεπικοινωνιακών δεκτών) για τα κόστη λ_{ij} είναι η *συμμετρική ή μηδέν – ένα (zero – one) επιλογή*:

$$\lambda(\alpha_i | \omega_j) = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases} \quad i, j = 1, 2, \dots, c \quad (\text{ίσος αριθμός κλάσεων και δράσεων})$$

Εχουμε:

$$\begin{aligned} R(\alpha_i | \underline{x}) &= \sum_j \lambda(\alpha_i | \omega_j) P(\omega_j | \underline{x}) \\ &= \sum_{j \neq i} P(\omega_j | \underline{x}) \\ &= 1 - P(\omega_i | \underline{x}) \end{aligned}$$

Επομένως, ο κανόνας ελάχιστης πιθανότητας λάθους είναι:

“Αποφάσισε ω_i αν $P(\omega_i | \underline{x}) > P(\omega_j | \underline{x}) \forall j \neq i$ ”

Ταξινομητές (classifiers), συναρτήσεις διάκρισης (discriminant functions) και επιφάνειες απόφασης (decision surfaces)

Όπως είδαμε, στην περίπτωση της ταξινόμησης σε μία από πολλές (c) κλάσεις, ο ταξινομητής λαμβάνει γενικά τη μορφή:

“Ταξινόμησε το διάνυσμα χαρακτηριστικών \underline{x} στην κλάση ω_i αν $g_i(\underline{x}) > g_j(\underline{x}) \forall j \neq i$ ”,

όπου οι συναρτήσεις $g_j(\underline{x})$, $j = 1, 2, \dots, c$ ονομάζονται συναρτήσεις διάκρισης. Ο ταξινομητής είναι ένας αλγόριθμος που υπολογίζει c συναρτήσεις διάκρισης και επιλέγει την κλάση για την οποία η αντίστοιχη συνάρτηση είναι μέγιστη.

Οι συναρτήσεις διάκρισης γενικά υπολογίζονται ως $g_i(\underline{x}) = -R(a_i|\underline{x})$, αλλά η επιλογή δεν είναι μοναδική. Γενικά, μπορούμε να αντικαταστήσουμε ένα σύνολο από $g_i(\underline{x})$ με $f(g_i(\underline{x}))$, όπου f είναι οποιαδήποτε αύξουσα συνάρτηση. Παρά ένα τέτοιο μετασχηματισμό, οι κανόνες απόφασης παραμένουν αναλλοίωτοι. Το αποτέλεσμα των κανόνων απόφασης είναι να χωριστεί ο χώρος (σύνολο δυνατών τιμών) του διάνυσματος χαρακτηριστικών σε c περιοχές απόφασης (decision region) R_1, R_2, \dots, R_c και, κάθε φορά που το \underline{x} εμπίπτει στην περιοχή R_j , αποφασίζει ω_j . Τα σύνορα απόφασης (decision boundaries) μεταξύ περιοχών απόφασης ορίζονται από εξισώσεις της μορφής $g_i(\underline{x}) = g_j(\underline{x})$ (σύνορο απόφασης μεταξύ R_i και R_j). Όταν το \underline{x} εμπίπτει σε ένα σύνορο απόφασης (γεγονός μηδενικής πιθανότητας για τυχαίο διάνυσμα \underline{x} με συνεχή κατανομή), τότε δεν έχει σημασία σε ποια από τις αντίστοιχες κλάσεις ανήκει.

Πιθανότητες λάθους

Ας θεωρήσουμε την περίπτωση δύο κλάσεων με αντίστοιχες περιοχές απόφασης R_1 και R_2 . Τότε η πιθανότητα λάθους είναι :

$$\begin{aligned} P_r \{\text{λάθος}\} &= P_r \{\underline{x} \in R_2, \omega_1\} + P_r \{\underline{x} \in R_1, \omega_2\} \\ &= P_r \{\underline{x} \in R_2 | \omega_1\}P(\omega_1) + P_r \{\underline{x} \in R_1 | \omega_2\}P(\omega_2) \\ &= P(\omega_1) \int_{R_2} p(\underline{x} | \omega_1) d\underline{x} + P(\omega_2) \int_{R_1} p(\underline{x} | \omega_2) d\underline{x} \end{aligned}$$

5. Η κανονική (Γκαουσιανή) κατανομή

Η δομή των μπεύζιανών ταξινομητών καθορίζεται κυρίως από τις δεσμευμένες συναρτήσεις πυκνότητας πιθανότητας $p(\underline{x}|\omega_i)$. Η πολυμεταβλητή κανονική (γκαουσιανή) κατανομή έχει μελετηθεί σε βάθος, γεγονός που οφείλεται κυρίως σε μαθηματική ευκολία στη χρήση της, αλλά και στη σχετικά μικρή διασπορά των χαρακτηριστικών στην ίδια κλάση.

Η μονοδιάστατη κανονική κατανομή

Έχει τη μορφή

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

για την οποία

$$E\{x\} = \int_{-\infty}^{\infty} xp(x)dx = \mu$$

και

$$E\{(x-\mu)^2\} = \int_{-\infty}^{\infty} (x-\mu)^2 p(x)dx = \sigma^2$$

Οι δύο παράμετροι μ (μέση τιμή) και σ^2 (διασπορά) καθορίζουν πλήρως την κανονική κατανομή. Δείγματα από την κανονική κατανομή τείνουν να ομαδοποιούνται γύρω από τη μέση τους τιμή: περίπου 95% των δειγμάτων θα βρίσκονται στο διάστημα $|x - \mu| \leq 2\sigma$.

Η πολυμεταβλητή κανονική κατανομή

Η γενική πολυμεταβλητή συνάρτηση πυκνότητας πιθανότητας είναι

$$p(\underline{x}) = \frac{1}{(2\pi)^{d/2} |\underline{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\underline{x} - \underline{\mu})^T \underline{\Sigma}^{-1} (\underline{x} - \underline{\mu})\right],$$

όπου \underline{x} είναι ένα διάνυσμα στήλη με d συνιστώσες, $\underline{\mu}$ είναι το μέσο διάνυσμα (με d συνιστώσες) και $\underline{\Sigma}$ είναι ένας $d \times d$ πίνακας συμμεταβλητότητας (covariance matrix). Επίσης, $(\underline{x} - \underline{\mu})^T$ είναι ο ανάστροφος του $\underline{x} - \underline{\mu}$, $\underline{\Sigma}^{-1}$ ο αντίστροφος του πίνακα $\underline{\Sigma}$ και $|\underline{\Sigma}|$ η ορίζουσα του $\underline{\Sigma}$.

Έχουμε:

$$\begin{aligned} E\{\underline{x}\} &= \underline{\mu} \\ E\{(\underline{x} - \underline{\mu})(\underline{x} - \underline{\mu})^T\} &= \underline{\Sigma} \end{aligned}$$

Γενικά, ο πίνακας $\underline{\Sigma}$ είναι συμμετρικός και θετικά ημιορισμένος ($\underline{u}^T \underline{\Sigma} \underline{u} \geq 0$ για κάθε \underline{u}). Στις πρακτικές περιπτώσεις που ενδιαφέρουν αυτό το μάθημα, θα υποθέσουμε ότι ο πίνακας $\underline{\Sigma}$ είναι θετικά ορισμένος ($\underline{u}^T \underline{\Sigma} \underline{u} > 0$ για κάθε \underline{u}). Το διαγώνιο στοιχείο σ_{ii} του πίνακα $\underline{\Sigma}$ είναι η διασπορά του x_i , ενώ το στοιχείο σ_{ij} είναι η συμμεταβλητότητα των x_i και x_j , δηλαδή

$$\sigma_{ij} = E\{(x_i - \mu_i)(x_j - \mu_j)\}.$$

Συναρτήσεις διάκρισης για την κανονική κατανομή

Είδαμε ότι η ταξινόμηση ελάχιστης πιθανότητας λάθους επιτυγχάνεται με τη χρήση συναρτήσεων διάκρισης της μορφής:

$$g_i(\underline{x}) = \log p(\underline{x} | \omega_i) + \log P(\omega_i).$$

Ο υπολογισμός των εκφράσεων αυτών είναι ιδιαίτερα εύκολος αν οι πυκνότητες $p(\underline{x} | \omega_i)$ είναι πολυμεταβλητές κανονικές. Έστω ότι $p(\underline{x} | \omega_i) \sim N(\underline{\mu}_i, \underline{\Sigma}_i)$. Τότε:

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \underline{\Sigma}_i^{-1} (\underline{x} - \underline{\mu}_i) - \frac{d}{2} \log(2\pi) - \frac{1}{2} \log |\underline{\Sigma}_i| + \log P(\omega_i)$$

Ας δούμε διάφορες ειδικές περιπτώσεις:

$$i) \underline{\underline{\Sigma}}_i = \sigma^2 \underline{\underline{I}}$$

Στην περίπτωση αυτή, τα χαρακτηριστικά είναι στατιστικώς ανεξάρτητα και το καθένα έχει διασπορά σ^2 . Βρίσκουμε ότι οι συναρτήσεις διάκρισης έχουν τη μορφή

$$\begin{aligned} g_i(\underline{x}) &= -\frac{(\underline{x} - \underline{\mu}_i)^T (\underline{x} - \underline{\mu}_i)}{2\sigma^2} + \log P(\omega_i) \\ &= -\frac{\|\underline{x} - \underline{\mu}_i\|^2}{2\sigma^2} + \log P(\omega_i) \end{aligned}$$

Αν επιπλέον, οι εκ των προτέρων πιθανότητες είναι ίδιες για όλες τις κλάσεις, τότε οι όροι $\log P(\omega_i)$ δεν έχουν σημασία. Ο κανόνας ταξινόμησης μετράει τις ευκλίδειες αποστάσεις $\|\underline{x} - \underline{\mu}_i\|$ του \underline{x} από κάθε ένα από τα μέσα διανύσματα $\underline{\mu}_i$ και αναθέτει το \underline{x} στην κατηγορία με τον πλησιέστερο μέσο. Ένας τέτοιος ταξινομητής ονομάζεται *ταξινομητής ελάχιστης απόστασης* (*minimum distance classifier*).

Επιστρέφοντας στην περίπτωση των άνισων εκ των προτέρων πιθανοτήτων, αναπτύσσουμε την τετραγωνική μορφή $(\underline{x} - \underline{\mu}_i)^T (\underline{x} - \underline{\mu}_i)$ και βρίσκουμε

$$g_i(\underline{x}) = -\frac{1}{2\sigma^2} [\underline{x}^T \underline{x} - 2\underline{\mu}_i^T \underline{x} + \underline{\mu}_i^T \underline{\mu}_i] + \log P(\omega_i)$$

Ο όρος $\underline{x}^T \underline{x}$ είναι κοινός για όλα τα i , άρα μπορούμε ισοδύναμα να χρησιμοποιήσουμε τις γραμμικές συναρτήσεις διάκρισης

$$g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{i0}$$

$$\text{με } \underline{w}_i = \frac{1}{\sigma^2} \underline{\mu}_i$$

$$\text{και } w_{i0} = -\frac{1}{2\sigma^2} \underline{\mu}_i^T \underline{\mu}_i + \log P(\omega_i).$$

$$2) \underline{\underline{\Sigma}}_i = \underline{\underline{\Sigma}}$$

Στην περίπτωση ίσων πινάκων συμμεταβλητότητας για όλες τις κλάσεις, ως συναρτήσεις διάκρισης μπορούν να χρησιμοποιηθούν οι επόμενες

$$g_i(\underline{x}) = -\frac{1}{2} (\underline{x} - \underline{\mu}_i)^T \underline{\underline{\Sigma}}^{-1} (\underline{x} - \underline{\mu}_i) + \log P(\omega_i)$$

Επειδή οι όροι $\underline{x}^T \underline{\underline{\Sigma}}^{-1} \underline{x}$ είναι ανεξάρτητοι των κλάσεων, μπορούμε να χρησιμοποιήσουμε ισοδύναμα τις γραμμικές συναρτήσεις διάκρισης:

$$g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{i0}$$

$$\text{με } \underline{w}_i = \underline{\Sigma}_i^{-1} \underline{\mu}_i$$

$$\text{και } w_{i0} = -\frac{1}{2} \underline{\mu}_i^T \underline{\Sigma}_i^{-1} \underline{\mu}_i + \log P(\omega_i)$$

3) Οποιαδήποτε $\underline{\Sigma}_i$

Στη γενική περίπτωση, ο μόνος όρος που είναι ανεξάρτητος από τις κλάσεις είναι ο όρος $\frac{d}{2} \log(2\pi)$. Επομένως, οι συναρτήσεις διάκρισης είναι:

$$g_i(\underline{x}) = \underline{x}^T \underline{W}_i \underline{x} + \underline{\omega}_i^T \underline{x} + \omega_{i0},$$

$$\text{όπου } \underline{W}_i = -\frac{1}{2} \underline{\Sigma}_i^{-1}$$

$$\underline{w}_i = \underline{\Sigma}_i^{-1} \underline{\mu}_i$$

$$w_{i0} = -\frac{1}{2} \underline{\mu}_i^T \underline{\Sigma}_i^{-1} \underline{\mu}_i - \frac{1}{2} \log |\underline{\Sigma}_i| + \log P(\omega_i).$$

6. Εκτίμηση Παραμέτρων και Εκμάθηση με Επιτήρηση

Σε προηγούμενα μαθήματα, είδαμε ότι μπορούμε να σχεδιάσουμε το βέλτιστο ταξινομητή αν γνωρίζουμε τις εκ των προτέρων πιθανότητες $P(\omega_j)$ και τις δεσμευμένες πυκνότητες $p(\underline{x}|\omega_j)$. Δυστυχώς, επακριβής και λεπτομερής γνώση των δεσμευμένων πυκνοτήτων $p(\underline{x}|\omega_j)$ είναι δύσκολη, ειδικά όταν το διάνυσμα χαρακτηριστικών \underline{x} είναι μεγάλης διάστασης. Η εκτίμηση της πυκνότητας $p(\underline{x}|\omega_j)$ απαιτεί μεγάλο αριθμό δειγμάτων που συνήθως δεν είναι διαθέσιμος. Είναι δυνατό, όμως, να διαθέτουμε γνώση του προβλήματος που να μας επιτρέπει να υποθέσουμε ότι γνωρίζουμε μια παραμετρική οικογένεια στην οποία ανήκουν οι συναρτήσεις πυκνότητας πιθανότητας και οι μόνιμοι άγνωστοι είναι οι συγκεκριμένες τιμές παραμέτρων. Έτσι, υποβιβάζεται το πρόβλημα από πρόβλημα εκτίμησης μιας συνάρτησης $p(\underline{x}|\omega)$ σε πρόβλημα εκτίμησης ενός μικρού αριθμού παραμέτρων.

Οι κοινότερες μέθοδοι εκτίμησης παραμέτρων είναι της μέγιστης πιθανοφάνειας (*maximum likelihood*) και η μπεϋζιανή. Αν και οδηγούν σε παρόμοιους κανόνες, διαφέρουν στη φιλοσοφία τους. Στη μπεϋζιανή προσέγγιση, θα παρατηρήσουμε το φαινόμενο της εκμάθησης (*learning*): κάθε νέα παρατήρηση οξύνει των εκ των υστέρων συνάρτηση πυκνότητας πιθανότητας.

Θα διακρίνουμε σε εκμάθηση με επιτήρηση (*supervised learning*) και σε εκμάθηση χωρίς επιτήρηση (*unsupervised learning*). Η διαφορά έγκειται στο ότι κατά την εκμάθηση με επιτήρηση γνωρίζουμε την κατάσταση της φύσης σε αντίθεση με την εκμάθηση χωρίς επιτήρηση.

Εκτίμηση μέγιστης πιθανοφάνειας

Υποθέτουμε ότι σε ένα πρόβλημα c κλάσεων διαθέτουμε c σύνολα δειγμάτων X_1, \dots, X_c , όπου τα δείγματα X_j έχουν ληφθεί ανεξάρτητα από την κατανομή $p(\underline{x}|\omega_j)$. Υποθέτουμε ότι η πυκνότητα $p(\underline{x}|\omega_j)$ έχει γνώστη μορφή που εξαρτάται από ένα διάνυσμα παραμέτρων $\underline{\theta}_j$. Για παράδειγμα, $p(\underline{x}|\omega_j) \sim N(\underline{\mu}_j, \underline{\Sigma}_j)$, όπου το $\underline{\theta}_j$ περιέχει τόσο τα στοιχεία του $\underline{\mu}_j$ όσο και τα στοιχεία του $\underline{\Sigma}_j$. Θα γράφουμε $p(\underline{x}|\omega_j, \underline{\theta}_j)$ για να υποδηλώσουμε την εξάρτηση από τις παραμέτρους $\underline{\theta}_j$. Το πρόβλημα τώρα είναι να εκτιμήσουμε τις παραμέτρους $\underline{\theta}$ από το σύνολο δειγμάτων X , γνωρίζοντας ότι τα δείγματα είναι στατιστικώς ανεξάρτητα με σ.π.π $p(\underline{x}|\omega, \underline{\theta})$. Ας υποθέσουμε ότι το σύνολο X περιέχει n δείγματα: $X = \{\underline{x}_1, \dots, \underline{x}_n\}$.

Αφού τα δείγματα είναι στατιστικώς ανεξάρτητα, έχουμε:

$$p(X | \underline{\theta}) = \prod_{k=1}^n p(\underline{x}_k | \underline{\theta}).$$

Το $p(X | \underline{\theta})$, ως συνάρτηση του $\underline{\theta}$, θα ονομάζεται *πιθανοφάνεια (likelihood) του $\underline{\theta}$ αναφορικά με το δείγμα X* .

Η *εκτίμηση μέγιστης πιθανότητας του $\underline{\theta}$* είναι εκείνη η τιμή που μεγιστοποιεί τη συνάρτηση $p(X|\underline{\theta})$. Διαισθητικά, αντιστοιχεί στην τιμή του $\underline{\theta}$ που «συμφωνεί» όσο το δυνατόν καλύτερα με τα δείγματα.

Πολλές φορές είναι ευκολότερο να κάνουμε υπολογισμούς με το λογάριθμο $l(\underline{\theta}) = \log p(X|\underline{\theta})$.

$$\text{Έστω ότι } \underline{\theta} = \begin{bmatrix} \theta_1 \\ \cdot \\ \cdot \\ \cdot \\ \theta_p \end{bmatrix} \text{ και } \nabla_{\theta} = \begin{bmatrix} \partial/\partial\theta_1 \\ \cdot \\ \cdot \\ \cdot \\ \partial/\partial\theta_p \end{bmatrix}.$$

Τότε, μια αναγκαία συνθήκη που ικανοποιεί η εκτίμηση $\hat{\theta}$ είναι $\nabla_{\theta} l = \sum_{k=1}^n \nabla_{\theta} \log p(\underline{x}_k | \underline{\theta}) = 0$.

Παράδειγμα: Πολυμεταβλητή γκαουσιανή κατανομή με άγνωστο μέσο και άγνωστο πίνακα συμμεταβλητότητας.

Ας υποθέσουμε ότι τα δείγματα προέρχονται από μια γκαουσιανή κατανομή με μέσο $\underline{\mu}$ και πίνακα συμμεταβλητότητας $\underline{\Sigma}$. Υποθέτουμε πρώτα ότι μόνο ο μέσος $\underline{\mu}$ είναι άγνωστος. Τότε:

$$\log p(\underline{x}_k | \underline{\mu}) = -\frac{1}{2} \log[(2\pi)^d |\underline{\Sigma}|] - \frac{1}{2} (\underline{x}_k - \underline{\mu})^T \underline{\Sigma}^{-1} (\underline{x}_k - \underline{\mu})$$

και

$$\nabla_{\underline{\mu}} \log p(\underline{x}_k | \underline{\mu}) = \sum_{k=1}^n \underline{\Sigma}^{-1} (\underline{x}_k - \underline{\mu}).$$

Επομένως, η εκτίμηση μέγιστης πιθανοφάνειας για το $\underline{\mu}$ ικανοποιεί την εξίσωση:

$$\sum_{k=1}^n \underline{\Sigma}^{-1} (\underline{x}_k - \hat{\underline{\mu}}) = 0,$$

απ' όπου: $\hat{\underline{\mu}} = \frac{1}{n} \sum_{k=1}^n \underline{x}_k$ (αριθμητικός μέσος των δειγμάτων).

Αν και ο πίνακας $\underline{\Sigma}$ είναι άγνωστος, τότε οι εκτιμήσεις μέγιστης πιθανοφάνειας γίνονται:

$$\hat{\underline{\mu}} = \frac{1}{n} \sum_{k=1}^n \underline{x}_k$$

$$\hat{\underline{\Sigma}} = \frac{1}{n} \sum_{k=1}^n (\underline{x}_k - \hat{\underline{\mu}}) (\underline{x}_k - \hat{\underline{\mu}})^T.$$

Από μαθήματα στατιστικής, οι παραπάνω εκτιμήσεις $\hat{\underline{\mu}}$ και $\hat{\underline{\Sigma}}$ είναι γνωστές. Είναι όμως επίσης γνωστό ότι ενώ η εκτιμήτρια $\hat{\underline{\mu}}$ αμερόληπτη (unbiased), η εκτιμήτρια $\hat{\underline{\Sigma}}$ μεροληπτεί. Συγκεκριμένα:

$$E\left\{\hat{\underline{\mu}}\right\} = \frac{1}{n} \sum_{k=1}^n E\{\underline{x}_k\} = \frac{1}{n} \sum_{k=1}^n \underline{\mu} = \underline{\mu}$$

$$E\left\{\hat{\underline{\Sigma}}\right\} = \dots = \frac{n-1}{n} \underline{\Sigma}$$

Η εκτιμήτρια

$$\hat{\underline{C}} = \frac{1}{n-1} \sum_{k=1}^n \left(\underline{x}_k - \hat{\underline{\mu}}\right) \left(\underline{x}_k - \hat{\underline{\mu}}\right)^T = \frac{n}{n-1} \hat{\underline{\Sigma}}$$

δεν μεροληπτεί:

$$E\left\{\hat{\underline{C}}\right\} = \underline{\Sigma}$$

7. Εκτίμηση παραμέτρων με τη μέθοδο του Bayes

Ενώ στη μέθοδο μέγιστης πιθανοφάνειας υποθέτουμε ότι οι άγνωστοι παράμετροι είναι μη τυχαίες μεταβλητές, στη μπεϋζιανή προσέγγιση θεωρούμε ότι οι παράμετροι είναι τυχαίες μεταβλητές με κάποια εκ των προτέρων (a priori) σ.π.π. $p(\underline{\theta})$. Πιο συγκεκριμένα υποθέτουμε τα εξής:

- 1) Γνωρίζουμε τη μορφή της σ.π.π. $p(\underline{x}|\underline{\theta})$, αλλά η τιμή του διανύσματος παραμέτρων $\underline{\theta}$ είναι άγνωστη.
- 2) Έχουμε κάποια αρχική γνώση για το $\underline{\theta}$ με τη μορφή της εκ των προτέρων (a priori) σ.π.π. $p(\underline{\theta})$.
- 3) Διαθέτουμε ένα σύνολο $X^n = \{\underline{x}_1, \dots, \underline{x}_n\}$ από n στατιστικώς ανεξάρτητα δείγματα.

Η μπεϋζιανή μέθοδος εκτίμησης του διανύσματος παραμέτρων $\underline{\theta}$ αποτελείται από δύο βήματα:

- 1) Υπολογισμό της εκ των υστέρων (a posteriori) σ.π.π. του $\underline{\theta}$ δεδομένων των μετρήσεων X^n :

$$p(\underline{\theta} | X^n) = \frac{p(X^n | \underline{\theta})p(\underline{\theta})}{\int p(X^n | \underline{\theta})p(\underline{\theta})d\underline{\theta}}$$

Έχουμε:

$$p(X^n | \underline{\theta}) = \prod_{k=1}^n p(x_k | \underline{\theta})$$

$$\int p(x^n | \underline{\theta}) p(\underline{\theta}) d\underline{\theta} = p(X^n): \text{σταθερά κανονικοποίησης, που δεν περιέχει το } \underline{\theta}.$$

- 2) Εύρεση του σημείου $\hat{\underline{\theta}}$ για το οποίο η συνάρτηση $p(X^n | \underline{\theta})p(\underline{\theta})$ ή, ισοδύναμα, η συνάρτηση: $\left[\sum_{k=1}^n \log p(x_k | \underline{\theta}) \right] + \log p(\underline{\theta})$ μεγιστοποιείται.

Παράδειγμα:

Ας υποθέσουμε ότι διαθέτουμε n δείγματα της τυχαίας μεταβλητής x , δηλαδή $X^n = \{x_1, \dots, x_n\}$. Η μεταβλητή x είναι $N(\mu, \sigma^2)$, όπου μόνο ο μέσος μ είναι άγνωστος. Υποθέτουμε ότι $\mu \sim N(\mu_0, \sigma_0^2)$ (μ_0, σ_0^2 : γνωστά), δηλαδή η καλύτερη αρχική εκτίμηση του μ είναι μ_0 με μια αβεβαιότητα της τάξεως του σ_0^2 . Έχουμε:

$$p(x|\mu) \sim N(\mu, \sigma^2) \text{ και } p(\mu) = N(\mu_0, \sigma_0^2).$$

$$\begin{aligned} \text{Επομένως: } p(\mu | X^n) &= \frac{p(X^n | \mu)p(\mu)}{\int p(X^n | \mu)p(\mu)d\mu} \\ &= (\text{σταθερά}) \times \prod_{k=1}^n p(x_k | \mu)p(\mu). \end{aligned}$$

Άρα:

$$\begin{aligned} p(\mu | X^n) &= (\text{σταθερά}) \times \prod_{k=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2} \frac{(x^k - \mu)^2}{\sigma^2}\right] \times \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left[-\frac{1}{2} \frac{(\mu - \mu_0)^2}{\sigma_0^2}\right] \\ &= (\text{νέα} - \text{σταθερά}) \times \exp\left[-\frac{1}{2} \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}\right) \mu^2 - 2 \left(\frac{1}{\sigma^2} \sum_{k=1}^n x_k + \frac{\mu_0}{\sigma_0^2}\right) \mu\right] \end{aligned}$$

Ορίζοντας

$$\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}$$

$$\frac{\mu_n}{\sigma_n^2} = \frac{n}{\sigma^2} m_n + \frac{\mu_0}{\sigma_0^2}, \quad m_n = \frac{1}{n} \sum_{k=1}^n x_k,$$

δηλαδή

$$\mu_n = \frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} m_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$$

$$\sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2},$$

βρίσκουμε

$$p(\mu | X^n) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-\frac{1}{2} \frac{(\mu - \mu_n)^2}{\sigma_n^2}\right]$$

Άρα $\hat{\mu}_n$ (από n δείγματα) $= \mu_n = \frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} m_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \mu_0$, με αβεβαιότητα της τάξεως του σ_n^2 .

Παρατηρούμε ότι:

- 1) Καθώς $\sigma_0^2 \rightarrow \infty$, έχουμε $\hat{\mu}_n \rightarrow m_n$ για κάθε n , δηλαδή επιστρέφουμε στην εκτίμηση μέγιστης πιθανοφάνειας.
- 2) Καθώς $n \rightarrow \infty$, έχουμε $\sigma_n^2 \rightarrow \sigma_0^2$ για κάθε σ^2 , δηλαδή η ακρίβεια της εκτίμησης του μ περιορίζεται μόνο από την αβεβαιότητα της εκ των προτέρων γνώσης μας.
- 3) Καθώς $n \rightarrow \infty$, η εκ των υστέρων σ.π.π. $p(\mu|X^n)$ γίνεται όλο και περισσότερο συγκεντρωμένη γύρω από το μέσο της. Το φαινόμενο αυτό λέγεται *μπεϋζιανή εκμάθηση (Bayesian learning)*.

8. Παραδείγματα εκτίμησης μέγιστης πιθανοφάνειας και μπεϋζιανής

1. Υποθέτουμε ότι η τ.μ. X έχει εκθετική κατανομή :

$$p(x | \theta) = \begin{cases} \theta e^{-\theta x}, & x \geq 0 \\ 0, & \text{αλλιώς} \end{cases}, \quad \theta > 0$$

Διαθέτουμε n δείγματα (στατιστικώς ανεξάρτητα) x_1, \dots, x_n της τ.μ. X . Να βρεθεί η εκτίμηση μέγιστης πιθανοφάνειας $\hat{\theta}$ μ.π. της παραμέτρου θ .

Σχηματίζουμε το διάνυσμα $\underline{x} = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$ και τη συνάρτηση πιθανοφάνειας

$$p(\underline{x} | \theta) = \prod_{\kappa=1}^n p(x_{\kappa} | \theta) = \begin{cases} \theta^n e^{-\theta \sum_{\kappa=1}^n x_{\kappa}}, & x_{\kappa} \geq 0, \kappa = 1, 2, \dots, n \\ 0, & \text{αλλιώς} \end{cases}$$

Έχουμε $l(\theta) \equiv \log p(\underline{x} | \theta) = \begin{cases} n \log \theta - \theta \sum_{\kappa=1}^n x_{\kappa}, & x_{\kappa} \geq 0, \kappa = 1, 2, \dots, n \\ -\infty, & \text{αλλιώς} \end{cases}$

και $\frac{d}{d\theta} l(\theta) = \frac{n}{\theta} - \sum_{\kappa=1}^n x_{\kappa}.$

Επομένως, $\hat{\theta}_{\mu.π.}$ τέτοιο ώστε $\left. \frac{dl}{d\theta} \right|_{\theta = \hat{\theta}_{\mu.π.}} = 0$

απ' όπου βρίσκουμε $\hat{\theta}_{\mu.π.} = \frac{1}{\frac{1}{n} \sum_{\kappa=1}^n x_{\kappa}}$

2. Υποθέτουμε ότι η τ.μ. X έχει ομοιόμορφη κατανομή

$$p(x | \theta) = \begin{cases} \frac{1}{\theta}, 0 \leq x \leq \theta \\ 0, & \text{αλλιώς} \end{cases}, \theta > 0$$

Διαθέτουμε n δείγματα (στατιστικώς ανεξάρτητα) x_1, \dots, x_n της τ.μ. X . Να βρεθεί η εκτίμηση μέγιστης πιθανοφάνειας $\hat{\theta}_{\mu.π.}$ της παραμέτρου θ .

Σχηματίζουμε το διάνυσμα $\underline{x} = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$ και τη συνάρτηση πιθανοφάνειας

$$p(\underline{x} | \theta) = \prod_{\kappa=1}^n p(x_{\kappa} | \theta) = \begin{cases} \frac{1}{\theta^n}, 0 \leq x_{\kappa} \leq \theta, \kappa = 1, 2, \dots, n \\ 0, \text{αλλιώς} \end{cases}$$

Παρατηρούμε ότι για $\theta < \max_{1 \leq \kappa \leq n} x_{\kappa}$ η συνάρτηση πιθανοφάνειας μηδενίζεται, ενώ για

$\theta > \max_{1 \leq \kappa \leq n} x_{\kappa}$ η συνάρτηση πιθανοφάνειας έχει θετική τιμή. Άρα, πρώτο συμπέρασμα

είναι ότι $\hat{\theta}_{\mu.π.}$ πρέπει να είναι μια τιμή μεγαλύτερη του $\max_{1 \leq \kappa \leq n} x_{\kappa}$. Παρατηρούμε ότι

καθώς αυξάνει το θ πέρα της τιμής $\max_{1 \leq \kappa \leq n} x_{\kappa}$, η συνάρτηση πιθανοφάνειας φθίνει. Άρα,

η συνάρτηση πιθανοφάνειας μεγιστοποιείται όταν $\theta = \max_{1 \leq \kappa \leq n} x_{\kappa}$ και $\hat{\theta}_{\mu.π.} = \max_{1 \leq \kappa \leq n} x_{\kappa}$.

3. Υποθέτουμε ότι η τ.μ. X έχει κατανομή Rayleigh :

$$p(x | \theta) = \begin{cases} 2\theta e^{-\theta x^2}, x \geq 0 \\ 0, \text{αλλιώς} \end{cases}, \theta > 0$$

Να βρεθεί η εκτίμηση μέγιστης πιθανοφάνειας από n δείγματα (στατιστικώς ανεξάρτητα) x_1, \dots, x_n .

Σχηματίζουμε το διάνυσμα $\underline{x} = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$ και τη συνάρτηση πιθανοφάνειας

$$p(\underline{x} | \theta) = \prod_{\kappa=1}^n p(x_{\kappa} | \theta) = \begin{cases} (2\theta)^n e^{-\theta \sum_{\kappa=1}^n x_{\kappa}^2}, & x_{\kappa} \geq 0, \kappa = 1, 2, \dots, n \\ 0, & \text{αλλιώς} \end{cases}$$

$$\text{Επομένως } l(\theta) \equiv \log p(\underline{x} | \theta) = \begin{cases} n \log(2\theta) - \theta \sum_{\kappa=1}^n x_{\kappa}^2, & x_{\kappa} \geq 0, \kappa = 1, 2, \dots, n \\ -\infty, & \text{αλλιώς} \end{cases}$$

$$\text{και } \hat{\theta}_{\mu.π.} = \frac{1}{\frac{1}{n} \sum_{\kappa=1}^n x_{\kappa}^2}.$$

4. Για την κατανομή Maxwell

$$p(x | \theta) = \begin{cases} \frac{4}{\sqrt{\pi}} \theta^{3/2} x^2 e^{-\theta x^2}, & x \geq 0 \\ 0, & \text{αλλιώς} \end{cases}, \theta > 0$$

η εκτίμηση μέγιστης πιθανοφάνειας $\hat{\theta}_{\mu.π.}$ του θ από n δείγματα (στατιστικώς ανεξάρτητα) είναι :

$$\hat{\theta}_{\mu.π.} = \frac{3/2}{\frac{1}{n} \sum_{\kappa=1}^n x_{\kappa}^2}$$

9. Γραμμικές συναρτήσεις διάκρισης

Υποθέσαμε στα προηγούμενα ότι σε προβλήματα ταξινόμησης ήταν γνωστές οι πιθανοτικές κατανομές των χαρακτηριστικών και παρουσιάσαμε δύο μεθόδους για την εκτίμηση των τιμών των παραμέτρων τους από δείγματα. Θα υποθέσουμε εδώ ότι γνωρίζουμε (μέχρι κάποιο αριθμό παραμέτρων) τη μορφή των συναρτήσεων διάκρισης και θα χρησιμοποιήσουμε δείγματα για την εκτίμηση των τιμών των παραμέτρων του ταξινομητή. Πιο συγκεκριμένα, θα ασχοληθούμε με γραμμικές συναρτήσεις διάκρισης,

αφενός γιατί είναι οι βέλτιστες για ορισμένες κατανομές και αφετέρου γιατί, κι όταν δεν είναι βέλτιστες, έχουν μια απλή μορφή που υλοποιείται εύκολα.

Δύο κλάσεις

Μια γραμμική συνάρτηση διάκρισης έχει τη μορφή

$$g(\underline{x}) = \underline{w}^T \underline{x} + w_0,$$

όπου \underline{w} είναι το διάνυσμα βαρών (*weight vector*) και w_0 είναι το βάρος κατωφλίου (*threshold weight*). Ο γραμμικός ταξινομητής δύο κατηγοριών χρησιμοποιεί τον κανόνα απόφασης: 'Αποφάσισε ω_1 αν $g(\underline{x}) > 0$ και ω_2 αν $g(\underline{x}) < 0$ '. Αν $g(\underline{x}) = 0$, το \underline{x} μπορεί να ανατεθεί σε οποιαδήποτε από τις δύο κλάσεις χωρίς σημασία.

Η εξίσωση $g(\underline{x}) = 0$ ορίζει την επιφάνεια απόφασης που χωρίζει τα σημεία που ανατίθενται στην κλάση ω_1 από τα σημεία που ανατίθενται στην κλάση ω_2 . Όταν η $g(\underline{x})$ είναι γραμμική, η επιφάνεια απόφασης είναι ένα *υπερεπίπεδο (hyperplane)* κάθετο στο διάνυσμα \underline{w} .

Πολλές κλάσεις

Ο γραμμικός ταξινομητής c κατηγοριών χρησιμοποιεί c γραμμικές συναρτήσεις διάκρισης

$$g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{i0}, \quad i = 1, 2, \dots, c$$

και αναθέτει το \underline{x} στην κλάση ω_i αν $g_i(\underline{x}) > g_j(\underline{x})$ για κάθε $j \neq i$. Έτσι, ο χώρος των διανυσμάτων χαρακτηριστικών \underline{x} χωρίζεται σε c περιοχές $R_i, i = 1, 2, \dots, c$, με $g_i(\underline{x}) > g_j(\underline{x})$ ($j \neq i$) για κάθε \underline{x} στην περιοχή R_i . Για δύο περιοχές R_i και R_j με κοινό σύνορο, η εξίσωση

$$g_i(\underline{x}) = g_j(\underline{x})$$

ορίζει το σύνορό τους ως τμήμα ενός υπερεπιπέδου κάθετου στο διάνυσμα $(\underline{w}_i - \underline{w}_j)$.

Οι περιοχές απόφασης για γραμμικές συναρτήσεις διάκρισης είναι *κυρτές* (*convex*) και, επομένως, οι γραμμικές συναρτήσεις διάκρισης είναι κατάλληλες για δεσμευμένες σ.π.π. που είναι *unimodal*.

Γενικευμένες γραμμικές συναρτήσεις διάκρισης

Οι γραμμικές συναρτήσεις διάκρισης που είδαμε ως τώρα είχαν τη μορφή :

$$g(\underline{x}) = w_0 + \sum_{i=1}^d w_i x_i,$$

όπου είχαμε $\underline{w} = \begin{bmatrix} w_1 \\ \cdot \\ \cdot \\ \cdot \\ w_d \end{bmatrix}$.

Μπορούμε να ορίσουμε γενικευμένες γραμμικές συναρτήσεις διάκρισης αντικαθιστώντας στην παραπάνω έκφραση τα x_i με συναρτήσεις $y_i(x)$:

$$g(\underline{x}) = \sum_{i=1}^d a_i y_i(\underline{x}) = \underline{a}^T \underline{y},$$

όπου το \hat{d} γενικά διαφέρει από το d . Για παράδειγμα, οι τετραγωνικές συναρτήσεις διάκρισης έχουν τη μορφή

$$g(\underline{x}) = w_0 + \sum_{i=1}^d w_i x_i + \sum_{i=1}^d \sum_{j=1}^d w_{ij} x_i x_j .$$

Για αυτές, οι διαχωριστικές επιφάνειες που ορίζονται από τη σχέση $g(\underline{x}) = 0$ είναι δεύτερης τάξης (υπερτετραγωνικές (hyperquadric) επιφάνειες). Ας δούμε το συγκεκριμένο παράδειγμα

$$g(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2$$

Στην περίπτωση αυτή $\underline{y}(x) = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$ και $\underline{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$.

Ενώ $d = 1$, έχουμε $\hat{d} = 3$.

Έχοντας ορίσει (γενικευμένες) συναρτήσεις διάκρισης, υπάρχουν διάφορες μεθοδολογίες για την εκτίμηση των παραμέτρων (δηλ. του διανύσματος βαρών \underline{a}) από δείγματα.

10. Δύο γραμμικά διαχωρίσιμες κλάσεις και ο αλγόριθμος perceptron

Γενικά

Ας υποθέσουμε ότι διαθέτουμε n δείγματα $\underline{y}_1, \dots, \underline{y}_n$ γενικευμένων χαρακτηριστικών, μερικά από τα οποία προέρχονται από την κλάση ω_1 και τα υπόλοιπα από την κλάση ω_2 . Ας υποθέσουμε επίσης ότι υπάρχει μια γραμμική συνάρτηση διάκρισης $g(\underline{x}) = \underline{a}^T \underline{y}$ που ταξινομεί με πολύ μικρή πιθανότητα λάθους. Το πρόβλημά μας ανάγεται στο να υπολογίσουμε το διάνυσμα βαρών \underline{a} . Η περίπτωση αυτή ονομάζεται *γραμμικά διαχωρίσιμη (linearly separable)*.

Παρατηρούμε ότι ένα δείγμα \underline{y} από την κλάση ω_1 ταξινομείται σωστά αν $\underline{a}^T \underline{y} > 0$, ενώ ένα δείγμα \underline{y} από την κλάση ω_2 ταξινομείται σωστά αν $\underline{a}^T \underline{y} < 0$ ή $\underline{a}^T (-\underline{y}) > 0$. Το διάνυσμα βαρών \underline{a} θα ονομάζεται *διαχωριστικό διάνυσμα (separating vector)* ή

διάνυσμα λύσης (*solution vector*). Προφανώς, το διάνυσμα \underline{a} δεν είναι μοναδικό. Επιπρόσθετοι περιορισμοί (π.χ. μοναδιαίου μήκους) μπορούν να επιβληθούν σ' αυτό.

Η γενική μεθοδολογία που θα ακολουθήσουμε για να προσδιορίσουμε το διάνυσμα λύσης \underline{a} συνίσταται στον ορισμό ενός κριτηρίου $J(\underline{a})$ και την εύρεση του διανύσματος \underline{a} που το ελαχιστοποιεί. Επομένως, το πρόβλημα ανάγεται στην ελαχιστοποίηση μιας βαθμωτής συνάρτησης, και μπορεί να επιλυθεί με τη μέθοδο της πιο απότομης καθόδου (*steepest descent*). Η μέθοδος αυτή συνίσταται στην αυθαίρετη επιλογή ενός αρχικού διανύσματος \underline{a}_1 . Στη συνέχεια, οι επόμενες προσεγγίσεις υπολογίζονται με μετακίνηση από την τρέχουσα προσέγγιση προς την κατεύθυνση της πιο απότομης καθόδου, δηλαδή κατά μήκος της αρνητικής βαθμίδας (*gradient*) του κριτηρίου :

$$\underline{a}_{k+1} = \underline{a}_k - \rho_k \nabla J(\underline{a}_k), \quad k = 1, 2, \dots,$$

όπου ρ_k είναι ένα θετικό βήμα (*step*).

Η διαδικασία αυτή έχει αρκετές δυσκολίες στη γενική περίπτωση, αλλά η επιλογή του κατάλληλου κριτηρίου μπορεί να αποφύγει τις δυσκολίες αυτές. Μία από τις δυσκολίες έγκειται στην επιλογή του βήματος ρ_k . Μικρό βήμα επιβραδύνει τη διαδικασία, ενώ μεγάλο βήμα μπορεί να προκαλέσει απόκλιση.

Τα κριτήρια τα οποία θα εξετάσουμε είναι:

1. Το κριτήριο perceptron
2. Το κριτήριο μέσου τετραγωνικού σφάλματος

Το κριτήριο perceptron

Το κριτήριο perceptron επιλέγει τη συνάρτηση

$$J_p(\underline{a}) = \sum_{\underline{y} \in Y(\underline{a})} \left[-\underline{a}^T (\delta_{\underline{y}} \underline{y}) \right], \quad \delta_{\underline{y}} = \begin{cases} 1, & \underline{y} \in w_1 \\ -1, & \underline{y} \in w_2 \end{cases}$$

όπου $Y(\underline{a})$ είναι το σύνολο των δειγμάτων που δεν έχουν ταξινομηθεί σωστά από τα \underline{a} . Αν δεν υπάρχουν δείγματα ταξινομημένα λάθος, τότε ορίζουμε το J_p να είναι μηδέν. Επειδή $\underline{a}^T (\delta_{\underline{y}} \underline{y}) \leq 0$ όταν το \underline{y} έχει ταξινομηθεί λάθος, το $J_p(\underline{a})$ δεν είναι ποτέ αρνητικός αριθμός, αλλά γίνεται μηδέν μόνο όταν το \underline{a} είναι ένα διάνυσμα λύσης. Γεωμετρικά, το $J_p(\underline{a})$ είναι ανάλογο του αθροίσματος των αποστάσεων των λάθος ταξινομημένων δειγμάτων από το σύνορο μεταξύ των δύο περιοχών απόφασης.

Έχουμε :

$$\frac{\partial J_p}{\partial a_j} = \sum_{\underline{y} \in Y} (-\delta_{\underline{y}} y_j)$$

όπου y_j είναι j-οστή συνιστώσα του διανύσματος \underline{y} . Επομένως :

$$\nabla J_p = \sum_{\underline{y} \in Y(\underline{a})} (-\delta_{\underline{y}} \underline{y})$$

και ο βασικός αλγόριθμος απότομης καθόδου γίνεται:

$$\underline{a}_{\kappa+1} = \underline{a}_{\kappa} + \rho_{\kappa} \sum_{\underline{y} \in Y_{\kappa}} \delta_{\underline{y}} \underline{y} ,$$

όπου Y_{κ} είναι το σύνολο των δειγμάτων που ταξινομήθηκαν λάθος από το \underline{a}_{κ} . Δηλαδή, το επόμενο διάνυσμα βάρους είναι το άθροισμα του τρέχοντος διανύσματος βάρους και ενός πολλαπλασίου του αθροίσματος των λάθος ταξινομημένων δειγμάτων.

Παράδειγμα

Έστω $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ διάνυσμα χαρακτηριστικών και γενικευμένα χαρακτηριστικά $\underline{y} =$

$\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$. Ας υποθέσουμε ότι μετά από κ επαναλήψεις, ο αλγόριθμος perceptron έχει

υπολογίσει $\underline{a}_{\kappa} = \begin{bmatrix} 1 \\ 1 \\ -1/2 \end{bmatrix}$ έχοντας χρησιμοποιήσει $\rho = 0,7$. Τα διανύσματα $\underline{x}_1 =$

$\begin{bmatrix} 0,4 \\ 0,05 \end{bmatrix}$ (από ω_1) και $\underline{x}_2 = \begin{bmatrix} -0,2 \\ 0,75 \end{bmatrix}$ (από ω_2) παραμένουν ταξινομημένα λάθος. Η

επόμενη αναδρομή δίνει

$$\underline{a}_{k+1} = \underline{a}_k - \rho(\underline{y}_1 - \underline{y}_2) = \begin{bmatrix} 1 \\ 1 \\ -1/2 \end{bmatrix} + 0,7 \left(\begin{bmatrix} 0,4 \\ 0,05 \\ 1 \end{bmatrix} - \begin{bmatrix} -0,2 \\ 0,75 \\ 1 \end{bmatrix} \right),$$

δηλαδή $\underline{a}_{k+1} = \begin{bmatrix} 1,42 \\ 0,51 \\ -0,5 \end{bmatrix}$. Επειδή $\underline{a}_{k+1}^T \cdot \underline{y}_1 > 0$ και $\underline{a}_{k+1}^T \cdot \underline{y}_2 < 0$, η ταξινόμηση γίνεται

σωστά και ο αλγόριθμος τερματίζεται.

11. Μέθοδοι ελαχίστων τετραγώνων

Αντίθετα από το κριτήριο *perceptron* που εστιάζεται στα λάθος ταξινομημένα δείγματα, εδώ θα θεωρήσουμε ένα κριτήριο που περιλαμβάνει όλα τα δείγματα. Ενώ προηγουμένως προσπαθήσαμε να βρούμε ένα διάνυσμα \underline{a} τέτοιο ώστε $\underline{a}^T \cdot \underline{y}_i > 0$,

τόρα θα προσπαθήσουμε να βρούμε \underline{a} τέτοιο ώστε $\underline{a}^T \cdot \underline{y}_i = b_i$, όπου b_i είναι αυθαίρετες θετικές σταθερές. Μετατρέπουμε, έτσι, το πρόβλημα επίλυσης ενός συνόλου γραμμικών ανισοτήτων σε πρόβλημα επίλυσης ενός συστήματος γραμμικών εξισώσεων. Το δεύτερο, αν και πιο περιοριστικό, είναι ευκολότερα κατανοητό.

Ας εισάγουμε συμβολισμό πινάκων για ευκολία. Έστω \underline{Y} ο $n \times d$ πίνακας του οποίου η i -οστή γραμμή είναι το διάνυσμα \underline{y}_i^T , δηλαδή

$$\underline{Y} = \begin{bmatrix} \underline{y}_1^T \\ \underline{y}_2^T \\ \cdot \\ \cdot \\ \underline{y}_\tau^T \end{bmatrix}. \quad \text{Επίσης, έστω } \underline{b} = \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{bmatrix}.$$

Θέλουμε να βρούμε ένα \underline{a} ($d \times 1$), έτσι ώστε

$$\underline{Y} \underline{a} = \underline{b}$$

Εάν ο πίνακας Y δεν είναι *ανώμαλος* (*singular*), τότε προφανώς

$\underline{a} = \underline{Y}^{-1} \underline{b}$. Συνήθως, όμως, ο πίνακας \underline{Y} έχει περισσότερες γραμμές από στήλες, δηλαδή περισσότερες εξισώσεις από αγνώστους ($n > d$). Λέμε, σε αυτή την περίπτωση, ότι το σύστημα γραμμικών εξισώσεων είναι *υπερπροσδιορισμένο* (*overdetermined*), και γενικά, δεν έχει ακριβή λύση. Εναλλακτικά, μπορούμε να βρούμε ένα διάνυσμα βάρους \underline{a} που να ελαχιστοποιεί την ‘απόσταση’ του $\underline{Y} \underline{a}$ από το \underline{b} . Ορίζουμε λοιπόν το διάνυσμα λάθους $\underline{e} = \underline{Y} \underline{a} - \underline{b}$ και προσπαθούμε να ελαχιστοποιήσουμε το τετράγωνο του μήκους του :

$$J_s(\underline{a}) = \|\underline{Y} \underline{a} - \underline{b}\|^2 = \sum_{i=1}^n (\underline{\alpha}^T \underline{y}_i - b_i)^2$$

Το $J_s(\underline{a})$ ονομάζεται *τετραγωνικό σφάλμα* (*squared error*) και αναγκαία συνθήκη για την ελαχιστοποίησή του είναι να μηδενίζεται η βαθμίδα του :

$$\nabla J_s(\underline{a}) = 0$$

ή

$$\sum_{i=1}^n 2(\underline{\alpha}^T \underline{y}_i - b_i) \underline{y}_i = 0$$

ή

$$2 \underline{Y}^T (\underline{Y} \underline{a} - \underline{b}) = 0$$

ή

$$\underline{Y}^T \underline{Y} \underline{a} = \underline{Y}^T \underline{b}$$

Οι εξισώσεις (αναγκαία συνθήκη)

$$\underline{Y}^T \underline{Y} \underline{a} = \underline{Y}^T \underline{b}$$

ονομάζονται *κανονικές εξισώσεις* (*normal equations*), και έχουν το πλεονέκτημα ότι είναι d εξισώσεις με d αγνώστους. Αν ο πίνακας $\underline{Y}^T \underline{Y}$ ($d \times d$) είναι ομαλός, τότε

$$\underline{a} = (\underline{Y}^T \underline{Y})^{-1} \underline{Y}^T \underline{b}$$

ή

$$\underline{\underline{a}} = \underline{\underline{Y}}^t \underline{\underline{b}}, \quad \text{με } \underline{\underline{Y}}^t = \left(\underline{\underline{Y}}^t \underline{\underline{Y}} \right)^{-1} \underline{\underline{Y}}^t$$

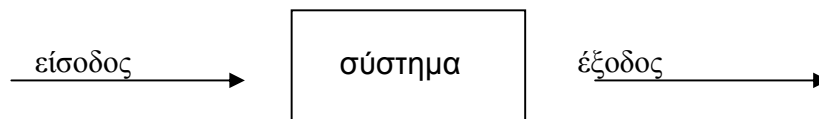
Ο πίνακας $\underline{\underline{Y}}^t$ ονομάζεται *ψευδοαντίστροφος (pseudoinverse)* (κατά Penrose – Moore) και συμπίπτει με το συνήθη αντίστροφο του $\underline{\underline{Y}}$ όταν ο $\underline{\underline{Y}}$ είναι ομαλός. Αν ο πίνακας $\underline{\underline{Y}}^t \underline{\underline{Y}}$ είναι ανώμαλος, τότε οι κανονικές εξισώσεις επιδέχονται περισσότερες από μία λύσεις. Μπορεί να αποδειχτεί ότι υπάρχει πάντα το όριο :

$$\lim_{\varepsilon \rightarrow 0} \left(\underline{\underline{Y}}^t \underline{\underline{Y}} + \varepsilon \underline{\underline{I}} \right)^{-1}$$

και το διάνυσμα $\lim_{\varepsilon \rightarrow 0} \left(\underline{\underline{Y}}^t \underline{\underline{Y}} + \varepsilon \underline{\underline{I}} \right)^{-1} \underline{\underline{Y}}^t \underline{\underline{b}}$ αποτελεί λύση ελαχίστων τετραγώνων του συστήματος $\underline{\underline{Y}} \underline{\underline{a}} = \underline{\underline{b}}$, δηλαδή ελαχιστοποιεί το τετραγωνικό σφάλμα $J_s(\underline{\underline{a}})$.

Παράδειγμα : Πολυωνυμικό μοντέλο εισόδου - εξόδου

Ας θεωρήσουμε ένα σύστημα που θέλουμε να μοντελοποιήσουμε με την προσέγγιση μαύρου κουτιού :



Θεωρούμε ότι ένα καλό μοντέλο είναι ένα πολυωνυμικό μοντέλο τάξης N χωρίς μνήμη, δηλαδή ένα μοντέλο που συνδέει το σήμα εξόδου y με το σήμα εισόδου x με μια σχέση :

$$y = a_0 + a_1 x + \dots + a_N x^N + \varepsilon$$

όπου ε είναι το σφάλμα ανάμεσα στο πραγματικό σύστημα και το μοντέλο του.

Ας υποθέσουμε ότι έχουμε ένα σύνολο d μετρήσεων σημάτων εισόδου – εξόδου (x_i, y_i) , $i=1, 2, \dots, d$. Θέλουμε να υπολογίσουμε τις παραμέτρους a_0, a_1, \dots, a_N του μοντέλου, χρησιμοποιώντας τις μετρήσεις και τη μέθοδο ελαχίστων τετραγώνων.

Έχουμε :

$$\underline{a} = \begin{bmatrix} a_0 \\ a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_N \end{bmatrix} \quad (N+1) \times 1$$

$$\underline{e} = \begin{bmatrix} e_1 \\ e_2 \\ \cdot \\ \cdot \\ \cdot \\ e_d \end{bmatrix} \quad d \times 1$$

$$\underline{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_d \end{bmatrix} \quad d \times 1$$

$$\underline{X} = \begin{bmatrix} 1 & x_1 & \cdot & \cdot & x_1^N \\ 1 & x_2 & \cdot & \cdot & x_2^N \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_d & \cdot & \cdot & x_d^N \end{bmatrix} \quad d \times (N+1)$$

και

$$y_i = \alpha_0 + \alpha_1 x_i + \dots + \alpha_N x_i^N + e_i, \quad i = 1, 2, \dots, d$$

ή

$$\underline{Y} = \underline{X} \underline{a} + \underline{e}.$$

Θέλουμε να επιλέξουμε το \underline{a} έτσι ώστε το $\|\underline{e}\|^2 = \sum_{i=1}^d e_i^2$ να ελαχιστοποιείται.

Επομένως, θέλουμε να επιλύσουμε τις κανονικές εξισώσεις

$$\left(\underline{X}^T \underline{X} \right) \underline{a} = \underline{X}^T \underline{Y}$$

12. Βέλτιστος εκτιμητής ελαχίστου μέσου τετραγωνικού σφάλματος

Ας θεωρήσουμε δύο τυχαία διανύσματα \underline{y} και \underline{x} , διαστάσεων κ και l αντίστοιχα, με από κοινού σ.π.π. $p(\underline{x}, \underline{y})$.

Θεώρημα:

Η εκτίμηση $\hat{\underline{y}}(\underline{x})$ που ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα $E\left\{\|\underline{y} - \hat{\underline{y}}(\underline{x})\|^2\right\}$ δίνεται ως

$$\hat{\underline{y}}(\underline{x}) = E\{\underline{y}|\underline{x}\} = \int \underline{y} p(\underline{y}|\underline{x}) d\underline{y},$$

όπου $p(\underline{y}|\underline{x})$ η δεσμευμένη σ.π.π. του \underline{y} δεδομένου του \underline{x} .

Το θεώρημα αυτό, αν και απόλυτα γενικό, δεν έχει πρακτική αξία τις περισσότερες φορές. Για το λόγο αυτό, θα εστιάσουμε σε γραμμικούς εκτιμητές

$$\hat{\underline{y}}(\underline{x}) = \underline{A}\underline{x},$$

όπου \underline{A} είναι ένας πίνακας διαστάσεων $\kappa \times l$.

Ορίζουμε το μέσο τετραγωνικό σφάλμα

$$E\left\{\|\underline{y} - \underline{A}\underline{x}\|^2\right\} = \sum_{i=1}^{\kappa} E\left\{\left(y_i - \underline{a}_i^T \underline{x}\right)^2\right\},$$

$$\text{όπου } \underline{A} = \begin{bmatrix} \underline{a}_1^T \\ \underline{a}_2^T \\ \vdots \\ \underline{a}_\kappa^T \end{bmatrix}, \quad \underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_l \end{bmatrix}, \quad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_\kappa \end{bmatrix}$$

Η ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος απαιτεί ελαχιστοποίηση κ κοστών της μορφής

$$J_i(\underline{a}) = E\left\{\left(y_i - \underline{a}_i^T \underline{x}\right)^2\right\}, \quad i = 1, 2, \dots, \kappa$$

που επιτυγχάνεται ως :

$$\hat{\underline{a}}_i = \left[E\{\underline{x}\underline{x}^T\}\right]^{-1} E\{\underline{x}y_i\}, \quad i = 1, 2, \dots, \kappa$$

Ορίζουμε τον πίνακα αυτοσυσχέτισης του διανύσματος \underline{x}

$$\underline{\underline{R}}_{\underline{x}\underline{x}} = E\{\underline{x}\underline{x}^T\}$$

και το διάνυσμα συσχέτισης του \underline{x} με το y_i

$$\underline{\underline{R}}_{\underline{x}y} = E\{\underline{x}y_i\}$$

και βρίσκουμε

$$\hat{\underline{a}}_i = \underline{\underline{R}}_{\underline{x}\underline{x}}^{-1} \underline{\underline{R}}_{\underline{x}y_i}, \quad i = 1, 2, \dots, \kappa$$

Ας δούμε τώρα τη σχέση των παραπάνω με το πρόβλημα της ταξινόμησης. Η βέλτιστη κατά Bayes συνάρτηση διάκρισης είναι $g(\underline{x})$ και επιθυμούμε να την προσεγγίσουμε με μια γραμμική συνάρτηση διάκρισης $\underline{a}^T \underline{x}$, σε τρόπο ώστε να ελαχιστοποιείται το μέσο τετραγωνικό σφάλμα

$$E\left\{\left(g(\underline{x}) - \underline{a}^T \underline{x}\right)^2\right\}$$

Είδαμε ότι ο βέλτιστος ταξινομητής δίνεται από τη σχέση (κανονικές εξισώσεις) :

$$\underline{\underline{R}}_{\underline{x}\underline{x}} \underline{a} = E\{\underline{x}g(\underline{x})\}$$

Το πρόβλημα εδώ είναι ότι δεν είναι εύκολος ο υπολογισμός του $g(\underline{x})$, και για το λόγο αυτό επιδιώκουμε τη γραμμική προσέγγισή του.

$$g(\underline{x}) = p(\omega_1|\underline{x}) - p(\omega_2|\underline{x}) \text{ και} \\ p(\underline{x}) p(\omega_i|\underline{x}) = p(\omega_i) p(\underline{x}|\omega_i)$$

Αλλά:

$$\begin{aligned} E\{\underline{x}g(\underline{x})\} &= \int \underline{x}g(\underline{x})p(\underline{x})d\underline{x} \\ &= p(\omega_1) \int_{\underline{x} \in \omega_1} p(\underline{x}|\omega_1)(+\underline{x})d\underline{x} + p(\omega_2) \int_{\underline{x} \in \omega_2} p(\underline{x}|\omega_2)(-\underline{x})d\underline{x}, \end{aligned}$$

δηλαδή, αντί για $g(\underline{x})$, χρειάζεται μόνο να χρησιμοποιηθεί το πρόσημό του, +1 αν $\underline{x} \in \omega_1$ και -1 αν $\underline{x} \in \omega_2$.

13. Μέθοδοι στοχαστικής προσέγγισης

Ανασκόπηση

$g(\underline{x})$: βέλτιστη κατά Bayes συνάρτηση διάκρισης

$\underline{a}^T \underline{x}$: γραμμική προσέγγιση της $g(\underline{x})$

$J(\underline{a}) = E\left\{\left(g(\underline{x}) - \underline{a}^T \underline{x}\right)^2\right\}$: κριτήριο βελτιστοποίησης του μέσου τετραγωνικού σφάλματος

$\underline{\underline{R}}_{xx} \underline{a} = E\{\underline{x}g(\underline{x})\}$: κανονικές εξισώσεις

$$E\{\underline{x}g(\underline{x})\} = p(\omega_1) \int_{\underline{x} \in \omega_1} p(\underline{x}|\omega_1)\underline{x}d\underline{x} - p(\omega_2) \int_{\underline{x} \in \omega_2} p(\underline{x}|\omega_2)\underline{x}d\underline{x},$$

δηλαδή χρειάζεται μόνο το πρόσημο του $g(\underline{x})$ κι όχι η πλήρης έκφρασή του.

Αν μπορούν να υπολογιστούν κάπως οι $\underline{\underline{R}}_{xx}$ και $E\{\underline{x}g(\underline{x})\}$, τότε $\hat{\underline{a}} = \underline{\underline{R}}_{xx}^{-1} E\{\underline{x}g(\underline{x})\}$.

Η δυσκολία έγκειται στο ότι δε γνωρίζουμε συνήθως τις πιθανοτικές κατανομές (σ.π.π.) $p(\underline{x}|\omega_1)$ και $p(\underline{x}|\omega_2)$ και επομένως δεν μπορούμε να υπολογίσουμε απευθείας την αναμενόμενη τιμή $E\{\underline{x}g(\underline{x})\}$.

Λύση: Αναδρομική – επαναληπτική διαδικασία (αλγόριθμος), όπου οι αναμενόμενες τιμές αντικαθίστανται από εκτιμήσεις με βάση τα υπάρχοντα δείγματα. Η βασική ιδέα είναι να χρησιμοποιηθεί ένας επαναληπτικός αλγόριθμος καθόδου κατά την κατεύθυνση της αρνητικής βαθμίδας (*gradient descent*):

$$\underline{a}_{\kappa+1} = \underline{a}_{\kappa} - \rho_{\kappa} \nabla J(\underline{a}_{\kappa}), \quad \kappa = 1, 2, 3, \dots, \quad \rho_{\kappa} > 0 : \text{ακολουθία βημάτων.}$$

Έχουμε :

$$\nabla J(\underline{a}) = \nabla E \left\{ \left(\underline{g}(\underline{x}) - \underline{a}^{\tau} \underline{x} \right)^2 \right\} = -2 E \left\{ \underline{x} \left(\underline{g}(\underline{x}) - \underline{a}^{\tau} \underline{x} \right) \right\}$$

Άρα :

$$\underline{a}_{\kappa+1} = \underline{a}_{\kappa} + 2\rho_{\kappa} E \left\{ \underline{x} \left(\underline{g}(\underline{x}) - \underline{a}^{\tau} \underline{x} \right) \right\}$$

Στη θέση της (άγνωστης) αναμενόμενης τιμής, χρησιμοποιείται η προσέγγιση :

$$E \left\{ \underline{x} \left(\underline{g}(\underline{x}) - \underline{a}^{\tau} \underline{x} \right) \right\} \approx \underline{x} \left(\underline{g}(\underline{x}) - \underline{a}^{\tau} \underline{x} \right)$$

και ο αλγόριθμος γίνεται :

$$\underline{a}_{\kappa+1} = \underline{a}_{\kappa} + 2\rho_{\kappa} \underline{x}_{\kappa} \left(\underline{g}(\underline{x}_{\kappa}) - \underline{a}^{\tau} \underline{x}_{\kappa} \right)$$

Αυτός ο αλγόριθμος ονομάζεται αλγόριθμος των Widrow – Hoff ή LMS (Least Mean Square).

Μπορεί ναδειχτεί ότι αν ο πίνακας $\underline{R}_{xx} = E \left\{ \underline{x} \underline{x}^{\tau} \right\}$ είναι ομαλός (αντιστρέψιμος) και

$$\lim_{m \rightarrow \infty} \sum_{\kappa=1}^m \rho_{\kappa} = \infty \quad \text{και} \quad \lim_{m \rightarrow \infty} \sum_{\kappa=1}^m \rho_{\kappa}^2 < \infty, \quad \text{τότε}$$

$$E \left\{ \left\| \underline{a}_{\kappa} - \hat{\underline{\alpha}} \right\|^2 \right\} \xrightarrow{\kappa \rightarrow \infty} 0$$

Οι συνθήκες για την ακολουθία βημάτων ρ_{κ} έχουν την εξής σημασία: Η πρώτη δεν επιτρέπει στη σύγκλιση του διανύσματος \underline{a}_{κ} να είναι τόσο γρήγορη ώστε να παραμείνει κάποιο συστηματικό λάθος. Η δεύτερη επιτρέπει τελικά να περιορίζονται οι τυχαίες διακυμάνσεις του διανύσματος \underline{a}_{κ} . Μια επιλογή είναι $\rho_{\kappa} = \frac{1}{\kappa}$ και ικανοποιεί

και τις δύο συνθήκες. Όμως, οδηγεί σε αργή σύγκλιση και, αν ληφθούν οι ιδιαιτερότητες του προβλήματος υπόψη, μπορεί να βελτιωθεί.

Άλλες στοχαστικές προσεγγίσεις μπορούν να δημιουργηθούν από τον αλγόριθμο του Newton. Πιο συγκεκριμένα :

$$\left[\frac{\partial^2 J}{\partial a_i \partial a_j} \right] = 2 E \{ \underline{x} \underline{x}^\tau \} = 2 \underline{\underline{R}}_{xx}$$

και $\underline{a}_{\kappa+1} = \underline{a}_\kappa + \underline{\underline{R}}_{xx}^{-1} E \{ (g(\underline{x}) - \underline{a}^\tau \underline{x}) \underline{x} \}.$

Η στοχαστική προσέγγιση είναι :

$$\underline{a}_{\kappa+1} = \underline{a}_\kappa + \underline{\underline{R}}_{\kappa+1}^{-1} (g(\underline{x}_\kappa) - \underline{a}_\kappa^\tau \underline{x}_\kappa) \underline{x}_\kappa$$

με $\underline{\underline{R}}_{\kappa+1} = \underline{\underline{R}}_\kappa - \frac{\underline{\underline{R}}_\kappa \underline{x}_\kappa (\underline{\underline{R}}_\kappa \underline{x}_\kappa)^\tau}{1 + \underline{x}_\kappa \underline{\underline{R}}_\kappa \underline{x}_\kappa}.$

Μπορεί να αποδειχτεί ότι και για αυτό τον αλγόριθμο $E \left\{ \left\| \underline{a}_\kappa - \hat{\underline{a}} \right\|^2 \right\} \xrightarrow{\kappa \rightarrow \infty} 0$. Μάλιστα, η σύγκλιση είναι ταχύτερη, αλλά απαιτεί περισσότερη υπολογιστική προσπάθεια ανά βήμα.

14. Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)

Τα ΤΝΔ είναι συστήματα με δομή εμπνευσμένη από τη λειτουργία του νευρικού συστήματος και του εγκεφάλου. Η βασική μονάδα ενός νευρωνικού δικτύου είναι ο νευρώνας (*neuron*). Ένας βιολογικός νευρώνας παρουσιάζεται στο επόμενο σχήμα.

Ο πυρήνας του νευρώνα δέχεται σήματα από άλλους νευρώνες, μέσω καναλιών εισόδου που ονομάζονται *δενδρίτες (dendrites)*, και τα επεξεργάζεται για τη δημιουργία ενός καινούριου σήματος. Αν το σήμα αυτό είναι αρκετά ισχυρό, ενεργοποιείται η έξοδος του νευρώνα και παράγεται ένα σήμα εξόδου που μεταδίδεται μέσω ενός καναλιού εξόδου. Το κανάλι εξόδου ονομάζεται *άξονας (axon)* και η σύνδεσή του με τους δενδρίτες των άλλων νευρώνων γίνεται μέσω *συνάψεων (synapses)*. Το μεταδιδόμενο σήμα μεταβάλλεται ανάλογα με την ισχύ της αντίστοιχης σύναψης. Ο ανθρώπινος εγκέφαλος αποτελείται από δεκάδες δισεκατομμυρίων νευρώνων.

Σε ένα ΤΝΔ, το κύριο επεξεργαστικό στοιχείο είναι μια απομίμηση του βιολογικού νευρώνα, ο τεχνητός νευρώνας. Είναι ένα χρονικά αναλλοίωτο σύστημα χωρίς μνήμη, με πολλές εισόδους και μία έξοδο.

Πιο συγκεκριμένα :

$$y_i(t) = f(x_i(t))$$

$$x_i(t) = \sum_{j=1}^M w_{ij} u_j(t)$$

Ο όρος $x_i(t)$ είναι το άθροισμα των σημάτων που προέρχονται από άλλους νευρώνες συνδεδεμένους με τον i -οστό νευρώνα μετά την επίδραση των συναπτικών βαρών w_{ij} , $j=1,2,\dots,M$.

Η συνάρτηση f μπορεί να είναι μια συνάρτηση κατωφλίου που παράγει μια μηδενική τιμή αν το άθροισμα $x_i(t)$ είναι αρκετά ισχυρό ή να είναι μια συνεχής συνάρτηση. Για παράδειγμα :

(α) μοντέλο McCulloph – Pitts :

$$f(\xi) = \begin{cases} 1, & \xi > 0 \\ 0, & \xi \leq 0 \end{cases}$$

(β) τμηματικά γραμμικό μοντέλο

$$f(\xi) = \begin{cases} 0, & \xi \leq 0 \\ \xi, & 0 \leq \xi \leq 1 \\ 1, & \xi \geq 1 \end{cases}$$

(γ) σιγμοειδές μοντέλο

$$f(\xi) = \frac{1}{1 + e^{-a\xi}}, \quad a > 0$$

ή

$$f(\xi) = \frac{1 - e^{-a\xi}}{1 + e^{-a\xi}}, \quad a > 0$$

Ένα ΤΝΔ αποτελείται από τη σειριακή, παράλληλη και με ανατροφοδότηση σύνδεση νευρώνων. Γενικά, τα ΤΝΔ διακρίνονται σε *νευρωνικά δίκτυα πολλαπλών επιπέδων (multilayer neural networks)* και *επαναληπτικά (αναδρομικά) νευρωνικά δίκτυα (recurrent neural networks)*.

Νευρωνικά δίκτυα πολλαπλών επιπέδων

Στα ΤΝΔ πολλαπλών επιπέδων, οι νευρώνες οργανώνονται σε μια ακολουθία επιπέδων. Τα επίπεδα διακρίνονται σε επίπεδα εισόδου, εξόδου και κρυφά (*input, output, and hidden*). Ένα παράδειγμα φαίνεται στο επόμενο σχήμα :

Το δίκτυο παραμετροποιείται από τα βάρη w_{ij} των συνάψεων. Μόλις αυτά καθοριστούν, τα χαρακτηριστικά του δικτύου καθορίζονται πλήρως, και σε κάθε σήμα εισόδου αντιστοιχεί συγκεκριμένο σήμα εξόδου.

Για κάθε επίπεδο με m σήματα εισόδου και k σήματα εξόδου, το διάνυσμα των σημάτων εξόδου ικανοποιεί μια σχέση της μορφής

$$\underline{y}(t) = \underline{f}(\underline{W}\underline{u}(t) + \underline{\theta}),$$

όπου το \underline{y} έχει μήκος k (όσα τα σήματα εξόδου), ο πίνακας \underline{W} ($k \times m$) είναι ο πίνακας των βαρών, $\underline{\theta}$ είναι ένα διάνυσμα (μήκους k) τιμών κατωφλίου και

$$\underline{f}(\underline{x}) = \begin{bmatrix} f(x_1) \\ \cdot \\ \cdot \\ \cdot \\ f(x_k) \end{bmatrix}_{k \times 1}, \quad \underline{x} = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_k \end{bmatrix}.$$

Θεώρημα:

Έστω $\psi : \mathbb{R}^m \rightarrow \mathbb{R}$ μια συνεχής πραγματική συνάρτηση ορισμένη σε ένα συμπαγές (δηλ. κλειστό και φραγμένο) υποσύνολο $I \subset \mathbb{R}^m$. Έστω, επίσης, $f(\cdot)$ μια συνεχής σιγμοειδής συνάρτηση, δηλαδή μια συνεχής συνάρτηση $f(\cdot)$ για την οποία υπάρχουν $A > 0$ και $B \leq 0$ τέτοια ώστε $f(\xi) \xrightarrow{\xi \rightarrow \infty} A$ και $f(\xi) \xrightarrow{\xi \rightarrow -\infty} B$. Τέλος, έστω αυθαίρετο $\varepsilon > 0$. Τότε, υπάρχει θετικός ακέραιος N και ένα νευρωνικό δίκτυο $g(\cdot)$ με ένα κρυφό επίπεδο και επίπεδο εξόδου χωρίς συνάρτηση ενεργοποίησης

$$g(\underline{u}) = \sum_{i=1}^N a_i f(\underline{w}_i^T \cdot \underline{u} + \theta_i)$$

έτσι ώστε $|\psi(\underline{u}) - g(\underline{u})| < \varepsilon$ για κάθε $\underline{u} \in I$.

Το θεώρημα γενικεύεται άμεσα για διανυσματικές πραγματικές συναρτήσεις $\psi : \mathbb{R}^m \rightarrow \mathbb{R}^k$.

15. Αναδρομικά (Επαναληπτικά) Τεχνητά Νευρωνικά Δίκτυα

Τα αναδρομικά τεχνητά νευρωνικά δίκτυα είναι συστήματα με μνήμη και, επομένως, αναπαρίστανται ως δυναμικά συστήματα. Η απλούστερη μορφή τους είναι αυτή του δικτύου Hopfield διακριτού χρόνου:

Τη χρονική στιγμή $t = 0$, το σήμα εισόδου $\underline{u}(n)$ αρχικοποιεί m επεξεργαστικές μονάδες, κάθε μία από τις οποίες υλοποιεί τη δυναμική εξίσωση

$$J_i(k+1) = f\left(\sum_{j=1}^m w_{ij} J_j(k)\right), \quad k = 0, 1, 2, \dots$$

με $J_i(0) = u_i(n)$. Σε συμπαγή μορφή:

$$\underline{J}(k+1) = \underline{f}(\underline{w} \underline{J}(k)), \quad k=0, 1, 2, \dots$$

με $\underline{J}(0) = \underline{u}(n)$, με \underline{w} και \underline{f} όπως και στην περίπτωση των νευρωνικών δικτύων πολλαπλών επιπέδων.

Το νευρωνικό δίκτυο δέχεται το σήμα εισόδου $\underline{u}(n)$, το οποίο προκαλεί επαναληπτικά υπολογισμό του $\underline{J}(k+1)$. Μετά από ορισμένες επαναλήψεις, το δίκτυο κατασταλάζει σε μία σταθερή κατάσταση, δηλαδή το \underline{J} δεν μεταβάλλεται από επανάληψη σε επανάληψη. Η τελική (ευσταθής) κατάσταση παρουσιάζεται ως έξοδος του δικτύου. Αν τα βάρη w_{ij} επιλεγούν κατάλληλα, τότε το δίκτυο παρουσιάζει μόνο έναν πεπερασμένο αριθμό ευσταθών καταστάσεων που αντιστοιχούν σε ``πρότυπα αποθηκευμένα στο δίκτυο''. Έτσι, κάθε σήμα εισόδου θεωρείται ως μία θορυβώδης μορφή κάποιου από τα αποθηκευμένα πρότυπα. Το δίκτυο δρα ως μία *συσχετιστική μνήμη (associative memory)* και ταυτοποιεί το σήμα εισόδου με κάποιο από τα αποθηκευμένα πρότυπα.

Δίκτυα ακτινικών συναρτήσεων βάσης (ΔΑΣΒ, radial basis function networks)

Τα ΔΑΣΒ μοιάζουν με τα ΤΝΔ ενός κρυφού στρώματος και έχουν την τυπική μορφή του επομένου σχήματος:

Η βασική μονάδα επεξεργασίας είναι η *μονάδα ακτινικής βάσης (radial basis unit)*. Κάθε τέτοια μονάδα χαρακτηρίζεται από ένα διάνυσμα \underline{w} . Ο σκοπός της είναι να παράγει μη αμελητέα απόκριση μόνο όταν η είσοδος της είναι "κοντά" στο \underline{w} . Επομένως, η απόκριση της μονάδας ακτινικής βάσης σε είσοδο \underline{u} έχει τη μορφή $r(d(\underline{u}, \underline{w}))$, όπου $r: \mathbb{R} \rightarrow \mathbb{R}$ είναι μια συνάρτηση με υψηλή τιμή κοντά στο μηδέν και ταχεία πτώση στο μηδέν μακριά από το μηδέν και $d(\cdot, \cdot)$ είναι ένα μέτρο της απόκλισης του \underline{u} από το \underline{w} . Μια συνηθισμένη επιλογή είναι:

$$r(\underline{u}, \underline{w}) = e^{-\frac{1}{2}(\underline{u}-\underline{w})^T \underline{c}^{-1}(\underline{u}-\underline{w})},$$

όπου \underline{c} είναι ένας διαγώνιος, θετικά ορισμένος (με θετικά στοιχεία στη διαγώνιο) πίνακας. Επομένως

$$y(n) = \sum_{i=1}^N a_i r(|\underline{u}(n) - \underline{w}_i|^2).$$

Σύμφωνα με το επόμενο θεώρημα, τα ΔΑΣΒ μπορούν να προσεγγίσουν οποιαδήποτε συνεχή συνάρτηση, όπως βέβαια και τα ΤΝΔ με ένα κρυφό επίπεδο. Η διαφορά έγκειται στο ότι, ενώ στα ΤΝΔ με ένα κρυφό επίπεδο οι σιγμοειδείς συναρτήσεις παράγουν απόκριση σε ένα απεριόριστο διάστημα τιμών, οι ακτινικές συναρτήσεις βάσεις είναι εξαιρετικά "συγκεντρωμένες".

ΘΕΩΡΗΜΑ:

Θεωρούμε τη γκαουσιανή συνάρτηση $g(\xi) = e^{-\xi^2/2}$ και ένα ΔΑΣΒ της μορφής

$$\sum_{i=1}^N a_i g((\underline{u} - \underline{w}_i)^T \underline{c}^{-1}(\underline{u} - \underline{w}_i)), \quad \text{όπου } N \geq 1, \quad a_i \in \mathbb{R}, \quad \underline{w}_i \in \mathbb{R}^m, \quad \text{και}$$

\underline{c} : διαγώνιος, θετικά ορισμένος πίνακας. Εστω αυθαίρετο $\varepsilon > 0$ και $\psi: \mathbb{R}^m \supset I \rightarrow \mathbb{R}$ μια συνεχής πραγματική συνάρτηση ορισμένη στο συμπαγές (κλειστό και φραγμένο) υποσύνολο I του \mathbb{R}^m . Τότε, υπάρχει θετικός ακέραιος N και κατάλληλα a_i, \underline{w}_i έτσι ώστε

$$\left| \psi(\underline{u}) - \sum_{i=1}^N a_i g((\underline{u} - \underline{w}_i)^T \underline{c}^{-1}(\underline{u} - \underline{w}_i)) \right| < \varepsilon \quad \text{για κάθε } \underline{u} \in I.$$

16. Ο αλγόριθμος οπισθοδιάδοσης (backpropagation algorithm)

Ας θεωρήσουμε ένα πολυστρωματικό νευρωνικό δίκτυο με L κρυφά στρώματα. Το στρώμα εισόδου και κάθε ένα από τα κρυφά στρώματα περιέχουν k_0 και k_i , $i = 1, 2, \dots, L$ νευρώνες αντίστοιχα. Κάθε νευρώνας είναι της μορφής:

Δηλαδή $y = f\left(\sum_{i=1}^n w_i u_i + \theta\right) = f(\underline{w}^T \underline{u} + \theta)$, όπου η συνάρτηση ενεργοποίησης είναι μια ομαλή σιγμοειδής συνάρτηση. Σε κάθε χρονική στιγμή k , η είσοδος είναι $\underline{u}(k)$ και η έξοδος $y(k)$ εμφανίζεται, μετά από αλληλεπιδράσεις από στρώμα σε στρώμα, στο τελικό στρώμα και έχει μήκος k_L . Δεδομένων των L , k_i , $i = 1, 2, \dots, L$, της δομής των νευρώνων και της αρχιτεκτονικής των διασυνδέσεων, θέλουμε να επιλέξουμε τα βάρη w_i σε κάθε νευρώνα έτσι ώστε η έξοδος να προσεγγίζει κάποια επιθυμητή μορφή. Η επιθυμητή έξοδος είναι ένα σήμα:

$$\underline{d}(k) = \begin{bmatrix} d_1(k) \\ d_2(k) \\ \dots \\ d_{k_L}(k) \end{bmatrix}$$

Το δίκτυο θα εκπαιδευτεί, έτσι ώστε να "μάθει" να παράγει την επιθυμητή απόκριση $\underline{d}(k)$ όταν δέχεται ως είσοδο $\underline{u}(k)$.

Τόσο η είσοδος όσο και η έξοδος προσδιορίζονται για ένα διάστημα $1 \leq k \leq N$. Για κάθε στρώμα l και κάθε κόμβο i του l -οστού στρώματος, θέλουμε να προσδιορίσουμε το διάνυσμα βαρών $w_i^{(l)}$ του νευρώνα i , έτσι ώστε να ελαχιστοποιείται το ολικό τετραγωνικό σφάλμα

$$J(\underline{w}) = \frac{1}{2} \sum_{k=1}^N \sum_{i=1}^{k_L} e_i^2(k) = \frac{1}{2} \sum_{k=1}^N |\underline{e}(k)|^2,$$

όπου $\underline{e}(k) = \underline{d}(k) - y(k)$.

Για την ελαχιστοποίηση του τετραγωνικού σφάλματος πρέπει να χρησιμοποιηθούν επαναληπτικοί αλγόριθμοι. Εστω $\underline{u}^{(l)}(k)$ ($\underline{y}^{(l)}(k)$), το σήμα εισόδου (εξόδου) τη στιγμή k στο l -οστό επίπεδο. Τα μήκη των $\underline{u}^{(l)}(k)$ και $\underline{y}^{(l)}(k)$ είναι k_{L-1} και k_L αντίστοιχα, ενώ ισχύει $\underline{u}^{(l)}(k) = \underline{y}^{(l-1)}(k)$.

Ο αλγόριθμος οπισθοδιάδοσης είναι ένας αλγόριθμος απότομης καθόδου. Χρησιμοποιεί τη βαθμίδα του κόστους

$$\frac{1}{2} \sum_{k=1}^N \frac{\vartheta}{\vartheta \underline{w}_i^{(l)}} |\underline{e}(k)|^2,$$

η οποία υπολογίζεται ως:

$$\frac{1}{2} \frac{\vartheta}{\vartheta \underline{w}_i^{(l)}} |\underline{e}(k)|^2 = \frac{1}{2} \frac{\vartheta}{\vartheta x_i^{(l)}(k)} |\underline{e}(k)|^2 \frac{\vartheta}{\vartheta \underline{w}_i^{(l)}} x_i^{(l)}(k)$$

$$\text{με } x_i^{(l)}(k) = \underline{w}_i^{(l)T} \underline{u}^{(l)}(k).$$

Θέτουμε

$$\frac{1}{2} \frac{\vartheta}{\vartheta x_i^{(l)}(k)} |\underline{e}(k)|^2 = \delta_i^{(l)}(k),$$

$$\text{ενώ έχουμε } \frac{\vartheta}{\vartheta \underline{w}_i^{(l)}} x_i^{(l)}(k) = \underline{u}^{(l)}(k).$$

$$\text{Άρα: } \frac{1}{2} \frac{\vartheta}{\vartheta \underline{w}_i^{(l)}(k)} |\underline{e}(k)|^2 = \delta_i^{(l)}(k) \underline{u}^{(l)}(k).$$

Οι παράμετροι $\delta_i^{(l)}(k)$ υπολογίζονται επαναληπτικά, αρχίζοντας από το τελευταίο στρώμα ($l=L$) και προχωρώντας προς το πρώτο. Αυτή η διαδικασία προσδίδει το όνομα οπισθοδιάδοση στον αλγόριθμο.

Έχουμε:

$$\delta_i^{(L)}(k) = \frac{1}{2} \frac{\vartheta}{\vartheta x_i^{(L)}(k)} \sum_{j=1}^{k_L} e_j^2(k) = e_i(k) \frac{\vartheta}{\vartheta x_i^{(L)}(k)} e_i(k)$$

$$\text{Άρα: } \delta_i^{(L)}(k) = -\left(d_i(k) - f(\underline{w}_i^{(L)}(k)^T \underline{u}^{(L)}(k))\right) f'(\underline{w}_i^{(L)}(k)^T \underline{u}^{(L)}(k))$$

Για $l < L$, βρίσκουμε :

$$\delta_i^{(l)}(k) = \left(\sum_{j=1}^{k_{l+1}} \delta_j^{(l+1)}(k) w_{ji}^{(l+1)} \right) f'(\underline{w}_i^{(l)}(k)^T \underline{u}^{(l)}(k))$$

Απλοποιήσεις επιτυγχάνονται αν το στιγμιαίο σφάλμα αντικαταστήσει το μέσο σφάλμα, όπως και στους αλγόριθμους perceptron και LMS

Συνοπτικά:

$$\underline{w}_i^{(l)}(k+1) = \underline{w}_i^{(l)}(k) + \gamma \delta_i^{(l)}(k) \underline{u}^{(l)}(k) \left. \vphantom{\underline{w}_i^{(l)}(k+1)} \right\} \text{ ενημέρωση βαρών}$$

$$\underline{u}^{(l)}(k) = \begin{bmatrix} u_1^{(l)}(k) \\ \dots \\ \dots \\ u_{k_l}^{(l)}(k) \end{bmatrix} \left. \vphantom{\underline{u}^{(l)}(k)} \right\} \text{ υπολογισμοί προς τα εμπρός στο } l\text{-οστό επίπεδο}$$

$$\begin{aligned} u_i^{(l)}(k) &= f(x_i^{(l-1)}(k)) \\ x_i^{(l-1)}(k) &= (\underline{w}_i^{(l-1)}(k))^T \underline{u}^{(l-1)}(k) \end{aligned}$$

$$\left. \begin{aligned} \delta_i^{(L)}(k) &= \epsilon_i^{(L)}(k) f'(x_i^{(L)}(k)) \\ \epsilon_i^{(L)}(k) &= d_i(k) - f(x_i^{(L)}(k)) \\ \text{για } l &= L - i, \dots, 1 \\ \delta_i^{(l)}(k) &= \epsilon_i^{(l)}(k) f'(x_i^{(l)}(k)) \\ \epsilon_i^{(l)}(k) &= \sum_{j=1}^{k_{l+1}} \delta_j^{(l+1)}(k) w_{ji}^{(l+1)}(k) \end{aligned} \right\} \text{ υπολογισμοί προς τα πίσω στο } l\text{-οστό επίπεδο.}$$

17. Το πολυστρωματικό perceptron

Ο αλγόριθμος perceptron ενός στρώματος θεωρεί δύο γραμμικά διαχωρίσιμες κλάσεις ω_1 και ω_2 , δηλαδή υποθέτει ότι υπάρχει \underline{w}_* , έτσι ώστε

$$\underline{w}_*^T \underline{x} > 0 \text{ όταν } \underline{x} \in \omega_1$$

$$\underline{w}_*^T \underline{x} < 0 \text{ όταν } \underline{x} \in \omega_2.$$

Για την εύρεση του \underline{w}_* , θεωρεί την συνάρτηση κόστους

$$J(\underline{w}) = \sum_{\underline{x} \in X_w} \delta_{\underline{x}}(\underline{w}^T \underline{x}),$$

όπου X_w είναι το σύνολο των διανυσμάτων χαρακτηριστικών που ταξινομούνται λάθος από το \underline{w} και

$$\delta_{\underline{x}} = \begin{cases} -1, & \underline{x} \in \omega_1 \\ +1, & \underline{x} \in \omega_2 \end{cases}$$

Για την ελαχιστοποίηση του κόστους $J(\underline{w})$, χρησιμοποιείται ο επαναληπτικός αλγόριθμος

\underline{w}_0 : αυθαίρετο

$$\underline{w}_{k+1} = \underline{w}_k - \rho_k \sum_{\underline{x} \in X_{w_k}} \delta_{\underline{x}} \underline{x}, \quad k = 0, 1, 2, \dots$$

$$\text{Όταν } \lim_{l \rightarrow \infty} \sum_{k=1}^l \rho_k = \infty \quad \text{και} \quad \lim_{l \rightarrow \infty} \sum_{k=1}^l \rho_k^2 < \infty$$

τότε $\underline{w}_k \rightarrow \underline{w}_*$ σε πεπερασμένο αριθμό βημάτων.

Όταν οι κλάσεις ω_1 και ω_2 δεν είναι γραμμικά διαχωρίσιμες, τότε εφαρμογή του μονοστρωματικού αλγορίθμου perceptron μπορεί να οδηγήσει σε εσφαλμένο ταξινομητή. Σε αντίθεση με το μονοστρωματικό perceptron, το πολυστρωματικό perceptron, δηλαδή ένα πολυστρωματικό νευρωνικό δίκτυο που χρησιμοποιεί τον αλγόριθμο οπισθοδιάδοσης για εκμάθηση, μπορεί να υλοποιήσει οποιαδήποτε συνάρτηση.

Όπως είδαμε, ο αλγόριθμος οπισθοδιάδοσης είναι ένας αλγόριθμος εκμάθησης με επιτήρηση, ο οποίος αποτελείται από 2 φάσεις : τους υπολογισμούς προς τα εμπρός (*feedforward computations*) και την ενημέρωση των βαρών (*weight update*). Όταν δοθεί ένα σήμα εισόδου

στο στρώμα εισόδου, υπολογίζεται με βήματα προς τα εμπρός το αντίστοιχο σήμα εξόδου και υλοποιείται ένα *πέρασμα προς τα εμπρός (forward pass)*.

Το τετραγωνικό σφάλμα ανάμεσα στο επιθυμητό και στο πραγματικό σήμα εξόδου ελαχιστοποιείται με μια μέθοδο απότομης καθόδου. Όταν σε κάθε πέρασμα ενημερώνονται τα βάρη προς την κατεύθυνση απότομης καθόδου του συνολικού τετραγωνικού σφάλματος (δηλαδή για όλα τα πρότυπα), τότε έχουμε τον αλγόριθμο *οπισθοδιάδοσης εκτός γραμμής (batch mode backpropagation)*. Συνήθως όμως σε κάθε πέρασμα ενημερώνονται τα βάρη για κάθε πρότυπο εισόδου χωριστά

Παράδειγμα

Ας θεωρήσουμε την υλοποίηση της λογικής πράξης του ``αποκλειστικού ή`` με ένα πολυστρωματικό ΤΝΔ

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

x_1	x_2	XOR	κλάση
1	1	0	ω_2
0	0	0	ω_2
1	0	1	ω_1
0	1	1	ω_1

Οι δύο κλάσεις, ω_1 και ω_2 , δεν είναι γραμμικά διαχωρίσιμες. Αντίθετα, οι λογικές πράξεις “και” και “ή” είναι γραμμικά διαχωρίσιμες:

x_1	x_2	AND	κλάση	OR	κλάση
0	0	0	ω_2	0	ω_2
0	1	0	ω_2	1	ω_1
1	0	0	ω_2	1	ω_1
1	1	1	ω_1	1	ω_1

Το μονοστρωματικό perceptron του σχήματος:

υλοποιεί τη λογική πράξη “ή”. Το μονοστρωματικό perceptron του σχήματος

υλοποιεί τη λογική πράξη “και”.

Για την υλοποίηση του XOR με πολυστρωματικό perceptron, θεωρούμε ένα πολυστρωματικό νευρωνικό δίκτυο με ένα κρυφό επίπεδο και 5 νευρώνες, καθένας από τους οποίους χρησιμοποιεί ως συνάρτηση ενεργοποίησης την \tanh^{-1} . Για την εκπαίδευση του ΤΝΔ, χρησιμοποιείται ο αλγόριθμος οπισθοδιάδοσης.

Αποδεικνύεται στην επόμενη βιβλιογραφική αναφορά ότι το πολυστρωματικό perceptron, όταν εκπαιδευτεί ως ταξινομητής με τον αλγόριθμο οπισθοδιάδοσης, προσεγγίζει τη βέλτιστη μπεϋζιανή συνάρτηση διάκρισης:

D.W.Ruck et al , “The multilayer perceptron as an approximation to a Bayes optimal discriminant function”, IEEE Transactions on Neural Networks, vol. NN-1, pp.296-298, 1990.

18. Εκμάθηση χωρίς επιτήρηση και ομαδοποίηση

Μέχρι τώρα υποθέταμε ότι τα δείγματα που χρησιμοποιούνταν στη φάση εκμάθησης ενός ταξινομητή είχαν μια ετικέτα που έδειχνε την κλάση στην οποία ανήκαν. Αυτές οι διαδικασίες εκμάθησης ονομάζονται *επιτηρούμενες (supervised)*. Θα εξετάσουμε τώρα διαδικασίες *μη επιτηρούμενες (unsupervised)* που χρησιμοποιούν δείγματα χωρίς ετικέτα. Θα εξετάσουμε, δηλαδή, τι μπορεί να γίνει αν όλη η πληροφορία που διαθέτουμε είναι μια συλλογή δειγμάτων για τα οποία αγνοούμε την ταξινόμησή τους.

Στην αρχή θα πίστευε κανείς ότι το παραπάνω πρόβλημα δεν θα είχε πρακτική αξία ή δεν θα ήταν καν επιλύσιμο. Όμως, υπάρχουν τουλάχιστον τρεις λόγοι για τους οποίους το εξετάζουμε:

1. Η συλλογή και η τοποθέτηση ετικετών σε μεγάλο αριθμό δειγμάτων είναι εξαιρετικά υψηλού κόστους και χρονοβόρα. Εναλλακτικά, ένας ταξινομητής μπορεί να σχεδιαστεί χονδρικά με βάση ένα μικρό αριθμό δειγμάτων και να τελειοποιηθεί αργότερα λειτουργώντας χωρίς επιτήρηση.
2. Σε πολλές εφαρμογές, τα χαρακτηριστικά των προτύπων αλλάζουν με το χρόνο και επομένως, χρειάζεται ο ταξινομητής να ακολουθεί αυτές τις αλλαγές χωρίς επιτήρηση.
3. Στις πρώτες φάσεις σχεδίασης ενός ταξινομητή, είναι χρήσιμο να αποκτούμε μια γενική εικόνα των χαρακτηριστικών και της δομής των δεδομένων

Το κατά πόσο είναι δυνατό κατ' αρχή να μάθουμε κάτι από δεδομένα χωρίς ετικέτα εξαρτάται από τις υποθέσεις μας. Αν υποθέσουμε αρχικά ότι οι σ.π.π. των δεδομένων είναι γνωστές, εκτός από έναν αριθμό παραμέτρων, τότε προκύπτει ένα πρόβλημα που μπορούμε να λύσουμε και η λύση μοιάζει πολύ με αυτή του αντίστοιχου προβλήματος για εκμάθηση με επιτήρηση. Στη συνέχεια θα εξετάσουμε το πρόβλημα της ταξινόμησης ως ένα πρόβλημα *ομαδοποίησης (clustering)* των δεδομένων. Θα δούμε ότι οι μέθοδοι ομαδοποίησης, ενώ δεν έχουν θεωρητική βάση, έχουν μεγάλη πρακτική σημασία.

Πυκνότητα μίγματος και αναγνωρισιμότητα

Υποθέτουμε αρχικά ότι γνωρίζουμε την πλήρη πιθανοτική δομή του προβλήματος, εκτός από έναν αριθμό παραμέτρων. Πιο συγκεκριμένα, κάνουμε τις εξής υποθέσεις:

1. Τα δείγματα προέρχονται από ένα γνωστό αριθμό c κλάσεων.
2. Οι εκ των προτέρων πιθανότητες $P(\omega_j)$ για κάθε κλάση είναι γνωστές ($j = 1, 2, \dots, c$)
3. Οι μορφές των σ.π.π. δεδομένων των κλάσεων $p(\underline{x} | \omega_j, \underline{\theta}_j)$, $j = 1, 2, \dots, c$, είναι γνωστές.
4. Οι μόνοι άγνωστοι είναι οι τιμές των c διανυσμάτων $\underline{\theta}_1, \underline{\theta}_2, \dots, \underline{\theta}_c$.

Υποθέτουμε ότι τα δείγματα συγκεντρώνονται επιλέγοντας την κατάσταση της φύσης ω_j με πιθανότητα $P(\omega_j)$ και στη συνέχεια επιλέγοντας ένα διάνυσμα χαρακτηριστικών \underline{x} σύμφωνα με τον πιθανοτικό νόμο $p(\underline{x} | \omega_j, \underline{\theta}_j)$. Επομένως η σ.π.π. των δειγμάτων είναι:

$$p(\underline{x} | \underline{\theta}) = \sum_{j=1}^c p(\underline{x} | \omega_j, \underline{\theta}_j) P(\omega_j),$$

όπου $\underline{\theta} = [\underline{\theta}_1, \underline{\theta}_2, \dots, \underline{\theta}_c]$.

Μια σ.π.π. αυτής της μορφής ονομάζεται *πυκνότητα μίγματος (mixture density)*. Οι δεσμευμένες σ.π.π. $p(\underline{x} | \omega_j, \underline{\theta}_j)$ ονομάζονται *συνιστώσες πυκνότητες* και οι εκ των προτέρων πιθανότητες $P(\omega_j)$ ονομάζονται *παράμετροι ανάμιξης (mixing parameters)*. Οι παράμετροι ανάμιξης θα μπορούσαν επίσης να συμπεριληφθούν στις άγνωστες παραμέτρους $\underline{\theta}$. Ο στόχος μας είναι βασικά, να χρησιμοποιήσουμε τα δείγματα της πυκνότητας μίγματος για να υπολογίσουμε τις παραμέτρους $\underline{\theta}$. Όταν εκτιμήσουμε το $\underline{\theta}$, μπορούμε να αποσυνθέσουμε το μίγμα στις συνιστώσες του και έχουμε λύσει το πρόβλημα.

Μια πρώτη ερώτηση είναι κατά πόσο μπορούμε κατ' αρχήν να εκτιμήσουμε το $\underline{\theta}$. Αν είχαμε διαθέσιμο έναν απεριόριστο αριθμό δειγμάτων και μόνο μια τιμή του $\underline{\theta}$ οδηγούσε στις παρατηρούμενες τιμές της σ.π.π. $p(\underline{x} | \underline{\theta})$, τότε, κατ' αρχήν, η λύση είναι εφικτή. Αν όμως διάφορες τιμές του $\underline{\theta}$ οδηγούν στις ίδιες τιμές του $p(\underline{x} | \underline{\theta})$, τότε δεν υπάρχει ελπίδα μοναδικής λύσης.

Με βάση αυτές τις σκέψεις, έχουμε τον επόμενο ορισμό: μια σ.π.π. $p(\underline{x} | \underline{\theta})$ είναι *αναγνωρίσιμη (identifiable)* αν η υπόθεση $\underline{\theta} \neq \underline{\theta}'$ υποδηλώνει ότι υπάρχει \underline{x} τέτοιο ώστε $p(\underline{x} | \underline{\theta}) \neq p(\underline{x} | \underline{\theta}')$. Είναι φανερό ότι η μελέτη της εκμάθησης χωρίς επιτήρηση απλοποιείται όταν περιορίζεται σε αναγνωρίσιμα μίγματα. Ευτυχώς, στις περισσότερες περιπτώσεις τα μίγματα είναι αναγνωρίσιμα. Μίγματα διακριτών κατανομών όμως δεν είναι πάντα αναγνωρίσιμα.

Εκτίμηση μέγιστης πιθανοφάνειας

Υποθέτουμε ότι διαθέτουμε ένα σύνολο $\mathbf{X} = \{\underline{x}_1, \dots, \underline{x}_n\}$ από n δείγματα χωρίς ετικέτα που έχουν ληφθεί ανεξάρτητα από τη σ.π.π. μίγματος

$$p(\underline{x} | \underline{\theta}) = \sum_{j=1}^c p(\underline{x} | \omega_j, \underline{\theta}_j) P(\omega_j),$$

όπου οι παράμετροι $\underline{\theta}$ είναι άγνωστες, αλλά όχι τυχαίες μεταβλητές. Η πιθανοφάνεια των δειγμάτων είναι εξ ορισμού η από κοινού πυκνότητα

$$p(\mathbf{X} | \underline{\theta}) = \prod_{k=1}^n p(\underline{x}_k | \underline{\theta}).$$

Η εκτίμηση μέγιστης πιθανοφάνειας $\hat{\underline{\theta}}$ είναι η τιμή του $\underline{\theta}$ που μεγιστοποιεί την $p(\mathbf{X} | \underline{\theta})$. Υποθέτοντας ότι η πιθανοφάνεια $p(\mathbf{X} | \underline{\theta})$ είναι διαφορίσιμη συνάρτηση του $\underline{\theta}$ και υποδεικνύοντας με $l(\underline{\theta})$ το λογάριθμό της

$$\left(l(\underline{\theta}) = \log p(\mathbf{X} | \underline{\theta}) = \sum_{k=1}^n \log p(\underline{x}_k | \underline{\theta}) \right)$$

έχουμε:

$$\nabla_{\underline{\theta}_i} l = \sum_{k=1}^n \frac{1}{p(\underline{x}_k | \underline{\theta})} \nabla_{\underline{\theta}_i} \left[p(\underline{x}_k | \omega_j, \underline{\theta}_j) P(\omega_j) \right].$$

$$\text{Θέτουμε } P(\omega_i | \underline{x}_k, \underline{\theta}) = \frac{p(\underline{x}_k | \omega_i, \underline{\theta}_i) P(\omega_i)}{p(\underline{x}_k | \underline{\theta})}$$

και έχουμε:

$$\nabla_{\underline{\theta}_i} l = \sum_{k=1}^n P(\omega_i | \underline{x}_k, \underline{\theta}) \nabla_{\underline{\theta}_i} \log p(\underline{x}_k | \omega_i, \underline{\theta}_i)$$

Αφού $\nabla_{\underline{\theta}_i} l = 0$ όταν $\underline{\theta}_i = \hat{\underline{\theta}}_i$, αναγκαίες συνθήκες για την εύρεση της εκτίμησης μέγιστης πιθανοφάνειας είναι:

$$\sum_{k=1}^n P(\omega_i | \underline{x}_k, \hat{\underline{\theta}}) \nabla_{\underline{\theta}_i} \log p(\underline{x}_k | \omega_i, \hat{\underline{\theta}}_i) = 0, \quad \text{όταν } i = 1, 2, \dots, c.$$

Γενικότερα, αν θεωρήσουμε τις εκ των προτέρων πιθανότητες $P(\omega_i)$ ως άγνωστες παραμέτρους, τότε επιθυμούμε την μεγιστοποίηση της πιθανοφάνειας

$$p(\underline{X} | \underline{\theta}, P(\omega_1), P(\omega_2), \dots, P(\omega_c))$$

ως προς $\underline{\theta}$ και $P(\omega_1), P(\omega_2), \dots, P(\omega_c)$ με τους περιορισμούς

$$P(\omega_i) \geq 0 \quad i=1, 2, \dots, c$$

και

$$\sum_{i=1}^c P(\omega_i) = 1.$$

Βρίσκουμε :

$$\hat{P}(\omega_i) = \frac{1}{n} \sum_{k=1}^n \hat{P}(\omega_i | \underline{x}_k, \hat{\underline{\theta}})$$

και

$$\sum_{k=1}^n \hat{P}(\omega_i | \underline{x}_k, \hat{\underline{\theta}}) \nabla_{\underline{\theta}_i} \log p(\underline{x}_k | \omega_i, \hat{\underline{\theta}}_i) = 0$$

$$\text{όπου } \hat{P}(\omega_i | \underline{x}_k, \hat{\underline{\theta}}) = \frac{p(\underline{x}_k | \omega_i, \hat{\underline{\theta}}_i) \hat{P}(\omega_i)}{\sum_{j=1}^c p(\underline{x}_k | \omega_j, \hat{\underline{\theta}}_j) \hat{P}(\omega_j)}.$$

19. Εκμάθηση χωρίς επιτήρηση με Γκαουσιανές πυκνότητες μίγματος

Ας εξετάσουμε το πρόβλημα της εκμάθησης χωρίς επιτήρηση όταν οι συνιστώσες πυκνότητες είναι πολυμεταβλητές γκαουσιανές, $p(\underline{x} | \omega_i, \underline{\theta}_i) \sim N(\underline{\mu}_i, \underline{\Sigma}_i)$.

Περίπτωση 1: $\underline{\theta}_i = \underline{\mu}_i$: άγνωστα, $\underline{\Sigma}_i$: γνωστά, $P(\omega_i)$: γνωστά

Από την γενική εξίσωση

$$\sum_{k=1}^n P(\omega_i | \underline{x}_k, \hat{\theta}) \nabla_{\theta_i} p(\underline{x}_k | \omega_i, \hat{\theta}) = 0, \quad i=1, 2, \dots, c,$$

έχουμε:

$$\sum_{k=1}^n P(\omega_i | \underline{x}_k, \hat{\underline{\mu}}) \sum_{i=1}^{-1} (\underline{x}_k - \hat{\underline{\mu}}_i) = 0, \quad \hat{\underline{\mu}} = (\hat{\underline{\mu}}_1, \dots, \hat{\underline{\mu}}_c)$$

αφού

$$\log p(\underline{x} | \omega_i, \underline{\mu}_i) = -\log \left[(m)^{d/2} |\underline{\Sigma}_i|^{1/2} \right] - \frac{1}{2} (\underline{x} - \underline{\mu}_i)^T \underline{\Sigma}_i^{-1} (\underline{x} - \underline{\mu}_i)$$

$$\nabla_{\underline{\mu}_i} \log p(\underline{x} | \omega_i, \underline{\mu}_i) = \underline{\Sigma}_i^{-1} (\underline{x} - \underline{\mu}_i)$$

Μετά από λίγες πράξεις, βρίσκουμε:

$$\hat{\underline{\mu}}_i = \frac{\sum_{k=1}^n P(\omega_i | \hat{\underline{x}}_k, \hat{\underline{\mu}}) \underline{x}_k}{\sum_{k=1}^n P(\omega_i | \underline{x}_k, \hat{\underline{\mu}})}$$

Η εξίσωση αυτή δείχνει ότι η εκτίμηση $\hat{\underline{\mu}}_i$ είναι ένας μέσος όρος των δειγμάτων, όπου το κάθε δείγμα έχει βάρος ανάλογο της πιθανότητας να ανήκει στην κλάση ω_i .

Δυστυχώς η έκφραση που βρήκαμε για το $\hat{\underline{\mu}}_i$ δεν είναι εκπεφρασμένη. Αν κάνουμε την αντικατάσταση.

$$P(\omega_i | \underline{x}_k, \hat{\underline{\mu}}) = \frac{p(\underline{x}_k | \omega_i, \hat{\underline{\mu}}_i) P(\omega_i)}{\sum_{j=1}^c p(\underline{x}_k | \omega_j, \hat{\underline{\mu}}_j) P(\omega_j)},$$

τότε καταλήγουμε σε ένα σύστημα μη γραμμικών εξισώσεων. Από όλες τις λύσεις τους, πρέπει να διαλέξουμε εκείνη που πραγματικά μεγιστοποιεί τη συνάρτηση πιθανοφάνειας. Αν ξεκινήσουμε από αρχικές εκτιμήσεις $\hat{\underline{\mu}}_i(0)$, $i=1, 2, \dots, c$, τότε μια επαναληπτική διαδικασία μπορεί να οριστεί ως:

$$\hat{\mu}_i(N+1) = \frac{\sum_{k=1}^n P(\omega_i | \underline{x}_k, \hat{\mu}(N)) \underline{x}_k}{\sum_{k=1}^n P(\omega_i | \underline{x}_k, \hat{\mu}(N))}, N=0, 1, 2, \dots$$

Βέβαια μια τέτοια επαναληπτική διαδικασία δεν οδηγεί κατ'ανάγκη στο ολικό μέγιστο της συνάρτησης πιθανοφάνειας.

Περίπτωση 2: $\underline{\mu}_i, \underline{\Sigma}_i, P(\omega_i)$: άγνωστα

Αν όλες οι παράμετροι $\underline{\mu}_i, \underline{\Sigma}_i, P(\omega_i)$ είναι άγνωστες, τότε, χωρίς άλλους περιορισμούς, η μέθοδος της μέγιστης πιθανοφάνειας οδηγεί σε ανώμαλες λύσεις χωρίς χρησιμότητα. Αυτό σημαίνει ότι εμφανίζεται το φαινόμενο η συνάρτηση πιθανοφάνειας να απειρίζεται καθώς κάποια παράμετρος τείνει προς το μηδέν ή το άπειρο. Προφανώς, τέτοιες ανώμαλες λύσεις δεν έχουν πρακτική χρησιμότητα.

Η παραπάνω παρατήρηση δεν σημαίνει ότι η μέθοδος μέγιστης πιθανοφάνειας πρέπει να απορριφθεί. Έχει παρατηρηθεί εμπειρικά ότι το μεγαλύτερο από τα πεπερασμένα μέγιστα της συνάρτησης πιθανοφάνειας μπορεί να οδηγήσει σε χρήσιμες εκτιμήσεις.

Η διαδικασία εκτίμησης μέγιστης πιθανοφάνειας προκύπτει από τις γενικές εξισώσεις και καταλήγει σε συστήματα μη γραμμικών εξισώσεων. Για την επίλυσή τους, χρησιμοποιούνται επαναληπτικοί αλγόριθμοι, στους οποίους επιβάλλεται ο επιπλέον περιορισμός της αποφυγής ανώμαλων λύσεων.

Περίπτωση 3: $\underline{\mu}_i, \underline{\Sigma}_i, P(\omega_i)$ και c : άγνωστα

Για την περίπτωση αυτή θα μιλήσουμε στο επόμενο μάθημα.

20. Μπεϋζιανή εκμάθηση χωρίς επιτήρηση

Οι μέθοδοι μέγιστης πιθανοφάνειας δεν θεωρούν το διάνυσμα παραμέτρων $\underline{\theta}$ ως τυχαίο, μόνο ως άγνωστο. Έτσι δεν εκμεταλλεύονται εκ των προτέρων γνώση για το $\underline{\theta}$ εκτός ίσως στην επιλογή καλών αρχικών τιμών σε επαναληπτικούς αλγορίθμους. Η μπεϋζιανή προσέγγιση θεωρεί το $\underline{\theta}$ ως τυχαίο διάνυσμα με εκ των προτέρων σ.π.π. $p(\underline{\theta})$ και χρησιμοποιεί τα δείγματα για να υπολογίσει την εκ των υστέρων σ.π.π. $p(\underline{\theta} | \mathbf{X})$.

Υποθέσεις :

1. Ο αριθμός c των κλάσεων είναι γνωστός
2. Οι εκ των προτέρων πιθανότητες $P(\omega_i)$, $i=1, 2, \dots, c$, για κάθε κλάση είναι γνωστές.
3. Η μορφή των δεσμευμένων σ.π.π. $p(\underline{x} | \omega_i, \underline{\theta}_i)$ είναι γνωστή εκτός από τις παραμέτρους $\underline{\theta} = [\underline{\theta}_1, \dots, \underline{\theta}_c]$.
4. Γνωρίζουμε την εκ των προτέρων σ.π.π. $p(\underline{\theta})$.
5. Διαθέτουμε ένα σύνολο X^n , n στατιστικώς ανεξαρτήτων δειγμάτων $\underline{x}_1, \dots, \underline{x}_n$ από την πυκνότητα μίγματος

$$p(\underline{x} | \underline{\theta}) = \sum_{i=1}^c p(\underline{x} | \omega_i, \underline{\theta}_i) P(\omega_i).$$

Από τον κανόνα του Bayes, έχουμε

$$p(\underline{\theta} | X^n) = \frac{p(X^n | \underline{\theta})p(\underline{\theta})}{\int p(X^n | \underline{\theta})p(\underline{\theta})d\underline{\theta}},$$

όπου $p(X^n | \underline{\theta}) = \prod_{k=1}^n p(\underline{x}_k | \underline{\theta})$.

Από την τελευταία σχέση βρίσκουμε

$$p(\underline{\theta} | X^n) = \frac{p(\underline{x}_n | \underline{\theta})p(\underline{\theta} | X^{n-1})}{\int p(\underline{x}_n | \underline{\theta})p(\underline{\theta} | X^{n-1})d\underline{\theta}}, \quad \text{με } X^{n-1} = \{\underline{x}_1, \dots, \underline{x}_{n-1}\}.$$

Η εξίσωση αυτή αποτελεί τη βασική σχέση της μπεϋζιανής εκμάθησης χωρίς επιτήρηση. Από πρώτη άποψη, η εκμάθηση χωρίς επιτήρηση δεν διαφέρει σημαντικά από την εκμάθηση με επιτήρηση. Υπάρχει όμως μια σημαντική διαφορά ως προς την αναγνωρισιμότητα. Μη αναγνωρισιμότητα στην περίπτωση της εκμάθησης με επιτήρηση σημαίνει μόνο ότι υπάρχουν περισσότερα από ένα διανύσματα παραμέτρων που οδηγούν στις ίδιες συνιστώσες πυκνότητας. Το γεγονός αυτό είναι δευτερεύουσας σημασίας. Μη αναγνωρισιμότητα στην εκμάθηση χωρίς επιτήρηση σημαίνει ότι το μίγμα δεν μπορεί να αναλυθεί σωστά στις συνιστώσες του. Μια άλλη διαφορά έγκειται στο ότι δεν υπάρχει τρόπος να βρεθούν απλές αναλυτικές λύσεις στην εκμάθηση χωρίς επιτήρηση.

Ομαδοποίηση Δεδομένων (Data Clustering)

Ας εξετάσουμε πάλι το αρχικό πρόβλημα όπου διαθέτουμε ένα σύνολο διανυσμάτων χαρακτηριστικών διάστασης d χωρίς ετικέτα. Τα διανύσματα αυτά αποτελούν ένα σύννεφο

σημείων στο χώρο των d διαστάσεων. Αν επιπλέον γνωρίζαμε ότι τα σημεία αυτά είχαν προέλθει από την κανονική κατανομή, τότε ο μέσος και ο πίνακας συνδιακύμανσή τους θα αποτελούσαν την πλήρη (και συμπαγή) περιγραφή των δεδομένων. Ο μέσος περιγράφει το “κέντρο βαρύτητας” των δεδομένων, ενώ ο πίνακας συνδιακύμανσης τη διασπορά των δεδομένων γύρω από το κέντρο. Αν τα δείγματα δεν είχαν προέλθει από την κανονική κατανομή, τότε η περιγραφή τους με βάση το μέσο και τον πίνακα συνδιακύμανσης θα μπορούσε να οδηγήσει σε εσφαλμένα συμπεράσματα. Κατά κάποιο τρόπο, θα προσπαθούσαμε στην περίπτωση αυτή να επιβάλουμε παρά να αναγνωρίσουμε κάποια δομή στα δεδομένα.

Μια εναλλακτική λύση θα ήταν να βρούμε ομάδες στα δεδομένα χρησιμοποιώντας μια διαδικασία ομαδοποίησης (*clustering procedure*). Σε γενικές γραμμές, η ομαδοποίηση περιγράφει τα δεδομένα σε σχέση με ομάδες τους που εσωτερικά διακρίνονται από έντονη ομοιότητα. Χρησιμοποιούνται κριτήρια όπως το άθροισμα των τετραγώνων από τα κέντρα των ομάδων και η ομαδοποίηση μεγιστοποιεί ή ελαχιστοποιεί το κριτήριο.

Ιεραρχική Ομαδοποίηση (Hierarchical Clustering)

Ας θεωρήσουμε μια ακολουθία διαμερισμών n δειγμάτων σε c ομάδες. Αρχικά διαμερίζουμε τα n δείγματα σε n ομάδες, καθεμία από τις οποίες περιέχει ακριβώς ένα δείγμα. Στη συνέχεια διαμερίζουμε τα n δείγματα σε $n-1$ ομάδες, σε $n-2$ ομάδες κλπ. μέχρι διαμερισμό σε μία ομάδα που περιέχει όλα τα n δείγματα. Θα λέμε ότι βρισκόμαστε στο επίπεδο k όταν $c=n-k+1$. Αν η ακολουθία των διαμερισμών σε ομάδες έχει την ιδιότητα δείγματα που ανήκουν στην ίδια ομάδα σε κάποιο επίπεδο να παραμένουν στην ίδια ομάδα σε υψηλότερα επίπεδα θα λέμε ότι η ακολουθία αποτελεί μια *ιεραρχική ομαδοποίηση*.

Κλασσικά παραδείγματα ιεραρχικής ομαδοποίησης εμφανίζονται στη βιολογική ταξινόμηση, όπου άτομα ομαδοποιούνται σε *είδη (species)*, είδη ομαδοποιούνται σε *γένη (genera)*, γένη ομαδοποιούνται σε *οικογένειες (families)*, κ.ο.κ.

Σε κάθε ιεραρχική ομαδοποίηση αντιστοιχεί ένα *δενδρόγραμμα (dendrogram)*.

Για να διαπιστώσουμε κατά πόσο η ομαδοποίηση είναι φυσική ή επιβεβλημένη, χρησιμοποιείται κάποιο μέτρο ομοιότητας το οποίο μειώνεται καθώς αυξάνεται το επίπεδο ομαδοποίησης.

Οι διαδικασίες ιεραρχικής ομαδοποίησης είναι εξαιρετικά δημοφιλείς και διακρίνονται σε *συγχωνευτικές (agglomerative)* και *διαιρετικές (divisive)*. Οι συγχωνευτικές (από κάτω προς τα πάνω) ξεκινούν από n ομάδες και δημιουργούν μια ακολουθία από διαδοχικές συγχωνεύσεις ομάδων. Οι διαιρετικές (από πάνω προς τα κάτω) ξεκινούν με όλα τα δείγματα σε μια ομάδα και δημιουργούν μια ακολουθία από διαδοχικές διαιρέσεις ομάδων.

Για παράδειγμα, η συγχωνευτική ιεραρχική ομαδοποίηση λειτουργεί ως εξής:

1. $\hat{c} = n$, $X_i = \{\underline{x}_i\}$, $i = 1, 2, \dots, n$
2. Αν $\hat{c} \leq c$, σταμάτα (c : ελάχιστος επιτρεπτός αριθμός ομάδων)
3. Βρες το πλησιέστερο ζευγάρι διακεκριμένων ομάδων, έστω X_i και X_j
4. Συγχώνευσε τις ομάδες X_i και X_j και μείωσε το \hat{c} κατά ένα
5. Πήγαινε στο 2.

Για τη μέτρηση της “απόστασης” μεταξύ των ομάδων, μπορούν να χρησιμοποιηθούν τα μέτρα

$$d_{\min}(X_i, X_j) = \min_{\underline{x} \in X_i, \underline{x}' \in X_j} \|\underline{x} - \underline{x}'\|$$

$$d_{\max}(X_i, X_j) = \max_{\underline{x} \in X_i, \underline{x}' \in X_j} \|\underline{x} - \underline{x}'\|$$

$$d_{\text{avg}}(X_i, X_j) = \frac{1}{n_i n_j} \sum_{\underline{x} \in X_i} \sum_{\underline{x}' \in X_j} \|\underline{x} - \underline{x}'\|$$

$$d_{\text{mean}}(X_i, X_j) = \|\underline{m}_i - \underline{m}_j\|$$

21. Νευρωνικά δίκτυα με εκμάθηση χωρίς επιτήρηση

Εκτός από τα ΤΝΔ όπου η εκμάθηση γίνεται καθαρά με επιτήρηση, υπάρχουν και πρότυπα ΤΝΔ όπου η εκμάθηση γίνεται *χωρίς επιτήρηση (unsupervised learning)* ή με *καταγραφή (recording learning)*. Στην *εκμάθηση χωρίς επιτήρηση*, διατίθεται μόνο ένα σύνολο

διανυσμάτων χαρακτηριστικών και το δίκτυο εξελίσσεται έτσι ώστε να εξάγει χαρακτηριστικά ή κανονικότητες που υπάρχουν σε αυτά χωρίς να γνωρίζει την κλάση στην οποία ανήκει κάθε διάνυσμα. Με άλλα λόγια, το δίκτυο κατηγοριοποιεί επίμονα χαρακτηριστικά χωρίς ανατροφοδότηση από το περιβάλλον. Πρότυπα ΤΝΔ με εκμάθηση χωρίς επιτήρηση αποτελούν η ανταγωνιστική εκμάθηση (*competitive learning*), ο αυτο-οργανωνόμενος χάρτης του Kohonen (*Kohonen self-organizing map*) και η ανάλυση κυρίων συνιστωσών (*principal component analysis*).

Στην εκμάθηση με καταγραφή χρησιμοποιούνται δίκτυα συσχετιστικής μνήμης (*associative memory networks*). Ένα δίκτυο συσχετιστικής μνήμης σχεδιάζεται με το να καταγραφούν διάφορα ιδανικά πρότυπα στις σταθερές καταστάσεις του δικτύου. Κατά την φάση λειτουργίας του δικτύου ως ταξινομητή, περιμένουμε ότι η κατάστασή του θα καταλήξει σε κάποιο από τα αποθηκευμένα πρότυπα μετά από ένα μικρό αριθμό επαναλήψεων. Για την εκπαίδευση του δικτύου, χρησιμοποιούνται τεχνικές στοχαστικής βελτιστοποίησης. (π.χ. προσομοιωμένη ψύξη (*simulated annealing*)). Παράδειγμα ΤΝΔ με εκμάθηση με καταγραφή αποτελεί το δίκτυο του Hopfield, το οποίο έχει την ικανότητα να ανακτά πρότυπα που έχουν “μολυνθεί” από θόρυβο.

22. Αρχές εξελικτικού υπολογισμού και εφαρμογές

Γενικά

Η αρχή της εξέλιξης αποτελεί την πρωταρχική ενοποιητική ιδέα της βιολογίας, συνενώνοντας κάθε οργανισμό μαζί σε μία ιστορική αλυσίδα από γεγονότα. Κάθε δημιουργία στην αλυσίδα είναι το προϊόν μιας σειράς από ``ατυχήματα`` τα οποία έχουν ταξινομηθεί λεπτομερώς κάτω από την επιλεκτική πίεση του περιβάλλοντος. Για πολλές γενιές, η τυχαία μεταβολή και η φυσική επιλογή διαμορφώνουν τις συμπεριφορές των ατόμων και των ειδών ώστε να προσαρμόζονται στις απαιτήσεις των περιβαλλόντων τους.



Σχήμα 1: Η εξέλιξη αντιμετωπίζει τα προβλήματα με τρόπους που δεν συμβαίνουν ποτέ σε ανθρώπους οι οποίοι σκοπεύουν να σκεφτούν γραμμικά. Μάρτυρας ο γεμάτος φύλλα θαλάσσιος δράκοντας του οποίου τα άκρα και το πίσω μέρος αναλαμβάνουν τον σχηματισμό τα βλάστησης που τον περιβάλλει, κρύβοντας έτσι τον εαυτό του από αρπακτικά.

Αυτή η προσαρμογή μπορεί να είναι αρκετά ασυνήθιστη και αναγκαστική (Σχήμα 1), γεγονός που αποτελεί μία καθαρή ένδειξη ότι η εξέλιξη είναι δημιουργική. Ενώ η εξέλιξη δεν έχει κανένα ουσιώδη σκοπό - είναι απλά το αποτέλεσμα των φυσικών νόμων που ενεργούν μέσα στους πληθυσμούς και στα είδη - είναι ικανή για μηχανικές λύσεις των προβλημάτων

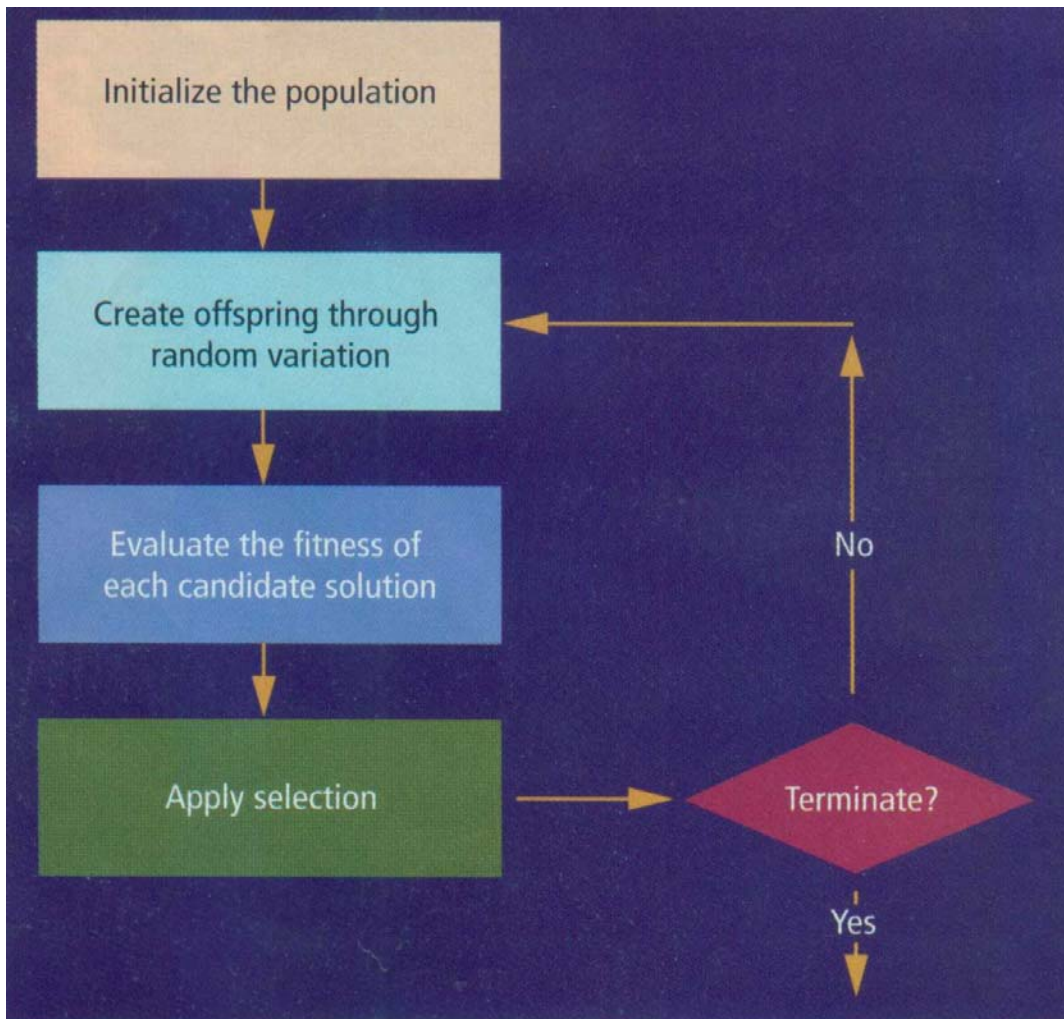
επιβίωσης τα οποία είναι μοναδικά για την κατάσταση του κάθε ατόμου και με κάθε μέτρο αρκετά εφευρετική.

Φανταστείτε τι αποτελέσματα μπορεί να έχει η εξελικτική διαδικασία μέσα σε ένα υπολογιστή. Θα μπορούσε να παρέχει ένα πλαίσιο για την αντιμετώπιση σύγχρονων τεχνολογικών προβλημάτων - εκείνων που περιέχουν χαοτικές αναταραχές, τυχαιότητα, και σύνθετες δυναμικές μεταβλητές - τα οποία οι παραδοσιακοί αλγόριθμοι αδυνατούν να λύσουν. Πράγματι, το πεδίο της εξελικτικής διεργασίας είναι από τα πεδία της επιστήμης και της τεχνολογίας των υπολογιστών που αναπτύσσονται πιο γρήγορα για τον εξής και μόνο λόγο: αντιμετωπίζει πολλά προβλήματα τα οποία προηγουμένως ήταν πέρα από κάθε λογική και απλησίαστα, όπως η άμεση δημιουργία φαρμάκων, ευέλικτες λύσεις σε αλυσιδωτά διοικητικά προβλήματα ή η άμεση ανάλυση των διάφορων τακτικών άμυνας στο πεδίο μάχης. Είναι πιθανό το πεδίο αυτό να μπορεί να εκπληρώσει το όνειρο της τεχνητής νοημοσύνης δηλαδή έναν υπολογιστή ο οποίος να μπορεί να μάθει από μόνος του και να γίνει ένας ειδικός σε μία επιλεγμένη περιοχή.

Σε πιο γενικές γραμμές, η εξέλιξη μπορεί να περιγραφεί ως μία επαναληπτική διαδικασία που αποτελείται από δύο βήματα και περιέχει τυχαία μεταβολή η οποία ακολουθείται από επιλογή. Η σύνδεση μεταξύ της περιγραφής της εξέλιξης και των βελτιστοποιημένων αλγορίθμων οι οποίοι αποτελούν το σήμα ποιότητας της εξελικτικής διεργασίας είναι απλή.

Όπως η φυσική εξέλιξη ξεκινάει από έναν αρχικό πληθυσμό πλασμάτων, έτσι και η αλγοριθμική προσέγγιση ξεκινάει επιλέγοντας ένα αρχικό σύνολο από ισχυρές λύσεις για ένα ιδιαίτερο πρόβλημα. Το σύνολο μπορεί να επιλεγθεί γεννώντας λύσεις τυχαία ή χρησιμοποιώντας για τον συγκεκριμένο σκοπό κάθε διαθέσιμη γνώση σχετικά με το πρόβλημα.

Αυτές οι ``γονικές`` λύσεις κατόπιν παράγουν τους ``απογόνους`` από μία προεπιλεγμένη σειρά από τυχαία μεταβολή. Οι λύσεις που προκύπτουν εκτιμούνται για την αποτελεσματικότητά τους -- για την ``προσαρμοστικότητά`` τους -- και για την υφιστάμενη επιλογή. Έτσι όπως η φύση επιβάλλει τον κανόνα της ``επιβίωσης των πιο κατάλληλων``, έτσι κι εδώ οι λύσεις που είναι λιγότερο εφικτές απομακρύνονται από την παραπέρα μελέτη και η διαδικασία επαναλαμβάνεται μόνο για επιτυχείς παραγωγές (Σχήμα 2).



Σχήμα 2: Ένας εξελικτικός αλγόριθμος ξεκινάει με αρχικοποίηση ενός πληθυσμού από υποψήφιες λύσεις σε ένα πρόβλημα. Κατόπιν δημιουργούνται νέες λύσεις με τυχαία μεταβολή αυτών που προέρχονται από τον αρχικό πληθυσμό. Όλες οι λύσεις μετριοούνται σχετικά με το πόσο καλά κατευθύνουν το θέμα. Τελικά, ένα κριτήριο επιλογής εφαρμόζεται για να ξεχωρίσει (και να απορρίψει) αυτές που βρίσκονται κάτω από ένα φυσιολογικό όριο. Η διαδικασία επαναλαμβάνεται χρησιμοποιώντας το επιλεγμένο σύνολο λύσεων μέχρι να συναντήσει ένα ειδικό κριτήριο.

Το πρόβλημα της υπόθεσης

Οι παραδοσιακοί αλγόριθμοι απαιτούν όσον αφορά στην ανακάλυψη των πιο κατάλληλων λύσεων – βελτιστοποιημένων αλγόριθμων -- να κάνουν οι χρήστες τους αρκετές υποθέσεις σχετικά με πώς να εκτιμούν την καταλληλότητα μιας λύσης. Σύμφωνα με αυτούς τους παραδοσιακούς αλγόριθμους, η εκτίμηση έχει να κάνει με πολλά ονόματα: συνάρτηση καταλληλότητας ή συνάρτηση κόστους, επιφάνεια απόκρισης, ή στην τεχνολογία, δείκτης απόδοσης.) Για παράδειγμα οι λεγόμενοι γραμμικοί προγραμματιστικοί αλγόριθμοι

προϋποθέτουν οι συναρτήσεις κόστους να είναι επίσης γραμμικές -- ένα άθροισμα από σταθμισμένους ανεξάρτητους όρους.

Μία άλλη παραδοσιακή προσέγγιση, στην οποία αναζητείται το σημείο μηδενισμού της παραγωγού, δηλαδή το μέγιστο ή το ελάχιστο, απαιτεί μια ομαλά διαφορίσιμη συνάρτηση κόστους. Είναι αδύνατο να αντιμετωπιστεί έτσι μία αιφνίδια και ασυνεχής αλλαγή.

Αλλά οι εξελικτικοί αλγόριθμοι δεν απαιτούν τέτοιες υποθέσεις. Βασικά, ο δείκτης αποτελεσματικότητας απαιτεί μόνο να είναι εφικτή η ταξινόμηση (κατάταξη) δύο ανταγωνιστικών λύσεων. Αυτό σημαίνει ότι πρέπει απλώς να καθορίζεται ότι μία λύση είναι κάπως καλύτερη από μία άλλη. Αυτό επιτρέπει να αντιμετωπιστεί μία μεγάλη σειρά προβλημάτων, τα οποία διαφέρουν από τα συμβατικά προβλήματα της εξελικτικής προσέγγισης. Με λίγα λόγια, οι εξελικτικοί αλγόριθμοι μπορούν συχνά να αναλύσουν προβλήματα τα οποία δεν μπορούν αντιμετωπιστούν με τις συνηθισμένες αριθμητικές τεχνικές.

Στην πραγματικότητα, μία εξελικτική προσέγγιση στην επίλυση τεχνολογικών προβλημάτων προσφέρει σημαντικά πλεονεκτήματα. Ένα τέτοιο πλεονέκτημα είναι η προσαρμοστικότητα των λύσεων σε αλλαγή των καταστάσεων. Για παράδειγμα ας υποθέσουμε ότι ένας μανάτζερ πρέπει να βρει το καλύτερο πρόγραμμα για την λειτουργία ενός εργοστασίου. Αναμφίβολα, θα υπάρχουν αρκετοί περιορισμοί: η διαθεσιμότητα του προσωπικού, ο αριθμός των μηχανών, ο χρόνος που απαιτείται για την αλλαγή των ρυθμίσεων των μηχανών, και τόσοι άλλοι. Ακόμα και αν ο μανάτζερ μπορούσε να βρει έναν βέλτιστο προγραμματισμό ο οποίος θα ήταν περισσότερο επικερδής, θα χρειαζόταν ακόμη να εξετάσει και το ενδεχόμενο ότι οι μηχανές θα μπορούσαν να χαλάσουν ή το προσωπικό να μην φτάσει στην εργασία την ώρα του. Στην καθημερινή ζωή, το πρόβλημα που εμφανίζεται κάθε στιγμή μπορεί να έχει εκτραπεί σημαντικά από το πρόβλημα που αρχικά αναμενόταν.

Δυστυχώς, σε πολλές παραδοσιακές διαδικασίες βελτιστοποίησης, ο υπολογισμός πρέπει να ξεκινήσει ξανά από την αρχή αν αλλάξει μία μεταβλητή του προβλήματος. Αυτό βέβαια είναι προγραμματιστικά πανάκριβο. Στην άλλη περίπτωση, με έναν εξελικτικό αλγόριθμο, ο τρέχων πληθυσμός μπορεί να χρησιμοποιηθεί σαν ρεζέρβα της αποθηκευμένης γνώσης, στον οποίο μπορεί να εφαρμοσθεί κατά την μετάβαση προς ένα δυναμικό περιβάλλον. Η εκκίνηση από την αρχή δεν είναι απαραίτητη.

Ένα άλλο πλεονέκτημα μιας εξελικτικής προσέγγισης στην επίλυση του προβλήματος είναι το ότι είναι ικανή να παράγει αρκετά καλές λύσεις και αρκετά γρήγορα για αυτούς οι οποίοι θα τους χρησιμοποιήσουν. Αυτή η δυνατότητα ίσως απεικονίζεται καλύτερα με το κλασικό πρόβλημα του περιοδεύοντος πωλητή, το οποίο είναι το εξής: «Υποθέστε έναν πωλητή ο οποίος θα πρέπει να επισκεφθεί τους πελάτες που βρίσκονται σε διαφορετικές πόλεις, και κατόπιν να επιστρέψει στο σπίτι του. Ποια είναι η μικρότερη διαδρομή που πρέπει να κάνει ο πωλητής μεταξύ αυτών των πόλεων, επισκεπτόμενος κάθε πόλη μία και μόνο φορά;»

Το πρόβλημα εύκολα δηλώνεται και αναλύεται αλλά δύσκολα λύνεται. Ανήκει στην κατηγορία εκείνων των προβλημάτων που αναφέρονται ως *μη πολυωνυμικά προβλήματα (NP-complete)*. Για τέτοια προβλήματα, δεν υπάρχουν γνωστοί αλγόριθμοι που να είναι ικανοί να

παράγουν την καλύτερη απάντηση σε ένα διάστημα χρόνου το οποίο αυξάνει μόνο σαν μια πολυωνυμική συνάρτηση του αριθμού των στοιχείων του προβλήματος.

Στην πραγματικότητα, ο αριθμός των πιθανών διαδρομών σε οποιοδήποτε πρόβλημα περιοδεύοντος πωλητή αυξάνει σαν μία παραγοντική συνάρτηση του αριθμού των πόλεων. Έτσι για 100 πόλεις υπάρχουν πάνω από 10^{155} διαφορετικά πιθανά μονοπάτια μεταξύ όλων των πόλεων από τα οποία θα μπορούσε κάποιος να επιλέξει. Παίρνοντας υπόψη ότι έως τώρα έχουν υπολογιστεί περίπου 10^{18} δευτερόλεπτα στην ιστορία του σύμπαντος, η απλή εφαρμογή ενός αλγόριθμου ``ωμής βίας`` (brute force) που θα ψάξει όλες τις πιθανές λύσεις σε ένα πρόβλημα του περιοδεύοντος πωλητή ενός αρκετά μετρίου μεγέθους είναι σίγουρο ότι θα αποτύχει.

Αντίθετα τώρα, ας θεωρήσουμε μία εξελικτική προσέγγιση στην ανακάλυψη μιας χρήσιμης λύσης σε ένα τέτοιο πρόβλημα. Η προσέγγιση αυτή προϋποθέτει τα εξής τέσσερα βήματα:

1. Επιλογή της αναπαράστασης της λύσης.
2. Επιλογή ενός συντελεστή τυχαίας μεταβολής.
3. Καθορισμό ενός κανόνα για μία επιβιώσιμη λύση.
4. Αρχικοποίηση του πληθυσμού.

Εφόσον καθοριστούν τα παραπάνω, ο εξελικτικός αλγόριθμος θα ξεκινήσει να παράγει λύσεις και θα συνεχίσει να παράγει καλύτερες λύσεις καθώς ο χρόνος περνά. Καθένα από αυτά τα βήματα αναλύεται και επεξηγείται σε ένα μεγάλο βαθμό στις παραγράφους που ακολουθούν.

Εξελικτικά βήματα

Για να αναπαραστήσουμε μία πιθανή λύση έχοντας υπόψη τους περιορισμούς ενός υπολογιστή, θα πρέπει να οριστεί αρχικά μία δομή για τα δεδομένα η οποία θα κωδικοποιεί κάθε πιθανή λύση που θα ήταν επιθυμητό να υπολογιστεί. Εδώ δεν υπάρχει άλλη πιο απλή και καλύτερη επιλογή για την αναπαράσταση (αυτό αποδεικνύεται και μαθηματικά) από λίγη δημιουργικότητα.

Μια πιθανή αναπαράσταση των δεδομένων για το πρόβλημα του περιοδεύοντος πωλητή είναι η εξακρίβωση κάθε διαφορετικής πιθανότητας σε μία μετάθεση. Πιο απλά, αν υπήρχαν έξι πόλεις (περιέχοντας και την αρχική θέση που βρίσκεται ο πωλητής), τότε μία πιθανή λύση θα μπορούσα να είναι η [1 2 3 4 5 6], η οποία δηλώνει την σειρά της προόδου. (Ας σημειωθεί, ότι καθώς το πρόβλημα απαιτεί μια κλειστή διαδρομή δηλαδή ένα κλειστό βρόγχο-το πρώτο στοιχείο στην σειρά είναι επίσης και το τελευταίο. Επίσης, ας σημειωθεί ότι εφόσον αυτός είναι ένας βρόγχος, δεν έχει σημασία στην πραγματικότητα από ποια πόλη ξεκινάει ο πωλητής. Κάθε μετάθεση αυτών των πόλεων θα μπορούσε να είναι μία ακόμα, περισσότερο ή λιγότερο επιθυμητή λύση στο πρόβλημα.

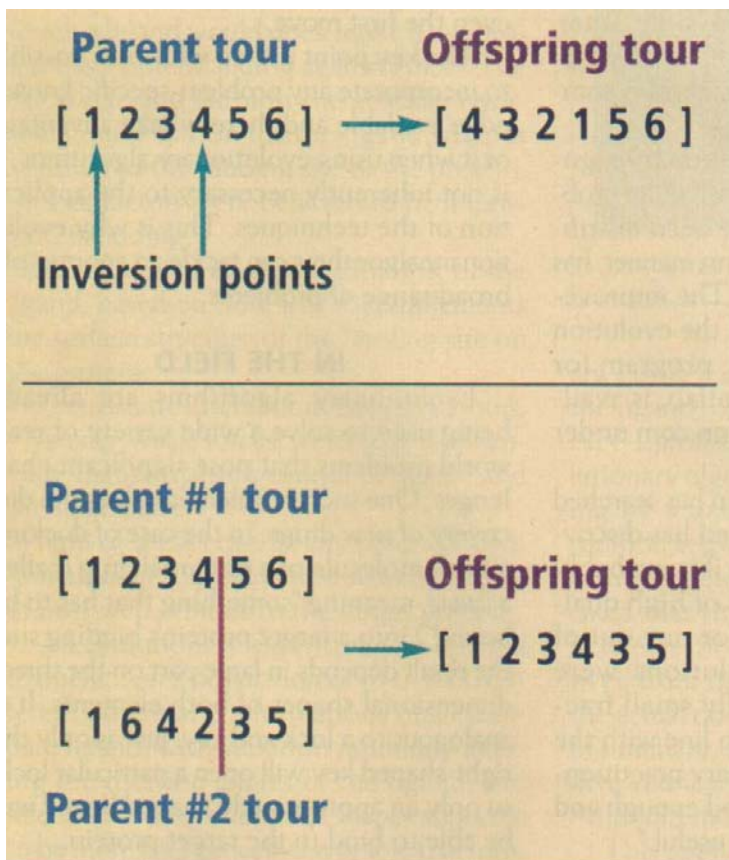
Υποθέστε ότι επιλέγεται αυτή η αναπαράσταση. Τότε η συνάρτηση κόστους, η οποία είναι ο μέσος για τον υπολογισμό κάθε υποψήφιας λύσης, θα πρέπει επίσης να καθοριστεί. Εδώ η αποστολή είναι εμφανής: να ταξιδέψει την μικρότερη δυνατή απόσταση. Έτσι το ``κόστος``

κάθε λύσης μπορεί να οριστεί ως ισοδύναμο με την απόσταση της διαδρομής, με τις συντομότερες διαδρομές να προτιμούνται περισσότερο από τις μακρύτερες. Φυσικά, τα πράγματα μπορούν να περιπλακούν ακόμα περισσότερο ενσωματώνοντας επιπρόσθετα ζητήματα λογικοφανών προβλημάτων όπως εκείνο της ελαχιστοποίησης του ταξιδιού που απαιτείται κατά την διάρκεια των ωρών μέγιστης κυκλοφορίας ή όπως εκείνο της απαίτησης να επισκεφθεί ο πωλητής συγκεκριμένους πελάτες για παράδειγμα το απόγευμα, κλπ. Έτσι λοιπόν ο υπολογισμός υποψηφίων λύσεων μπορεί να γίνει ακόμα πιο περίπλοκος. Η πολυπλοκότητα κάνει μόνο την εφαρμογή του υπολογισμού ενός εξελικτικού αλγορίθμου πιο σχετική, διότι μετατοπίζει γρήγορα το πρόβλημα από το πεδίο των πιο παραδοσιακών προβλημάτων βελτιστοποίησης. Για χάρη αυτού του παραδείγματος, ας υποθέσουμε ότι η απλή συνολική απόσταση της διαδρομής αποτελεί το μέτρο της ποιότητας μιας λύσης.

Το δεύτερο βήμα είναι το να επινοήσουμε έναν συντελεστή (ή συντελεστές) τυχαίας μεταβολής ο οποίος μπορεί να χρησιμοποιηθεί για να παράγει απογονικές λύσεις. Υπάρχουν πολλές επιλογές. Στη φύση υπάρχουν δύο γενικοί τρόποι αναπαραγωγής: ο σεξουαλικός και ο μη σεξουαλικός. Στην σεξουαλική αναπαραγωγή, οι δύο γονείς επιτελούν ένα είδος ανταλλαγής γενετικού υλικού το οποίο ανασυνδυάζεται για να σχηματίσει έναν απόγονο. Η μη σεξουαλική αναπαραγωγή είναι βασικά κλωνοποίηση, αλλά αλλαγές χρωμοσωμάτων από διάφορους σχηματισμούς μπορούν να εισχωρήσουν μέσα στην γενετική πληροφορία που μεταβιβάζεται από τον γονέα στον απόγονο. Αυτοί οι συντελεστές αξίζει να ενσωματωθούν μέσα σε έναν εξελικτικό αλγόριθμο.

Σκεπτόμενος κάποιος ακόμα πιο γενικά θα μπορούσε να αναρωτηθεί γιατί δεν επινοούμε άλλα σχήματα μεταβολής τα οποία δεν υπάρχουν στην φύση. Παραδείγματα περιλαμβάνουν τον ανασυνδυασμό του γενετικού υλικού από τρεις ή περισσότερους γονείς, επιτρέποντας μια δημοκρατική ψηφοφορία μεταξύ των γονέων που εμπλέκονται στην αναπαραγωγή. Δεν υπάρχει κατ' ουσία κανένα όριο στους τύπους των συντελεστών μεταβολής οι οποίοι μπορούν να επινοηθούν, ούτε κανέναν λόγο να περιοριστούν από τη Φύση.

Η τελική επιτυχία ενός εξελικτικού αλγορίθμου εξαρτάται βασικά από το πόσο καλά ταιριάζουν οι συντελεστές μεταβολής, η απεικόνιση και η συνάρτηση υπολογισμού. Διαφορετικοί συντελεστές θα ποικίλλουν στην χρησιμότητα ανά κατάσταση. Όπως ακριβώς και με την απεικόνιση, δεν υπάρχει κάποιος απλός και τέλειος συντελεστής μεταβολής για όλα τα προβλήματα (και αυτό επίσης μπορεί να αποδειχτεί και μαθηματικά).



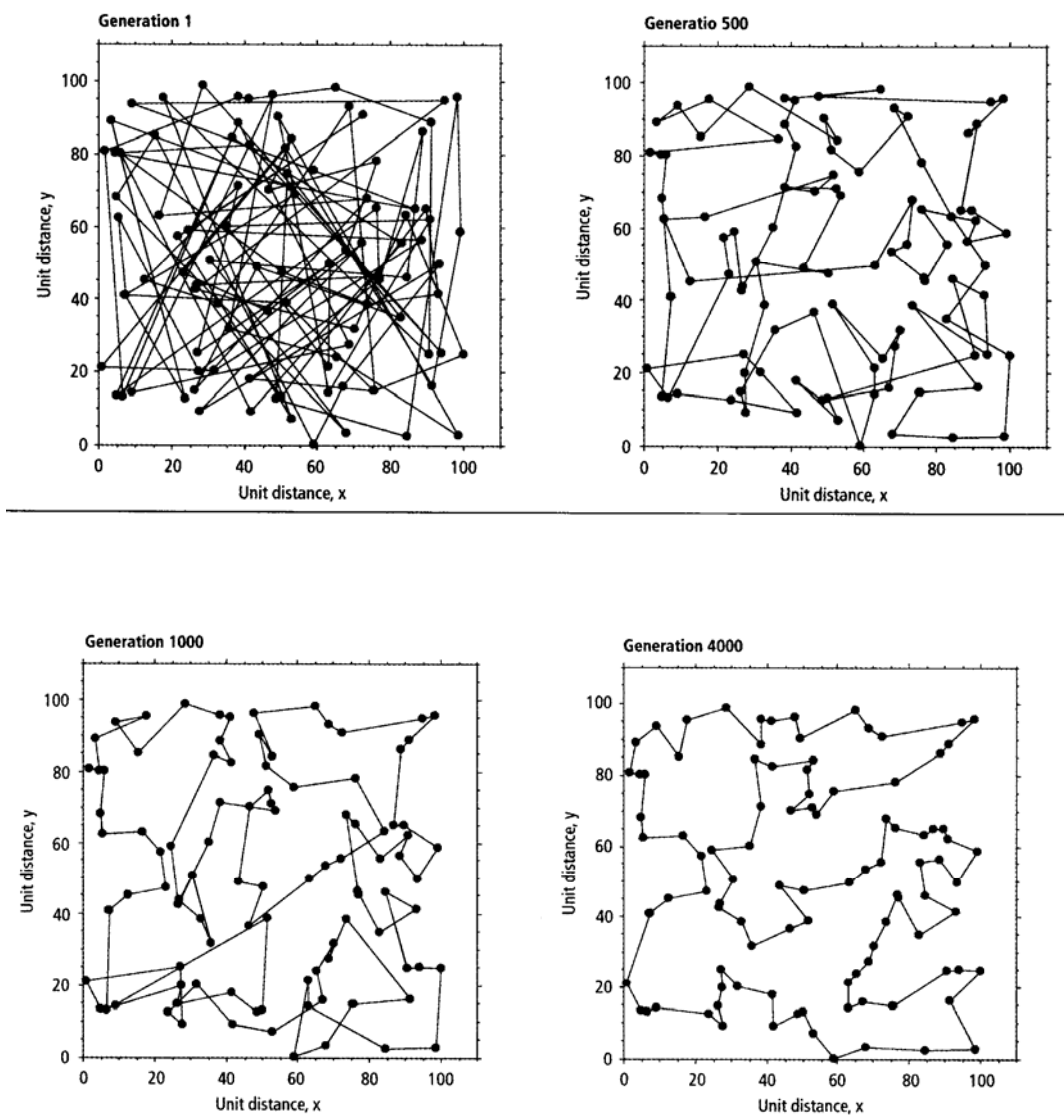
Σχήμα 3: Το πρόβλημα του περιοδεύοντος πωλητή είναι το εξής: να βρεθεί η μικρότερη διαδρομή για την επίσκεψη όλων των πόλεων που επιλέχθηκαν. Με έξι πόλεις [1 2 3 4 5 6], ένας απλός γονικός τελεστής θα μπορούσε να δημιουργήσει μια λύση (απόγονο) αντιστρέφοντας την σειρά επίσκεψης μεταξύ δύο τυχαία επιλεγμένων σημείων (κορυφή). Με έναν διπλό γονικό τελεστή, δύο διαδρομές θα μπορούσαν να διαιρεθούν σε κάποιο τυχαίο σημείο και τα κομμάτια θα ενώνονταν για να δημιουργήσουν έναν απόγονο (βάση). Η διαδρομή εδώ δεν είναι ολοκληρωμένη και επιπρόσθετοι κανόνες είναι αναγκαίοι για να την διορθώσουν ή να την αποτρέψουν από το να πραγματοποιηθεί.

Θεωρήστε δύο πιθανότητες για το πρόβλημα του περιοδεύοντος πωλητή οι οποίες αναπαριστούνται με τη χρήση μεταθέσεων. Μία επιλογή είναι να χρησιμοποιήσουμε έναν απλό γονέα και να παράγουμε έναν απόγονο με τυχαία επιλογή δύο θέσεων προς τον γονέα αντιστρέφοντας την λίστα με τις πόλεις σε αυτό το τμήμα (Σχήμα 3). Μια άλλη επιλογή είναι να χρησιμοποιήσουμε δύο γονείς, επιλέγοντας ένα τυχαίο σημείο κατά μήκος της μετάθεσης και έπειτα παίρνοντας το πρώτο τμήμα των πόλεων από τον πρώτο γονέα και το επακόλουθο τμήμα των πόλεων από τον δεύτερο γονέα.

Ο πρώτος συντελεστής φαίνεται να μοιάζει λίγο με την μη σεξουαλική αναπαραγωγή, ενώ ο δεύτερος φαίνεται να μοιάζει περισσότερο με την σεξουαλική αναπαραγωγή. Ενώ η πρώτη μεταβολή θα παράγει πάντοτε μία νόμιμη διαδρομή (επισκέπτεται κάθε πόλη μία και μόνο φορά), η δεύτερη μεταβολή μπορεί να παράγει μη νόμιμες διαδρομές, επειδή κάποιοι απόγονοι μπορεί να περιέχουν περισσότερα από ένα αντίγραφα των ίδιων πόλεων και κανένα αντίγραφο

από άλλες πόλεις. Αυτό δεν σημαίνει ότι γίνεται κανόνας η μη χρήση της σεξουαλικής αναπαραγωγής. Απλά υπαγορεύει το περιεχόμενο των επιπρόσθετων λειτουργιών για την επιδιόρθωση τέτοιων λύσεων έτσι ώστε να μπορούν να προσαρμοστούν πριν υπολογιστούν. Η επιδιόρθωση μπορεί να είναι το να εντοπίσουμε κάθε πόλη η οποία παρουσιάζεται πάνω από μία φορά σε μία λύση και έπειτα να αντικαταστήσουμε αυτήν την πόλη με κάποια η οποία δεν εμφανίζεται καθόλου. Αφού γίνει αυτό για κάθε διπλή πόλη, ο απόγονος θα έχει επιδιορθωθεί και θα έχει γίνει μια ισχυρά βιώσιμη λύση. Πολλοί συντελεστές μεταβολής θα μπορούσαν να χρησιμοποιηθούν για το πρόβλημα του περιοδεύοντος πωλητή, αλλά για την απεικόνιση θα χρησιμοποιήσουμε εδώ τον συντελεστή αντιστροφής, όπως αυτός περιγράφηκε παραπάνω.

Το τρίτο βήμα είναι ο καθορισμός ενός κανόνα για το ποιες λύσεις θα επιλέξουμε ως βιώσιμες ώστε να γίνουν οι γονείς για την επόμενη παραγωγή. Όπως με την μεταβολή, έτσι κι εδώ μπορούμε να θεωρήσουμε αρκετά σχήματα για την επιλογή. Ένας απλός κανόνας είναι η επιβίωση του πιο κατάλληλου: μόνο το σύνολο των πολύ καλών λύσεων παραμένει, ενώ όλες οι άλλες εξολοθρεύονται. Ένας εναλλακτικός κανόνας είναι να χρησιμοποιήσουμε ένα είδος τουρνουά, όπου τυχαία ανά δύο οι λύσεις συναγωνίζονται για το ποια θα επιβιώσει. Ακριβώς όπως στα επαγγελματικά αθλήματα, όπου οι πιο αδύνατοι παίκτες ή ομάδες μερικές φορές νικούν διότι παίρνουν μια τυχερή ισοπαλία στο τουρνουά, έτσι κι εδώ μερικές πιο αδύνατες λύσεις σε έναν πληθυσμό επιβιώνουν για λίγες παραγωγές κάτω από αυτό το σχήμα. Αυτό μπορεί να είναι καλό σε πολύπλοκα προβλήματα, όπου μπορεί να είναι πιο εύκολο να βρεθούν καινούριες βέλτιστες λύσεις κάνοντας μεταβολές από αντίστοιχα πιο αδύνατες, παρά να γίνει το ίδιο μένοντας μόνο με τις πολύ καλές. Οι πιθανότητες αφθονούν, αλλά κάθε κανόνας που γενικά ευνοεί για επιβίωση τις πιο καλές λύσεις από τις πιο αδύνατες είναι σίγουρα πιο λογικός. Πιο απλά, μόνο η πιο βασική προσέγγιση -- επιβίωση των πιο κατάλληλων -- θα χρησιμοποιηθεί εδώ.



Σχήμα 4: Ένας απλός επαναληπτικός εξελικτικός αλγόριθμος χρησιμοποιήθηκε για να λύσει ένα πρόβλημα περιοδεύοντος πωλητή με 100 πόλεις. Στην αρχή, ο ερευνητής επέλεξε 100 πιθανές λύσεις, κάθε μία από τις οποίες χρησιμοποιήθηκε για να παράγει έναν απόγονο με αναστροφή [Σχήμα 3], καθορίζοντας ένα σύνολο από 200 λύσεις. Οι καλύτερες 100 λύσεις (με κριτήριο σύγκρισης την μικρότερη διαδρομή) επιλέχθηκαν, με την πρώτη λύση από αυτήν την πρώτη παραγωγή να φαίνεται πάνω αριστερά. Οι καλύτερες λύσεις μετά από 500, 1000 και 4000 επαναλήψεις, υποδεικνύουν την πρόοδο που υπήρχε καθώς η διαδικασία επαναλαμβανόταν.

Γέννηση

Το τελικό βήμα είναι η επιλογή του αρχικού πληθυσμού. Αν τίποτα δεν είναι γνωστό σχετικά με την λύση του προβλήματος, τότε οι λύσεις μπορούν αποκλειστικά να επιλεγούν τυχαία από το διάστημα όλων των πιθανών λύσεων. Στην περίπτωση του προβλήματος του περιοδεύοντος

πωλητή, αυτό σημαίνει την τυχαία παραγωγή ενός αριθμού από μεταθέσεις ακεραίων, όπου κάθε μετάθεση αναπαριστά μία πιθανή λύση.

Εναλλακτικά, μπορούν να υπάρχουν μερικοί υπαινιγμοί για διαθέσιμες καλές λύσεις -- ίσως επειδή κάποιος άλλος αλγόριθμος ή κάποια προηγούμενη γνώση μπορεί να χρησιμοποιηθεί για να παράγει κάποια εναρκτήρια πρωτοβουλία -- οι οποίες μπορούν να ενσωματωθούν στον αρχικό πληθυσμό. Αν αυτές οι λύσεις αποδειχτεί ότι αξίζουν, θα επιβιώσουν και θα παράγουν νέες μεταβλητές, ενώ αν είναι άστοχες για αρχή τότε θα χαθούν μαζί με τις άλλες τις πιο αδύνατες λύσεις. Για το παράδειγμα εδώ, υποθέστε ότι ο αρχικός πληθυσμός επιλέγεται αποκλειστικά τυχαία.

Μια τυπική εκτέλεση ενός εξελικτικού αλγορίθμου σε ένα πρόβλημα περιοδεύοντος πωλητή με 100 πόλεις, στο οποίο οι πόλεις έχουν κατανεμηθεί τυχαία αλλά με ένα ομοιόμορφο τρόπο, έχει τα αποτελέσματα που φαίνονται στο Σχήμα 4. Η βελτιστοποίηση σε επιτυχημένα στάδια στην εξέλιξη των διαδρομών είναι προφανής. (Το πρόγραμμα για αυτό το παράδειγμα, γραμμένο σε MATLAB, είναι διαθέσιμο στην διεύθυνση www.natural-selection.com).

Ο εξελικτικός αλγόριθμος έχει ψάξει το διάστημα των πιθανών διαδρομών και έχει ανακαλύψει μία πολύ καλή διαδρομή. Ενώ είναι ίσως πολύ πιθανό η λύση να μην είναι η τέλεια, είναι όμως αρκετά ποιοτική. Συνολικά, εξετάστηκαν μόνο 400.000 -- ή μία για κάθε 10^{150} -- πιθανές λύσεις, ένα απειροελάχιστο κλάσμα από το ολόκληρο. Αυτό είναι σύμφωνο με το λειτουργικό δόγμα των εξελικτικών επαγγελματιών: «Οι λύσεις θα πρέπει να είναι αρκετά καλές και να παράγονται αρκετά γρήγορα για να είναι χρήσιμες».

Παραδεχόμαστε ότι για την επίλυση του προβλήματος του περιοδεύοντος πωλητή υπάρχουν αλγόριθμοι πολύ καλύτεροι από την παραπάνω εξελικτική διαδικασία. Αυτό οφείλεται στο ότι οι μαθηματικοί έχουν ήδη ξοδέψει αρκετό χρόνο και προσπάθεια προσπαθώντας να το λύσουν. Αλλά όλες αυτές οι άλλες τεχνικές στηρίζονται σε κάποια συγκεκριμένη γνώση σχετικά με το πρόβλημα για να βελτιστοποιήσουν την εκτέλεσή τους. Θυσιάζουν την γενικότητα με σκοπό να πετύχουν την εκτέλεση. Όπως ακριβώς ο Deep Blue, το πρόγραμμα που έγινε παγκόσμιος πρωταθλητής στο σκάκι, ακτινοβολούν στο στενό πεδίο της εφαρμογής, αλλά δρουν πολύ απογοητευτικά έξω από αυτό το πεδίο. Φανταστείτε τον Deep Blue να παίζει ένα παιχνίδι σκάκι: δεν θα μπορούσε να κάνει καν την πρώτη κίνηση.

Το σημείο κλειδί είναι το ότι ενώ είναι πιθανό να συγχωνευτεί κάθε πρόβλημα με συγκεκριμένη διαθέσιμη γνώση και ως εκ τούτου να λάβουμε θετικά αποτελέσματα από αυτό, όταν χρησιμοποιούμε εξελικτικούς αλγορίθμους δεν είμαστε υποχρεωτικά αναγκασμένοι στην εφαρμογή αυτών των τεχνικών. Για αυτό και οι εξελικτικοί αλγόριθμοι μπορούν να αντιμετωπίσουν μια εξαιρετικά μεγάλη σειρά από θεμελιώδη προβλήματα.

Εφαρμογές

Οι εξελικτικοί αλγόριθμοι ήδη χρησιμοποιούνται για να λύσουν μια μεγάλη ποικιλία από λογικοφανή προβλήματα τα οποία οριοθετούν σημαντικές προκλήσεις. Ένα τέτοιο πρόβλημα περικλείει την ανακάλυψη νέων φαρμάκων. Στην περίπτωση της περικοπής ενός μικρού

μορίου από ένα πιθανό φάρμακο (που ονομάζεται ``ligand`` που σημαίνει: κάτι το οποίο πρέπει να περιοριστεί) σε ένα σημείο πρόσδεσης πρωτεϊνών, το αποτέλεσμα εξαρτάται σε μεγάλο μέρος από τα τρισδιάστατα σχήματα και των δύο στοιχείων. Είναι ανάλογο με ό,τι συμβαίνει με ένα κλειδί και μια κλειδαριά: όπως μόνο το δεξιόμορφο κλειδί θα ανοίξει μια συγκεκριμένη κλειδαριά έτσι και μόνο ένα κατάλληλα διαμορφωμένο ligand θα είναι ικανό να προσδεθεί στην πρωτεΐνη.

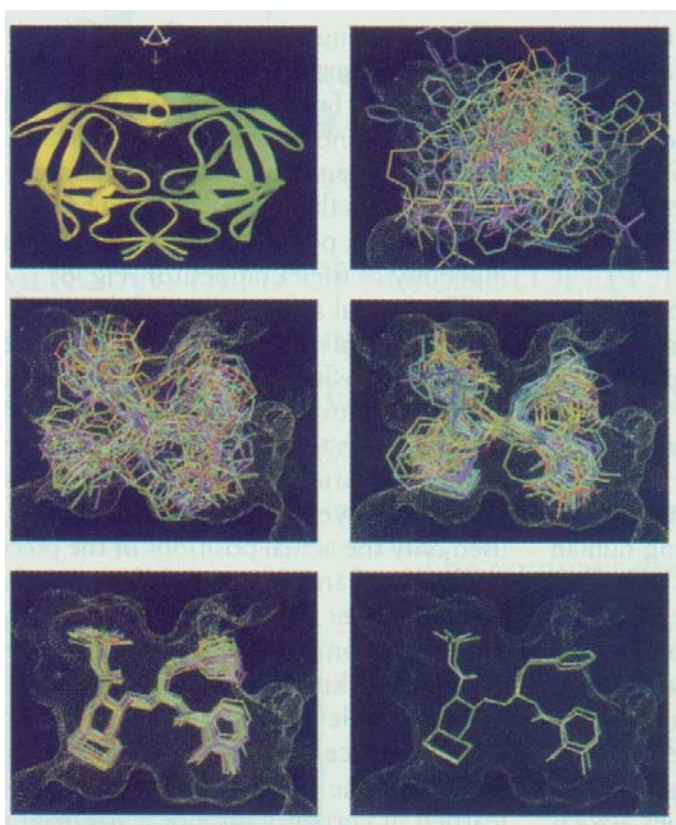
Έχει βρεθεί ότι το ligand είναι εξαιρετικά δύσκολο να επιτευχθεί. Όχι μόνο υπάρχουν δεκάδες χιλιάδες υποψήφια, αλλά καθένα από αυτά έχει πολλούς δεσμούς οι οποίοι μπορούν να περιστραφούν σε πολλές διαφορετικές θέσεις. Βασικά κάθε ligand έχει αναμφίβολα πολλά σχήματα. Είναι σίγουρα αλήθεια ότι πολλές από αυτές τις πιθανές δομές είναι υπερβολικά απίθανο να ανήκουν στις φυσικές και χημικές ιδιότητες των μορίων. Ωστόσο όμως, ο αριθμός των πιθανών διατομών τις οποίες μπορεί να παρουσιάσει ένα ligand είναι συχνά τεράστιος. Αυτό κάνει την πρόβλεψη της καλύτερης δομής ενός ligand όπως αυτό προσδέεται σε μία πρωτεΐνη αρκετά δύσκολη.

Σε μία μελέτη που δημοσιεύτηκε στο *Chemistry & Biology*, ερευνητές στο Argouron Pharmaceuticals Inc. υποδεικνύουν πως ένας εξελικτικός αλγόριθμος θα μπορούσε να χρησιμοποιηθεί για την πρόβλεψη του τρόπου με τον οποίο ένα ligand θα εισερχόταν στον ιό HIV-1 σαν ένα δυνατό φάρμακο κατά του AIDS. Το θέμα ήταν να βρουν την πιο ευνοϊκή δομή για το ligand όταν αυτό θα προσέγγιζε στο σημείο πρόσδεσης της πρωτεΐνης. Διάφοροι παράγοντες πιστεύεται ότι είναι σημαντικοί όπως:

- Η τριδιάστατη διάταξη του ligand που βασίζεται στο πόσο καλά συμπληρώνει την δομή της επιφάνειας ενός σημείου πρόσδεσης στην πρωτεΐνη.
- Ηλεκτροστατικές επιδράσεις μεταξύ των ατόμων.
- Οι δυνάμεις Van der Waal's οι οποίες υπαγορεύουν βασικά ότι δύο άτομα δεν μπορούν να βρίσκονται συγχρόνως σε ένα και ίδιο σημείο.

Στο πρίσμα αυτών των κριτηρίων, κάθε πιθανή δομή ενός ligand μπορεί να επιτευχθεί ανάλογα με το πόσο καλά προσδέεται στην πρωτεΐνη. Ένας εξελικτικός αλγόριθμος επινοήθηκε για να λειτουργήσει σε έναν πληθυσμό με πάνω από 1000 διαφορετικές δυνατές δομές ενός υποψήφιου ligand, με τυχαία μεταβολή των γωνιών περιστροφής του ligand και επιλογή από τους σχηματισμούς που εμφανίζονται αυτών που είναι πιο ευνοϊκοί από άλλους. Το Σχήμα 5 δείχνει μια επιτυχημένη εξέλιξη του ligand που ονομάζεται AG-1343 και χρησιμοποιείται για τον ιό HIV-1 μετά την ολοκλήρωση του εξελικτικού αλγόριθμου και μετά την μετα-διαδικασία της βαθμιαίας ελαχιστοποίησης σε κάθε γωνία περιστροφής. Η σύγκριση αυτού του αποτελέσματος με την πειραματικά καθορισμένη δομή του κρυστάλλου υποδεικνύει ότι υπάρχει μια μικρή ασυμφωνία μεταξύ της προβλέψιμης δομής που παράγεται από τον εξελικτικό αλγόριθμο και της πραγματικής παρατήρησης που σταχυολογείται από το περιβάλλον. Περιληπτικά, τα πειράματα δίνουν ανεξάρτητη επαλήθευση την οποία ο εξελικτικός αλγόριθμος έκανε βρίσκοντας πραγματικά την σωστή απάντηση. Αυτή η εφαρμογή είναι σημαντική διότι εξοικονομεί χρόνο και εργασία, τα οποία και τα δύο τυπικά δεν βρίσκονται και τόσο εύκολα. Όταν μια φαρμακευτική εταιρεία στοχεύει σε μια πρωτεΐνη, υπάρχουν πολλά περισσότερα πιθανά ligands που θα πρέπει να υποθέσει ότι θα μπορούσαν να εξεταστούν με απαρίθμηση. Κάθε πιθανό ligand μπορεί να έχει διάφορους περιστρεφόμενους

δεσμούς, προσφέροντας ένα πολυδιάστατο πρόβλημα βελτιστοποίησης για την εύρεση της καλύτερης δομής από την μία άκρη ως την άλλη. Από το να προσπαθήσει κανείς να ερευνήσει την κάθε περιστροφή ανεξάρτητα, ένας εξελικτικός αλγόριθμος μπορεί γρήγορα να ανακαλύψει ποια ligands ταιριάζουν καλύτερα στο σημείο πρόσδεσης με την πρωτεΐνη και επομένως έχουν την δυνατότητα να είναι υποψήφια φάρμακα. Οι χημικοί μπορούν κατόπιν να επικεντρώσουν την προσοχή τους σε αυτούς τους υποψήφιους. Το αποτέλεσμα είναι μια βελτιωμένη και αποτελεσματική σε κόστος διαδικασία για την προστασία φαρμακευτικών μόλυβδων.



Σχήμα 5: Για να εξετάσουμε την ικανότητά του στον σχεδιασμό φαρμάκων, ένας επαναστατικός αλγόριθμος χρησιμοποιήθηκε για να επιλέξει μόρια ή ligands, τα οποία θα εισερχόταν στο σημείο υποδοχής μιας πρωτεΐνης HIV-1 (αυτό φαίνεται, πάνω αριστερά ως ένα διάγραμμα ταινίας ενός βιοχημικού). Οι καλύτερες 150 λύσεις από τις 1000 πιθανές λύσεις στην δεύτερη παραγωγή φαίνονται ανακατεμένες (πάνω δεξιά). Ο αλγόριθμος όμως φαίνεται να έχει στην 70ή (στην μέση αριστερά), στην 92^η (στην μέση δεξιά) και στην 149^η (κάτω αριστερά) παραγωγή.

Η καλύτερη δομή που παράχθηκε (κάτω δεξιά, με πράσινο χρώμα) ταίριαζε πάρα πολύ με εκείνη που βρέθηκε πειραματικά (λευκή), υποδεικνύοντας την χρησιμότητα των εξελικτικών τεχνικών.

Άλλες πραγματικές εφαρμογές των εξελικτικών αλγορίθμων πραγματοποιούνται στον προγραμματισμό του ελέγχου της αλυσίδας προμηθειών και της ιατρικής διάγνωσης. Μπορεί ακόμα να χρησιμοποιηθούν ως βάση για προσομοιώσεις μαχών για εκπαίδευση στρατιωτικού

προσωπικού. Ένας εξελικτικός αλγόριθμος μπορεί να συμπεριφερθεί σαν αληθινός εχθρός ο οποίος μαθαίνει και προσαρμόζεται στις τακτικές του καθώς αλλάζει η στρατηγική κατάσταση. Ακόμα, η εξέλιξη χρησιμοποιείται ως μία βασική αρχή για την δημιουργία νέων σχεδιασμών υλικού (hardware) που περιλαμβάνουν συγχρόνως και τις φυσικές μηχανές αλλά και τα ηλεκτρονικά κυκλώματα.

Στο μακρινό μέλλον, οι εξελικτικοί αλγόριθμοι μπορεί να παίξουν έναν κεντρικό ρόλο στην επινοήση αληθινά έξυπνων μηχανών--υπολογιστών που θα μπορούν να μάθουν μόνοι τους. Η αληθινή πρόκληση δεν είναι τόσο το να φτιάξουμε έναν υπολογιστή που να μπορεί να συναγωνίζεται τους ανθρώπους. Σε πολλές περιορισμένες εφαρμογές όπως τα παιχνίδια, αυτό κατορθώνεται εύκολα μεταφράζοντας την ανθρώπινη ειδικότητα σε προγραμματιστικούς κανόνες συμπεριφοράς. Η αληθινή πρόκληση είναι να δώσουμε στον υπολογιστή έναν τρόπο να μάθει πώς να συμπεριφέρεται όπως ένας ειδικός χωρίς να στηρίζεται στον άνθρωπο.

Ένα πρόσφατο βήμα προς αυτήν την κατεύθυνση είναι ένας εξελικτικός αλγόριθμος ο οποίος έμαθε να παίζει ντάμα σε ένα πολύ υψηλό σημείο ανταγωνισμού χωρίς να στηρίζεται σε καμία τεχνική που να συνδέεται με την ανθρώπινη εμπειρία. Ο αλγόριθμος χρησιμοποίησε έναν πληθυσμό από τεχνητά νευρωνικά δίκτυα (ανταγωνιστικά μοντέλα που βασίζονται χαλαρά στην αρχιτεκτονική ενός εγκεφάλου) για να μετρήσει την ποιότητα κάθε θέσης του πίνακα του παιχνιδιού η οποία θα μπορούσε να υπολογιστεί στο παιχνίδι. Κάθε νευρωνικό δίκτυο ήταν στην ουσία μια στρατηγική για να παίξει κάποιος, διότι παρήγαγε ένα αποτέλεσμα που υποδείκνυε ποιές θέσεις ήταν πιο ευνοϊκές από κάποιες άλλες. Ο εξελικτικός αλγόριθμος έβαλε σε συναγωνισμό αυτά τα νευρωνικά δίκτυα (το ένα με το άλλο) ξεκινώντας από μία απόλυτα τυχαία αρχική κατάσταση και εφαρμόζοντας την μεταβολή και την επιλογή επαναληπτικά.

Ύστερα από μόνο 10 παραγωγές, το καλύτερο δίκτυο στο πληθυσμό ήταν ικανό να νικήσει τους αντιπάλους του εύκολα. Μετά από 100 παραγωγές, το καλύτερο αναπτυσσόμενο δίκτυο έπαιξε εναντίον ανθρώπων στη θέση www.zone.com (Microsoft Network Gaming Zone Web Site) βασιζόμενο στον αριθμό των παιχνιδιών που παίχτηκαν και στην αποτελεσματικότητα του δικτύου. Τελικά, κέρδισε ένα χαρακτηρισμό ``κλάσης Α'' -- ένα επίπεδο κάτω από τον χαρακτηρισμό του 'expert' σύμφωνα με το σύστημα της Ομοσπονδίας Σκακιού των ΗΠΑ (United States Chess Federation). Φυσικά, οι αντίπαλοι δεν ήξεραν ότι έπαιζαν εναντίον ενός προγράμματος και κανένας δεν ήξερε ότι ο αντίπαλός τους ήταν ένα πρόγραμμα. Στην πραγματικότητα, κάποιοι άνθρωποι επαίνεσαν την προφανή ιδιοφυΐα του αντιπάλου τους (Σχήμα 6).



Σχήμα 6: Σε ένα πρόσφατο παιχνίδι ντάμας που παίχτηκε on-line (στο Internet) οι κινήσεις του Davids1101 καθορίζονταν μυστικά από ένα αναπτυσσόμενο νευρωνικό δίκτυο ενώ ο Decepticon12345 ήταν ένας άνθρωπος (συναγωνιστής) επιπέδου expert.

Στο σημείο του παιχνιδιού που φαίνεται στο σχήμα, το νευρωνικό δίκτυο που παίζει με τα λευκά πιόνια, έχει μόλις κερδίσει έναν βασιλιά και ο άνθρωπος επιπέδου expert έχει αναγνωρίσει την κίνηση γράφοντας 'gm' που σημαίνει 'καλή κίνηση (good move)'.

Το κρίσιμο κατόρθωμα αυτής της εργασίας είναι ότι τα νευρωνικά δίκτυα δεν υπολόγισαν την θέση τους χρησιμοποιώντας την ανθρώπινη πρακτική -- κινητικότητα κομματιών, έλεγχο του κέντρου του πίνακα, ύπαρξη μονοπατιού που να μπορεί να σώσει τον βασιλιά ή οποιαδήποτε άλλη εμπειρική τεχνική που θα χρησιμοποιούσε ένας άνθρωπος-παίκτης. Χρησιμοποίησαν μόνο τις πραγματικές θέσεις των κομματιών στον πίνακα και το διαφορεικό κομμάτι, το οποίο είναι ο αριθμός των κομματιών εμπρός ή πίσω. Ο εξελικτικός αλγόριθμος συνεπαίρανε ο,τιδήποτε χρειαζόταν να γνωρίζει σχετικά με το πώς θα παίζει σε επίπεδο κοντά στο τέλειο απλά παίζοντας το παιχνίδι και λαμβάνοντας ανάδραση στην τελική έξοδο κάθε παιχνιδιού. Ο,τιδήποτε έμαθε, οπωσδήποτε δεν το έμαθε από τους δημιουργούς του οι οποίοι, όπως σημειώθηκε παραπάνω, γνωρίζουν πολύ λίγα σχετικά με το πώς να παίζουν πολύ καλή ντάμα. Τα πειράματα συνεχίζονται σε μια προσπάθεια με σκοπό να επιτευχθεί ένα επίπεδο τελειότητας που θα θέσει την εξελισσόμενη στρατηγική στην κορυφή 1% όλων των παιχτών οι οποίοι έχουν εγγραφεί στο site του παιχνιδιού στο Internet.

Η ικανότητα ενός υπολογιστή του να πετύχει επιδεξιότητα σε ένα παιχνίδι τέτοιου επιπέδου ανάλογη του ανθρώπου απλά μαθαίνοντας μόνος του ανοίγει για αυτόν πολύ μεγαλύτερες δυνατότητες: την ικανότητα των υπολογιστών να μαθαίνουν νέες πλευρές του αληθινού κόσμου χωρίς να στηρίζονται στους ανθρώπους για προγραμματισμό με όλη την απαιτούμενη γνώση. Καθώς το κόστος του προγραμματισμού μειώνεται, η υπευθυνότητα των εξελικτικών

αλγορίθμων στην επίλυση λογικοφανών προβλημάτων θα αναδυθεί. Αυτό οφείλεται στην παράλληλη φύση των εξελικτικών προσεγγίσεων. Με τη φυσική εξέλιξη, τα άτομα πάντοτε εκτιμούνται παράλληλα με το περιβάλλον τους. Έτσι, στηριζόμενος σε μία ομάδα από υπολογιστές (ονομάζεται και 'σωρός από PCs'), ένας επαγγελματίας μπορεί να ξοδέψει λιγότερο χρόνο για να λύσει δύσκολα προβλήματα ή να εκτιμήσει ανεξάρτητες λύσεις παράλληλα παρά σε σειρά. Ο πολύ μεγάλος πληθυσμός που επιτρέπει αυτό το πρωτόκολλο παράγει καλύτερες λύσεις γρηγορότερα από ό,τι κάνουν μικρότεροι πληθυσμοί.

Τα προϊόντα τα οποία παράγονται από αυτά τα μαζικά σε όγκο παράλληλα σχέδια παραμένουν ως ένας τρόπος συλλογισμού. Κάποιοι ήδη πρόβλεψαν την χρήση των εξελικτικών αλγορίθμων στον σχεδιασμό τεχνητών εγκεφάλων που θα είναι αντικαταστάτες και θα παραγκωνίσουν την ανθρώπινη γνώση, όπως έκανε ο Ray Kurzweil στο *The Age of Spiritual Computing* (Viking, 1999). Άσχετα του αν γίνει ή όχι αυτή η πρόβλεψη πραγματικότητα, δεν υπάρχει αμφιβολία ότι οι εξελικτικοί αλγόριθμοι θα γίνουν ένα κύριο στήριγμα στην επίλυση προβλημάτων στα χρόνια που θα έρθουν.

23. Το προγραμματιστικό περιβάλλον MATLAB

Τι είναι το MATLAB;

Το MATLAB είναι ένα υπολογιστικό περιβάλλον για αριθμητικούς υπολογισμούς και οπτικοποίηση. Το MATLAB παρέχει αριθμητική ανάλυση, υπολογισμούς με πίνακες, επεξεργασία σήματος και γραφικά σε ένα εύχρηστο περιβάλλον στο οποίο προβλήματα και λύσεις εκφράζονται όπως ακριβώς διατυπώνονται με μαθηματικό τρόπο, χωρίς δηλαδή παραδοσιακό προγραμματισμό.

MATLAB σημαίνει MATrix LABoratory (Εργαστήριο Πινάκων) και αρχικά αναπτύχθηκε για να παρέχει εύκολη πρόσβαση σε λογισμικό πινάκων που είχε αναπτυχθεί στα πλαίσια των προγραμμάτων LINPACK και EISPACK.

Το MATLAB είναι ένα αλληλεπιδραστικό σύστημα στο οποίο το βασικό στοιχείο δεδομένων είναι ένας πίνακας που δεν απαιτεί προσδιορισμό των διαστάσεών του. Αυτό επιτρέπει την επίλυση αριθμητικών προβλημάτων σε ένα κλάσμα του χρόνου που θα απαιτούσε ο προγραμματισμός σε μια γλώσσα όπως η FORTRAN, η BASIC, η C κλπ.

Το MATLAB συμπληρώνεται από ένα σύνολο πακέτων εφαρμογών που ονομάζονται *εργαλειοθήκες (toolboxes)*. Οι εργαλειοθήκες είναι πλήρεις συλλογές από συναρτήσεις MATLAB που επεκτείνουν το περιβάλλον του MATLAB έτσι ώστε να επιλύονται συγκεκριμένες κλάσεις προβλημάτων. Περιοχές για τις οποίες διατίθενται εργαλειοθήκες περιλαμβάνουν την επεξεργασία σήματος, το σχεδιασμό συστημάτων ελέγχου, την προσομοίωση δυναμικών συστημάτων, την αναγνώριση συστημάτων, τα νευρωνικά δίκτυα, κλπ.

Το πιο σημαντικό χαρακτηριστικό του MATLAB είναι ίσως η επεκτασιμότητά του. Αυτό επιτρέπει στον κάθε χρήστη να συνεισφέρει τις δικές του εφαρμογές. Ως αποτέλεσμα, επιστήμονες, μαθηματικοί και μηχανικοί έχουν συνεισφέρει νέες και ενδιαφέρουσες εφαρμογές χωρίς να γράψουν ούτε μια γραμμή κώδικα σε γλώσσα χαμηλού επιπέδου.

Το σύστημα του MATLAB

Το σύστημα του MATLAB αποτελείται από πέντε κύρια μέρη:

1. *Τη γλώσσα MATLAB:* Αυτή είναι μία υψηλού επιπέδου γλώσσα μήτρας/πίνακα με έλεγχο ροής δηλώσεων, συναρτήσεων, δομών δεδομένων, εισόδου/εξόδου, και των αντικειμενοστρεφών (*object-oriented*) χαρακτηριστικών προγράμματος. Επιτρέπει τόσο τη δημιουργία γρήγορων και “βρώμικων” προγραμμάτων (“*programming in the small*”), όσο και τη δημιουργία ολοκληρωμένων μεγάλων και πολύπλοκων προγραμματιστικών εφαρμογών (“*programming in the large*”).

2. *Το λειτουργικό περιβάλλον του MATLAB:* Αυτό είναι το σύνολο των εργαλείων και των διευκολύνσεων με το οποίο δουλεύει κανείς με το MATLAB ως χρήστης ή ως προγραμματιστής. Περιλαμβάνει διευκολύνσεις για τον έλεγχο των μεταβλητών στο χώρο εργασίας και την είσοδο και έξοδο δεδομένων. Επίσης, περιλαμβάνει εργαλεία για την εφαρμογή, έλεγχο, διόρθωση λαθών, και κατανομή των Μ-αρχείων.

3. *Το σύστημα γραφικών του MATLAB:* Αυτό είναι το γραφικό σύστημα του MATLAB. Περιλαμβάνει υψηλού επιπέδου εντολές για διδιάστατη και τριδιάστατη απεικόνιση δεδομένων, επεξεργασία εικόνας, δημιουργία κινουμένων σχεδίων (animation), και παρουσίαση γραφικών. Επίσης, περιλαμβάνει χαμηλού επιπέδου εντολές που επιτρέπουν την κατασκευή και εμφάνιση γραφικών, όπως την κατασκευή πλήρους διεπαφής προς το χρήστη (user interface).

4. *Τη βιβλιοθήκη μαθηματικών συναρτήσεων του MATLAB:* Αυτή είναι μια απέραντη συλλογή από υπολογιστικούς αλγόριθμους που ξεκινά από στοιχειώδεις συναρτήσεις όπως άθροισμα, ημίτονο, συνημίτονο, και πολύπλοκη αριθμητική, και φτάνει μέχρι πιο πολύπλοκες συναρτήσεις όπως μήτρες, αντίστροφες μητρώων, ιδιοτιμές μήτρας, συναρτήσεις Bessel, και μετασχηματισμούς Fourier.

5. *Τη διεπαφή προς προγράμματα εφαρμογών (Application Program Interface, API) του MATLAB:* Αυτή είναι μία βιβλιοθήκη που σου επιτρέπει να γράψεις προγράμματα σε C, Fortran κλπ. τα οποία αλληλεπιδρούν με το MATLAB. Επίσης, περιέχει διευκολύνσεις για το κάλεσμα ρουτινών της MATLAB (δυναμική συνάρμοση), κάλεσμα του MATLAB σαν μία υπολογιστική μηχανή, και για το διάβασμα και γράψιμο MAT-αρχείων.

Το Simulink: Το Simulink, ένα συνοδευτικό πρόγραμμα του MATLAB, είναι ένα σύστημα που αλληλεπιδρά με προσομοιωμένα μη γραμμικά δυναμικά συστήματα. Είναι ένα γραφικό πρόγραμμα – χρησιμοποιεί ποντίκι – που επιτρέπει τη μοντελοποίηση ενός συστήματος σχεδιάζοντας ένα ιστόγραμμα στην οθόνη και μεταβάλλοντας το δυναμικά. Μπορεί να δουλέψει με γραμμικά, μη γραμμικά, συνεχούς χρόνου, διακριτού χρόνου, πολυμεταβλητά και πολυρυθμικά (multirate) συστήματα.

Ξεκινώντας με το MATLAB

Για να εκκινήσετε το MATLAB σε ένα PC, κάνετε διπλό κλικ στο εικονίδιο του MATLAB. Για να εκκινήσετε το MATLAB σε ένα σύστημα με UNIX, γράψτε matlab στην προτροπή του λειτουργικού συστήματος. Για να κλείσετε το MATLAB οποιαδήποτε στιγμή, γράψτε quit στην προτροπή του MATLAB.

Εάν νιώθετε ότι χρειάζεστε περισσότερη βοήθεια, γράψτε help στην προτροπή του MATLAB ή ανοίξτε το μενού HELP σε ένα PC. Θα μιλήσουμε περισσότερο για την βοήθεια και των κειμένων εντός γραμμής (on line) αργότερα.

Μήτρες και μαγικά τετράγωνα

Ο καλύτερος τρόπος για να ξεκινήσετε με το MATLAB είναι να μάθετε πώς να χειρίζεστε πίνακες. Αυτή η ενότητα θα σας δείξει πώς να το κάνετε αυτό. Στο MATLAB, ένας πίνακας είναι ένα τετραγωνικό διάνυσμα αριθμών. Μερικές φορές δίνουμε ειδική σημασία στους 1 επί 1 πίνακες, που είναι βαθμωτά μεγέθη, και στους πίνακες γραμμής ή στήλης, που είναι διανύσματα. Το MATLAB έχει και άλλους τρόπους να αποθηκεύει τόσο αριθμητικά όσο και μη αριθμητικά δεδομένα, αλλά αρχικά είναι καλύτερα να σκεπτόμαστε τα πάντα ως πίνακες. Οι λειτουργίες του είναι σχεδιασμένες να είναι όσο τον δυνατόν πιο φυσικές. Εκεί που οι άλλες γλώσσες προγραμματισμού δουλεύουν με ένα αριθμό τη φορά, το MATLAB σας επιτρέπει να δουλεύετε με ολόκληρους πίνακες γρήγορα και εύκολα.

Μπορείτε να εισάγετε πίνακες στο MATLAB με διάφορους τρόπους.

1. Εισάγοντας μία λίστα στοιχείων.
2. Φορτώνοντας πίνακες από εξωτερικά αρχεία δεδομένων.
3. Παράγοντας πίνακες χρησιμοποιώντας συναρτήσεις του MATLAB.
4. Δημιουργώντας πίνακες με δικές σας συναρτήσεις σε M-αρχεία.

Αρχίστε εισάγοντας τον πίνακα του Dürer ως λίστα των στοιχείων του. Αρκεί απλά να ακολουθήσετε μερικούς βασικούς κανόνες:

1. Χωρίστε τα στοιχεία κάθε γραμμής με κενά ή κόμμα.
2. Χρησιμοποιήστε το ελληνικό ερωτηματικό, ; , για να δηλώσετε το τέλος κάθε γραμμής.
3. Τοποθετήστε όλη την λίστα με αγκύλες, [].

Για να εισάγετε τον πίνακα του Dürer, απλά γράψτε:

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

Το MATLAB προβάλλει τη μήτρα που μόλις εισάγατε,

```
A =  
16  3  2 13  
 5 10 11  8  
 9  6  7 12  
 4 15 14  1
```

sum, transpose, and diag

Πιθανόν να γνωρίζετε ήδη ότι οι ξεχωριστές ιδιότητες ενός μαγικού τετραγώνου έχουν σχέση με τους διάφορους τρόπους με τους οποίους αθροίζουμε τα στοιχεία του. Το άθροισμα

οποιαδήποτε γραμμής ή στήλης, καθώς και των δύο κύριων διαγωνίων του, είναι πάντα ο ίδιος αριθμός. Ας το εξακριβώσουμε αυτό χρησιμοποιώντας το MATLAB. Η πρώτη συνάρτηση που θα δοκιμάσουμε είναι

```
sum(A)
```

Το MATLAB απαντάει με το

```
ans =  
    34    34    34    34
```

Όταν δεν καθορίζετε μια μεταβλητή εξόδου, το MATLAB χρησιμοποιεί την μεταβλητή ans, συντομογραφία του answer, για να αποθηκεύσει το αποτέλεσμα ενός υπολογισμού. Έχετε υπολογίσει ένα διάνυσμα που περιέχει τα αθροίσματα των στηλών του A. Φυσικά, κάθε στήλη έχει το ίδιο άθροισμα, τον μαγικό άθροισμα, 34.

Και τα αθροίσματα των γραμμών; Το MATLAB προτιμάει να δουλεύει με τις στήλες ενός πίνακα, έτσι ο ευκολότερος τρόπος για να πάρετε τα αθροίσματα των γραμμών είναι να αντιμεταθέσετε τις γραμμές του πίνακα με τις στήλες, να υπολογίσετε τα αθροίσματα των νέων στηλών και μετά να αντιμεταθέσετε το αποτέλεσμα. Η πράξη της αντιμετάθεσης δηλώνεται με μία απόστροφο, '. Περιστρέφει ένα πίνακα γύρω από την κύρια διαγώνιο του και μετατρέπει ένα πίνακα γραμμής σε ένα πίνακα στήλης. Έτσι το

```
A'
```

παράγει

```
ans =  
    16     5     9     4  
     3    10     6    15  
     2    11     7    14  
    13     8    12     1
```

Και το

```
sum(A)'
```

παράγει ένα πίνακα στήλη ο οποίος περιέχει τα αθροίσματα των γραμμών

```
ans =  
    34  
    34  
    34  
    34
```

Εύκολα βρίσκουμε το άθροισμα των στοιχείων της κύριας διαγωνίου με τη βοήθεια της συνάρτησης `diag` η οποία τα ξεχωρίζει από τον πίνακα. Έτσι το

```
diag(A)
```

παράγει

```
ans =  
    16  
    10  
     7  
     1
```

και

```
sum(diag(A))
```

παράγει

```
ans =  
    34.
```

Η άλλη διαγώνιος, η αντιδιαγώνιος, δεν είναι πολύ σημαντική μαθηματικά και έτσι το MATLAB δεν έχει μία έτοιμη συνάρτηση για αυτή. Υπάρχει όμως μία συνάρτηση, αρχικά προοριζόμενη για χρήση στα γραφικά, η `fliplr`, η οποία «περιστρέφει» ένα πίνακα γύρω από ένα κατακόρυφο άξονα.

```
sum(diag(fliplr(A)))
```

```
ans =  
    34
```

Έχετε δείξει ότι ο πίνακας στη ξυλογραφία του Dürer είναι πράγματι ένα μαγικό τετράγωνο και, στην πορεία, δοκιμάσατε μερικές από τις συναρτήσεις πινάκων που σας προσφέρει το MATLAB. Το επόμενο τμήμα συνεχίζει να χρησιμοποιεί τους πίνακες για να δείξει μερικές ακόμα δυνατότητες του MATLAB.

Subscripts

Μπορούμε να αναφερθούμε στο στοιχείο γραμμής j και στήλης j με το $A(i,j)$. Για παράδειγμα, το $A(4,2)$ είναι ο αριθμός στην τέταρτη γραμμή και στην δεύτερη στήλη. Στο μαγικό μας τετράγωνο ο $A(4,2)$ είναι το 15. Έτσι είναι δυνατό να υπολογίσουμε το άθροισμα της τέταρτης στήλης γράφοντας

$$A(1,4) + A(2,4) + A(3,4) + A(4,4).$$

Αυτό μας δίνει

ans =
34

αλλά δεν είναι ο πιο κομψός τρόπος για να βρούμε το άθροισμα μίας στήλης.

Είναι επίσης δυνατό να αναφερθούμε στα στοιχεία ενός πίνακα με ένα μονό δείκτη, το $A(k)$. Αυτός είναι ο συνηθής τρόπος με τον οποίο αναφερόμαστε σε πίνακες γραμμής και στήλης. Μπορούμε όμως να τον εφαρμόσουμε και σε διδιάστατους πίνακες, όπου θεωρούμε ένα διάνυσμα σαν ένα πίνακα στήλης με στοιχεία τις γραμμές του αρχικού πίνακα. Έτσι, στο μαγικό μας τετράγωνο, το $A(8)$ είναι άλλος ένας τρόπος με τον οποίο μπορούμε να αναφερθούμε στην τιμή 15 που είναι αποθηκευμένη στο $A(4,2)$.

Εάν προσπαθήσετε να χρησιμοποιήσετε την τιμή ενός στοιχείου εκτός της μήτρας, αυτό είναι λάθος. Πράγματι το

$t = A(4,5)$

υπερβαίνει τις διαστάσεις του πίνακα.

Από την άλλη πλευρά, εάν αποθηκεύσετε μία τιμή σε ένα στοιχείο εκτός των διαστάσεων του πίνακα αυτός θα διευρυνθεί για να συμπεριλάβει το νέο στοιχείο.

$X = A;$
 $X(4,5) = 17$

$X =$
16 3 2 13 0
5 10 11 8 0
9 6 7 12 0
4 15 14 1 17

Ο τελεστής άνω και κάτω τελεία

Η άνω και κάτω τελεία, $:$, είναι ένας από τους πιο σημαντικούς τελεστές στο MATLAB. Την συναντάμε σε πολλές διαφορετικές περιπτώσεις. Η έκφραση

$1:10$

είναι ένας πίνακας γραμμής που περιέχει τους ακέραιους από 1 ως 10

1 2 3 4 5 6 7 8 9 10

Μπορείτε επίσης να καθορίσετε το διάστημα στο οποίο θα διαφέρουν οι αριθμοί. Για παράδειγμα το

```
100:-7:50
```

δίνει

```
100 93 86 79 72 65 58 51
```

και το

```
0:pi/4:pi
```

μας δίνει

```
0 0.7854 1.5708 2.3562 3.1416
```

Οι εκφράσεις δεικτών που χρησιμοποιούν άνω και κάτω τελεία αναφέρονται σε τμήματα ενός πίνακα. Το

```
A( 1 : k, j )
```

Είναι τα πρώτα k στοιχεία της στήλης j του πίνακα A . Έτσι το

```
sum(A(1:4,4))
```

υπολογίζει το άθροισμα της τέταρτης στήλης. Υπάρχει όμως και ένας καλύτερος τρόπος. Η άνω και κάτω τελεία μόνη της αναφέρεται στα στοιχεία μίας γραμμής ή στήλης ενός πίνακα και η λέξη-κλειδί `end` αναφέρεται στην τελευταία γραμμή ή στήλη. Έτσι το

```
sum(A(:,end))
```

υπολογίζει το άθροισμα των στοιχείων της τελευταίας στήλης του A .

```
ans =  
    34
```

Γιατί το μαγικό άθροισμα ενός 4 επί 4 πίνακα είναι 34; Εάν χωρίσουμε τους ακέραιους από το 1 ως το 16 σε τέσσερις ομάδες με ίσο άθροισμα αυτό το άθροισμα θα είναι

```
sum(1:16)/4
```

το οποίο, φυσικά, είναι

```
ans =  
    34
```


Η Συνάρτηση *magic*

Το MATLAB έχει μία ενσωματωμένη συνάρτηση η οποία δημιουργεί μαγικά τετράγωνα σχεδόν κάθε μεγέθους. Αυτή η συνάρτηση ονομάζεται – πώς αλλιώς – *magic*.

```
B = magic(4)
B =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

Ο πίνακας αυτός είναι σχεδόν ο ίδιος με εκείνον στη ξυλογραφία του Dürer και έχει όλες τις «μαγικές» ιδιότητες: η μόνη τους διαφορά τους είναι ότι οι δύο μεσαίες στήλες είναι η μία στην θέση της άλλης. Για να μετατρέψουμε τον B στον πίνακα του Dürer A, ανταλλάσσουμε τις δύο μεσαίες στήλες.

```
A = B(:,[1 3 2 4])
```

Αυτό σημαίνει «για κάθε μία από τις γραμμές του πίνακα B, αναδιάταξε τα στοιχεία με την σειρά 1,3,2,4». Έτσι παίρνουμε

```
A =
    16     3     2    13
     5    10    11     8
     9     6     7    12
     4    15    14     1
```

Γιατί ο Dürer να έμπαινε στον κόπο να αναδιατάξει τις στήλες όταν θα μπορούσε να χρησιμοποιήσει τις υπηρεσίες του MATLAB; Χωρίς αμφιβολία ήθελε να συμπεριλάβει την ημερομηνία της ξυλογραφίας, 1514, στον πάτο του μαγικού τετραγώνου.

Εκφράσεις

Όπως και οι περισσότερες προγραμματιστικές γλώσσες, το MATLAB παρέχει μαθηματικές εκφράσεις, αλλά αντίθετα από τις γλώσσες αυτές, οι εκφράσεις περιλαμβάνουν ολόκληρους πίνακες. Τα δομικά στοιχεία των εκφράσεων είναι :

- Μεταβλητές
- Αριθμοί
- Τελεστές
- Συναρτήσεις

Μεταβλητές

Το MATLAB δεν χρησιμοποιεί δηλώσεις κανενός είδους ή δηλώσεις διαστάσεων. Όταν το MATLAB συναντάει ένα καινούργιο όνομα μεταβλητής, αυτόματα δημιουργεί την μεταβλητή και δεσμεύει τον απαραίτητο χώρο στην μνήμη. Εάν η μεταβλητή ήδη υπάρχει τότε αλλάζει τα περιεχόμενα της και, αν χρειαστεί, δεσμεύει νέο χώρο στην μνήμη. Για παράδειγμα το

```
num_students = 25
```

δημιουργεί μία 1-επί-1 μήτρα με το όνομα num_students και τοποθετεί την τιμή 25 στο μοναδικό στοιχείο της.

Τα ονόματα των μεταβλητών αποτελούνται από ένα γράμμα, ακολουθούμενο από οποιοδήποτε αριθμό γραμμάτων, ψηφίων και κάτω παυλών. Το MATLAB χρησιμοποιεί μόνο τους 31 πρώτους χαρακτήρες του ονόματος μίας μεταβλητής. Το MATLAB ξεχωρίζει τα μικρά γράμματα από τα κεφαλαία. Το A και το a δεν είναι το ίδιο. Για να δείτε τον πίνακα που έχει ανατεθεί σε μία μεταβλητή, απλά τυπώστε το όνομα της μεταβλητής.

Αριθμοί

Το MATLAB χρησιμοποιεί τον συμβατικό τρόπο δήλωσης δεκαδικών αριθμών, με προαιρετική μια τελεία για το δεκαδικό μέρος και το συν ή το πλην στην αρχή. Η επιστημονική δήλωση αριθμών γίνεται με το e να ακολουθείται από τον παράγοντα μίας δύναμης του 10. Οι φανταστικοί αριθμοί χρησιμοποιούν είτε το i είτε το j σαν πρόθεμα. Παραδείγματα σωστών αριθμών είναι

```
3          -99          0.0001
9.6397238  1.60210e-20  6.02252e23
1i         -3.14159j   3e5i
```

Όλοι οι αριθμοί αποθηκεύονται εσωτερικά χρησιμοποιώντας την μακριά μορφοποίηση όπως αυτή ορίζεται από το πρότυπο αριθμών μεταβλητής υποδιαστολής IEEE. Οι αριθμοί κινητής υποδιαστολής έχουν μία πεπερασμένη ακρίβεια περίπου 16 σημαντικών δεκαδικών ψηφίων και μία πεπερασμένη ακτίνα περίπου από το 10^{-308} ως το 10^{+308} . (Ο υπολογιστής VAX χρησιμοποιεί διαφορετική μορφοποίηση για του αριθμούς κινητής υποδιαστολής, αλλά η ακρίβεια και η ακτίνα είναι πάνω κάτω η ίδια.)

Load

Η εντολή load διαβάζει δυαδικά αρχεία που περιέχουν πίνακες δημιουργημένους προηγουμένως από το MATLAB ή διαβάζει αρχεία κειμένου τα οποία περιέχουν αριθμητικά δεδομένα. Το αρχείο κειμένου θα πρέπει να είναι οργανωμένο σαν ένας πίνακας αριθμών χωρισμένων με κενά, με μία γραμμή της μήτρας σε κάθε γραμμή κειμένου και ίσο αριθμό

στοιχείων σε κάθε γραμμή. Για παράδειγμα, δημιουργείστε, εκτός MATLAB, ένα αρχείο κειμένου που να περιέχει τις εξής τέσσερις σειρές :

```
16.0  3.0  2.0  13.0
5.0  10.0  11.0  8.0
9.0  6.0  7.0  12.0
4.0  15.0  14.0  1.0
```

Αποθηκεύστε το αρχείο με το όνομα magik.dat. Τότε η εντολή

```
load magik.dat
```

διαβάζει το αρχείο και δημιουργεί μία μεταβλητή, την magik, η οποία περιέχει την παραπάνω μήτρα.

M-Αρχεία

Μπορείτε να δημιουργείτε δικούς σας πίνακες χρησιμοποιώντας M-αρχεία, τα οποία είναι αρχεία κειμένου που περιέχουν κώδικα σε MATLAB. Απλά δημιουργείστε ένα αρχείο κειμένου το οποίο θα περιέχει τις ίδιες εντολές που θα χρησιμοποιούσατε στην γραμμή εντολής του MATLAB. Αποθηκεύστε το αρχείο με όνομα που τελειώνει σε .m.

Για παράδειγμα, δημιουργείστε ένα αρχείο που να περιέχει τις εξής πέντε γραμμές

```
A = [ ...
16.0  3.0  2.0  13.0
5.0  10.0  11.0  8.0
9.0  6.0  7.0  12.0
4.0  15.0  14.0  1.0 ];
```

αποθηκεύστε το αρχείο με το όνομα magik.m. Τότε η εντολή

```
magik
```

διαβάζει το αρχείο και δημιουργεί μια μεταβλητή, A, η οποία περιέχει το παράδειγμα μας.

Συνένωση

Συνένωση είναι η διαδικασία με την οποία ενώνουμε μικρούς πίνακες για να δημιουργήσουμε μεγαλύτερους. Στην πραγματικότητα δημιουργήσατε την πρώτη σας μήτρα συνενώνοντας ξεχωριστά της στοιχεία. Το ζευγάρι αγκυλών, [], είναι ο τελεστής της συνένωσης. Για παράδειγμα ξεκινήστε με το 4-επί-4 μαγικό τετράγωνο και πληκτρολογήστε

```
B = [A A+32; A+48 A+16]
```

Αυτό θα δώσει ένα 8-επί-8 πίνακα, ο οποίος προήλθε από τους τέσσερις πίνακες

B =

```
16  3  2  13  48  35  34  45
 5 10 11  8  37  42  43  40
 9  6  7 12  41  38  39  44
 4 15 14  1  36  47  46  33
64 51 50 61  32  19  18  29
53 58 59 56  21  26  27  24
57 54 55 60  25  22  23  28
52 63 62 49  20  31  30  17
```

Ο πίνακας αυτός είναι σχεδόν ένα μαγικό τετράγωνο. Τα στοιχεία είναι μία ανακατάταξη των ακεραίων 1:64. Το άθροισμα κάθε στήλης έχει την σωστή τιμή για ένα 8-επί-8 μαγικό τετράγωνο.

Εχουμε:

```
sum(B)
```

```
ans =
```

```
260 260 260 260 260 260 260 260
```

αλλά τα αθροίσματα των σειρών του, $\text{sum}(B')$, δεν είναι όλα τα ίδια. Χρειάζεται περισσότερη επεξεργασία για να κάνουμε τον πίνακα ένα έγκυρο 8-επί-8 μαγικό τετράγωνο.

Διαγραφή Γραμμών και Στηλών

Μπορείτε διαγράψετε γραμμές και στήλες από ένα πίνακα χρησιμοποιώντας απλώς ένα ζευγάρι αγκύλες. Ξεκινήστε με το

```
X = A;
```

Στην συνέχεια, για να διαγράψετε την δεύτερη στήλη του X, χρησιμοποιήσε

```
X(:,2) = []
```

Αυτό μετατρέπει το X σε

```
X =
```

```
16  2  13
 5 11  8
 9  7 12
 4 14  1
```

Εάν διαγράψετε ένα μόνο στοιχείο από μία μήτρα, το αποτέλεσμα δεν είναι μήτρα πια. Έτσι εκφράσεις όπως

$$X(1,2) = []$$

θα οδηγήσουν σε ένα λάθος. Πάντως, χρησιμοποιώντας ένα δείκτη, διαγράφουμε ένα στοιχείο ή μια ακολουθία στοιχείων και διαμορφώνουμε τα υπόλοιπα στοιχεία σε ένα διάλυσμα γραμμής. Έτσι το

$$X(2:2:10) = []$$

έχει ως αποτέλεσμα

$$X = \begin{matrix} 16 & 9 & 2 & 7 & 13 & 12 & 1 \end{matrix}$$

Αριθμητικοί Τελεστές + - * / \ ^ `

+ *Πρόσθεση:* $A + B$ προσθέτει τους πίνακες A και B . Οι πίνακες A και B πρέπει να είναι των ιδίων διαστάσεων, εκτός αν ο ένας είναι βαθμωτό μέγεθος. Ένα βαθμωτό μέγεθος μπορεί να προστεθεί σε πίνακα οποιωνδήποτε διαστάσεων.

- *Αφαίρεση:* $A - B$ αφαιρεί τον πίνακα B από τον πίνακα A . Οι πίνακες A και B πρέπει να είναι των ιδίων διαστάσεων, εκτός αν ο ένας είναι βαθμωτό μέγεθος. Ένα βαθμωτό μέγεθος μπορεί να αφαιρεθεί από πίνακα οποιωνδήποτε διαστάσεων.

* *Πολλαπλασιασμός:* $A * B$ είναι το γραμμικό αλγεβρικό γινόμενο των πινάκων A και B . Ο αριθμός των στηλών του A πρέπει να ισούται με τον αριθμό των γραμμών του B , εκτός αν ο ένας είναι βαθμωτό μέγεθος. Ένα βαθμωτό μέγεθος μπορεί να πολλαπλασιάσει πίνακα οποιωνδήποτε διαστάσεων.

.* *Διατεταγμένος πολλαπλασιασμός:* $A .* B$ είναι το στοιχείο προς στοιχείο γινόμενο των πινάκων A και B . Οι πίνακες A και B πρέπει να είναι των ιδίων διαστάσεων, εκτός αν ο ένας είναι βαθμωτό μέγεθος.

\ *Αριστερή διαίρεση πινάκων:* Αν A είναι τετραγωνικός πίνακας, $A \setminus B$ είναι σε γενικές γραμμές το ίδιο με $\text{inv}(A) * B$. Αν A είναι ένας $m \times n$ πίνακας και B μια στήλη, τότε $A \setminus B$ είναι η λύση ελαχίστων τετραγώνων του συστήματος εξισώσεων $AX = B$.

.\ *Διατεταγμένη αριστερή διαίρεση πινάκων:* $A .\ B$ είναι ο πίνακας με στοιχεία $B(i,j)/A(i,j)$. Οι πίνακες A και B πρέπει να είναι των ιδίων διαστάσεων, εκτός αν ο ένας είναι βαθμωτό μέγεθος.

/ *Δεξιά διαίρεση πινάκων:* $B/A = (A \setminus B)'$.

./ Διατεταγμένη δεξιά διαίρεση πινάκων: $A ./ B$ είναι ο πίνακας με στοιχεία $A(i,j)/B(i,j)$. Οι πίνακες A και B πρέπει να είναι των ίδιων διαστάσεων, εκτός αν ο ένας είναι βαθμωτό μέγεθος.

^ Δύναμη: X^p είναι ο πίνακας X στην p -οστή δύναμη, αν p είναι βαθμωτό μέγεθος. Στη γενική περίπτωση, χρησιμοποιούνται ιδιοτιμές και ιδιοδιανύσματα για τον υπολογισμό. Αν x είναι βαθμωτό μέγεθος και P ένας πίνακας, το x^P υπολογίζεται με χρήση ιδιοτιμών και ιδιοδιανυσμάτων. Η έκφραση X^P , όπου και X και P είναι πίνακες, είναι λάθος.

.^ Διατεταγμένη δύναμη: $A.^B$ είναι ο πίνακας με στοιχεία $A(i,j)$ στη δύναμη $B(i,j)$. Οι πίνακες A και B πρέπει να είναι των ίδιων διαστάσεων, εκτός αν ο ένας είναι βαθμωτό μέγεθος.

` Ανάστροφος πίνακα: A' είναι ο αλγεβρικός ανάστροφος του A . Για πίνακες με μιγαδικά στοιχεία υπολογίζεται ο μιγαδικός συζυγής.

.' Διατεταγμένος ανάστροφος: $A.'$ είναι ο αλγεβρικός ανάστροφος του A . Για πίνακες με μιγαδικά στοιχεία δεν υπολογίζεται ο μιγαδικός συζυγής.

Τελεστές Συσχέτισης

< > <= >= == ~=

Λογικοί Τελεστές

& | ~

Ειδικοί Χαρακτήρες

[] () = ` . ; % !

Το Παράθυρο Εντολών

Ως τώρα, έχετε χρησιμοποιήσει την γραμμή εντολών του MATLAB, γράφοντας εντολές και εκφράσεις, και βλέποντας τα αποτελέσματα να εμφανίζονται στο παράθυρο εντολών. Αυτός ο τομέας περιγράφει μερικούς τρόπους να αλλάξουμε την εμφάνιση του παράθυρου εντολών. Εάν το σύστημά σας σας επιτρέπει να επιλέξετε την γραμματοσειρά του παράθυρου εντολών σας συμβουλεύουμε να χρησιμοποιήσετε μία σταθερού πλάτους, όπως η Fixedsys ή Courier, ώστε να υπάρχουν κατάλληλα διαστήματα.

H Εντολή format

Η εντολή `format` καθορίζει τον τρόπο με το οποίο εμφανίζονται οι αριθμητικές τιμές στο MATLAB. Η εντολή επηρεάζει μόνο τον τρόπο με τον οποίο εμφανίζονται οι αριθμοί και όχι τον τρόπο με τον οποίο το MATLAB κάνει τους υπολογισμούς ή τους αποθηκεύει. Εδώ παρατίθενται οι διάφορες μορφοποιήσεις καθώς και το αποτέλεσμα που παράγεται από ένα πίνακα `x`.

```
x = [ 4/3 1.2345e-6]
```

```
format short
```

```
1.3333 0.0000
```

```
format short e
```

```
1.3333e+000 1.2345e-006
```

```
format short g
```

```
1.3333 1.2345e-006
```

```
format long
```

```
1.333333333333333 0.00000123450000
```

```
format long e
```

```
1.333333333333333e+000 1.234500000000000e-006
```

```
format long g
```

```
1.333333333333333 1.2345e-006
```

```
format bank
```

```
1.33 0.00
```

```
format rat
```

```
4/3 1/810045
```

```
format hex
```

```
3ff5555555555555 3eb4b6231abfd271
```

Εάν το μεγαλύτερο στοιχείο μίας μήτρας είναι μεγαλύτερο από 10^3 ή μικρότερο από 10^{-3} , το MATLAB χρησιμοποιεί κοινή κλίμακα για τις μορφοποιήσεις short και long.

Μαζί με τις εντολές format που είδατε παραπάνω η

```
format compact
```

δεν εμφανίζει πολλές από τις κενές γραμμές που εμφανίζονται. Αυτό σας επιτρέπει να βλέπεται περισσότερες πληροφορίες σε μία οθόνη ή ένα παράθυρο. Εάν θέλετε πλήρη έλεγχο στην μορφή της εξόδου, χρησιμοποιήστε τις συναρτήσεις sprintf και fprintf.

Καταπίεση της Εξόδου

Εάν απλώς τυπώσετε μία εντολή ή έκφραση και πατήσετε το Enter, το MATLAB αυτόματα παρουσιάζει το αποτέλεσμα στην οθόνη. Εάν όμως τελειώσετε την γραμμή με ένα ελληνικό ερωτηματικό, το MATLAB κάνει τους υπολογισμούς αλλά δεν παρουσιάζει το αποτέλεσμα. Αυτό είναι ιδιαίτερα χρήσιμο όταν δημιουργείτε μεγάλες πίνακες, π.χ.

```
A = magic(100)
```

Μακροσκελείς εκφράσεις

Εάν μία έκφραση δεν χωράει σε μία γραμμή, χρησιμοποιήστε τρεις τελείες, ..., ακολουθούμενες από το Return ή το Enter για να δείξετε ότι η έκφραση συνεχίζεται στην επόμενη γραμμή. Για παράδειγμα

```
s = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 ...  
    - 1/8 + 1/9 - 1/10 + 1/11 - 1/12;
```

Τα κενά γύρω από τα =, +, - είναι προαιρετικά, αλλά κάνουν πιο ευανάγνωστο το κείμενο.

Δουλεύοντας στην Γραμμή Εντολών

Τα διάφορα βέλη και πλήκτρα ελέγχου στο πληκτρολόγιο σας επιτρέπουν να ανακαλέσετε, να επεξεργαστείτε και να ξαναχρησιμοποιήσετε εντολές που τυπώσατε προηγουμένως. Π.χ. ας πούμε ότι γράψατε κατά λάθος την εντολή

```
rho = (1 + sqrt(5))/2.
```

Έχετε γράψει λάθος την εντολή sqrt. Το MATLAB απαντάει με:

```
Undefined function or variable 'sqrt', δηλαδή, μη καθορισμένη συνάρτηση ή μεταβλητή 'sqrt'
```

Αντί να ξαναγράψετε ολόκληρη την εντολή, απλώς πατήστε το πάνω βέλος. Η λανθασμένη εντολή θα εμφανιστεί. Χρησιμοποιήστε το αριστερό βέλος για να μετακινήσετε τον κέρσορα

και συμπληρώστε το r. Το πάνω βέλος καλεί προηγούμενες εκφράσεις. Αν τυπώσετε μερικούς χαρακτήρες τότε το πάνω βέλος βρίσκει τις εντολές που ξεκινούν με αυτούς τους χαρακτήρες.

Η λίστα των διαθέσιμων πλήκτρων διαφέρει σε διαφορετικούς υπολογιστές. Πειραματιστείτε για να δείτε ποια από τα παρακάτω πλήκτρα είναι διαθέσιμα στον υπολογιστή σας.

Πάνω βέλος	Ctrl + p	Ανάκληση προηγούμενης εντολής
Κάτω βέλος	Ctrl + n	Ανάκληση επόμενης εντολής
Αριστερό βέλος	Ctrl + b	Μπροστά ένας χαρακτήρας
Δεξιό βέλος	Ctrl + f	Πίσω ένας χαρακτήρας
Ctrl + αριστερό βέλος	Ctrl + r	Μπροστά μία λέξη
Ctrl + δεξιό βέλος	Ctrl + l	Πίσω μία λέξη
Home	Ctrl + a	Κίνηση στην αρχή της γραμμής
End	Ctrl + e	Κίνηση στο τέλος της γραμμής
Esc	Ctrl + u	Καθαρίζει την γραμμή
Del	Ctrl + d	Διαγραφή του χαρακτήρα στον κέρσορα
Backspace	Ctrl + h	Διαγραφή του χαρακτήρα πριν από τον κέρσορα
	Ctrl + k	Διαγραφή ως το τέλος της γραμμής

Γραφικά

Η MATLAB έχει εκτενείς δυνατότητες για την αναπαράσταση διανυσμάτων και πινάκων σε μορφή γραφικών παραστάσεων, όπως επίσης και για την προσθήκη υποσημειώσεων σε αυτά καθώς και για την εκτύπωση τους. Αυτό το κεφάλαιο περιγράφει μερικές από τις πιο σημαντικές συναρτήσεις γραφικών και παραθέτει παραδείγματα κάποιων τυπικών εφαρμογών.

Κατασκευή ενός Σχεδιαγράμματος

Η συνάρτηση plot εμφανίζεται με διαφορετικές μορφές ανάλογα με τις εισαγόμενες παραμέτρους. Εάν u είναι ένα διάνυσμα τότε η plot(y) παράγει ένα τμηματικά γραμμικό

γράφημα των στοιχείων του y ως προς τον δείκτη του y . Εάν δώσετε δυο διανύσματα ως παραμέτρους, η `plot(x,y)` παράγει ένα γράφημα του y σε συνάρτηση με το x .

Για παράδειγμα, για να σχεδιάσετε τις τιμές της συνάρτησης \sin από το 0 έως το 2π χρησιμοποιείτε το ακόλουθο:

```
t = 0:pi/100:2*pi;
y = sin(t);
plot(t,y)
y2 = sin(t-.25);
y3 = sin(t-.5);
plot(t,y,t,y2,t,y3)
```

Είναι δυνατό να προσδιορίσουμε το χρώμα, το στυλ της γραμμής και σύμβολα σήμανσης όπως κύκλους με την εντολή

```
plot(x , y , 'color_style_marker')
```

Το `color_style_marker` είναι μία συμβολοσειρά από 1,2 ή τρεις χαρακτήρες (το οποίο παρατίθεται με απλά εισαγωγικά) που κατασκευάζετε από ένα χρώμα, ένα στυλ γραμμής και ένα τύπο σημείου σήμανσης

Οι συμβολοσειρές χρώματος είναι 'c' , 'm' , 'y' , 'r' , 'g' , 'b' , 'w' και 'k'. Αυτές αντιστοιχούν στα χρώματα κυανό, κίτρινο, κόκκινο, πράσινο, άσπρο και μαύρο.

Οι συμβολοσειρές του στυλ γραμμής είναι '-' για μια συμπαγή γραμμή , '--' για διακεκομμένη γραμμή , ':' για γραμμή από τέλειες , '-.' για γραμμή από παύλα-τέλεια και 'none' για την μη ύπαρξη γραμμής

Οι πιο συχνοί τύποι σημείων σήμανσης περιλαμβάνουν τα '+', 'o', '*', 'x'
Για παράδειγμα η δήλωση

```
plot( x, y, 'y:+' )
```

σχεδιάζει μια κιτρίνη γραμμή από τέλειες και τοποθετεί ένα σημείο συν σε κάθε σημείο δεδομένων. Εάν δώσετε το σημείο σήμανσης αλλά όχι το στυλ γραμμής το MATLAB σχεδιάζει μόνο το σημείο σήμανσης.

Παράθυρα Σχεδίασης

Η συνάρτηση `plot` αυτόματα ανοίγει ένα νέο παράθυρο σχεδίασης εάν δεν υπάρχουν άλλα παράθυρα σχεδίασης ήδη ανοιχτά στην οθόνη. Εάν υπάρχει ένα παράθυρο σχεδίασης, η `plot` χρησιμοποιεί το παράθυρο αυτό εάν δεν της δοθεί εντολή να χρησιμοποιήσει κάποιο συγκεκριμένο παράθυρο. Για να ανοίξετε ένα νέο παράθυρο σχεδίασης και να το κάνετε το ενεργό σχεδιάγραμμα, πληκτρολογήστε

figure

Για να κάνετε ένα ήδη υπάρχον παράθυρο το ενεργό παράθυρο πληκτρολογείστε

figure(n)

όπου (n) είναι ο αριθμός στην ραβδο τίτλου του σχεδιαγράμματος. Τα αποτελέσματα μετέπειτα εντολών γραφικών εμφανίζονται σε αυτό το παράθυρο.

Προσθήκη Σχεδίων σε Υπαρκτό Γράφημα

Η εντολή hold επιτρέπει την προσθήκη σχεδίων σε ένα υπαρκτό γράφημα. Όταν πληκτρολογήσετε

hold on

δεν αφαιρεί το ήδη υπάρχον γράφημα, αλλά προσθέτει τα νέα δεδομένα στο παρόν γράφημα ανακαλώντας το εάν χρειάζεται. Παραδείγματος χάριν οι ακόλουθες δηλώσεις πρώτα δημιουργούν ένα σχεδιάγραμμα περιγράμματος της συνάρτησης peaks και ύστερα τοποθετεί ένα σχεδιάγραμμα της ίδιας συνάρτησης σε ' ψευδοχρώμα '

```
[x,y,z] = peaks;  
contour(x,y,z,20,'k')  
hold on  
pcolor(x,y,z)  
shading interp
```

Η εντολή hold on συνδυάζει το σχεδιάγραμμα pcolor με το σχεδιάγραμμα contour σε μια παράσταση.

Τμηματικά Σχεδιαγράμματα

Η συνάρτηση subplot επιτρέπει την εμφάνιση πολλαπλών σχεδίων στο ίδιο παράθυρο ή την εκτύπωση τους στο ίδιο φύλλο χαρτιού. Πληκτρολογώντας

subplot(m,n,p)

χωρίζει το παράθυρο σχεδίασης σε έναν πίνακα m επί n ο οποίος αποτελείται από μικρά τμηματικά σχεδιαγράμματα και επιλέγει το p-οστό ως το ενεργό σχεδιάγραμμα. Τα σχεδιαγράμματα είναι αριθμημένα αρχικά κατά μήκος της πιο πάνω σειράς του παραθύρου σχεδίασης, κατόπιν κατά μήκος της δεύτερης, κ.ο.κ. Παραδείγματος χάριν για να σχεδιάσετε δεδομένα σε τέσσερις επιμέρους περιοχές του παραθύρου σχεδίασης πληκτρολογήστε:

```
t = 0:pi/10:2*pi;  
[X,Y,Z] = cylinder(4*cos(t));
```

```
subplot(2,2,1)
mesh(X)
subplot(2,2,2); mesh(Y)
subplot(2,2,3); mesh(Z)
subplot(2,2,4); mesh(X,Y,Z)
```

Φανταστικά και Μιγαδικά Δεδομένα

Όταν οι τιμές των μεταβλητών της plot είναι μιγαδικές, το φανταστικό μέρος αγνοείται εκτός εάν δοθεί στην plot μία μόνο μιγαδική τιμή. Σε αυτήν την ειδική περίπτωση, η εντολή είναι μια συντόμευση για ένα σχεδιάγραμμα του πραγματικού ως προς το φανταστικό μέρος.

Συνεπώς η εντολή :

```
plot(z),
```

όπου z είναι ένα μιγαδικό διάνυσμα ή μήτρα, είναι ισοδύναμο με την :

```
plot (real(z),imag(z) )
```

Παραδείγματος χάριν το

```
t = 0:pi/10:2*pi;
plot(exp(I*t),'-o')
```

σχεδιάζει ένα εικοσάπλευρο πολύγωνο με μικρούς κύκλους στις κορυφές.

Έλεγχος των Αξόνων

Η συνάρτηση axis έχει έναν αριθμό από επιλογές για την προσαρμογή της διαβάθμισης, της κατεύθυνσης και του λόγου μεταξύ των κλιμάκων των αξόνων των σχεδιαγραμμάτων.

Κανονικά, το MATLAB βρίσκει το μέγιστο και ελάχιστο από τα δεδομένα και επιλέγει το κατάλληλο κουτί σχεδίασης και της κατάλληλες επιγραφές αξόνων. Η συνάρτηση axis υπερπηδά την εξ ορισμού ισχύουσα κατάσταση με το να θέτει επιλεγμένα από τον χρήστη όρια αξόνων:

```
axis([xmin xmax ymin ymax])
```

Η axis επίσης δέχεται έναν αριθμό από λέξεις κλειδιά για τον έλεγχο των αξόνων. Για παράδειγμα, η εντολή

```
axis square
```

κάνει ολόκληρους τους άξονες x και y ισομήκεις και η εντολή

```
axis equal
```

κάνει τις αποστάσεις των διαβαθμίσεων των x και y επί των αξόνων ίσες. Έτσι η

```
plot(exp(i*t)),
```

ακολουθούμενη είτε από την εντολή axis square είτε από την axis equal, μετατρέπει το οβάλ σε κανονικό κύκλο. Η εντολή

```
axis auto
```

επαναφέρει τις διαβαθμίσεις στις εξ ορισμού ισχύουσες τιμές τους. Η εντολή

```
axis on
```

ενεργοποιεί της επιγραφές των αξόνων και τις γραμμές διαβαθμίσεων, ενώ η

```
axis off
```

απενεργοποιεί τις επιγραφές των αξόνων και τις γραμμές διαβαθμίσεων.

Η εντολή

```
grid off
```

απενεργοποιεί τις γραμμές πλέγματος και η

```
grid on
```

τις επανενεργοποιεί.

Επιγραφές και Τίτλοι Αξόνων

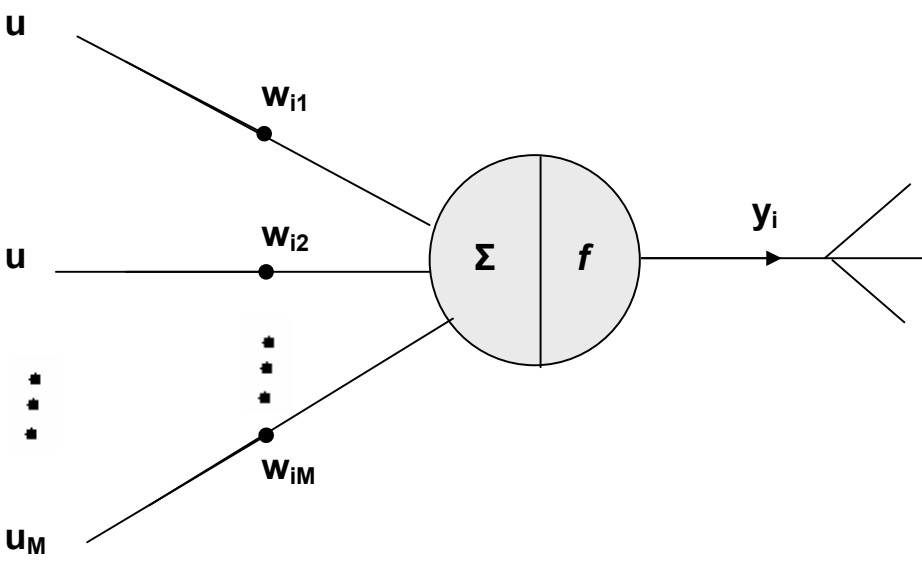
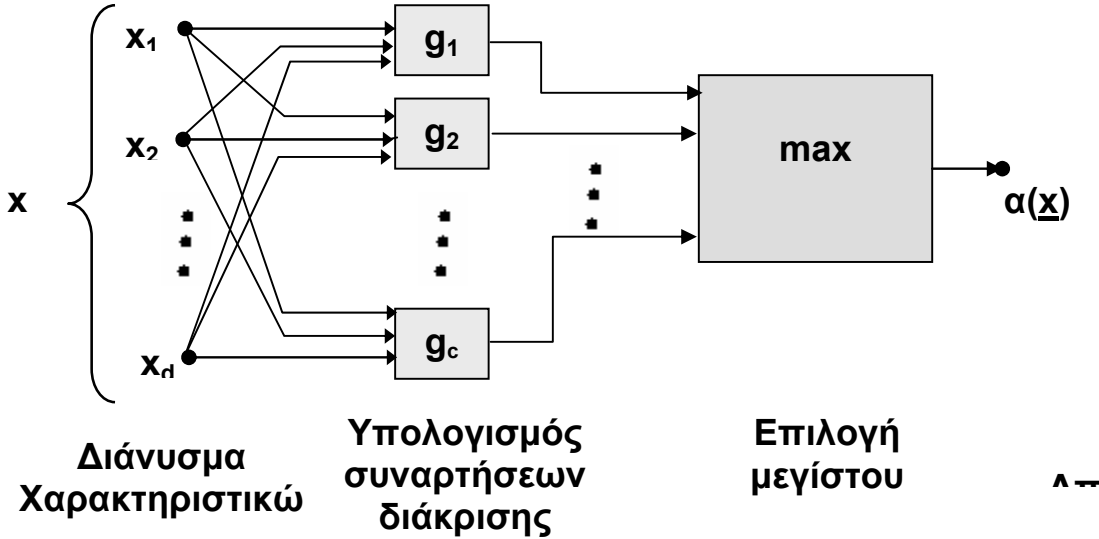
Οι συναρτήσεις xlabel, ylabel και zlabel προσθέτουν επιγραφές στους άξονες x, y και z. Η συνάρτηση title προσθέτει έναν τίτλο στην κορυφή του σχεδίου και η συνάρτηση text εισάγει κείμενο οπουδήποτε στο σχεδιάγραμμα. Ένα υποσύνολο του συμβολισμού του TeX παράγει ελληνικά γράμματα, μαθηματικά σύμβολα και εναλλακτικές γραμματοσειρές. Το ακόλουθο παράδειγμα χρησιμοποιεί \leq για \leq , \pi για π και \it για πλάγιου τύπου γραμματοσειρά:

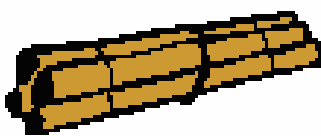
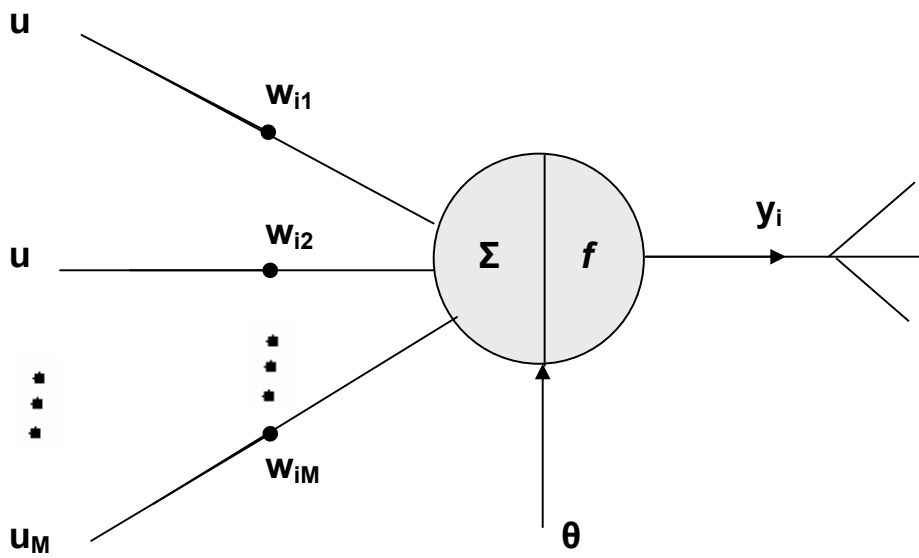
```
t = -pi:pi/100:pi;  
y = sin(t);  
plot( t , y );  
axis([-pi pi -1 1 ])  
xlabel('-\pi \leq \{itt\} \leq \pi')
```

```

ylabel('sin(t)')
title('Graph of the sine function');
text(1,-1/3,'\it{Note the odd symmetry}')

```





Ξυλεία



Αισθητήριο

Εξαγωγέας
Χαρακτηριστικών



1.1 Τεχνολογία



Απόφαση

