



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΕΧΝΟΛΟΓΙΕΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗΝ ΙΑΤΡΙΚΗ ΚΑΙ ΤΗ ΒΙΟΛΟΓΙΑ

Αναγνώριση Προτύπων

Πρακτική Άσκηση 1

Επιβλέπων: **Περαντώνης Σταύρος**, Διευθυντής Ερευνών, ΕΚΕΦΕ-Δημόκριτος

ΑΘΗΝΑ

ΙΑΝΟΥΑΡΙΟΣ 2013

Αναγνώριση Προτύπων

Πρακτική Άσκηση 1

Κωνσταντόπουλος Γ. Δημήτριος
Μπεγέτης Ι. Νικόλαος

ΑΜ: ΠΙΒ0112 / ΠΙΒ0111

ΕΠΙΒΛΕΠΩΝ :

Περαντώνης Σταύρος , Διευθυντής Ερευνών, ΕΚΕΦΕ-Δημόκριτος

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	Πρακτική Άσκηση 1	11
1.1	Θέμα 1	11
1.2	Θέμα 2	19
2	Υλοποιήσεις σε Matlab	31
2.1	Θέμα 1	31
2.1.1	k-NN classifier	32
2.1.2	Bayesian classifier	37
2.2	Θέμα 2	40
2.2.1	Bayesian Nets Toolkit	40
3	Ενδεικτικές Εικόνες και samIam	43
4	Κατακλείδα	47
	Ορολογία	49
	Ακρωνύμια - Αρκτικόλεξα	51
	Βιβλιογραφία	53

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

1.1	k-NN classification of Iris Database	14
1.2	k-NN classification of Iris Database with normalized data	15
1.3	k-NN classification of Pima Indian Database	16
1.4	k-NN classification of Pima Indian Database with normalized data	16
1.5	Bayesian classification of Pima Indian Database	17
1.6	bayesian classification of Pima Indian Database with normalized data	17
1.7	naive bayesian classification of Pima Indian Database	19
1.8	naive bayesian classification of Pima Indian Database with normalized data	19
1.9	Metastatic Cancer Bayesian Network	21
1.10	Metastatic Cancer Bayesian Network - samIam	22
1.11	Direct Probability Distribution e.g.1	23
1.12	Direct Probability Distribution e.g.1 - bar	23
1.13	Direct Probability Distribution e.g.1 - output	23
1.14	Direct Probability Distribution e.g.1 - samIam	24
1.15	Direct Probability Distribution e.g.2 - bar	24
1.16	Direct Probability Distribution e.g.2 - output	24
1.17	Direct Probability Distribution e.g.2 - samIam	25
1.18	Retrograded Probability Distribution e.g.1 - bar	25
1.19	Retrograded Probability Distribution e.g.1 - output	26
1.20	Retrograded Probability Distribution e.g.1 - samIam	26

1.21 Retrograded Probability Distribution e.g.2 - bar	26
1.22 Retrograded Probability Distribution e.g.2 - output	27
1.23 Retrograded Probability Distribution e.g.2 - samlam	27
1.24 CPT monitor for MC	29
3.1 Inferencing of Metastatic Cancer Bayesian Nets	43
3.2 Inferencing of Metastatic Cancer Bayesian Nets with B present	44
3.3 Inferencing of Metastatic Cancer Bayesian Nets with C present	44
3.4 Inferencing of Metastatic Cancer Bayesian Nets with C absent	44
3.5 Inferencing of Metastatic Cancer Bayesian Nets with MC and B present . . .	45
3.6 Inferencing of Metastatic Cancer Bayesian Nets with MC present	45
3.7 Inferencing of Metastatic Cancer Bayesian Nets with SC incr and SH present	45
3.8 Inferencing of Metastatic Cancer Bayesian Nets with SC not incr and B absent	46
3.9 Inferencing of Metastatic Cancer Bayesian Nets with SC not incr	46
3.10 Inferencing of Metastatic Cancer Bayesian Nets with SH present	46

ΚΕΦΑΛΑΙΟ 1

Πρακτική Άσκηση 1

Η Αναγνώριση Προτύπων (Pattern Recognition) είναι μία επιστημονική περιοχή που έχει στόχο την απόδοση κάποιας τιμής ή διακριτικού στοιχείου σε εισαγόμενα δεδομένα. Οι άνθρωποι και τα άλλα όντα έχουν την ικανότητα να ταυτοποιούν πραγματικά δεδομένα χρησιμοποιώντας τις αισθήσεις τους και την αντιληπτική τους ικανότητα (cognition) προκειμένου να λάβουν τις κατάλληλες αποφάσεις ώστε να επιβιώσουν στο περιβάλλον τους.

Μία μηχανή, όπως ένας ηλεκτρονικός υπολογιστής, πρέπει να εκπαιδευθεί κατάλληλα ώστε να αναγνωρίζει πρότυπα (patterns) και να τα κατηγοριοποιεί αυτόματα σε κατηγορίες. Ανάλογα με την εφαρμογή γίνεται κατάταξη των αντικειμένων σε κλάσεις με τη βοήθεια αλγορίθμων ταξινόμησης.

Το ερευνητικό ενδιαφέρον για αυτά τα ζητήματα ξεκίνησε από τη δεκαετία του 1960, κατά την πρώτη περίοδο της ανάπτυξης της επιστήμης των υπολογιστών. Βασισμένη στο θεωρητικό υπόβαθρο που παρείχε η επιστήμη της Στατιστικής, η πρώιμη έρευνα επικεντρώθηκε στην ανάπτυξη θεωρητικών μεθόδων. Ήδη από το 1970 γίνονταν προσπάθειες για την καλύτερη κατεύθυνση των προσπαθειών και το 1976 ιδρύεται η Παγκόσμια Ένωση για την Αναγνώριση Προτύπων (IARP). Σε πολλά επιστημονικά πεδία αξιοποιούνται εφαρμογές της αναγνώρισης προτύπων, όπως στην Ιατρική (υποβοηθούμενη από Η/Υ διάγνωση, ανάλυση δεδομένων DNA και άλλες εφαρμογές της βιοπληροφορικής) και την επιστήμη υπολογιστών (υπολογιστική όραση, αναγνώριση χαρακτηριστικών ή φωνής, νευρωνικά δίκτυα, εξόρυξη δεδομένων και ανάκτηση γνώσης, τεχνητή νοημοσύνη και μηχανική μάθηση, συστήματα υποστήριξης αποφάσεων). Στον σύγχρονο κόσμο, πολλές βιομηχανικές εφαρμογές ενσωματώνουν ανάλογα συστήματα για την αποδοτική και αυτόματη επεξεργασία πληροφοριών¹.

1.1 Θέμα 1

Στα πλαίσια της πρώτης Πρακτικής Άσκησης στο μάθημα της Αναγνώρισης Προτύπων, κλήθηκαμε να υλοποιήσουμε, να εκτιμήσουμε και να αξιολογήσουμε 3 είδη ταξινομητών, (α) τον ταξινομητή πλησιέστερων γειτόνων (K-NN), (β) τον ταξινομητή Bayes και (γ) τον απλό ταξινομητή

¹http://el.wikipedia.org/wiki/Αναγνώριση_προτύπων

Bayes(Naive Bayes), με εφαρμογή πάνω σε 2 πολύ γνωστά από την βιβλιογραφία datasets, το Iris plant Database και το Pima Indians Diabetes Database. Εμείς επιλέξαμε να πραγματοποιήσουμε την υλοποίηση μας στο προγραμματιστικό περιβάλλον της Matlab. Η εκτίμηση των αποτελεσμάτων για την κάθε υλοποίηση των ταξινομητών, για τα συγκεκριμένα datasets, έγινε με την χρήση της μεθόδου tenfold validation. Κατά την συγκεκριμένη μέθοδο, χωρίζονται τα δοθέντα δεδομένα σε 10 ισόποσα μέρη, folds. Σε κάθε βήμα του ελέγχου, τα δεδομένα των 9 folds, training fold, θα αποτελούν τα δεδομένα εκπαίδευσης του ταξινομητή που υλοποιείται, ενώ τα δεδομένα του 1 fold, validation fold, θα αποτελούν αυτά που είναι προς ταξινόμηση.

Πραγματοποιώντας επαναληπτικά την διαδικασία, όλα τα folds, με την σειρά θα έχουν από μία φορά τον ρόλο του validation fold, και 9 φορές θα αποτελούν μέρος του training fold. Για να υπάρχει τυχαιότητα στην θέση των δεδομένων μέσα στο dataset που μας δίνεται, αποφασίσαμε να κάνουμε μια τυχαία αναταξινόμηση των δεδομένων, ώστε να μην έχουν τις ίδιες θέσεις κάθε φορά που θα πραγματοποιείται μια υλοποίηση. Η αξιολόγηση των ταξινομητών γίνεται με βάση την επίδοσή τους, και πιο συγκεκριμένα με το ποσοστό πραγματοποίησης ορθής ταξινόμησης. Μετά από κάθε υλοποίηση των ταξινομητών, συγκρίνονται τα αποτελέσματα ταξινόμησης που προέκυψαν με αυτά που είναι καταχωρημένα στα dataset που έχουν δοθεί και έτσι υπολογίζεται το παρακάτω ποσοστό.

$$\frac{\text{Number of good classifications}}{\text{Number of classifications}} \times 100$$

Όπου ζητείται, γίνεται και αξιολόγηση με κριτήριο την σύγκριση των επιδόσεων με κάποιον άλλο ταξινομητή που έχει υλοποιηθεί.

Τα εκτελέσιμα προγράμματα Matlab καθώς και τα dataset που απαιτούνται για την κάθε υλοποίηση, βρίσκονται στους ομώνυμους φακέλους για τα υποερωτήματα α, β και γ αντίστοιχα του αρχικού φακέλου. Όλα τα υποερωτήματα έχουν εκτελεστεί υπεραρκετές φορές ώστε τα αποτελέσματα που εξάγουμε να είναι αντιπροσωπευτικά της υλοποίησης μας.

Για τα dataset που μας έχουν δοθεί, τα έχουμε χρησιμοποιήσει σε 2 μορφές, (1) όπως δίνονται στις default τιμές τους και (2) κανονικοποιημένα. Η κανονικοποίηση που έχουμε πραγματοποιήσει γίνεται με την standardizing method η οποία ακολουθεί τον τύπο:

$$z = \frac{\chi - \mu}{\sigma}$$

Όπου το μ είναι η μέση τιμή του συγκεκριμένου χαρακτηριστικού, feature, ενώ το σ είναι η αντίστοιχη τυπική απόκλιση. Το χ είναι η τιμή που είχε πριν την κανονικοποίηση το feature του κάθε διανύσματος από features. Ουσιαστικά μέσω της συγκεκριμένης κανονικοποίησης το σεν των features του κάθε dataset αποκτούν κοινή μέση τιμή $\mu = 0$ και κοινή τυπική απόκλιση $\sigma = 1$. Στην Matlab η συγκεκριμένη κανονικοποίηση πραγματοποιείται με την εντολή `zscore(X)`. Στο Pima Indians Diabetes Database έχουμε προσθέσει (πριν την κανονικοποίηση) άλλα 2 διανύσματα από features, 2 θάδες, οι οποίες παίρνουν τιμές για το κάθε χαρακτηριστικό τους ίση με την μέση τιμή όλης της κατηγορίας του κάθε χαρακτηριστικού και είναι ταξινομημένα σε μια ουδέτερη κλάση, την κλάση 2. Έτσι δεν επηρεάζουν τα αποτελέσματα και μας βοηθάνε να διαμερίσουμε τα δεδομένα μας σε 10 folds καθώς ο αριθμός των διανυσμάτων ήταν πριν 768 και μετά από αυτήν την μετατροπή είναι 770.

- (α') **Με χρήση του Matlab ή με άλλο τρόπο να υλοποιήσετε έναν ταξινομητή πλησιέστερων γειτόνων (k-NN). Να τον χρησιμοποιήσετε για να επιλύσετε τα εξής γνωστά από τη βιβλιογραφία προβλήματα :**

1 IRIS PLANT DATABASE (ταξινόμηση φυτών Iris σε τρία είδη).

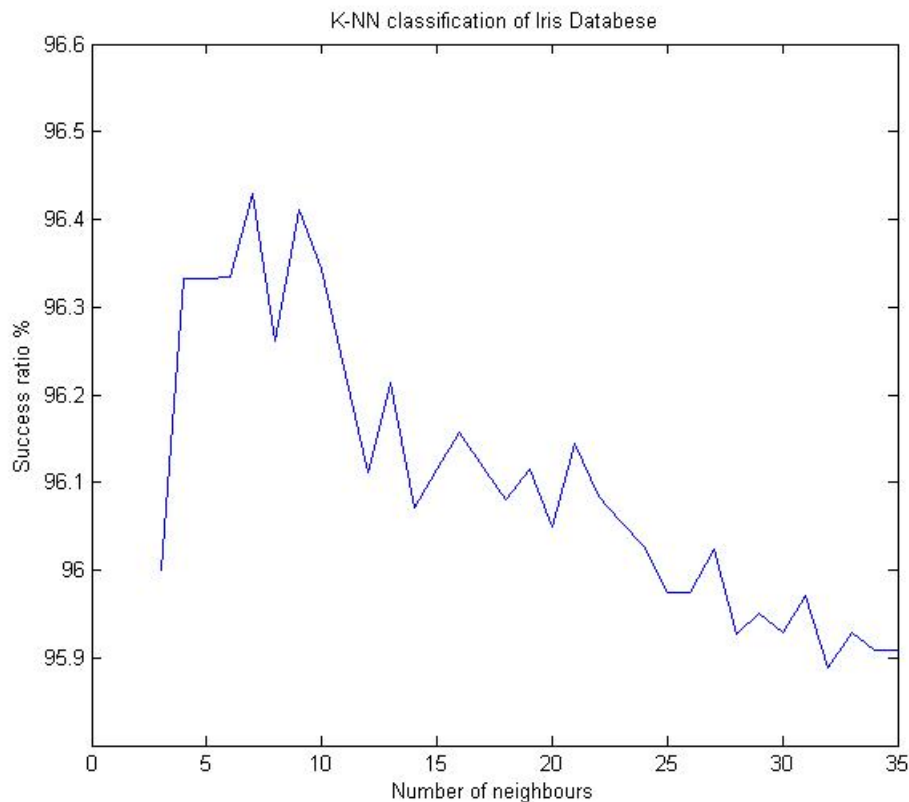
2 PIMA INDIANS DIABETES DATABASE (ταξινόμηση εγγύων ινδιάνων της φυλής Pima σε άτομα που έχουν ή δεν έχουν διαβήτη).

Τα δεδομένα των προβλημάτων θα τα βρείτε στο αρχείο UCIdata-exercise1.rar. Όπου κρίνετε σκόπιμο, κανονικοποιήστε τα δεδομένα ανά στήλη πριν εφαρμόσετε τον ταξινομητή. Να μελετήσετε το ποσοστό ορθής ταξινόμησης ως συνάρτηση του αριθμού των πλησιέστερων γειτόνων. Η εκτίμηση του αποτελέσματος να γίνει με χρήση της μεθόδου tenfold crossvalidation.

Στο πρώτο υποερώτημα υλοποιήσαμε έναν ταξινομητή πλησιέστερων γειτόνων, K-NN classifier, για τα 2 dataset που μας έχουν δοθεί. Στον φάκελο K-NN περιέχονται 2 υποφάκελοι, ο Iris και ο Indians, στους οποίους υπάρχει αντίστοιχα η υλοποίηση του ταξινομητή K-NN για το κάθε dataset. Τα εκτελέσιμα αρχεία είναι αντίστοιχα το Iris.m και το Indian.m. Για κάθε αρχείο διαβάζεται το αντίστοιχο dataset και η κάθε κατηγορία χαρακτηριστικών αποθηκεύεται σε ένα vector. Έτσι για το Iris έχουμε 5 vectors, όλα μεγέθους 150×1 , με τα 4 πρώτα να εκφράζουν τα 4 χαρακτηριστικά των φυτών, και το 5ο να περιέχει την κατηγορία στην οποία κατατάσσεται το κάθε φυτό. Αντίστοιχα και στο Pima έχουμε 9 vectors μεγέθους 770×1 , με τα 8 πρώτα να εκφράζουν τα 8 χαρακτηριστικά των ινδιάνων, και το 9ο να περιέχει την κατηγορία στην οποία κατατάσσεται ο κάθε ινδιάνος. Για το Iris database τρέξαμε το script πολλές φορές και μελετήσαμε την συμπεριφορά του ταξινομητή για 1 μέχρι 35 γείτονες. Τα δεδομένα αναταξινομήθηκαν με τυχαίο τρόπο μέσω της συνάρτησης randperm() της Matlab και έπειτα χωρίστηκαν σε 10 folds ώστε να εφαρμοστεί η μέθοδος tenfold validation όπως περιγράφεται παραπάνω. Έτσι τα 9 folds αποτέλεσαν την εκπαίδευση του αλγορίθμου, ενώ το 1 fold τα φυτά που έπρεπε να ταξινομηθούν. Για κάθε φυτό του validation fold, υπολογίσαμε την Ευκλείδεια απόστασή του από κάθε φυτό του training set, μέσω της συνάρτησης Euclidean που υλοποιήσαμε, και τις αποθηκεύσαμε σε ένα vector που ονομάσαμε dist. Ταξινομούμε τις αποστάσεις αυτές σε αύξουσα σειρά, με την συνάρτηση sort() της Matlab. Στην συνέχεια χρησιμοποιούμε την συνάρτηση KNN για να ταξινομήσουμε το φυτό που εξετάζουμε. Συγκεκριμένα, μέσα στην συνάρτηση χωρίζουμε το ταξινομημένο με βάση την απόσταση dataset σε 3 κατηγορίες, τις κατηγορίες των φυτών, και διαλέγουμε τον πιο απομακρυσμένο από τους κοντινότερους γείτονες. Π.χ. Αν είμαστε στην περίπτωση που εξετάζουμε τον KNN των 15 πλησιέστερων γειτόνων, διαλέγουμε για κάθε κατηγορία τον 15ο πιο απομακρυσμένο από αυτούς. Έτσι υπολογίζουμε 3 κύκλους μ ακτίνα τις αποστάσεις αυτές, έναν κύκλο για κάθε κατηγορία. Τέλος ταξινομούμε το φυτό με βάση τον τύπο 2.116 που δίνεται στο βιβλίο σελ. 57. :

$$z = \frac{V_2}{V_1} > (<) \frac{N_1 P(\omega_2)}{N_2 P(\omega_1)}$$

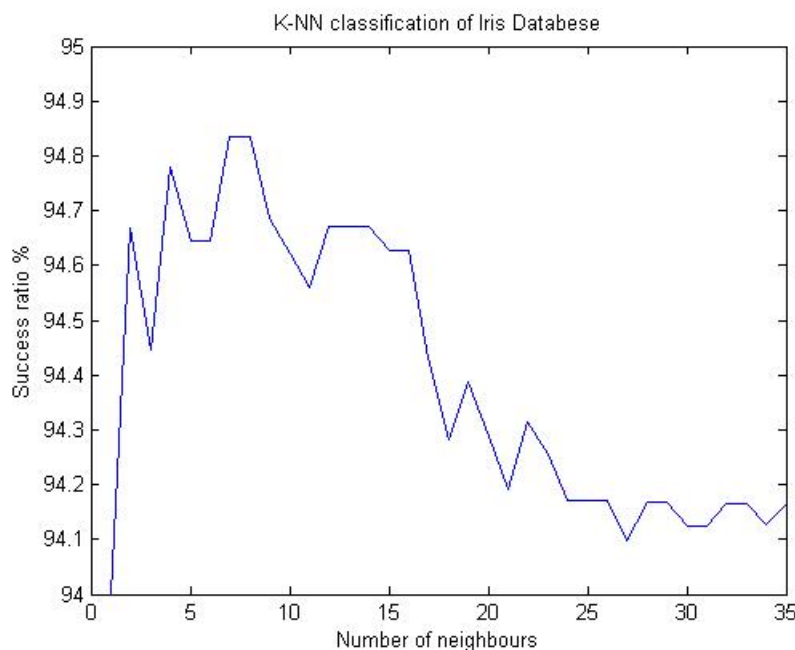
Όπου V_i ο όγκος του κάθε κύκλου, N_i ο αριθμός των φυτών της κατηγορίας i και $P(w_i)$ η a - priori πιθανότητα της κατηγορίας i . Στα παρακάτω γραφήματα 1(α) και 2(α) βλέπουμε τα αποτελέσματα επιτυχίας του ταξινομητή για μη - κανονικοποιημένα και κανονικοποιημένα δεδομένα αντίστοιχα.



Σχήμα 1.1: *k*-NN classification of Iris Database

Παρατηρείται ότι το ποσοστό ορθής ταξινόμησης για μη - κανονικοποιημένα δεδομένα είναι από 96 μέχρι 96.4, ποσοστό αρκετά μεγάλο και σταθερό για όλους τους αριθμούς των γειτόνων που έχει δοκιμαστεί. Παρατηρείται ότι έχουμε μικρή βελτίωση όταν ο αριθμός των γειτόνων κυμαίνεται από 6 μέχρι 8 γείτονες.

Παρατηρείται ότι το ποσοστό ορθής ταξινόμησης για κανονικοποιημένα δεδομένα είναι επίσης υψηλό, καθώς κυμαίνεται από 94.2 μέχρι 94.8, παραμένοντας όμως λίγο χαμηλότερο από το ποσοστό επιτυχίας των μη - κανονικοποιημένων δεδομένων. Μπορούμε να πούμε ότι τα δεδομένα του Iris Database είναι ιδανικά για το συγκεκριμένο είδος ταξινόμησης και δεν χρειάζεται να κανονικοποιηθούν. Η ίδια διαδικασία ακολουθείται και για το Pima Indian Database. Παρακάτω παρατίθενται τα γραφήματα 3(α) και 4(α) που περιγράφουν τα αποτελέσματα του KNN ταξινομητή για το συγκεκριμένο database.



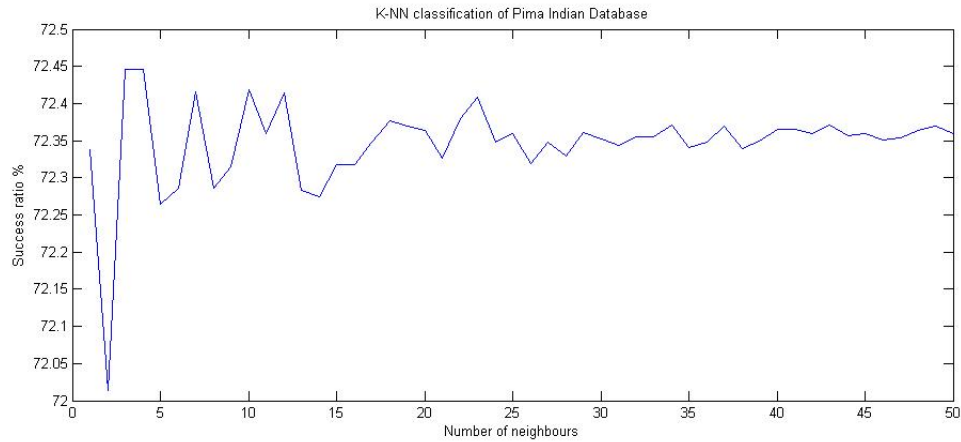
Σχήμα 1.2: *k*-NN classification of Iris Database with normalized data

Παρατηρείται ότι το ποσοστό ορθής ταξινόμησης για μη - κανονικοποιημένα δεδομένα είναι από 72 μέχρι 72.45, με μέση τιμή το 72.35. Βλέπουμε ότι όταν ο αριθμός των γειτόνων είναι μικρός, παρατηρούνται κάποιες μικρές διακυμάνσεις στο ποσοστό επιτυχίας, ενώ όταν οι γείτονες γίνουν 20, το ποσοστό ορθής πρόβλεψης σταθεροποιείται.

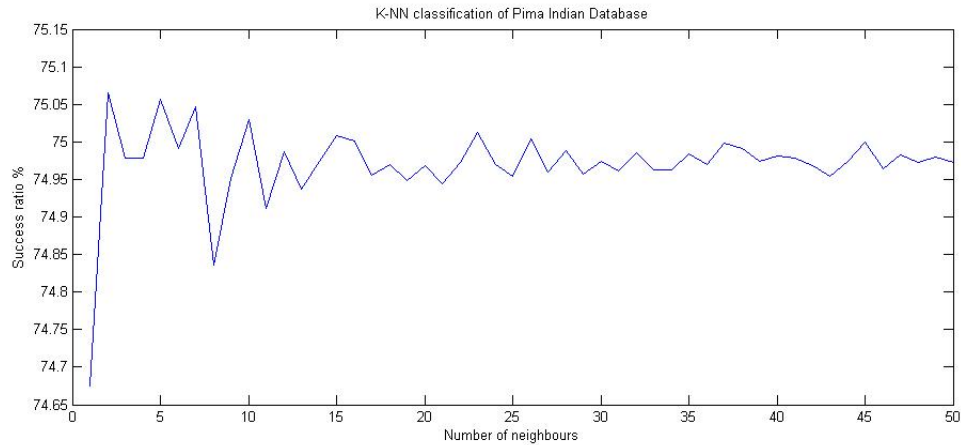
Παρατηρείται ότι το ποσοστό ορθής ταξινόμησης για κανονικοποιημένα δεδομένα είναι από 74.6 μέχρι 75, με μέση τιμή το 75. Εμφανίζεται μικρή βελτίωση στα ποσοστά ορθής πρόβλεψης σε σχέση με τα μη κανονικοποιημένα δεδομένα. Φαίνεται ότι για την Pima Indian Database, η κανονικοποίηση των δεδομένων ήταν απαραίτητη.

- (β') **Για το δεύτερο πρόβλημα, σχεδιάστε έναν ταξινομητή Bayes με υποκείμενες κανονικές κατανομές για τις πυκνότητες πιθανότητας των 2 κατηγοριών (με δικές σας παραδοχές για τους πίνακες συνδιασπορών, βασισμένες στα δεδομένα) και συγκρίνετε την επίδοσή του με αυτή του ταξινομητή k-NN.**

Στο δεύτερο υποερώτημα υλοποιήσαμε ένα ταξινομητή Bayes πάνω στο Pima Indian database με σκοπό να συγκρίνουμε τις επιδόσεις του με τον K-NN ταξινομητή του προηγούμενου υποερωτήματος. Στον φάκελο Bayesian περιέχεται το εκτελέσιμο αρχείο Matlab Bayesian.m το οποίο περιέχει την υλοποίηση του ταξινομητή Bayes για το dataset που ζητείται. Τα δεδομένα του dataset χωρίζονται σε 2 μέρη: (α) Έναν πίνακα διαστάσεων 770×8 , όπου οι στήλες εκφράζουν τα 8 φεαυρες του κάθε ινδιάνου, και οι γραμμές τον κάθε ινδιάνο και (β) Ένα vector 770×1 που εκφράζει την κατηγορία στην οποία έχει ταξινομηθεί ο κάθε ινδιάνος.



Σχήμα 1.3: *k*-NN classification of Pima Indian Database



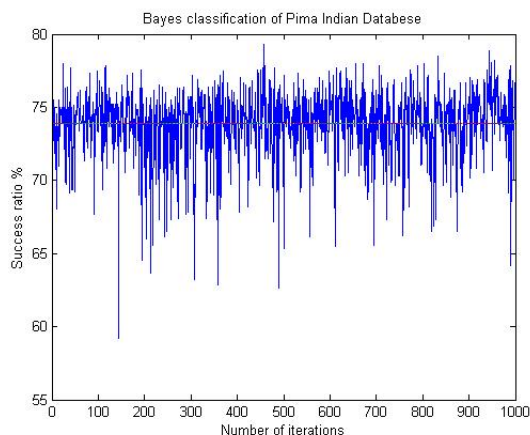
Σχήμα 1.4: *k*-NN classification of Pima Indian Database with normalized data

Εκτελέσαμε το script και πάλι αρκετές φορές ώστε να έχουμε αντιπροσωπευτικά αποτελέσματα, και το training και validation του ταξινομητή έγινε με την μέθοδο tenfold validation που έχει περιγραφεί παραπάνω. Χωρίσαμε τον πίνακα των χαρακτηριστικών μας σε 2 πίνακες, έναν για κάθε κατηγορία. Υπολογίσαμε τους μέσους του κάθε χαρακτηριστικού, της κάθε κατηγορίας με την συνάρτηση της Matlab `mean()`, και υπολογίσαμε επίσης του πίνακες συνδιασποράς της κάθε κατηγορίας με την συνάρτηση της Matlab `cov()`. Αυτό έγινε για να αποκτήσουμε τις παραμέτρους των κανονικών κατανομών που περιγράφουν την κάθε κατηγορία. Για sample του validation fold, υπολογίζω την posterior πιθανότητα του ως εξής:

$$P(\omega_i|X) = \frac{P(X, \omega_i) \times P(\omega_i)}{P(X)}$$

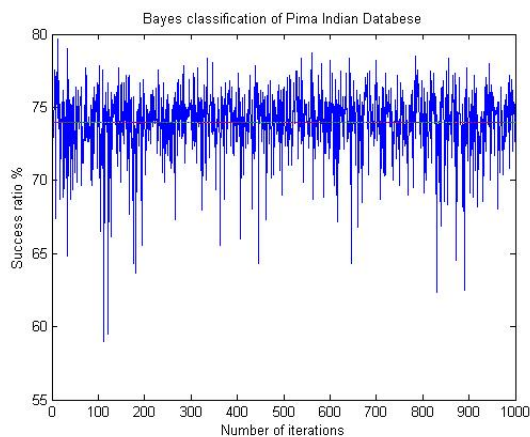
Όπου το $P(X, \omega_i)$ υπολογίζεται από την multivariate gaussian κατανομή της κάθε κατηγορίας

με X το sample του validation fold που εξετάζεται και μέση τιμή και πίνακα συνδιασποράς που έχουν υπολογιστεί όπως περιγράφεται παραπάνω, με την συνάρτηση της Matlab `mnpdf()`. Το $P(w_i)$ είναι η a - priori πιθανότητα της κάθε κατηγορίας που υπολογίζεται εύκολα από τα αντίστοιχα training set. Το $P(X)$ δεν χρειάζεται να υπολογιστεί καθώς απαλείφεται στις παρακάτω συγκρίσεις. Τελικά το sample του validation θα ταξινομηθεί στην κατηγορία από την οποία προκύπτει η μεγαλύτερη posterior probability. Παρακάτω παρατίθενται τα γραφήματα 1(β) και 2(β) που περιγράφουν τα γραφήματα του ταξινομητή Bayes για μη - κανονικοποιημένα και κανονικοποιημένα δεδομένα αντίστοιχα.



Σχήμα 1.5: *Bayesian classification of Pima Indian Database*

Παρατηρείται ότι το ποσοστό ορθής ταξινόμησης για μη κανονικοποιημένα δεδομένα είναι από 72 μέχρι 76, με μέση τιμή το 74, όπως φαίνεται από την κόκκινη γραμμή. Σε σχέση με τον ταξινομητή K - NN για μη κανονικοποιημένα δεδομένα, παρατηρούμε μια ελαφρά καλύτερη επίδοση ως προς την ορθή ταξινόμηση.



Σχήμα 1.6: *bayesian classification of Pima Indian Database with normalized data*

Παρατηρείται ότι το ποσοστό ορθής ταξινόμησης για κανονικοποιημένα δεδομένα είναι από 72 μέχρι 76, με μέση τιμή το 74, όπως φαίνεται από την κόκκινη γραμμή. Σε σχέση με τον ταξινομητή K-NN για κανονικοποιημένα δεδομένα, παρατηρούμε ότι οι επιδόσεις είναι παρόμοιες, με τον K-NN να είναι ελαφρά καλύτερος. Επίσης βλέπουμε ότι δεν παρατηρείται κάποια αλλαγή στην επίδοση του ταξινομητή όσον αφορά κανονικοποιημένα ή μη κανονικοποιημένα δεδομένα, καθώς και στις 2 περιπτώσεις η μέση τιμή είναι ίση με 74.

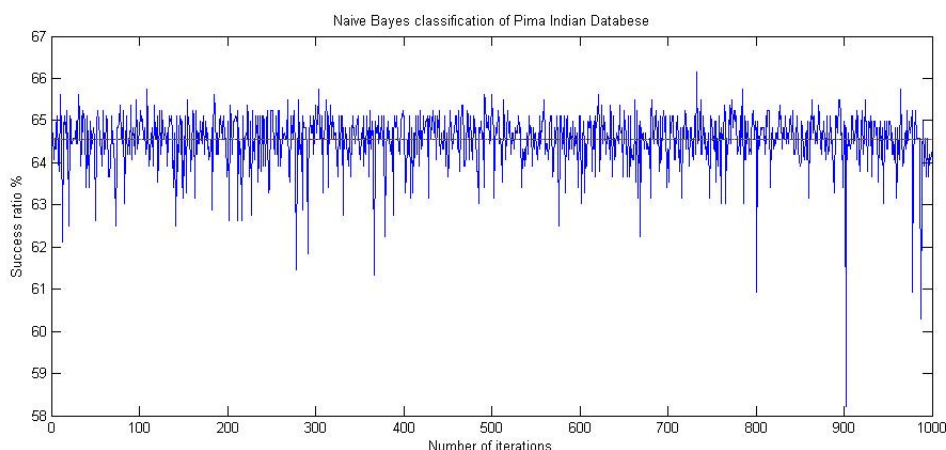
(γ') Τέλος, να υλοποιήσετε έναν απλοϊκό ταξινομητή Bayes για το δεύτερο πρόβλημα, να αποτιμήσετε την επίδοσή του και να τη συγκρίνετε με τους προηγούμενους ταξινομητές.

Στο τρίτο υποερώτημα υλοποιήσαμε ένα ταξινομητή Naive Bayes πάνω στο Pima Indian database με σκοπό να συγκρίνουμε τις επιδόσεις του με τον K-NN ταξινομητή όπως κάναμε και στο προηγούμενο υποερώτημα. Στον φάκελο Naive περιέχεται το εκτελέσιμο αρχείο Matlab Naive.m το οποίο περιέχει την υλοποίηση του ταξινομητή Naive Bayes για το dataset που ζητείται. Τα δεδομένα του dataset χωρίζονται όπως στο προηγούμενο υποερώτημα.

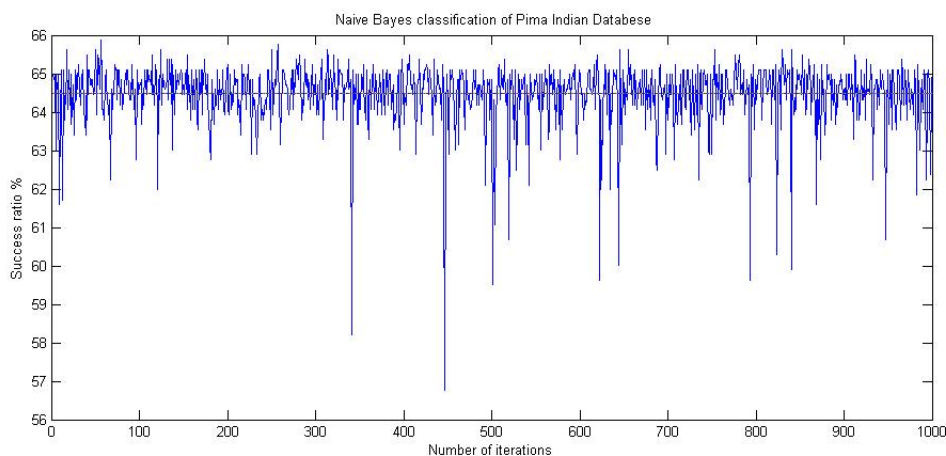
Εκτελέσαμε το script και πάλι τόσες φορές ώστε να έχουμε αντιπροσωπευτικά αποτελέσματα, και το training και validation του ταξινομητή έγινε με την μέθοδο tenfold validation που έχει περιγραφεί παραπάνω. Χωρίσαμε τον πίνακα των χαρακτηριστικών όπως προηγούμενα. Υπολογίσαμε τους μέσους και την τυπική απόκλιση του κάθε χαρακτηριστικού για την κάθε κατηγορία με την συνάρτηση της Matlab mean() και std(). Έτσι αποκτήσαμε τις παραμέτρους των κανονικών κατανομών που περιγράφουν το κάθε χαρακτηριστικό - στήλη, της κάθε κατηγορίας. Για κάθε sample του validation fold, υπολογίζω την a posterior πιθανότητα με παρόμοιο τρόπο με το προηγούμενο υποερώτημα, χρησιμοποιώντας αυτήν την φορά την τυπική απόκλιση αντί για τον πίνακα συνδιασπορών. Παρακάτω παρατίθενται τα γραφήματα 1(γ) και 2(γ) που περιγράφουν τα γραφήματα του ταξινομητή Naive Bayes για μη-κανονικοποιημένα και κανονικοποιημένα δεδομένα αντίστοιχα.

Παρατηρείται ότι το ποσοστό ορθής ταξινόμησης για μη κανονικοποιημένα δεδομένα κυμαίνεται από 64 μέχρι 65, με μέση τιμή το 64,5, όπως φαίνεται από την κόκκινη γραμμή. Σε σχέση με τον ταξινομητή K-NN για μη κανονικοποιημένα δεδομένα, παρατηρούμε ότι η επίδοση του ταξινομητή Naive Bayes ως προς την ορθή ταξινόμηση είναι χειρότερη. Το ίδιο ισχύει και για την σύγκριση με τον ταξινομητή Bayes.

Παρατηρείται ότι το ποσοστό ορθής ταξινόμησης για κανονικοποιημένα δεδομένα κυμαίνεται από 64 μέχρι 65, με μέση τιμή το 64,5, όπως φαίνεται από την κόκκινη γραμμή. Σε σχέση με τον ταξινομητή K-NN για κανονικοποιημένα δεδομένα, παρατηρούμε ξανά ότι η επίδοση του ταξινομητή Naive Bayes ως προς την ορθή ταξινόμηση είναι χειρότερη. Τα ίδια ισχύουν και για τον ταξινομητή Bayes. Όσον αφορά κανονικοποιημένα ή μη κανονικοποιημένα δεδομένα, δεν παρατηρείται κάποια αλλαγή στην επίδοση του ταξινομητή, καθώς και στις 2 περιπτώσεις η μέση τιμή είναι ίση με 64,5.



Σχήμα 1.7: *naive bayesian classification of Pima Indian Database*



Σχήμα 1.8: *naive bayesian classification of Pima Indian Database with normalized data*

1.2 Θέμα 2

- (α') **Ξεκινώντας με αναζήτηση στο internet, ή με άλλο τρόπο, σχεδιάστε ένα δίκτυο Bayes που να αντιστοιχεί σε ένα πρόβλημα της αρεσκείας σας. Το δίκτυο θα πρέπει να είναι σχετικά απλό (της τάξεως των 5-10 κόμβων) και τουλάχιστον ένας κόμβος θα πρέπει να έχει 2 τουλάχιστον γονείς. Αποδώστε a priori πιθανότητες στους κόμβους και τους συνδέσμους του σχετικού γραφήματος.**

Θεωρήστε έναν πρωτογενή όγκο με μία αβέβαιη πρόγνωση σε έναν τυχαίο ασθενή. Είναι γνωστό ότι ο καρκίνος μπορεί να εξαπλωθεί στον εγκέφαλο² αλλά και σε άλλα σημεία του

²http://en.wikipedia.org/wiki/Brain_metastasis

σώματος³. Στο πρόβλημα που επιλέξαμε να παρουσιάσουμε ενδιαφερόμαστε για την πορεία εξάπλωσης του καρκίνου για την χρονική περίοδο τριών χρόνων μετά την διάγνωσή του, και ειδικότερα σε σχέση με την ανάπτυξη/εμφάνιση του εγκεφαλικού όγκου και των συναφών προβλημάτων του [1], [2].

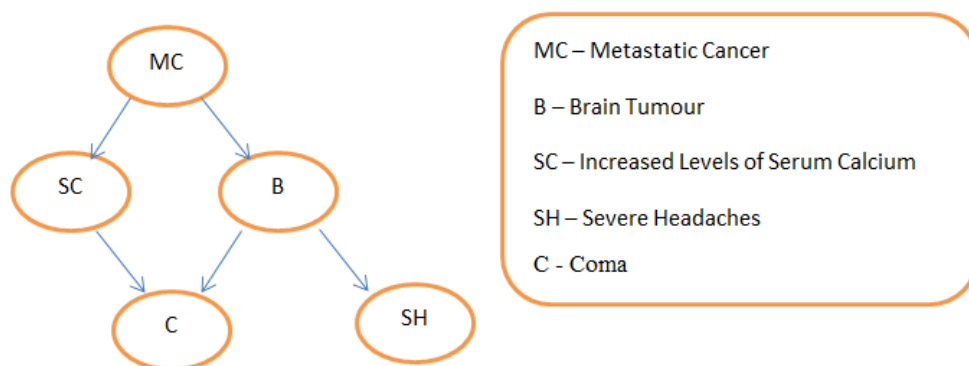
Σύμφωνα με ιατρικές πηγές[3] που βρήκαμε στο διαδίκτυο ^{4,5}:

- ◇ η πιθανότητα ο μεταστατικός καρκίνος να αναπτυχθεί από τον αρχικό όγκο του εγκεφάλου είναι 20%, ενώ ταυτόχρονα η πιθανότητα ο μεταστατικός καρκίνος να αναπτυχθεί στον εγκέφαλο είναι και πάλι 20%.
- ◇ Επιπρόσθετα, ακόμα και στην απουσία του μεταστατικού καρκίνου, υπάρχει μία μικρή πιθανότητα της τάξεως του 5% να αναπτυχθεί ένας (πρωτογενής) όγκος στο εγκέφαλο.
- ◇ Ο μεταστατικός καρκίνος μπορεί να ανιχνευτεί αν διαπιστωθούν αυξημένα επίπεδα ασβεστίου του ορού. Πράγματι, το ασθέσιο του ορού μπορεί να αυξηθεί με πιθανότητα 80% σε περίπτωση παρουσίας ενός μεταστατικού καρκίνου και μόνο με πιθανότητα 20% σε περίπτωση απουσίας του.
- ◇ Υπολογίζεται ότι ένας ασθενής μπορεί να πέσει σε κώμα μέσα στα επόμενα τρία χρόνια με πιθανότητα 80% όταν αυτός έχει προσβληθεί από κάποιο είδος καρκίνου ή/και όταν τα επίπεδα του ασβεστίου του ορού είναι ανεβασμένα (υπερασβεστιαϊμία). Από την άλλη, αν δεν υπάρχει κανένα από αυτά τα δύο συμπτώματα, υπάρχει μόνο μία μικρή πιθανότητα της τάξεως του 5% ο ασθενής να πέσει σε κώμα.
- ◇ Σοβαρές κεφαλαλγίες είναι πιθανό να αναπτυχθούν και όταν υπάρχει όγκος στον εγκέφαλο, με πιθανότητα 80%, αλλά και όταν δεν υπάρχει, με πιθανότητα 60%.

Θα παρουσιάσουμε το πρόβλημα αυτό με χρήση των Bayesian Networks και εφαρμογή τους μέσω του BNT "Bayesian Nets Toolkit" της Matlab και του προγράμματος SamIam που αναπτύχθηκε από το UCLA. Στο πρόβλημά μας έχουμε 5 αναγνωρισμένους κόμβους σημεία:

- Α' Την ύπαρξη μεταστατικού καρκίνου (MC),
- Β' Την ύπαρξη εγκεφαλικού όγκου (B),
- Γ' Την αύξηση των επιπέδων του ασβεστίου του ορού (SC),
- Δ' Το ενδεχόμενο ο ασθενής να πέσει σε κώμα μέσα στα επόμενα 3 χρόνια (C) και
- Ε' Την παρουσία σοβαρών κεφαλαλγιών (πονοκεφάλων) (SH).

Οι σχέσεις μεταξύ των παραπάνω κόμβων και οι επιμέρους πιθανότητες φαίνονται στο Σχήμα [αριτημοσ], το οποίο διαμορφώθηκε με το πρόγραμμα samIam του UCLA ⁶ και παρατίθεται στο παραδοτέο της άσκηση μαζί με ένα readme , μερικές εικόνες και το αρχείο από το οποίο προέκυψαν τα αποτελέσματα και από το οποίο μπορούμε διαδραστικά να δούμε όλες τις πιθανότητες για κάθε περίπτωση.

Σχήμα 1.9: *Metastatic Cancer Bayesian Network*

Στο παρόν πρόβλημα, η μόνη ενέργεια που αναφέρεται ρητά είναι η μέτρηση του ασβεστίου του ορού. Παρόλα αυτά, υπάρχουν σιωπηρές έμμεσες "παρενέργειες", ήτοι "παρενέργειες" στα δεδομένα που παρουσιάζονται εξαιτίας κάποιων θεραπευτικών επιλογών. Μετά από μία εγχείρηση, οι πιθανότητες όλων των κόμβων αλλάζουν, και η εξάρτηση των πονοκεφάλων και του κώματος δεδομένου του μεταστατικού καρκίνου αλλάζει και αυτή. Αυτές οι αλλαγές στις πιθανότητες εξάρτησης θα μπορούσαν να είναι διαφορετικές έπειτα από κάποια χημειοθεραπεία ή έπειτα από θεραπείες με ακτινοβολία. Οπότε, δεδομένων και των παραπάνω ο επιδιωκόμενος στόχος σε αυτό το πρόβλημα είναι αυτός στο οποίο ο ασθενής δεν έχει μεταστατικό καρκίνο, έχει ένα μειωμένο αριθμό σοβαρών κεφαλαλγιών και δεν είναι σε κώμα (ή νεκρός).

Συνεπώς για να επιτευχθεί αυτός ο στόχος χρειάζονται σύνθετοι αλγόριθμοι που λαμβάνουν υπόψη τους όλο το ιστορικό του συστήματος, που στην περίπτωσή μας είναι ο ανθρώπινος οργανισμός. Επομένως τα Bayesian δίκτυα είναι ένας καλός τρόπος για να υπολογίζονται τέτοιες καταστάσεις μιας και η υπολογιστική πολυπλοκότητα των υπολογισμών είναι πολυωνυμικής τάξης μεγέθους ως προς το πλήθος των κόμβων και τη σωστή διάταξή τους, αν και γενικότερα τα BNs ανήκουν στα NP-hard προβλήματα⁹.

- (β') Στη διεύθυνση: <http://bnt.googlecode.com> θα βρείτε το "Bayesian Nets Toolkit" (BNT). Κατεβάστε το, και διαβάστε τις οδηγίες χρήσης και το βασικό παράδειγμα κατασκευής ενός απλού δικτύου και των τρόπων με τους οποίους μπορούμε να κάνουμε συμπερασμό (inference) πιθανοτήτων σ' αυτό.

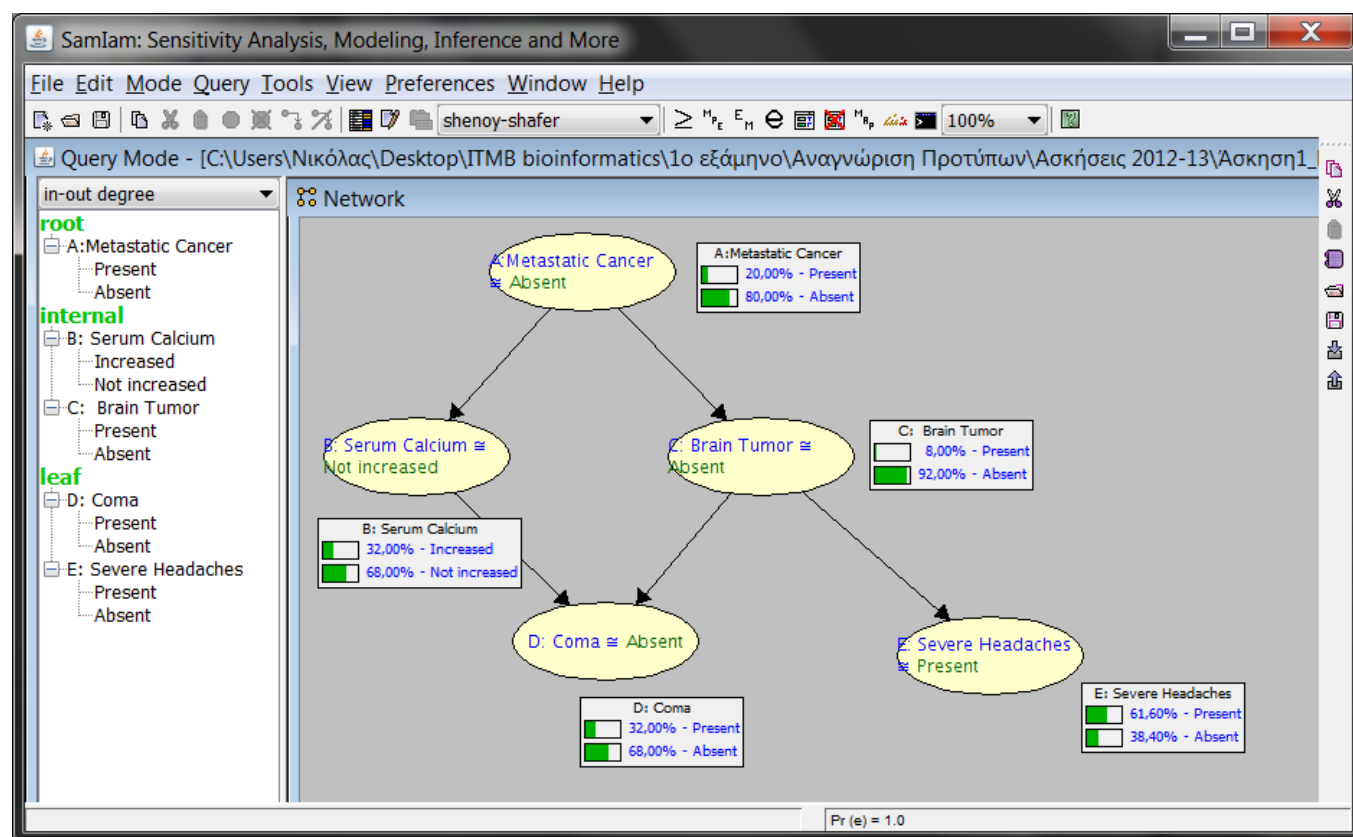
³<http://en.wikipedia.org/wiki/Metastasis>

⁴<http://www.nlm.nih.gov/medlineplus/ency/article/000769.htm>

⁵<http://www.cancer.gov/cancertopics/factsheet/Sites-Types/metastatic>

⁶<http://reasoning.cs.ucla.edu/samiam/>

⁹<http://users.cecs.anu.edu.au/jinbo/pr/lecture05.pdf>



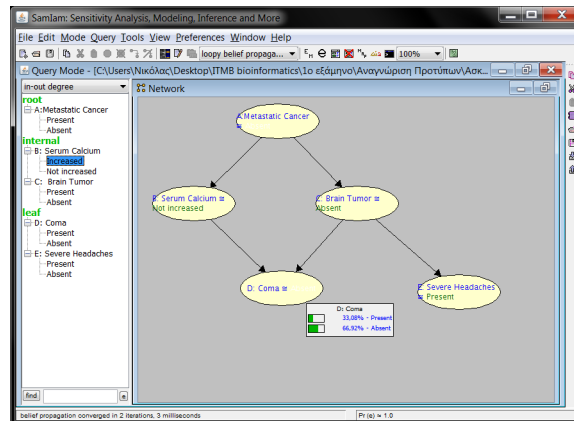
Σχήμα 1.10: Metastatic Cancer Bayesian Network, όπως φαίνεται από το πρόγραμμα samIAM που αναπτύχθηκε στο UCLA⁸

(γ) Στη συνέχεια να υλοποιήσετε με τη βοήθεια του BNT το δίκτυο που σχεδιάσατε στο (α). Να δώσετε παραδείγματα συμπερασμού που να χρησιμοποιούν τόσο ευθεία όσο και ανάδρομη διάδοση πληροφορίας και να χρησιμοποιήσετε το BNT για να υπολογίσετε τις αντίστοιχες δεσμευμένες πιθανότητες.

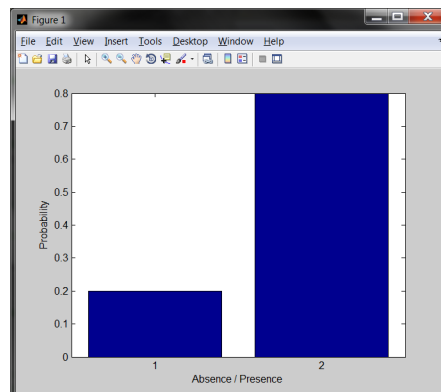
Ακολουθώντας τις οδηγίες της εκφώνησης κατεβάσαμε το "Bayesian Nets Toolkit" (BNT), και κάναμε συμπερασμό (inference) με βάση αυτό στο πρόβλημα της μετάστασης του καρκίνου. Τα αποτελέσματα που πήραμε από τη χρήση του BNT τα χρησιμοποιήσαμε τόσο σε ευθεία όσο και σε ανάδρομη διάδοση πληροφορίας και υπολογίσαμε τις αντίστοιχες δεσμευμένες πιθανότητες. Ο κώδικας Matlab που συντάξαμε παρατίθεται στο επόμενο κεφάλαιο που περιέχει τις υλοποιήσεις 2.2 αυτές.

Το παραδείγματα που επιλέξαμε να σας παρουσιάσουμε για ευθεία διάδοση πληροφορίας είναι ο υπολογισμός της πιθανότητας $p=P(C=2 \mid SC=2)$, δηλαδή ο υπολογισμός της πιθανότητας δεδομένης της αύξησης του ασβεστίου του ορού (υπερασβεστιαϊμία) να πέσει ο ασθενής σε κώμα μέσα στα πρώτα τρία χρόνια, και ο υπολογισμός της πιθανότητας $p=P(C=2 \mid MC=2)$,

Στα σχήματα Σχήμα 1.11 - 1.17 φαίνεται ότι η πρώτη πιθανότητα είναι $p=P(C=2 | SC=2)=0.8$. ενώ η δεύτερη πιθανότητα είναι $p=P(C=2 | MC=2)=0.68$.



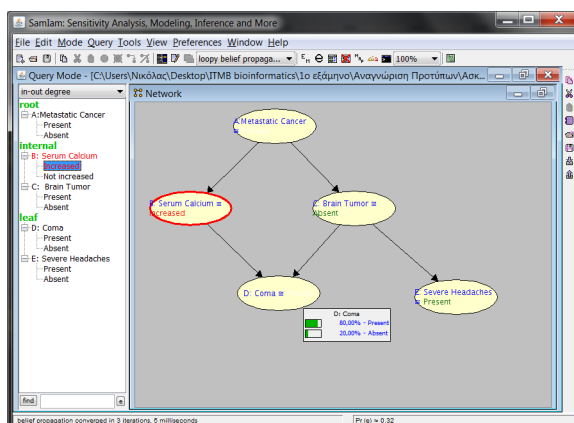
Σχήμα 1.11: *Direct Probability Distribution: Probability of C*



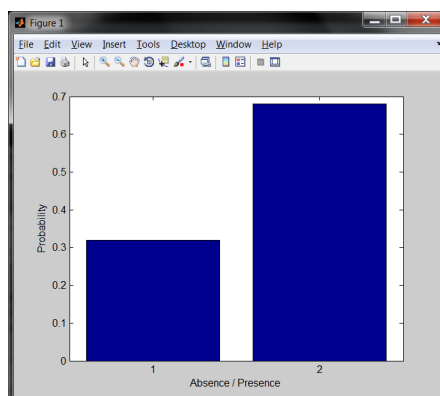
Σχήμα 1.12: *Direct Probability Distribution: Probability of C given SC as increased. Matlab bar diagram*

Name	Value	Min	Max
bnet	<1x1 struct>	4	4
c	<5x5 double>	0	1
dag	[1,2,3,4,5]	1	5
discrete_nodes	<1x1 jtree_inf_en...>	-1.13..	-1.13..
evidence	<1x5 cell>	1	1
loglik	<1x1 struct>	2	2
marg	[]	0.80..	0.80..
mc	0.6000	2	2
node_sizes	[2,2,2,2,2]	5	5
onodes	[]		
sc	2		
sh	5		

Σχήμα 1.13: *Direct Probability Distribution: Probability of C given SC as increased. Matlab probability output*



Σχήμα 1.14: *Direct Probability Distribution: Probability of C given SC as increased. SamIam*

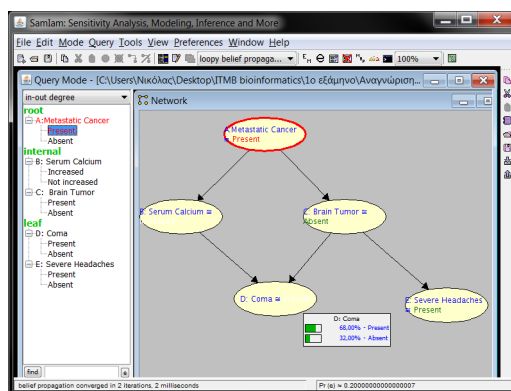


Σχήμα 1.15: *Direct Probability Distribution: Probability of C given MC as increased. Matlab bar diagram*

Name	Value	Min	Max
bnet	<1x1 struct>		
c	4	4	4
dag	<5x5 double>	0	1
discrete_nodes	[1,2,3,4,5]	1	5
engine	<1x1 jitree_inf_en...>		
evidence	<1x5 cell>		
loglik	-1.6094	-1.60	-1.6
marg	<1x1 struct>		
mc	1	1	1
node_sizes	[2,2,2,2,2]	2	2
onodes	[]		
p	0.6800	0.68	0.68
sc	2	2	2
sh	5	5	5

Σχήμα 1.16: *Direct Probability Distribution: Probability of C given MC as increased. Matlab probability output*

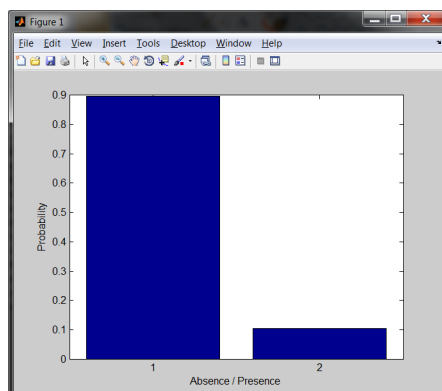
Ενώ ως παραδείγματα συμπερασμού που επιλέξαμε για ανάδρομη διάδοση πληροφορίας επιλέξαμε τον υπολογισμό της πιθανότητας $p=P(B=2|SH=2)$, δηλαδή ο υπολογισμός της πιθανότητας δεδομένων των σοβαρών κεφαλαλγιών να οφείλονται αυτές σε εγκεφαλικό όγκο. Δείτε τα σχήματα Σχήμα 1.18 - 1.20, όπου φαίνεται ότι αυτή η πιθανότητα είναι



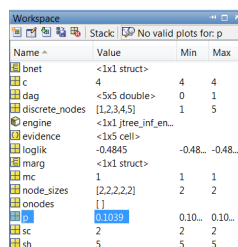
Σχήμα 1.17: *Direct Probability Distribution: Probability of C given MC as increased. SamIam*

$$p=P(B=2 \mid SH=2)=0.1039.$$

Επίσης επιλέξαμε και τον υπολογισμό της πιθανότητας $p=P(MC=2 \mid C=2)$, δηλαδή την πιθανότητα δεδομένου του κώματος στο οποίο έχει πέσει ο ασθενής, αυτό να οφείλεται στην ανάπτυξη του μεταστατικού καρκίνου. Δείτε τα σχήματα Σχήμα 1.21 - 1.23, όπου φαίνεται ότι αυτή η πιθανότητα είναι $p=P(MC=2 \mid C=2)=0.4250$.

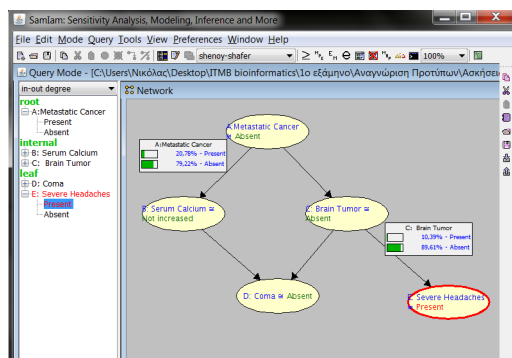


Σχήμα 1.18: *Retrograded Probability Distribution: Probability of B given SH as present. Matlab bar diagram*

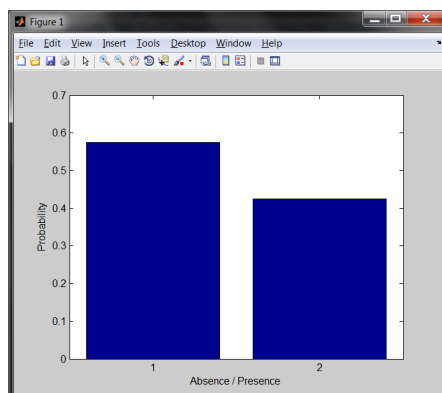


Name	Value	Min	Max
bnet	<1x1 struct>		
c	4	4	4
dag	<5x5 double>	0	1
discrete_nodes	[1,2,3,4,5]	1	5
engine	<1x1 jtree_inf.en...		
evidence	<1x5 cell>		
loglik	-0.4845	-0.48...	-0.48...
marg	<1x1 struct>		
mc	1	1	1
node_sizes	[2,2,2,2,2]	2	2
onodes	[]		
p	0.1039	0.10...	0.10...
sc	2	2	2
sh	5	5	5

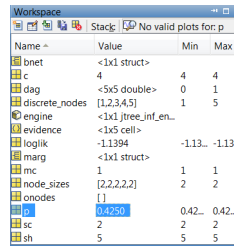
Σχήμα 1.19: *Retrograded Probability Distribution: Probability of B given SH as present. Matlab probability output*



Σχήμα 1.20: *Retrograded Probability Distribution: Probability of B given SH as present. samIam*

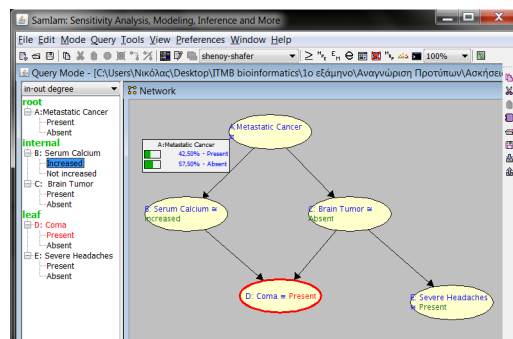


Σχήμα 1.21: *Retrograded Probability Distribution: Probability of MC given C as present. Matlab bar diagram*



Name	Value	Min	Max
bnet	<1x1 struct>		
c	4	4	4
dag	<5x5 double>	0	1
discrete_nodes	[1,2,3,4,5]	1	5
engine	<1x1 jtree_inf_en...>		
evidence	<1x5 cell>		
loglik	-1.1394	-1.13...	-1.13...
margin	<1x1 struct>		
mc	1	1	1
node_sizes	[2,2,2,2,2]	2	2
onodes	[]		
p	0.4250	0.42...	0.42...
sc	2	2	2
sh	5	5	5

Σχήμα 1.22: *Retrograded Probability Distribution: Probability of MC given C as present. Matlab probability output*



Σχήμα 1.23: *Retrograded Probability Distribution: Probability of MC given C as present. samlarn*

- (δ') **Τέλος, για ένα τουλάχιστον παράδειγμα συμπερασμού με ευθεία διάδοση πληροφορίας και για ένα τουλάχιστον παράδειγμα με ανάδρομη διάδοση πληροφορίας, να υποδείξετε εσείς ποιες πράξεις πιστεύετε ότι έκανε το BNT για να καταλήξει στα συμπεράσματά του.**

Από τις πιθανότητες που φαίνονται αναλυτικά στο Σχήμα. 1.24 μπορούμε να εξαγάγουμε μέσω υπολογισμών ότι η υπό συνθήκη πιθανότητα ευθείας διάδοσης πληροφορίας $p=P(C=2 | MC=2) = 0.68$. και η υπό συνθήκη πιθανότητα ανάδρομης διάδοσης πληροφορίας $p=P(MC=2 | C=2) = 0.4250$. Ακολουθούν οι υπολογισμοί για καθε μία από αυτές τις πιθανότητες: Ξεκινώντας από την ευθεία διάδοση πληροφορίας έχουμε: Από τον τύπο:

$$P(C|MC) = \frac{P(C, MC)}{P(MC)}$$

Υπολογίζουμε την από κοινού πιθανότητα:

$$P(C, MC) = P(C, SC, B, MC) + P(C, SC', B, MC) + P(C, SC, B', MC) + P(C, SC', B', MC) =$$

$$\begin{aligned}
 &P(C|SC, B)P(SC|MC)P(B|MC)P(MC) + P(C|SC', B)P(SC'|MC)P(B|MC)P(MC) + \\
 &P(C|SC, B')P(SC|MC)P(B'|MC)P(MC) + P(C|SC', B')P(SC'|MC)P(B'|MC)P(MC) = \\
 &\quad (0.8 \cdot 0.8 \cdot 0.2 \cdot 0.2) + (0.8 \cdot 0.2 \cdot 0.2 \cdot 0.2) + \\
 &\quad (0.8 \cdot 0.8 \cdot 0.8 \cdot 0.2) + (0.05 \cdot 0.2 \cdot 0.8 \cdot 0.2) = 0.136
 \end{aligned}$$

Και άρα τώρα έχουμε :

$$P(C|MC) = \frac{P(C, MC)}{P(MC)} \Rightarrow P(C|MC) = \frac{0.136}{0.2} = 0.68$$

Οπότε και βρέθηκε η υπό συνθήκη πιθανότητα ευθείας διάδοσης.

Τώρα για να βρούμε την πιθανότητα ανάδρομης διάδοσης

$$P(MC|C)$$

χρησιμοποιώντας τον τύπο του Bayes και την προηγούμενη υπο συνθήκη πιθανότητα που βρήκαμε έχουμε τον τύπο :

$$P(MC|C) = \frac{P(C|MC) \cdot P(MC)}{P(C)}$$

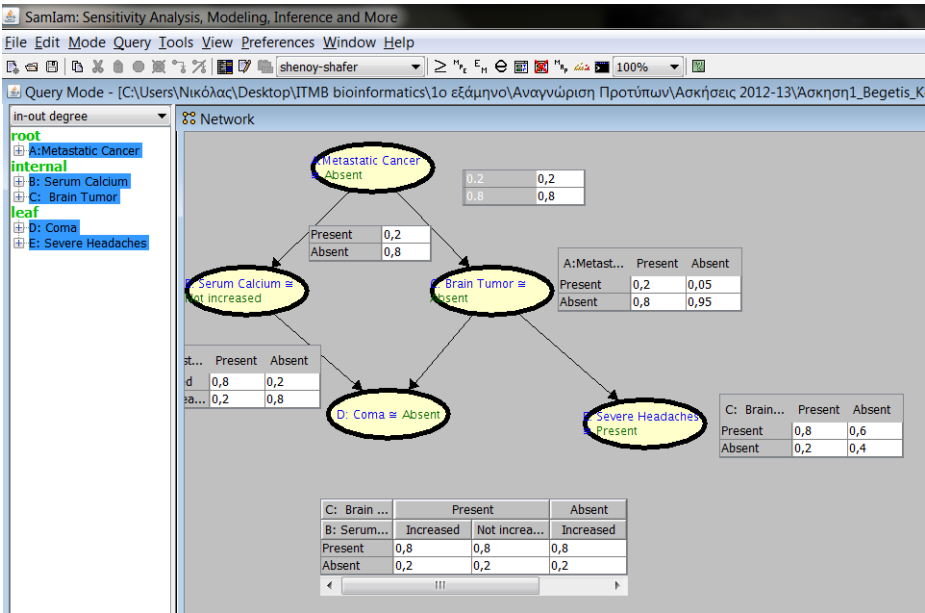
Τώρα υπολογίζουμε την ολική πιθανότητα για να επιτύχει το ενδεχόμενο Σ και έχουμε :

$$\begin{aligned}
 P(C) &= P(C, SC, B, MC) + P(C, SC, B, MC') + P(C, SC, B', MC) + P(C, SC', B, MC) + \\
 &\quad P(C, SC, B', MC') + P(C, SC', B, MC') + P(C, SC', B', MC) + P(C, SC', B', MC') = \\
 &P(C|SC, B)P(SC|MC)P(B|MC)P(MC) + P(C|SC, B)P(SC|MC')P(B|MC')P(MC') + \\
 &P(C|SC', B)P(SC'|MC)P(B|MC)P(MC) + P(C|SC', B)P(SC'|MC')P(B|MC')P(MC') + \\
 &P(C|SC, B')P(SC|MC)P(B'|MC)P(MC) + P(C|SC, B')P(SC|MC')P(B'|MC')P(MC') + \\
 &P(C|SC', B')P(SC'|MC)P(B'|MC)P(MC) + P(C|SC', B')P(SC'|MC')P(B'|MC')P(MC') = \\
 &\quad (0.8 \cdot 0.8 \cdot 0.2 \cdot 0.2) + (0.8 \cdot 0.2 \cdot 0.05 \cdot 0.8) + (0.8 \cdot 0.2 \cdot 0.2 \cdot 0.2) + (0.8 \cdot 0.2 \cdot 0.05 \cdot 0.8) + \\
 &\quad (0.8 \cdot 0.8 \cdot 0.8 \cdot 0.2) + (0.8 \cdot 0.2 \cdot 0.95 \cdot 0.8) + (0.05 \cdot 0.2 \cdot 0.8 \cdot 0.2) + (0.05 \cdot 0.8 \cdot 0.95 \cdot 0.8) = 0.3008
 \end{aligned}$$

Και οπότε τώρα έχουμε :

$$P(MC|C) = \frac{P(C|MC) \cdot P(MC)}{P(C)} \Rightarrow P(MC|C) = \frac{0.68 \cdot 0.2}{0.3008} = 0.452127$$

το οποίο, όπως και το προηγούμενο, συμβαδίζει με τα αποτελέσματα που πήραμε από το MatLab και το samlam και άρα επαληθεύτηκε ότι είναι σωστό.



Σχήμα 1.24: Conditional Probability Table monitor for Metastatic Cancer

ΚΕΦΑΛΑΙΟ 2

Υλοποιήσεις σε Matlab

Στο παρόν Κεφάλαιο παραθέτουμε τις υλοποιήσεις σε κώδικα Matlab για το κάθε θέμα της πρώτης πρακτικής άσκησης στην Αναγνώριση Προτύπων.

2.1 Θέμα 1

- (α) Με χρήση του Matlab ή με άλλο τρόπο να υλοποιήσετε έναν ταξινομητή πλησιέστερων γειτόνων (k-NN). Να τον χρησιμοποιήσετε για να επιλύσετε τα εξής γνωστά από τη βιβλιογραφία προβλήματα :

1 IRIS PLANT DATABASE (ταξινόμηση φυτών Iris σε τρία είδη).

2 PIMA INDIANS DIABETES DATABASE (ταξινόμηση εγγύων ινδιάνων της φυλής Pima σε άτομα που έχουν ή δεν έχουν διαβήτη).

Τα δεδομένα των προβλημάτων θα τα βρείτε στο αρχείο UCIdata-exercise1.rar. Όπου κρίνετε σκόπιμο, κανονικοποιήστε τα δεδομένα ανά στήλη πριν εφαρμόσετε τον ταξινομητή. Να μελετήσετε το ποσοστό ορθής ταξινόμησης ως συνάρτηση του αριθμού των πλησιέστερων γειτόνων. Η εκτίμηση του αποτελέσματος να γίνει με χρήση της μεθόδου **tenfold crossvalidation**.

- (β) Για το δεύτερο πρόβλημα, σχεδιάστε έναν ταξινομητή Bayes με υποκείμενες κανονικές κατανομές για τις πυκνότητες πιθανότητας των 2 κατηγοριών (με δικές σας παραδοχές για τους πίνακες συνδιασπορών, βασισμένες στα δεδομένα) και συγκρίνετε την επίδοσή του με αυτή του ταξινομητή k-NN.
- (γ) Τέλος, να υλοποιήσετε έναν απλοϊκό ταξινομητή Bayes για το δεύτερο πρόβλημα, να αποτιμήσετε την επίδοσή του και να τη συγκρίνετε με τους προηγούμενους ταξινομητές.

2.1.1 k-NN classifier

2.1.1.1 k-NN classifier: IRIS-iris.m

```
1
2 clc
3 clear all
4 %Diavasma tou arxeiou
5 [slength,swidth,plength,pwidth,name] = textread("iris.data", "%f %f %f %f ...
    %s","delimiter","");
6
7 %Ypologizw ta z-score
8 slength = zscore(slength);
9 swidth = zscore(swidth);
10 plength = zscore(plength);
11 pwidth = zscore(pwidth);
12 %Kataskevazw enan pinaka me ta stoixeia ta indicies to pinakwn tou arxeiou
13 indicies = [];
14 for i = 1:1:length(name)
15     indicies = [indicies;i];
16 end
17 %anakatevw me tuxaio tropo tis times
18 indicies = indicies(randperm(length(indicies))));
19 k = 0;
20 %Thetw ton ari8mo tw n geitonwn
21 times = 0;
22 true_predicted = 0;
23 %Se kathe loop allazoun ta folds
24 for i = 1:1:10
25     validation = indicies(k+1:1:k+length(indicies)/10);
26     training = indicies(1:1:k);
27     training = ...
        [training;indicies(k+length(indicies)/10+1:1:length(indicies))];
28     %Ypologizw posa futa apo to ka8e eidos periexontai sto validation fold
29     %wste na ta afairesw otan upologizw to classification
30     count_iris_virg = 50 - sum(strcmp(name(validation),"Iris-virginica"));
31     count_iris_setosa = 50 - sum(strcmp(name(validation),"Iris-setosa"));
32     count_iris_versi = 50 - sum(strcmp(name(validation),"Iris-versicolor"));
33     %Ypologizw tiw a priori pi8anothtes gia thn ka8e class
34     iris_virg_prob = count_iris_virg/length(training);
35     iris_setosa_prob = count_iris_setosa/length(training);
36     iris_versi_prob = count_iris_versi/length(training);
37     %Gia ka8e ena futo tou validation fold,taksinomw me vash ton KNN
38     for j = 1:1:length(validation)
39         times = times + 1;
40         index = validation(j);
41         %vrhskw tis Euclidean distances tou validation futou pou eksetazw
42         %me ka8e training futo
43         dist = Euclidean(slength,swidth,plength,pwidth,training,index);
```



```

44     %kanw afksousa taksinomhsh wste na exw tis elaxistes apostaseis sta
45     %prwta kelia tou dist kratwntas kai ta arxika indicies
46     [min_dist,orig_indicies] = sort(dist);
47     classified = ...
         KNN(dist,name,neighbours,min_dist,orig_indicies,training, ...
         count_iris_virg,count_iris_setosa,count_iris_versi,iris_virg_prob, ...
         iris_setosa_prob,iris_versi_prob);
48     if(strcmp(name(index),classified))
49         true_predicted = true_predicted +1;
50     end
51 end
52 k = k + length(indicies)/10;
53 end
54 percentage = (true_predicted / times)*100;
55 disp (["The percentage of the succeeded clasiifcation is ...
        ",num2str (percentage) ,"%"]);

```

2.1.1.2 k-NN classifier: IRIS-knn.m

```

1
2 function [ classified ] = ...
    KNN(dist,name,neighbours,min_dist,orig_indicies,training,count_iris_virg,
3 count_iris_setosa,count_iris_versi,iris_virg_prob,iris_setosa_prob,
4 iris_versi_prob)
5     found = 0;
6     %Gia ka8e futo,vrhskw ton pio apomakrhsmeno apo tous k geitones
7     for i = 1:1:length(min_dist)
8         if(strcmp(name(training(dist == ...
            dist(orig_indicies(i)))), "Iris-setosa"))
9             found = found + 1;
10        end
11        if (found == neighbours)
12            furthest_w1 = dist(orig_indicies(i));
13            found = 0;
14            break;
15        end
16    end
17    for i = 1:1:length(min_dist)
18        if(strcmp(name(training(dist == ...
            dist(orig_indicies(i)))), "Iris-versicolor"))
19            found = found + 1;
20        end
21        if (found == neighbours)
22            furthest_w2 = dist(orig_indicies(i));
23            found = 0;
24            break;
25        end
26    end

```

```

27     for i = 1:length(min_dist)
28         if (strcmp(name(training(dist == ...
29             dist(orig_indicies(i))), "Iris-virginica"))
30             found = found + 1;
31         end
32         if (found == neighbours)
33             furthest_w3 = dist(orig_indicies(i));
34             break;
35         end
36         %Υπολογισμός των 3 κύκλων των γειτονιών
37         w1_circle = pi*furthest_w1^2;
38         w2_circle = pi*furthest_w2^2;
39         w3_circle = pi*furthest_w3^2;
40         %αναλογισμός ταξινόμησης
41         if (w1_circle/w2_circle < ...
42             (count_iris_setosa/count_iris_versi)*(iris_versi_prob/iris_setosa_prob))
43             if (w1_circle/w3_circle < ...
44                 (count_iris_setosa/count_iris_virg)*(iris_virg_prob/iris_setosa_prob))
45                 classified = "Iris-setosa";
46             else
47                 classified = "Iris-virginica";
48             end
49         else
50             if (w2_circle/w3_circle < ...
51                 (count_iris_versi/count_iris_virg)*(iris_virg_prob/iris_versi_prob))
52                 classified = "Iris-versicolor";
53             else
54                 classified = "Iris-virginica";
55             end
56         end
57     end
58 end

```

2.1.1.3 k-NN classifier: Indians-Indian.m

```

1
2 clc
3 clear all
4 %Diavasma tou arxeiou
5 [pregn,plasmac,pressure,skinthick,insulin,mass,diabete,age,class] = ...
6     textread("pima-indians-diabetes.data", "%u %u %u %u %u %f %f %u ...
7         %u", "delimiter", ",");
8
9 %normalization
10 pregn = zscore(pregn);
11 plasmac = zscore(plasmac);
12 pressure = zscore(pressure);
13 skinthick = zscore(skinthick);

```

```
12 insulin = zscore(insulin);
13 mass = zscore(mass);
14 diabete = zscore(diabete);
15 age = zscore(age);
16 %Kataskevazw enan pinaka m
17
18 %Kataskevazw enan pinaka me ta stoixeia ta indicies to pinakwn tou arxeiou
19 indicies = [];
20 for i = 1:1:length(class)
21     indicies = [indicies;i];
22 end
23 %anakatevw me tuxaio tropo tis times
24 indicies = indicies(randperm(length(indicies))));
25
26 k = 0;
27 neighbours = 25;
28 times = 0;
29 true_predicted = 0;
30 %Se kathe loop allazoun ta folds
31 for i = 1:1:10
32     validation = indicies(k+1:1:k+length(indicies)/10);
33     training = indicies(1:1:k);
34     training = ...
        [training;indicies(k+length(indicies)/10+1:1:length(indicies))];
35     count_class0 = 500 - sum((class(validation) == 0));
36     count_class1 = 268 - sum((class(validation) == 1));
37     %Ypologizw tiw a priori pi8anothtes gia thn ka8e class
38     prob_class0 = count_class0/(count_class0 + count_class1);
39     prob_class1 = count_class1/(count_class0 + count_class1);
40     %Gia ka8e ena futo tou validation fold,taksinomw me vash ton KNN
41     for j = 1:1:length(validation)
42         times = times + 1;
43         index = validation(j);
44         %vrhskw tis Euclidean distances tou validation futou pou eksetazw ...
            me ka8e training
45         dist = ...
            Euclidean(pregn,plasmac,pressure,skinthick,insulin,mass,diabete, ...
            age,training,index);
46         %kanw afksousa taksinomhsh wste na exw tis elaxistes apostaseis sta
47         %prwta kelia tou dist kratwntas kai ta arxika indicies
48         [min_dist,orig_indicies] = sort(dist);
49         classified = ...
            KNN(dist,class,neighbours,min_dist,orig_indicies,training, ...
            count_class0,count_class1,prob_class0,prob_class1);
50         if((class(index) == classified))
51             true_predicted = true_predicted +1;
52         end
53     end
54     k = k + length(indicies)/10;
55 end
```

```

56 percentage = (true_predicted / times)*100;
57 disp (["The percentage of the succeeded classification is ", ...
        num2str(percentage), "%"]);

```

2.1.1.4 k-NN classifier: Indians-KNN.m

```

1
2 function [ classified ] = ...
    KNN(dist,class,neighbours,min_dist,orig_indicies,training,count_class0, ...
    count_class1,prob_class0,prob_class1)
3     found = 0;
4     %Για ka8e futo,vrhskw ton pio apomakrhsmeno apo tous k geitones
5     for i = 1:1:length(min_dist)
6         if((class(training(dist == dist(orig_indicies(i))))== 0))
7             found = found + 1;
8         end
9         if (found == neighbours)
10            furthest_w1 = dist(orig_indicies(i));
11            found = 0;
12            break;
13        end
14    end
15    for i = 1:1:length(min_dist)
16        if((class(training(dist == dist(orig_indicies(i))))== 1))
17            found = found + 1;
18        end
19        if (found == neighbours)
20            furthest_w2 = dist(orig_indicies(i));
21            found = 0;
22            break;
23        end
24    end
25    %Υπολογizw tous 2 kuklous twn geitonwn
26    w1_circle = pi*furthest_w1^2;
27    w2_circle = pi*furthest_w2^2;
28    %analogws taksinomw
29    if (w1_circle/w2_circle < ...
        (count_class1/count_class0)*(prob_class0/prob_class1))
30        classified = 0;
31    else
32        classified = 1;
33    end
34
35
36 end

```

2.1.2 Bayesian classifier

2.1.2.1 Bayesian classifier: bayesian.m

```
1
2 clc
3 clear all
4 %Diavasma tou arxeiou
5 [pregn,plasmac,pressure,skinthick,insulin,mass,diabete,age,class] = ...
    textread("pima-indians-diabetes.data", "%u %u %u %u %u %f %f %u ...
    %u","delimiter",",");
6 %normalization
7 pregn = zscore(pregn);
8 plasmac = zscore(plasmac);
9 pressure = zscore(pressure);
10 skinthick = zscore(skinthick);
11 insulin = zscore(insulin);
12 mass = zscore(mass);
13 diabete = zscore(diabete);
14 age = zscore(age);
15 data = [pregn plasmac pressure skinthick insulin mass diabete age];
16
17 %Kataskevazw enan pinaka me ta stoixeia ta indicies to pinakwn tou arxeiou
18 indicies = [];
19 for i = 1:1:length(class)
20     indicies = [indicies;i];
21 end
22 %anakatevw me tuxaio tropo tis times
23 indicies = indicies(randperm(length(indicies))));
24
25 k = 0;
26 times = 0;
27 true_predicted = 0;
28 %Se kathe loop allazoun ta folds
29 for i = 1:1:10
30     validation = indicies(k+1:1:k+length(indicies)/10);
31     training = indicies(1:1:k);
32     training = ...
        [training;indicies(k+length(indicies)/10+1:1:length(indicies))];
33     count_class0 = 500 - sum((class(validation) == 0));
34     count_class1 = 268 - sum((class(validation) == 1));
35     data0 = data(class(training) == 0,:);
36     data1 = data(class(training) == 1,:);
37     %Ypologizw tous mesous ka8e kathgorias kai tous pinakes sundiasporas ...
        gia ka8e class
38     mean0 = mean(data0);
39     mean1 = mean(data1);
40     cov0 = cov(data0);
41     cov1 = cov(data1);
```

```

42 %Ypologizw tis a priori pi8anothtes gia thn ka8e class
43 prob_class0 = count_class0/(count_class0 + count_class1);
44 prob_class1 = count_class1/(count_class0 + count_class1);
45 %Gia ka8e ena indiano tou validation fold
46 for j = 1:length(validation)
47     times = times + 1;
48     %Ypologizw tis posterior probabilities gia thn ka8e kathgotia
49     %agnowntas to P(x) ka8ws stis sugkriseis 8a fygei
50     prob0 = mvnpdf(data(validation(j),:),mean0,cov0)*prob_class0;
51     prob1 = mvnpdf(data(validation(j),:),mean1,cov1)*prob_class1;
52     if(prob0 > prob1)
53         classified = 0;
54     else
55         classified = 1;
56     end
57
58     if((class(validation(j)) == classified))
59         true_predicted = true_predicted +1;
60     end
61 end
62 k = k + length(indicies)/10;
63 end
64 percentage = (true_predicted / (times-2))*100;

```

2.1.2.2 Naive Bayesian classifier: bayesian.m

```

1
2 clc
3 clear all
4 %Diavasma tou arxeiou
5 [pregn,plasmac,pressure,skinthick,insulin,mass,diabete,age,class] = ...
    textread("pima-indians-diabetes.data", "%u %u %u %u %u %f %f %u ...
    %u", "delimiter", ",", "");
6
7 %normalization
8 pregn = zscore(pregn);
9 plasmac = zscore(plasmac);
10 pressure = zscore(pressure);
11 skinthick = zscore(skinthick);
12 insulin = zscore(insulin);
13 mass = zscore(mass);
14 diabete = zscore(diabete);
15 age = zscore(age);
16 data = [pregn plasmac pressure skinthick insulin mass diabete age];
17
18
19 %Kataskevazw enan pinaka me ta stoixeia ta indicies to pinakwn tou arxeiou
20 indicies = [];

```

```
21 for i = 1:1:length(class)
22     indicies = [indicies;i];
23 end
24 %anakatevw me tuxaio tropo tis times
25 indicies = indicies(randperm(length(indicies))));
26
27 k = 0;
28 times = 0;
29 true_predicted = 0;
30 %Se kathe loop allazoun ta folds
31 for i = 1:1:10
32     validation = indicies(k+1:1:k+length(indicies)/10);
33     training = indicies(1:1:k);
34     training = ...
        [training;indicies(k+length(indicies)/10+1:1:length(indicies))];
35     count_class0 = 500 - sum((class(validation) == 0));
36     count_class1 = 268 - sum((class(validation) == 1));
37     data0 = data(class(training) == 0,:);
38     data1 = data(class(training) == 1,:);
39     %Ypologizw tous mesous kai tis diaspores ka8e kathgorias gia ka8e class
40     mean0 = mean(data0);
41     mean1 = mean(data1);
42     std0 = std(data0);
43     std1 = std(data1);
44     %Ypologizw tis a priori pi8anothtes gia thn ka8e class
45     prob_class0 = count_class0/(count_class0 + count_class1);
46     prob_class1 = count_class1/(count_class0 + count_class1);
47     for j = 1:1:length(validation)
48         times = times + 1;
49         %Ypologizw tis posterior pi8anothtes gia thn ka8e class
50         prob0 = normcdf(data(validation(j),:),mean0,std0)*prob_class0;
51         prob1 = normcdf(data(validation(j),:),mean1,std1)*prob_class1;
52         if(prob0 > prob1)
53             classified = 0;
54         else
55             classified = 1;
56         end
57         if((class(validation(j))) == classified)
58             true_predicted = true_predicted +1;
59         end
60     end
61     k = k + length(indicies)/10;
62 end
63 percentage = (true_predicted / (times-2))*100;
64 disp (["The percentage of the succeeded classification is ...
        ",num2str(percentage),"%"]);
```

2.2 Θέμα 2

- (α') Ξεκινώντας με αναζήτηση στο internet, ή με άλλο τρόπο, σχεδιάστε ένα δίκτυο Bayes που να αντιστοιχεί σε ένα πρόβλημα της αρεσκείας σας. Το δίκτυο θα πρέπει να είναι σχετικά απλό (της τάξεως των 5-10 κόμβων) και τουλάχιστον ένας κόμβος θα πρέπει να έχει 2 τουλάχιστον γονείς. Αποδώστε a priori πιθανότητες στους κόμβους και τους συνδέσμους του σχετικού γραφήματος.
- (β') Στη διεύθυνση: <http://bnt.googlecode.com> θα βρείτε το "Bayesian Nets Toolkit" (BNT). Κατεβάστε το, και διαβάστε τις οδηγίες χρήσης και το βασικό παράδειγμα κατασκευής ενός απλού δικτύου και των τρόπων με τους οποίους μπορούμε να κάνουμε συμπερασμό (inference) πιθανοτήτων σ' αυτό.
- (γ') Στη συνέχεια να υλοποιήσετε με τη βοήθεια του BNT το δίκτυο που σχεδιάσατε στο (α). Να δώσετε παραδείγματα συμπερασμού που να χρησιμοποιούν τόσο ευθεία όσο και ανάδρομη διάδοση πληροφορίας και να χρησιμοποιήσετε το BNT για να υπολογίσετε τις αντίστοιχες δεσμευμένες πιθανότητες.

2.2.1 Bayesian Nets Toolkit

```

1
2 clc; echo on; warning on;
3
4 % check this url: http://bnt.googlecode.com/svn/trunk/docs/usage.html for ...
   more info
5
6 % number of nodes in graph
7 N=5;
8
9 % specify the directed acyclic graph for metastatic cancer(dag)
10 dag=zeros(N,N);
11
12 % the nodes must always be numbered in topological order, i.e., ancestors ...
   before descendants.
13 mc=1; sc=2; b=3; c=4; sh=5;
14
15
16 % specifying the graph structure.
17 dag(mc,[sc b])=1;
18 dag(sc,c)=1;
19 dag(b,[c sh])=1;
20

```



```

21 % we must specify the size and type of each node. In this case, all nodes ...
    are discrete and binary.
22 discrete_nodes=1:N;
23 node_sizes = 2*ones(1,N);
24
25
26
27 % we are now ready to make the Bayes net
28 bnet = mk_bnet(dag, node_sizes, "discrete", discrete_nodes);
29
30 % specify which nodes will be observed. Because it is not fixed in ...
    advance, we just use the empty list (the default).
31 onodes = [];
32 bnet = mk_bnet(dag, node_sizes, "discrete", discrete_nodes, "observed", ...
    onodes);
33
34
35
36 % define the probability distribution of a node given its parents.
37 % CPD objects (CPD = Conditional Probability Distribution)
38 bnet.CPD{mc}=tabular_CPD(bnet, mc, "CPT", [0.8 0.2]); ...
    % 1
39 bnet.CPD{sc}=tabular_CPD(bnet, sc, "CPT", [0.8 0.2 0.2 0.8]); ...
    % 2
40 bnet.CPD{b}=tabular_CPD(bnet, b, "CPT", [0.95 0.8 0.05 0.2]); ...
    % 3
41 bnet.CPD{c}=tabular_CPD(bnet, c, "CPT", [0.95 0.2 0.2 0.2 0.05 0.8 0.8 ...
    0.8]); % 4
42 bnet.CPD{sh}=tabular_CPD(bnet, sh, "CPT", [0.4 0.2 0.6 0.8]); ...
    % 5
43
44 % how did the above come out
45 % 1 - Metastatic Cancer
46 % 80% probability of metastatic cancer(MC) absence and 20% ...
    probability of its presence
47 % 2 - Serum Calcium
48 % if Serum Calcium(SC) is not increased and
49 % ( MC is absent then 80%, MC is present then 20% ),
50 % else if SC is increased and
51 % ( MC is absent then 20%, MC is present then 80%)
52 % 3 - Brain Tumor
53 % if Brain Tumor(B) is absent and
54 % ( MC is absent then 95%, MC is present then 80% ),
55 % else if B is present and
56 % ( MC is absent then 5%, MC is present then 20% )
57 % 4 - Coma
58 % if Coma(C) is absent and
59 % [ B is absent and
60 % ( SC is not increased then 95%, SC is increased then 20% ),
61 % B is present and

```

```

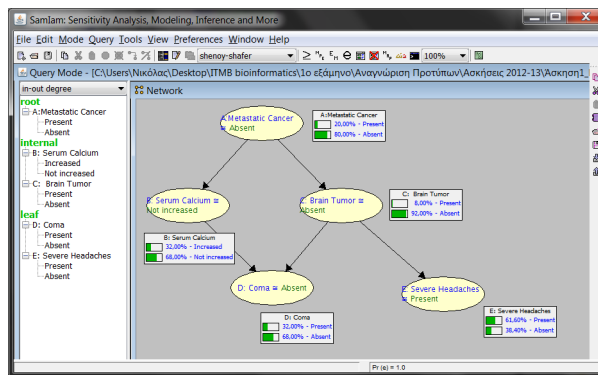
62         % ( SC is not increased then 20%, SC is increased then 20% ...
        % ) ],
63     % else if C is present and
64     % [ B is absent and
65     % ( SC is not increased then 5%, SC is increased then 80% ),
66     % B is present and
67     % ( SC is not increased then 80%, SC is increased then 80% )]
68 % 5 - Severe Headaches
69 % if Severe Headaches(SH) are absent and
70 % ( B is absent then 40%, B is present then 20% ),
71 % else if SH is present and
72 % ( B is absent then 60%, B is present then 80% )
73
74
75 G=bnet.dag;
76 draw_graph(G);          %visualization of the arbitrary graph
77
78
79 % use of the junction tree engine to inference on the graph
80 engine = jtree_inf_engine(bnet);
81
82
83 % compute the probability that Brain Tumor(B) was on given that Severe ...
    Headaches(SH) were present
84 evidence = cell(1,N);
85 evidence{sc}=2;          % sc=true
86
87 [engine,loglik] = enter_evidence(engine, evidence);
88
89
90 % these three below (2 reverse and 1 direct) conditional probabilities ...
    where used for our examples
91 % compute probability p=P(B=2|SH=2)
92 % compute probability p=P(MC=2|C=2)
93 % compute probability p=P(C=2|SC=2)
94 marg = marginal_nodes(engine,c);
95 marg.T
96 p = marg.T(2);
97
98 % plot discrete variable as a barchart
99 bar(marg.T)
100 xlabel("Absence / Presence");
101 ylabel("Probability");

```

ΚΕΦΑΛΑΙΟ 3

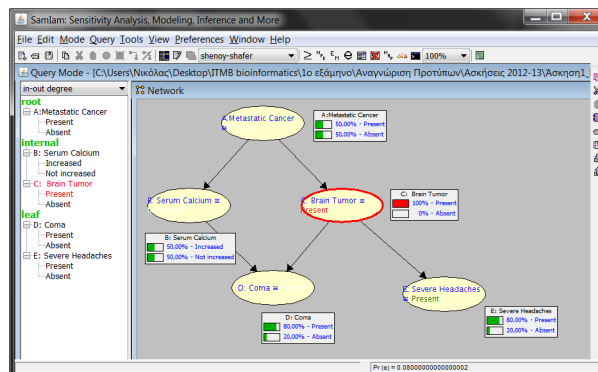
Ενδεικτικές Εικόνες και samIam

Σε αυτό το κεφάλαιο παραθέτουμε κάποιες ενδεικτικές εικόνες που δεν παρουσιάσαμε στην εργασία νωρίτερα. Επίσης σας προτρέπουμε να χρησιμοποιήσετε το πρόγραμμα samIam το οποίο κάνει inference bayesian networks και αναπτύχθηκε από το UCLA¹. Αξίζει να σημειωθεί ότι η ομάδα μας χρησιμοποίησε το samIam και ως εργαλείο επαλήθευσης για την επιβεβαίωση της σωστής λειτουργίας του BNT.

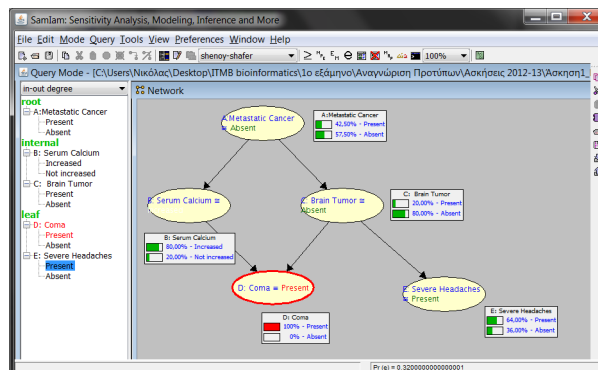


Σχήμα 3.1: *Inferencing of Metastatic Cancer Bayesian Nets*

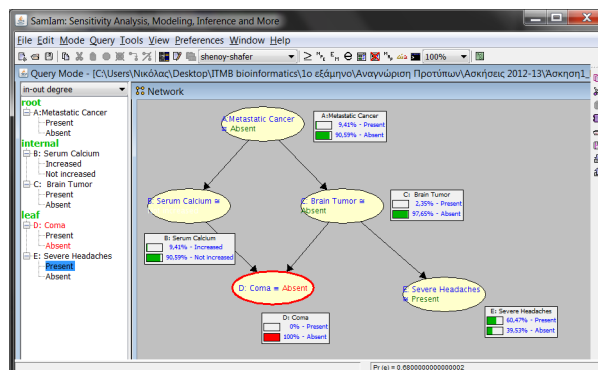
¹<http://reasoning.cs.ucla.edu/samiam/>



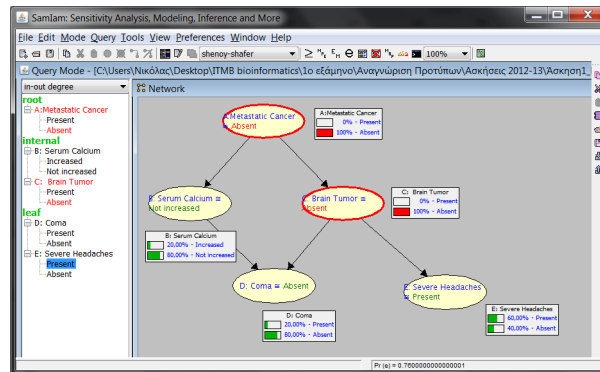
Σχήμα 3.2: *Inferencing of Metastatic Cancer Bayesian Nets with B present*



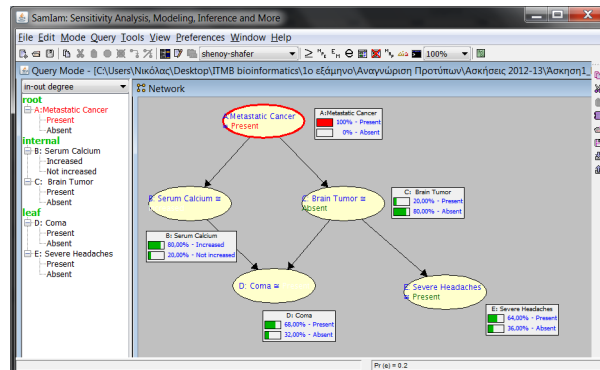
Σχήμα 3.3: *Inferencing of Metastatic Cancer Bayesian Nets with C present*



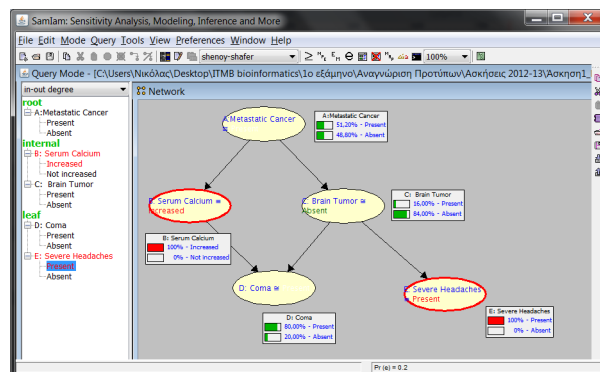
Σχήμα 3.4: *Inferencing of Metastatic Cancer Bayesian Nets with C absent*



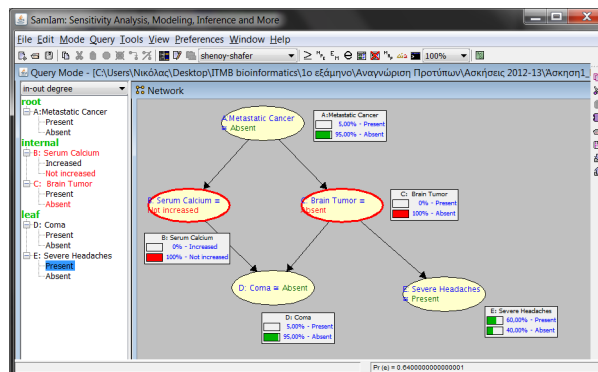
Σχήμα 3.5: *Inferencing of Metastatic Cancer Bayesian Nets with MC and B present*



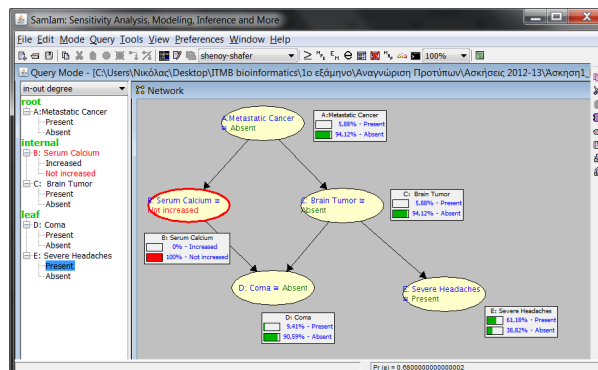
Σχήμα 3.6: *Inferencing of Metastatic Cancer Bayesian Nets with MC present*



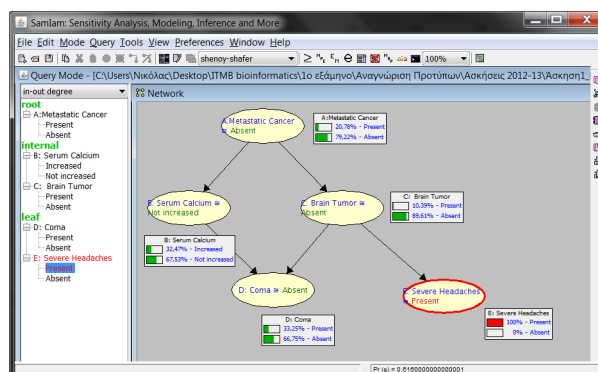
Σχήμα 3.7: *Inferencing of Metastatic Cancer Bayesian Nets with SC incr and SH present*



Σχήμα 3.8: *Inferencing of Metastatic Cancer Bayesian Nets with SC not incr and B absent*



Σχήμα 3.9: *Inferencing of Metastatic Cancer Bayesian Nets with SC not incr*



Σχήμα 3.10: *Inferencing of Metastatic Cancer Bayesian Nets with SH present*

ΚΕΦΑΛΑΙΟ 4

Κατακλείδα

Στα πλαίσια της πρώτης Πρακτικής Άσκησης στο μάθημα της Αναγνώρισης Προτύπων, κλιθήκαμε να υλοποιήσουμε, να εκτιμήσουμε και να αξιολογήσουμε 3 είδη ταξινομητών, (α) τον ταξινομητή πλησιέστερων γειτόνων (K-NN), (β) τον ταξινομητή Bayes και (γ) τον απλό ταξινομητή Bayes (Naive Bayes), με εφαρμογή πάνω σε 2 πολύ γνωστά από την βιβλιογραφία datasets, το Iris plant Database και το Pima Indians Diabetes Database. και επίσης να χρησιμοποιήσουμε τη βιβλιοθήκη Bayesian Nets Toolkit, της Matlab με σκοπό να συμπαιράνουμε πάνω σε υπό συνθήκη πιθανότητες για ένα πρόβλημα της αρεσκείας μας και συγκεκριμένα του Μεταστατικού Καρκίνου. Εμείς επιλέξαμε να πραγματοποιήσουμε την υλοποίηση μας στο προγραμματιστικό περιβάλλον της Matlab. Η εκτίμηση των αποτελεσμάτων για την κάθε υλοποίηση των ταξινομητών, για τα συγκεκριμένα datasets, έγινε με την χρήση της μεθόδου tenfold validation. Κατά την συγκεκριμένη μέθοδο, χωρίστηκαν τα δοθέντα δεδομένα σε 10 ισόποσα μέρη, folds. Έτσι, σε κάθε βήμα του ελέγχου, τα δεδομένα των 9 folds, training fold, αποτελούσαν τα δεδομένα εκπαίδευσης του ταξινομητή που υλοποιείται, ενώ τα δεδομένα του 1 fold, validation fold, αποτελούσαν αυτά που είναι προς ταξινόμηση.

Πραγματοποιώντας επαναληπτικά την διαδικασία, όλα τα folds, με την σειρά, είχαν από μία φορά τον ρόλο του validation fold, και 9 φορές αποτελούσαν μέρος του training fold. Για να υπάρχει τυχαιότητα στην θέση των δεδομένων μέσα στο dataset που μας δίνεται, αποφασίσαμε να κάνουμε μια τυχαία αναταξινόμηση των δεδομένων, ώστε να μην έχουν τις ίδιες θέσεις κάθε φορά που θα πραγματοποιείται μια υλοποίηση. Η αξιολόγηση των ταξινομητών έγινε με βάση την επίδοσή τους, και πιο συγκεκριμένα με το ποσοστό πραγματοποίησης ορθής ταξινόμησης.

Όπου ζητήθηκε, έγινε και αξιολόγηση με κριτήριο την σύγκριση των επιδόσεων με κάποιον άλλο ταξινομητή που έχει υλοποιηθεί.

Τέλος, σε πολλά σημεία της άσκησής μας χρησιμοποιήσαμε το εργαλείο samIam που αναπτύχθηκε στο UCLA ¹ και κάνει συμπερασμό(inference) πάνω σε δίκτυα Bayes.

¹<http://reasoning.cs.ucla.edu/samiam/>

Ορολογία

Ξένος όρος	Ελληνικός όρος
Bayesian Networks	Δίκτυα Bayes
Classifier	Ταξινομητής
Inference	Συμπερασμός
Retrograde	Ανάδρομος
Direct	Ευθύς
DataBase	Βάση Δεδομένων
Metastatic Cancer	Μεταστατικός Καρκίνος
Brain Tumor	Εγγκεφαλικός όγκος
Serum Calcium	Ασβέστιο του ορού
Severe Headaches	Σοβαρές Κεφαλαλγίες
8Coma	Κώμα
Probability Distribution	Κατανομή
Presence	Παρουσία
Absence	Απουσία
Increased	Αυξημένος
Decreased	Μειωμένος
Validation	Εγκυρότητα
Normalization	Κανονικοποίηση

Ακρωνύμια - Αρκτικόλεξα

Ακρωνύμια/Αρκτικόλεξα	Full Evolvent
k-NN	k Nearest Neighbours
BNT	Bayesian Nets Toolkit
MC	Metastatic Cancer
SC	Serum Calcium
B	Brain Tumor
C	C
SH	Severe Headaches
BN	Bayesian Network
ΤΠΙΒ	Τεχνολογίες Πληροφορικής στην Ιατρική και στη Βιολογία
IARP	International Association of REIKI Professionals
UCLA	University of California Los Angeles
DNA	Deoxyribonucleic acid

Βιβλιογραφία

Βιβλιογραφία

- [1] Arthur Choi, Lu Zheng, Adnan Darwiche and Ole J. Mengshoel, “*A Tutorial on Bayesian Networks for System Health Management*” Silicon Valley Campus, November 2011.
Available: http://repository.cmu.edu/silicon_valley/66/
- [2] Victor Tse, MD, PhD Associate Professor, Department of Neurosurgery, Stanford University Medical Center, Santa Clara Valley Medical Center , “*Brain Metastasis Clinical Presentation*” Oct 2011.
Available: <http://emedicine.medscape.com/article/1157902-clinical>
- [3] Alex Dekhtyar, Judy Goldsmith, Janice L. Pearce, “*When Plans Distinguish Bayes Nets*” May 2001.
Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.2896>