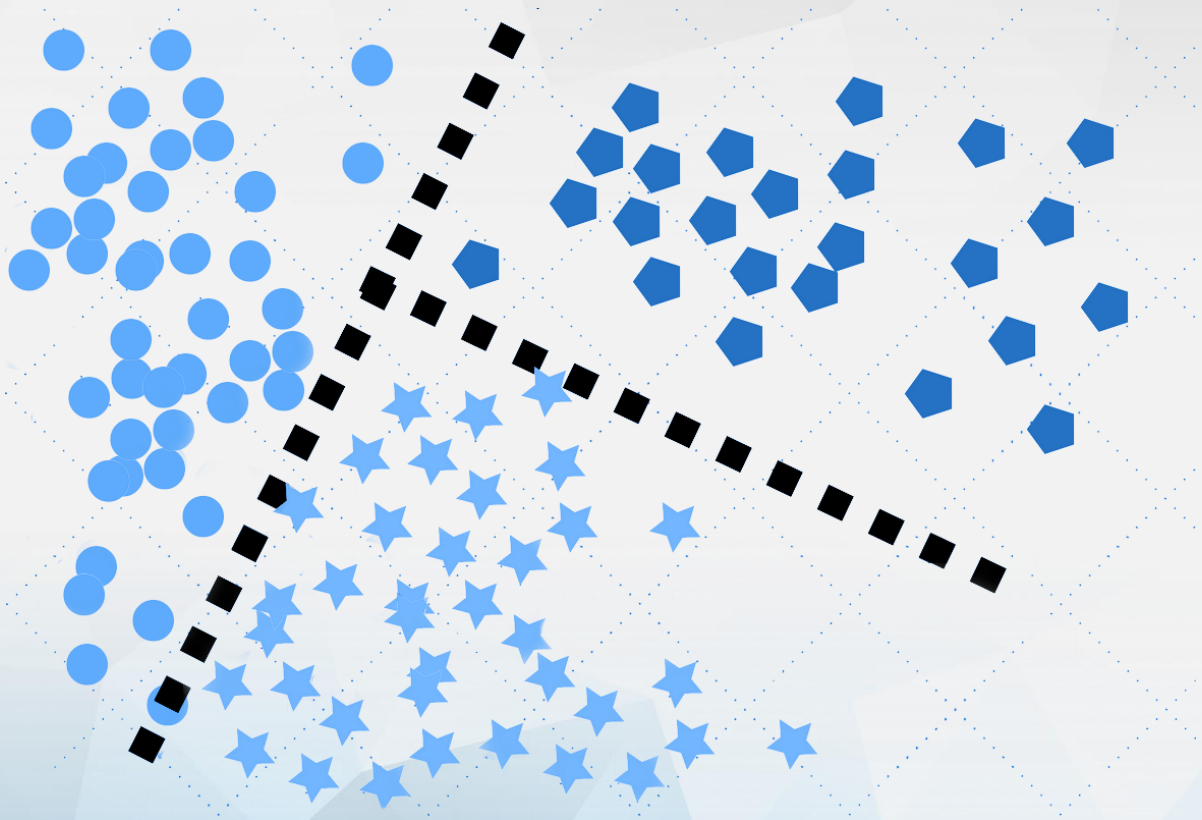
ΣΤΑΤΙΣΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

2ο σετ θεωρητικών ασκήσεων



Γαλάνης Μιχάλης

2016030036

ΠΕΡΙΕΧΟΜΕΝΑ

Οι σύνδεσμοι για τις παρακάτω ενότητες είναι διαδραστικοί.
Πιέστε πάνω στο επιθυμητό τμήμα της άσκησης για τη μετάβαση σε αυτό.

Άσκηση 1 (Logistic Regression)

Στην πρώτη άσκηση, ασχολούμαστε με τη **Λογιστική παλινδρόμηση**, ένας από τους πιο συνηθισμένους αλγόριθμους που χρησιμοποιούνται για supervised learning. Θα τη χρησιμοποιήσουμε για να μοντελοποιήσουμε τη πιθανότητα κάποιων φοιτητών να γίνουν δεκτοί στην εισαγωγή τους στο πανεπιστήμιο βάσει των ιστορικών αποτελεσμάτων δύο διαγωνισμάτων σε προηγούμενους φοιτητές.

Υλοποίηση

Αρχικά φορτώνουμε το dataset από το αρχείο δεδομένων που μας δίνεται. Αυτό περιέχει ως γραμμές τα δείγματα μας και ως στήλες την βαθμολογία του πρώτου διαγωνίσματος, του δεύτερου διαγωνίσματος καθώς και το label για την αποδοχή ή όχι του συγκεκριμένου φοιτητή.

Το πρώτο μας βήμα είναι να συμπληρώσουμε το αρχείο **`sigmoid.m`** το οποίο υλοποιεί τη συνάρτηση της λογιστικής παλινδρόμησης:

$$f(z) = \frac{1}{1 + e^{-z}}$$

Επόμενο βήμα είναι η συμπλήρωση του αρχείου **`costFunction.m`** έτσι ώστε να υπολογίσουμε το αρχικό κόστος και τη κλήση της παλινδρόμησης. Αυτά τα μεγέθη ορίζονται ως:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}ln(h_\theta(x^{(i)})) - (1 - y^{(i)})ln(1 - h_\theta(x^{(i)}))\right)$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

όπου $h_\theta(x^{(i)})$ το αποτέλεσμα της λογιστικής συνάρτησης.

Ύστερα θα χρησιμοποιήσουμε τη συνάρτηση **`fminunc()`** η οποία βρίσκει τη βέλτιστη τιμή του κόστους για τις παραμέτρους θ. Η τιμή του θ που θα προκύψει, θα χρησιμοποιηθεί αργότερα για την απεικόνιση του hyperplane.

Τελευταίο βήμα είναι η συμπλήρωση του αρχείου **`predict.m`** το οποίο θα μας δείξει ποια ήταν η ακρίβεια της εκπαίδευσης των δεδομένων παρατηρώντας ποια αποτελέσματα ήταν τελικά σωστά. Πιο συγκεκριμένα για ένα vector X επιστρέφουμε ένα Boolean vector p με μηδενικά και άσσους σύμφωνα με ένα threshold.

Για την απεικόνιση χρησιμοποιούμε τις ήδη υλοποιημένες συναρτήσεις **`plotData.m`** και **`plotDecisionBoundary.m`**.

Αποτελέσματα

Εκτελώντας τα συμπληρωμένα scripts του matlab παράγουμε τα ακόλουθα αποτελέσματα.

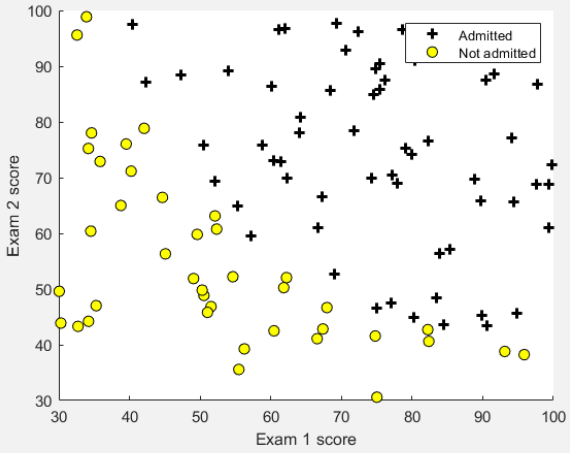
```
Cost at initial theta (zeros): 0.693147
Gradient at initial theta (zeros):
 -0.100000
 -12.009217
 -11.262842
```

```
Cost at theta found by fminunc: 0.203498
theta:
 -25.161343
 0.206232
 0.201472
```

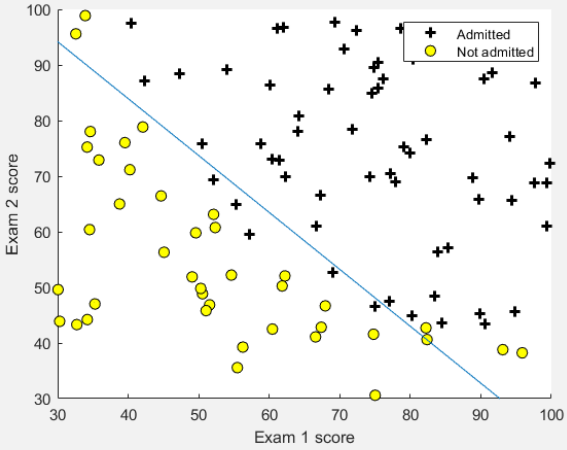
Επίσης, για έναν φοιτητή με βαθμολογίες **45** και **85** στα διαγωνίσματα του, προβλέψαμε μια πιθανότητα **0.776291** να εισαχθεί στο πανεπιστήμιο.

Η ακρίβεια της εκπαιδευτικής διαδικασίας των δεδομένων υπολογίστηκε **89%**. Παρακάτω φαίνονται τα αποτελέσματα που παράχθηκαν από το εργαλείο matlab.

Αρχικά Δεδομένα



Δεδομένα με boundary line



Άσκηση 2 (Reg. Log. Regression)

Στην άσκηση αυτή εφαρμόζουμε ομαλοποιημένη λογιστική παλινδρόμηση για να προβλέψουμε αν τα μικροτσίπ από μια μονάδα κατασκευής περνούν τον έλεγχο ποιότητας στον οποίο υπάρχουν διάφορες δοκιμές για την εξασφάλιση λειτουργείας τους. Έτσι αποφασίζεται αν τα μικροτσίπ θα αποδεχθούν ή θα απορριφθούν.

Υλοποίηση

Όπως θα αποδειχθεί αργότερα στην άσκηση αυτή, τα δεδομένα μας αυτή τη φορά δε μπορούν να διαχωριστούν με μια ευθεία γραμμή. Συνεπώς θα χρειαστεί μια παραλλαγή της λογιστικής παλινδρόμησης. Αρχικά λοιπόν θα δημιουργήσουμε περισσότερα features από κάθε σημείο. Αυτό επιτυγχάνεται με την απεικόνιση των δεδομένων σε χώρο μεγαλύτερης διάστασης έτσι ώστε να διαχωριστούν πιο εύκολα

με τη λογιστική παλινδρόμηση. Συμπληρώνουμε λοιπόν τη συνάρτηση του αρχείου mapFeature.m και δημιουργούμε ένα vector με χαρακτηριστικά σε όλους τους όρους πολυωνύμων μέχρι και 6ου βαθμού έτσι ώστε:

$$\boldsymbol{P(x_1, x_2)} = \sum_{i=0}^{6}\sum_{j=0}^{i} x_1^j x_2^{i-j}$$

$$\boldsymbol{X} = \left[1, x_1, x_2, x_1^2, x_2^2, x_1x_2, \ldots, x_1x_2^5, x_2^6\right]^T$$

Όπως και στη προηγούμενη άσκηση, υλοποιούμε τη **`sigmoid.m`** συνάρτηση καθώς και το costFunctionReg.m αλλά αυτή τη φορά η συνάρτηση κόστους και gradient δίνονται από τις σχέσεις:

$$\boldsymbol{J(\theta)} = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}ln(h_\theta(x^{(i)})) - (1-y^{(i)})ln(1-h_\theta(x^{(i)}))\right) + \frac{\lambda}{2m}\sum_{i=1}^{m}\theta_j^2$$

$$\frac{\boldsymbol{\partial J(\theta)}}{\boldsymbol{\partial \theta_j}} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m}\theta_j$$

όπου λ μια παράμετρος ομαλοποίησης και $h_\theta(x^{(i)})$ το αποτέλεσμα της λογιστικής συνάρτησης.

Ύστερα χρησιμοποιούμε και πάλι τη συνάρτηση βελτιστοποίησης **`fminunc()`** και παρομοίως υπολογίζουμε την ακρίβεια του training process με την **`predict()`**.

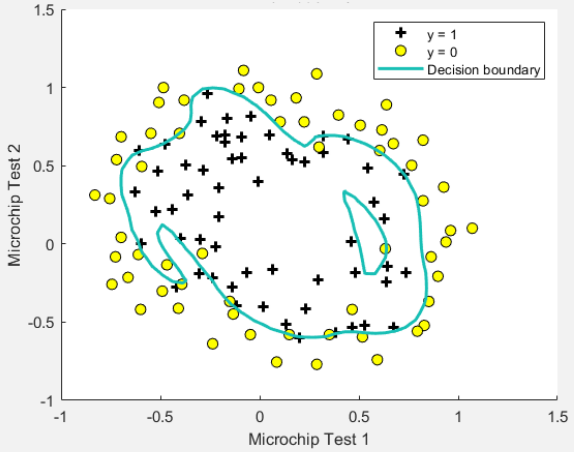
Αποτελέσματα

Αυτή τη φορά βρίσκουμε τα παρακάτω αποτελέσματα για τις διάφορες τιμές του λ.

Decision Boundary ($\lambda = 0$)

$init\_cost = 0.693147$

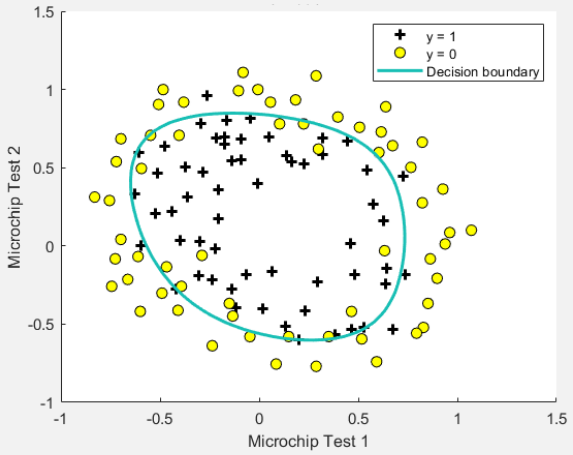
$acc = 88.983\%$



Decision Boundary ($\lambda = 1$)

$init\_cost = 0.693147$

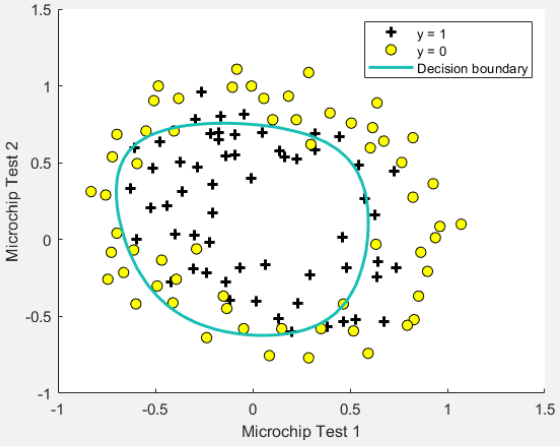
$acc = 82.203\%$



Decision Boundary ($\lambda = 10$)

$init\_cost = 0.693147$

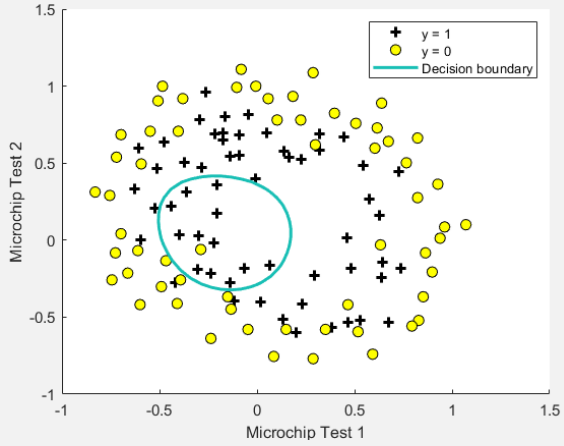
$acc = 74.576\%$



Decision Boundary ($\lambda = 100$)

$init\_cost = 0.693147$

$acc = 60.169\%$



Παρατηρούμε ότι βάζοντας μικρές τιμές του λ ο ταξινομητής βρίσκει σωστά τις τιμές του dataset αλλα δημιουργεί ένα πολύπλοκο decision boundary. Όσο μεγαλώνουμε το λ, δημιουργεί ένα απλούστερο και ομαλοποιημένο σύνορο απόφασης. Βέβαια για πολύ μεγάλες τιμές, έχουμε underfitting και αρχίζουμε να χάνουμε πολλή ακρίβεια.

Άσκηση 3 (Maximum Likelihood)

Υλοποίηση

Έχουμε $\boldsymbol{n}$ δείγματα $D = \{x_1, \ldots, x_n\}$ από μία κατανομή Poisson με παράμετρο λ τέτοιο ώστε:

$$\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\lambda}) = \frac{\lambda^x e^{-\lambda}}{x!}, \qquad x = 0,1,2,\ldots, \qquad \lambda > 0$$

Εφαρμόζουμε MLE με τον ακόλουθο τρόπο:

$$L(\lambda) = \prod_{i=1}^{n} f(x_i|\lambda) = \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} = \frac{\lambda^{\sum_{i=1}^{n} x_i} e^{-n\lambda}}{\prod_{i=1}^{n} x_i!} \overset{log}{\Leftrightarrow}$$

$$logL(\lambda) = \sum_{i=1}^{n} x_i \, log\lambda - n\lambda - \sum_{i=1}^{n} log(x_i!) \Leftrightarrow$$

$$\frac{d}{dx} \, logL(\lambda) = \frac{\sum_{i=1}^{n} x_i}{\lambda} - n = 0$$

Αφού λογαριθμίσουμε και θέσουμε τη παράγωγο ίση με το μηδέν, βρίσκουμε το argmax με την ακόλουθη σχέση:

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Άσκηση 4 (Naïve Bayes Classifier)

Στη 4η άσκηση υλοποιούμε έναν **Naïve Bayes Classifier** για την αναγνώριση χειρόγραφων αριθμητικών ψηφίων.

Υλοποίηση και Αποτελέσματα

Αρχικά φορτώνουμε τα δεδομένα μας από το αρχείο **`digits.mat`** που μας δόθηκε. Αυτό περιλαμβάνει 10 κλάσεις για training set και 10 για test set. Η κάθε κλάση περιγράφει ένα ψηφίο και περιέχει 500 δείγματα. Η κάθε ασπρόμαυρη εικόνα διαθέτει 28 οριζόντια αλλά και κάθετα pixel με αριθμητικές τιμές 0 ή 1.

1ο Ερώτημα

Θα χρησιμοποιήσουμε MLE για να υπολογίσουμε τη παράμετρο p της κατανομής Bernoulli. Αρχικά γράφουμε τη συνάρτηση πιθανοφάνειας:

$$\boldsymbol{L(\theta)} = \prod_{i=1}^{n} p^{X_i}(1-p)^{1-X_i}$$

Λογαριθμίζουμε τη παραπάνω σχέση για να απλουστεύσουμε τις πράξεις μας:

$$LL(\theta) = log\prod_{i=1}^{n} p^{X_i}(1-p)^{1-X_i} = \sum_{i=1}^{n} logp^{X_i}(1-p)^{1-X_i} \Leftrightarrow$$

$$LL(\theta) = \sum_{i=1}^{n} X_i logp + 1 - (1-X_i)log(1-p) \Leftrightarrow$$

$$\boldsymbol{LL(\theta)} = Ylogp + (n-Y)log(1-p)$$

όπου $Y = \sum_{i=1}^{n} X_i$. Αρκεί τώρα να βρούμε εκείνη την τιμή του p που μεγιστοποιεί την λογαριθμική συνάρτηση πιθανοφάνειας. Παραγωγίζοντας και θέτοντας τη παράγωγο ίση με το μηδέν βρίσκουμε την τιμή εκείνη:

$$\frac{\delta LL(p)}{\delta p} = \frac{Y}{p} + \frac{Y-n}{1-p} = 0 \Leftrightarrow$$

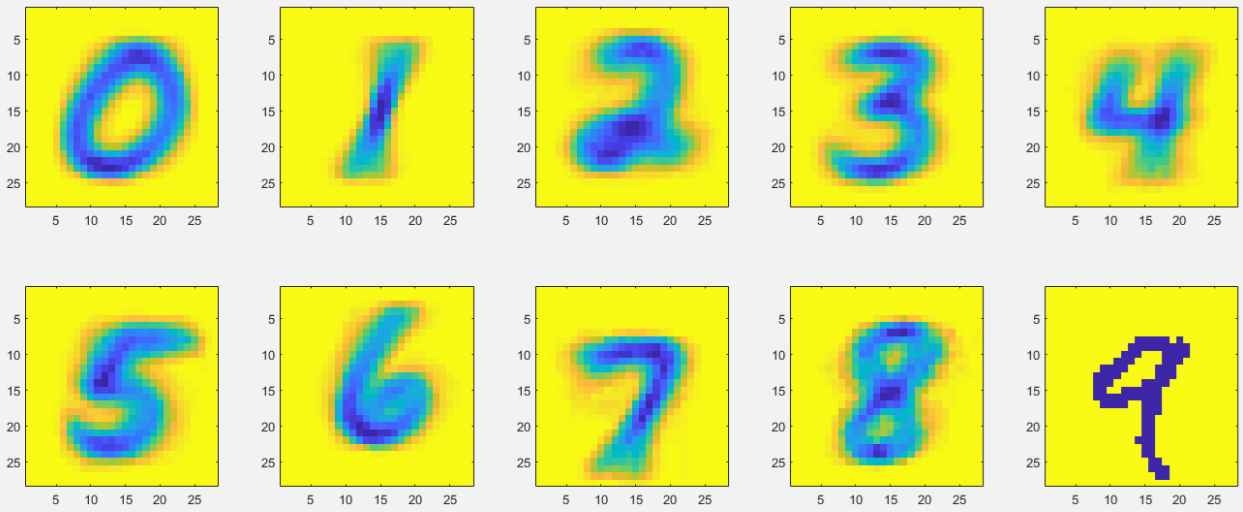
$$\boldsymbol{\hat{p}} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Άρα καταλήγουμε στο γεγονός ότι το MLE για Bernoulli είναι απλώς το mean των δειγμάτων. Μπορούμε τώρα να εισάγουμε τα δεδομένα μας στο matlab και να

εκτιμήσουμε τις μέγιστες πιθανοφάνειες για κάθε κλάση. Διατρέχουμε λοιπόν τα training data και για κάθε κλάση υπολογίζουμε τα mean των αντίστοιχων pixel μεταξύ τους. Σε αυτό το σημείο για την αποφυγή αριθμητικών προβλημάτων αντικαθιστούμε τις μηδενικές τιμές παραμέτρου πιθανότητας p με έναν πολύ μικρό αριθμό $\varepsilon = 0.00001$.

2ο Ερώτημα

Σε αυτό το ερώτημα χρησιμοποιήθηκε ο πίνακας με τους εκτιμητές προκειμένου να εκπαιδεύσουμε τα μοντέλα μας για κάθε ψηφίο. Πιο συγκεκριμένα για κάθε ψηφίο υπολογίστηκε η δεσμευμένη πιθανότητα και αποθηκεύτηκε στον πίνακα `p_tr`. Ύστερα χρησιμοποιήθηκε ο πίνακας αυτός για να απεικονίσουμε τα αποτελέσματα της εκπαίδευσης. Τα αποτελέσματα αυτά παρουσιάζονται παρακάτω:



3ο Ερώτημα

Στο τρίτο και τελευταίο ερώτημα υπολογίσαμε για κάθε δείγμα στα test set μας 10 δεσμευμένες πιθανότητες (μια για κάθε αριθμητικό ψηφίο) η οποίες μας πληροφορούν πόσο πιθανό είναι ένα συγκεκριμένο δείγμα να είναι κάποιο από τα 10 ψηφία. Το ψηφίο που ευθύνεται για τη μεγαλύτερη από αυτές πιθανότητα επιλέγεται και εισάγεται στο **confusion matrix**.

Αυτός ο πίνακας δηλώνει πόσο συχνά μια εικόνα ταξινομήθηκε ως ένα συγκεκριμένο ψηφίο και μας πληροφορεί για το πόσο επιτυχημένη ήταν η εκπαίδευση του μοντέλου μας αλλα και για τις μεγαλύτερες αστοχίες. Για παράδειγμα από το confusion matrix που παράχθηκε λόγω του matlab παρατηρούμε ότι γενικώς είχαμε καλή ευστοχία, υπήρχαν όμως και περιπτώσεις που η πρόβλεψη ήταν λανθασμένη όπως το ψηφίο 4 που πολλές φορές αναγνωριζόταν ως 9.

Η ακρίβεια των προβλέψεων για κάθε κλάση υπολογίστηκε ως την τιμή της διαγωνίου προς το άθροισμα της οριζόντιας γραμμής. Υπολογίστηκε επίσης και η συνολική ακρίβεια του test set ως τον μέσο όρο της ακρίβειας όλων των κλάσεων.

Conf. Matrix		Ψηφίο προβλεφθηκε ως									
		0	1	2	3	4	5	6	7	8	9
Ψηφίο πραγματικά είναι	0	441	0	1	0	2	30	13	0	12	1
	1	0	472	2	3	0	13	4	0	6	0
	2	8	12	375	37	7	2	9	13	33	4
	3	1	10	4	416	5	23	5	13	10	13
	4	1	1	5	0	374	4	15	2	5	93
	5	14	2	3	67	20	345	7	6	15	21
	6	7	10	28	0	7	33	412	0	3	0
	7	1	21	9	3	15	0	0	395	10	46
	8	6	16	12	49	11	18	1	3	349	35
	9	3	8	3	8	51	7	0	6	5	409

```
total_accuray =

    0.7976
```

Άσκηση 5 (Bayesian Inference)

Στην άσκηση αυτή κάνουμε Bayesian εκτίμηση παραμέτρων για ένα πείραμα n μετρήσεων.

Υλοποίηση και Αποτελέσματα

Φορτώνουμε αρχικά το αρχείο '**data_2_5.txt**' το οποίο περιέχει ένα vector 25 στοιχείων. Η συνάρτηση πυκνότητας πιθανότητας των μετρήσεων είναι Γκαουσιανή $p(x) \sim N(\mu, \sigma^2)$ με $\sigma = 1.25$. Η κατανομή της μέσης τιμής είναι επίσης $p(\mu) \sim N(\mu_0, \sigma_0^2)$.

1ο Ερώτημα

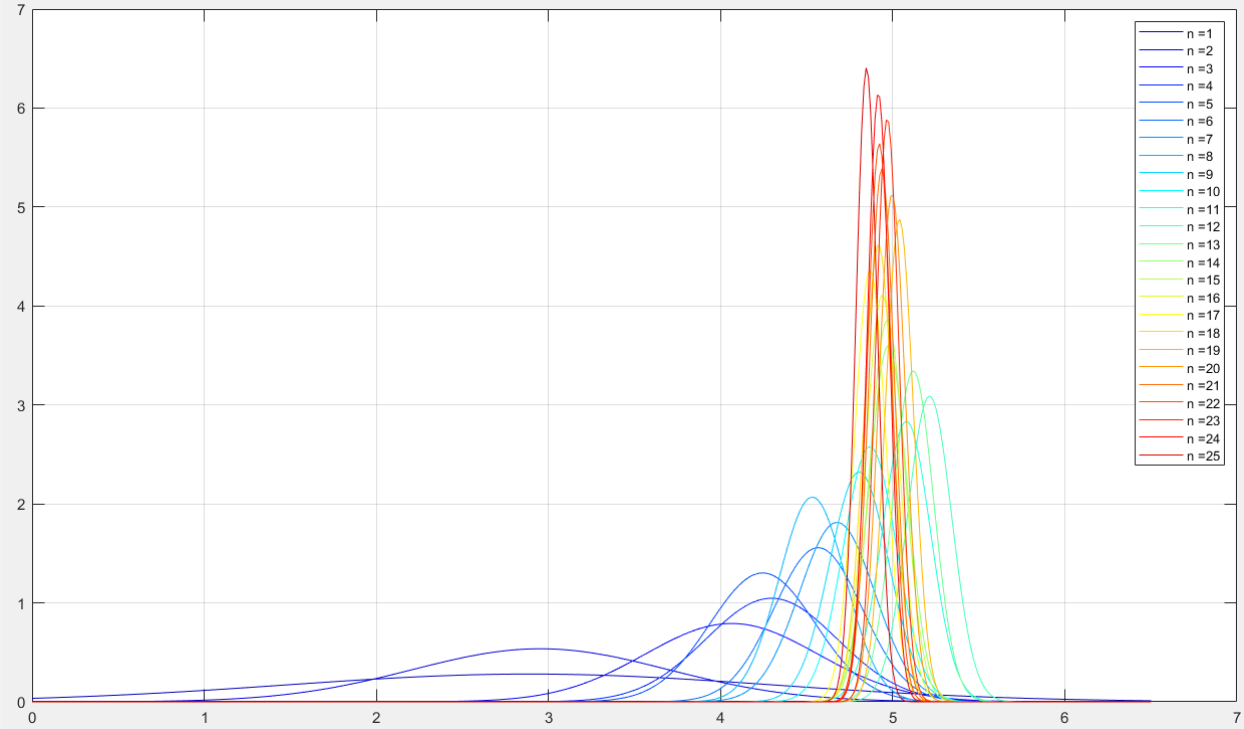
Εφόσον $\mu_0 = 0$ και $\sigma_0^2 = 10\sigma^2$, μας ζητείται να απεικονιστεί η δεσμευμένη πυκνότητα $p(\mu|H(n))$ καθώς το $\boldsymbol{n}$ αυξάνεται από 1 εώς 25.

Από τη θεωρία προκύπτει ότι μπορούμε να υπολογίσουμε τις παραμέτρους μ_n και σ_n^2 με τις ακόλουθες σχέσεις:

$$\boldsymbol{\mu_n} = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2}\right)\bar{x}_n + \left(\frac{\sigma^2}{n\sigma_0^2 + \sigma^2}\right)\mu_0$$

$$\boldsymbol{\sigma_n^2} = \left(\frac{\sigma_0^2\sigma^2}{n\sigma_0^2 + \sigma^2}\right)$$

Συνεπώς, σε έναν βρόγχο επανάληψης υπολογίζουμε επαναληπτικά τις παραπάνω παραμέτρους, δημιουργούμε τη συνάρτηση πυκνότητας πιθανότητας με μέση τιμή μ_n και διασπορά σ_n^2 παίρνοντας υπόψιν όλο και περισσότερες μετρήσεις και ύστερα απεικονίσουμε την a posteriori πυκνότητα $p(\mu|H(n))$. Το παραγόμενο διάγραμμα εμφανίζεται παρακάτω:

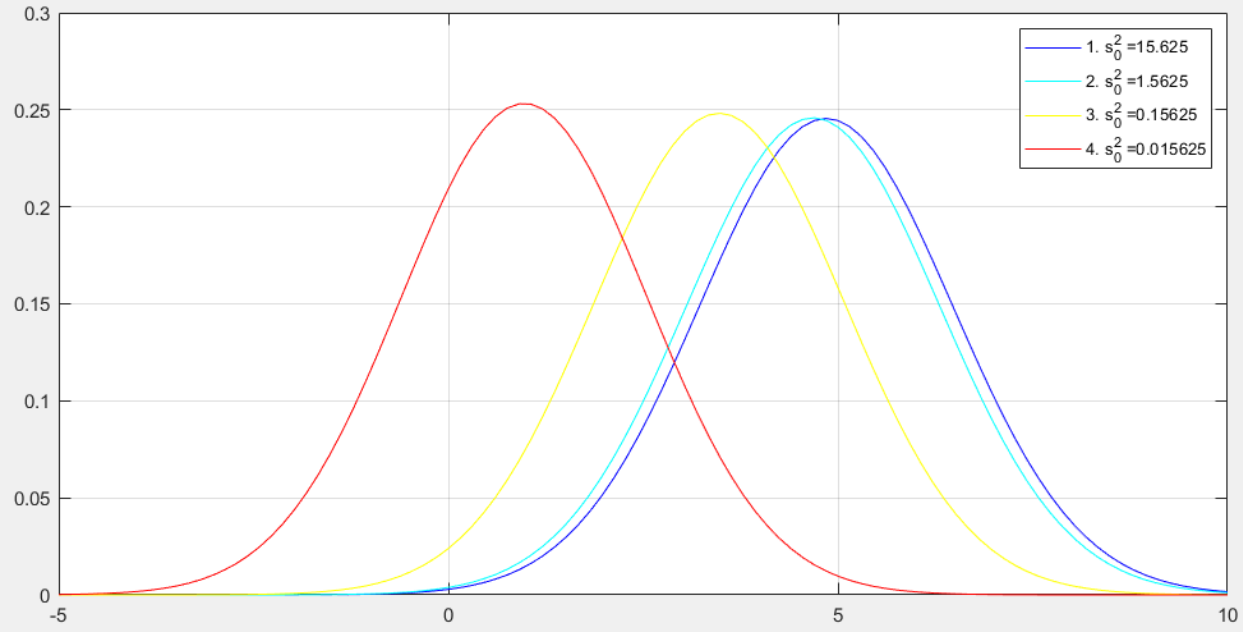


Παρατηρούμε ότι όσο αυξάνεται το $\boldsymbol{n}$, δηλαδή ο όγκος της διαθέσιμης πληροφορίας, μειώνεται σημαντικά η διασπορά (αβεβαιότητα) και η $p(\mu|H(n))$ γίνεται ολοένα και πιο αιχμηρή, προσεγγίζοντας τη συνάρτηση μοναδιαίας ώσης. Αυτή η διαδικασία είναι γνωστή και ως **Bayesian Learning**.

2ο Ερώτημα

Αφήνουμε τώρα σταθερό τον αριθμό των δειγμάτων και ίσο με 25. Ζητείται να σχεδιαστεί η δεσμευμένη πυκνότητα $p(x|H(n))$ για διάφορες τιμές διασποράς σ_0^2.

Γνωρίζουμε από τη θεωρία, ότι έχοντας βρει την $p(\mu|H(n))$, μπορούμε να υπολογίσουμε την $p(x|H(n))$. Προκύπτει ότι και η $p(x|H(n))$ είναι γκαουσιανή με μέση τιμή μ_n και διασπορά $\sigma^2 + \sigma_n^2$. Παρομοίως με το προηγούμενο ερώτημα, σε έναν βρόγχο επανάληψης υπολογίζουμε επαναληπτικά τις παραμέτρους μ_n και σ_n^2 παίρνοντας υπόψιν τις διαφορετικές τιμές του σ_0^2. Δημιουργούμε κατόπιν τη συνάρτηση πυκνότητας $p(x|H(n))$ και την απεικονίζουμε. Τα αποτελέσματα φαίνονται παρακάτω:



Παρατηρούμε ότι όσο αυξάνεται η τιμή του σ_0^2, το διάγραμμα της κατανομής μετακινείται προς τα δεξιά και τείνει να έχει μέση τιμή την $\bar{x}_n$ καθώς και μια ελάχιστα μεγαλύτερη διασπορά, ενώ όσο μειώνεται το σ_0^2, η συνάρτηση πυκνότητας τείνει να χει μηδενική μέση τιμή καθώς και μικρότερη διασπορά.