
# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

## ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

---

# Μηχανική Μάθηση σε Μεταβαλλόμενα Περιβάλλοντα: Ανίχνευση Μεταβολών του Εννοιολογικού Πλαισίου σε Βίντεο

---

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Σεφέρης Εμμανουήλ (Μανώλης)

**Επιβλέπων:** Κόλλιας Στέφανος
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016


# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

## ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

---

# Μηχανική Μάθηση σε Μεταβαλλόμενα Περιβάλλοντα: Ανίχνευση Μεταβολών του Εννοιολογικού Πλαισίου σε Βίντεο

---

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Σεφέρης Εμμανουήλ

**Επιβλέπων:** Κόλλιας Στέφανος
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 20 Ιουλίου 2016.

| ..... | ..... | ..... |
|---|---|---|
| Κόλλιας Στέφανος<br>Καθηγητής ΕΜΠ | Σταφυλοπάτης Γεώργιος-Ανδρέας<br>Καθηγητής ΕΜΠ | Γεώργιος Στάμου<br>Επίκουρος Καθηγητής ΕΜΠ |

Αθήνα, Ιούλιος 2016

.....

**Σεφέρης Εμμανουήλ**
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

# *Ευχαριστίες*

Αρχικά θα ήθελα να ευχαριστήσω τον κ. Στέφανο Κόλλια, Καθηγητή της Σχολής Ηλεκτρολόγων μηχανικών και μηχανικών υπολογιστών, που δέχθηκε να επιβλέψει τη διπλωματική μου εργασία, καθώς και για την ευκαιρία που μου έδωσε να μελετήσω έναν καινούριο και πολύ ενδιαφέροντα τομέα της μηχανικής μάθησης, τη μάθηση σε μεταβαλλόμενα περιβάλλοντα (learning in nonstationary environments). Χωρίς την βοήθειά και τις συμβουλές του η εργασία αυτή θα ήταν αδύνατο να ολοκληρωθεί. Επίσης, θα ήθελα να ευχαριστήσω τον κ. Σταφυλοπάτη, καθώς και τον κ. Στάμου, οι οποίοι δέχθηκαν να συμμετάσχουν στην τριμελή επιτροπή. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για την αμέριστη ηθική και υλική υποστήριξη που μου παρείχε όλα αυτά τα χρόνια.

# Περίληψη

Σκοπός της εργασίας αυτής είναι η μελέτη ενός σχετικά νέου τομέα της μηχανικής μάθησης, της μάθησης σε μεταβαλλόμενα περιβάλλοντα (learning in nonstationary environments), και η εφαρμογή κάποιων από τις μεθοδολογίες του σε ένα πρόβλημα αναγνώρισης συναισθήματος από εικόνα.
Οι συνήθεις αλγόριθμοι μηχανικής μάθησης (νευρωνικά δίκτυα, SVM, Random forests, κλπ.) λειτουργούν με τον εξής τρόπο: αρχικά εκπαιδεύονται σε ένα γνωστό σύνολο δεδομένων (train data) και στη συνέχεια εφαρμόζονται στα υπόλοιπα δεδομένα που θέλουμε να ταξινομήσουμε (test data). Η διαδικασία αυτή κάνει μία θεμελιώδη παραδοχή: ότι η στατιστική κατανομή των test data είναι η ίδια με τη στατιστική κατανομή των train data. Με τον τρόπο αυτό, ένας αλγόριθμος μηχανικής μάθησης μαθαίνει τις ιδιότητες των δεδομένων απ' τα test data, και τις εφαρμόζει μετά στα προς ταξινόμηση δεδομένα.
Υπάρχουν όμως κάποιες περιπτώσεις, όπου η παραπάνω υπόθεση είναι λανθασμένη. Σε μία αρκετά μεγάλη ποικιλία φαινομένων, η στατιστική κατανομή των δεδομένων μεταβάλλεται με το χρόνο. Μερικά παραδείγματα είναι οι κλιματολογικές συνθήκες, δημογραφικά δεδομένα, εισηγητικά συστήματα όπου οι προτιμήσεις του χρήστη αλλάζουν όσο αυτός μεγαλώνει, ρομπότ που αλληλεπιδρούν με το περιβάλλον, κλπ. Σε αυτές τις περιπτώσεις χρειαζόμαστε κάποιες μεθόδους, οι οποίες να μπορούν να ανιχνεύουν τις στατιστικές μεταβολές των δεδομένων, και να τροποποιούν ανάλογα τον ταξινομητή, προσθέτοντας σε αυτόν τη νέα γνώση. Ο κλάδος αυτός της μηχανικής μάθησης ονομάζεται adaptive learning, ή learning in nonstationary environments, ενώ το φαινόμενο της στατιστικής μεταβολής των φαινομένων ονομάζεται concept drift (μεταβολή εννοιολογικού πλαισίου, ή απλώς μεταβολή πλαισίου).
Στην παρούσα εργασία, θα παρουσιάσουμε και θα ανακεφαλαιώσουμε τις σημαντικότερες ιδέες και τεχνικές του adaptive learning. Στη συνέχεια, στο πειραματικό μέρος, θα προσπαθήσουμε να εφαρμόσουμε κάποιες απ' τις τεχνικές αυτές στο πεδίο της αναγνώρισης συναισθήματος από εικόνα (emotion recognition). Χρησιμοποιώντας μία βάση με φωτογραφίες χρηστών και αντίστοιχων συναισθημάτων, θα εκπαιδεύσουμε ένα deep neural net στο να αναγνωρίζει τα συναισθήματα αυτά. Στη συνέχεια, θα προκαλέσουμε drift στο dataset μας, εισάγοντας στον ταξινομητή εικόνες των χρηστών με άλλες συνθήκες φωτισμού, περιβάλλοντος, κλπ., τροποποιημένες – αλλοιωμένες εικόνες των χρηστών, εικόνες άλλων χρηστών, κλπ., και θα δούμε οι αλγόριθμοί μας είναι σε θέση να αντιληφθούν τις μεταβολές αυτές, και αν μπορούν να βελτιώσουν την απόδοση της ταξινόμησης, και σε ποιο βαθμό. Τέλος, ανακεφαλαιώνουμε τα συμπεράσματά που βγάλαμε, και προτείνουμε πιθανές βελτιώσεις.

**Λέξεις κλειδιά**

Προσαρμοζόμενη μάθηση, μάθηση σε μεταβαλλόμενα περιβάλλοντα, μεταβολή πλαισίου, νευρωνικά δίκτυα, βαθιά νευρωνικά δίκτυα, συνελικτικά νευρωνικά δίκτυα.

# Abstract

The goal to this thesis is the study of a relatively new branch of machine learning, called learning in nonstationary environments, and to apply some of its techniques and methods to an emotion recognition by image problem.
Most usual machine learning algorithms (neural networks, SVMs, random forests, etc.) operate in the following way: initially, they are being trained on a known data set (train data), and then we apply them to the rest of the data we want to classify (test data). This process makes a fundamental assumption: that the statistical distribution of test data is the same with the statistical distribution of train data. In this way, a machine learning algorithm learns the properties of the data from the training dataset, and then it applies this knowledge to the rest of the data.
Yet there are some cases where the above assumption is erroneous. There is a quite large variety of phenomena, where the statistical distribution of their data changes with time. Some examples include climate changes, demographic data, recommender systems, where the user's preferences change over time as he grows, robots that interact with their environment, etc. In such cases, we need methods which are able to detect the statistical variations of the data, and then adjust the classifier accordingly, by adding the new knowledge to it. This branch of machine learning is called adaptive learning, of learning in nonstationary environments, whereas the phenomenon of the variation of the dataset's statistical parameters is called concept drift.
In this work, we will present and recapitulate the most important ideas and techniques of adaptive learning. Subsequently, in the experimental part, we will try to apply some of these techniques in the area of emotion recognition by image. By using a database containing photos of some users and the respective emotions, we will train a deep neural net to recognize these emotions. Then we will cause a drift to our dataset, by introducing user images taken in other light – environment conditions, modifying the users images, adding photos from new users, etc., and we shall see whether our algorithms will be able to detect the drifts, and improve the classification performance, and to what extent. Finally, we will sum up the conclusions we made, and suggest possible improvements.

**Key words**

Adaptive learning, learning in nonstationary environments, concept drift, neural networks, deep neural networks, convolutional neural networks.

# Περιεχόμενα

# 1 Τεχνητά νευρωνικά δίκτυα και Βαθιά μάθηση

## 1.1 Νευρωνικά δίκτυα πολλών επιπέδων

Τα τεχνητά νευρωνικά δίκτυα (artificial neural networks) είναι μία τεχνική της μηχανικής μάθησης, η οποία προσπαθεί να μιμηθεί τη λειτουργία του εγκεφάλου. Συγκεκριμένα, ένα νευρωνικό δίκτυο είναι ένα δίκτυο από απλές μονάδες – νευρώνες, οι οποίοι είναι συνδεμένοι μεταξύ τους. Κάθε νευρώνας δέχεται στις εισόδους του σήματα, είτε από το περιβάλλον, είτε από άλλους νευρώνες, τα οποία στη συνέχεια επεξεργάζεται εσωτερικά. Τέλος, το αποτέλεσμα οδηγείται στην έξοδο του νευρώνα, και διοχετεύεται είτε σε άλλους νευρώνες, είτε στην έξοδο του δικτύου. Οι συνδέσεις μεταξύ των νευρώνων ονομάζονται συνάψεις, και χαρακτηρίζονται από μία τιμή βάρους. Οι τιμές των βαρών όλων των συνάψεων αποτελούν τη διαθέσιμη γνώση του δικτύου. Το μοντέλο αυτό εμπνέεται απ' τον εγκέφαλο, όπου θεωρείται γενικά ότι αποθηκεύει και επεξεργάζεται την πληροφορία μεταβάλλοντας τις συνάψεις μεταξύ των νευρώνων του.

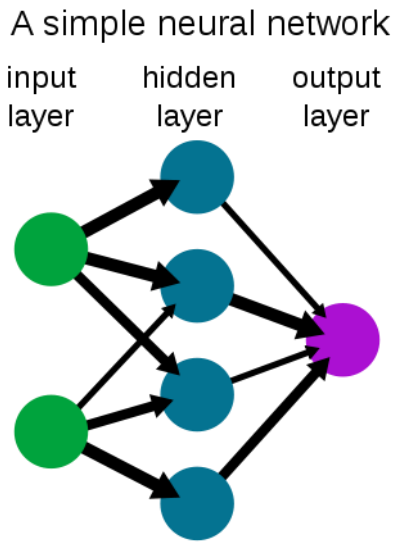
Το υπολογιστικό μοντέλο που χρησιμοποιεί ο κάθε νευρώνας είναι αρκετά απλό. Συγκεκριμένα, ο νευρώνας πολλαπλασιάζει τις τιμές που λαμβάνει στις εισόδους του με το αντίστοιχο βάρος της κάθε εισόδου – σύναψης, αθροίζει τα γινόμενα αυτά, και τα θέτει ως είσοδο σε μία εσωτερική συνάρτηση, τη λεγόμενη συνάρτηση ενεργοποίησης. Η τιμή που προκύπτει αποτελεί την έξοδο του νευρώνα, η οποία διοχετεύεται είτε σε άλλους νευρώνες για περαιτέρω επεξεργασία, είτε στο περιβάλλον. Συγκεκριμένα, εάν συμβολίσουμε με $x_{ki}$ την $i$-οστή είσοδο του $k$ νευρώνα, $w_{ki}$ το $i$-οστό συναπτικό βάρος του $k$ νευρώνα, και $\varphi(\cdot)$ τη συνάρτηση ενεργοποίησης, τότε η έξοδος $y_k$ του νευρώνα δίνεται απ' τη σχέση:

$$y_k = \varphi\left(\sum_{i=0}^{N} x_{ki} w_{ki}\right). \tag{1.1.1}$$

Εδώ πρέπει να σημειωθεί ότι στον $k$-οστό νευρώνα υπάρχει επιπλέον ένα συναπτικό βάρος $w_{k0}$, το οποίο καλείται πόλωση ή κατώφλι. Τέλος, συνηθισμένες συναρτήσεις ενεργοποίησης είναι η σιγμοειδής συνάρτηση, η γραμμική συνάρτηση, η βηματική, η συνάρτηση προσήμου, κλπ.

Η πιο συνήθης αρχιτεκτονική νευρωνικών δικτύων είναι τα πολυεπίπεδα δίκτυα εμπρόσθιας τροφοδότησης (feedforward multilayer neural networks). Σε αυτά τα δίκτυα, οι νευρώνες κατανέμονται σε διαδοχικά επίπεδα ή στρώματα. Στο πρώτο επίπεδο βρίσκονται οι νευρώνες εισόδου, οι οποίοι λαμβάνουν τις εισόδους τους από το περιβάλλον, ενώ στο τελευταίο επίπεδο βρίσκονται οι νευρώνες εξόδου, οι

οποίοι παράγουν τη συνολική έξοδο του δικτύου. Τέλος, οι νευρώνες που βρίσκονται στα ενδιάμεσα επίπεδα ονομάζονται υπολογιστικοί νευρώνες, και είναι υπεύθυνοι για την εσωτερική επεξεργασία των δεδομένων. Επίσης, σε ένα δίκτυο εμπρόσθιας τροφοδότησης, οι συνδέσεις μεταξύ των νευρώνων γίνονται από το ένα επίπεδο στο επόμενο, δηλ. ένας νευρώνας στο επίπεδο $k$ λαμβάνει εισόδους μόνο απ' το επίπεδο $k-1$. Άλλες συνδέσεις απαγορεύονται (συναντώνται φυσικά σε άλλες αρχιτεκτονικές νευρωνικών δικτύων).
Ένα απλό δίκτυο εμπρόσθιας τροφοδότησης φαίνεται στην παρακάτω εικόνα:




*Σχήμα 1.1.1: Ένα απλό νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης.*

## 1.2 Η εκπαίδευση των νευρωνικών δικτύων – ο αλγόριθμος Back – Propagation

Το κύριο χαρακτηριστικό των νευρωνικών δικτύων, όπως άλλωστε και των περισσοτέρων αλγορίθμων μηχανικής μάθησης, είναι η ικανότητά τους να εκπαιδεύονται, με στόχο την επίλυση ενός προβλήματος. Στην περίπτωση των νευρωνικών δικτύων εμπρόσθιας τροφοδότησης, η εκπαίδευση συνίσταται στην εύρεση των σωστών τιμών των συναπτικών βαρών, έτσι ώστε, για κάθε είσοδο, το δίκτυο να δίνει την επιθυμητή έξοδο, ή μία καλή προσέγγισή της.
Ο πλέον διαδεδομένος αλγόριθμος εκπαίδευσης των νευρωνικών δικτύων είναι ο αλγόριθμος της ανάστροφης διάδοσης σφάλματος (Error Back – propagation, ή απλά back – propagation, [1], [3]). Ο στόχος του αλγόριθμου αυτού είναι, με δεδομένο ένα σύνολο εκπαίδευσης, δηλ. ένα σύνολο ζευγών εισόδου – επιθυμητής εξόδου, να βρει τις κατάλληλες τιμές των βαρών, έτσι ώστε να ελαχιστοποιήσει μία συνάρτηση κόστους, η οποία μετράει την απόκλιση της εξόδου του δικτύου απ' την επιθυμητή. Αυτό επιτυγχάνεται μέσω της εφαρμογής μίας συνήθους τεχνικής της θεωρίας βελτιστοποίησης, του αλγορίθμου της ταχύτερης καθόδου (gradient descend), στην περίπτωση της συνάρτησης κόστους του δικτύου.

Ας δούμε τα πράγματα αναλυτικά: Έστω ότι μας δίνεται ένα σύνολο εκπαίδευσης που αποτελείται από $N$ ζεύγη εισόδων – επιθυμητών εξόδων της μορφής $(\boldsymbol{x}, \boldsymbol{d})$, όπου το $\boldsymbol{x}(i)$, $i = 1,2, \dots, N$ είναι ένα διανύσμα εισόδου, και $\boldsymbol{d}(i)$ το επιθυμητό διάνυσμα εξόδου που αντιστοιχεί στο $\boldsymbol{x}(i)$ (τα δύο διανύσματα δεν έχουν απαραίτητα την ίδια διάσταση). Ας θεωρήσουμε τώρα έναν νευρώνα του επιπέδου εξόδου, έστω τον νευρώνα $j$, όπου $j = 1,2, \dots, M$, και $M$ ο αριθμός των νευρώνων εξόδου του δικτύου. Το σφάλμα στην έξοδο του νευρώνα αυτού, όταν παρουσιάζεται στην είσοδο του δικτύου το πρότυπο $\boldsymbol{x}(n)$, με $n = 1,2, \dots, N$, δίνεται από τη σχέση:

$$e_j(n) = d_j(n) - y_j(n), \tag{1.2.1}$$

όπου $y_j(n)$ η έξοδος του νευρώνα $j$, όταν στο δίκτυο δίνεται ως είσοδος το πρότυπο $\boldsymbol{x}(n)$.
Συνήθως, είναι μαθηματικά πιο εύκολο να δουλεύουμε με το τετραγωνικό σφάλμα,

$$\frac{1}{2} {e_j}^2(n). \tag{1.2.2}$$

Με βάση αυτό, το συνολικό σφάλμα του δικτύου όταν παρουσιάζεται στη είσοδο το πρότυπο $\boldsymbol{x}(n)$ δίνεται από τη σχέση:

$$G(n) = \frac{1}{2} \sum_j {e_j}^2(n). \tag{1.2.3}$$

Η συνάρτηση $G(n)$ είναι η συνάρτηση κόστους του προβλήματος, την οποία έχει σαν στόχο να ελαχιστοποιήσει ο αλγόριθμος back – propagation. Συγκεκριμένα, για κάθε $n$, ο αλγόριθμος μεταβάλλει λίγο τα βάρη του δικτύου απ' τις προηγούμενες τιμές τους, με στόχο την ελαχιστοποίηση της $G(n)$. Συγκεκριμένα, σε κάθε επανάληψη, ο αλγόριθμος back – propagation προκαλεί μία ανανέωση – διόρθωση $\Delta w_{ji}(n)$ στο βάρος $w_{ji}(n)$ της σύνδεσης μεταξύ του νευρώνα $j$ κάποιου επιπέδου του δικτύου και του νευρώνα $i$ του προηγούμενου επιπέδου. Η ανανέωση γίνεται από μπροστά προς τα πίσω: πρώτα ανανεώνονται τα βάρη του επιπέδου εξόδου, και στη συνέχεια, με βάση τις τιμές που προκύπτουν, ανανεώνονται διαδοχικά τα βάρη των εσωτερικών επιπέδων, μέχρι να φτάσουμε στο επίπεδο εισόδου. Εξ' ου και το όνομα του αλγορίθμου.

Αυτό που θέλει να πετύχει στην ουσία ο αλγόριθμος, είναι να βρει τις τιμές των βαρών $w_{ji}(n)$ για τις οποίες θα ισχύει

$$\frac{\partial G(n)}{\partial w_{ji}(n)} = 0, \tag{1.2.4}$$

αφού στο σημείο αυτό η συνάρτηση $G(n)$ γίνεται ελάχιστη, όπως πρέπει. Και εδώ έρχεται και εφαρμόζεται ο αλγόριθμος gradient descend: σύμφωνα με τον

αλγόριθμο αυτό, αποδεικνύεται ότι, εάν $w_{ji}(n)$ τα αρχικά βάρη – ορίσματα της $G(n)$ τότε, με κατάλληλη επιλογή της παραμέτρου $\eta$, τα νέα ορίσματα $w_{ji}'(n)$, με

$$w_{ji}^{\prime(n)} = w_{ji}(n) - \eta \frac{\partial G(n)}{\partial w_{ji}(n)} \tag{1.2.5}$$

θα βρίσκονται πιο κοντά στο ελάχιστο της $G(n)$. Η παράμετρος $\eta$ ονομάζεται ρυθμός μάθησης, και πρέπει να επιλεγεί με προσοχή: μικρή τιμή οδηγεί σε αργή σύγκλιση προς το ελάχιστο – άρα όχι καλή εκπαίδευση, ενώ μία μεγάλη τιμή οδηγεί κάποιες φορές στο να προσπεράσουμε το ορθό ελάχιστο, και να καταλήξουμε σε ένα άλλο σημείο, πράγμα που σημαίνει επίσης κακή εκπαίδευση. Για το λόγο αυτό, υπάρχουν συνήθως στα πακέτα νευρωνικών δικτύων ενσωματωμένες μέθοδοι, οι οποίες επιλέγουν αρχικά μία σχετικά μεγάλη τιμή του $\eta$, ώστε να έχουμε γρήγορη σύγκλιση προς το ελάχιστο, και κατόπιν σταδιακά το μειώνουν, ώστε ο αλγόριθμος να μην το παρακάμψει.
Σε κάθε περίπτωση, από την παραπάνω εξίσωση βρίσκουμε ότι η ανανέωση του κάθε βάρους δίνεται απ' τη σχέση:

$$\Delta w_{ji}(n) = w_{ji}^{\prime(n)} - w_{ji}(n) \Leftrightarrow \Delta w_{ji}(n) = -\eta \frac{\partial G(n)}{\partial w_{ji}(n)}. \tag{1.2.6}$$

Κάνοντας τις απαραίτητες παραγωγίσεις, μπορούμε εύκολα να δείξουμε ότι η παραπάνω εξίσωση γράφεται ως εξής:

$$\Delta w_{ji}(n) = -\eta \delta_j(n) y_i(n), \tag{1.2.7}$$

όπου το $\delta_j$ είναι το σφάλμα στην έξοδο του νευρώνα $j$ του επιπέδου $m$, και $y_i$ είναι η έξοδος του νευρώνα $i$ του επιπέδου $m-1$, ή το $i$-οστό στοιχείο του διανύσματος εισόδου, αν $m = 1$.
Συνεπώς, βλέπουμε ότι οι τιμές των $\delta_j$ υπολογίζονται με βάση τις τιμές των εξόδων των νευρώνων του δικτύου, καθώς και τις τιμές των βαρών των συνδέσεων του κατά την επανάληψη $n$. Για παράδειγμα, στην περίπτωση που η συνάρτηση ενεργοποίησης των νευρώνων είναι η σιγμοειδής συνάρτηση, τότε, για το επίπεδο της εξόδου έχουμε τη σχέση

$$\delta_k(n) = y_k(n)[1 - y_k(n)][d_k(n) - y_k(n)], \tag{1.2.8}$$

για τον νευρώνα $k$, ενώ για τα ενδιάμεσα επίπεδα το $\delta$ γράφεται στη μορφή

$$\delta_k(n) = y_k(n)[1 - y_k(n)] \sum_j \delta_j(n) w_{jk}, \tag{1.2.9}$$

όπου το άθροισμα του δεξιού μέλους αναφέρεται στα σφάλματα $\delta_j$ του ανώτερου επιπέδου.

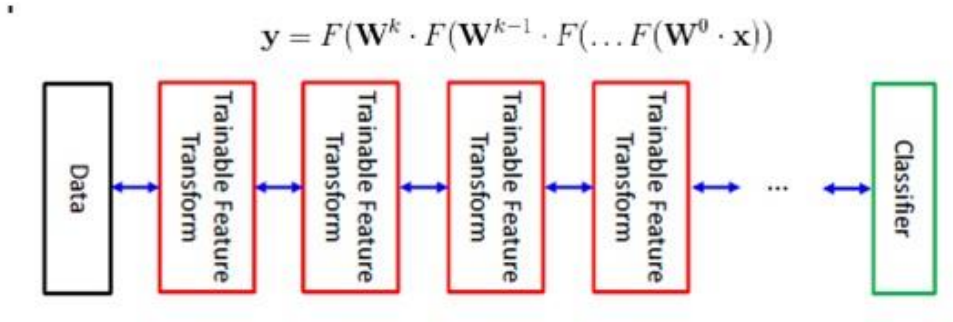
Με βάση τα παραπάνω, ο αλγόριθμος back – propagation λειτουργεί ως εξής: αρχικά, αρχικοποιούμε τα βάρη του δικτύου σε τυχαίες θετικές τιμές. Στη συνέχεια, για κάθε πρότυπο εκπαίδευσης, εκτελούμε τον αλγόριθμο σε δύο φάσεις. Στην πρώτη φάση (forward pass) υπολογίζουμε, με βάση το πρότυπο εισόδου, διαδοχικά την έξοδο του δικτύου. Στη συνέχεια, στη δεύτερη φάση (reverse pass), ο αλγόριθμος ξεκινά απ' την έξοδο και διαδίδει τα σήματα σφάλματος προς τα προηγούμενα επίπεδα, ανανεώνοντας τα βάρη, μέχρι να φτάσουμε στο επίπεδο εισόδου.

## 1.3 Βαθιά νευρωνικά δίκτυα

Όπως γνωρίζουμε, ένα νευρωνικά δίκτυο εμπρόσθιας τροφοδότησης με τουλάχιστον δύο επίπεδα μπορεί θεωρητικά να προσεγγίσει οποιαδήποτε συνεχή συνάρτηση $\boldsymbol{d} = f(\boldsymbol{x})$ εισόδου – εξόδου. Παρόλα αυτά, στην πράξη αποδεικνύεται ότι τα νευρωνικά δίκτυα δύο ή τριών επιπέδων δεν είναι τόσο αποτελεσματικά όταν εκπαιδεύονται σε σύνθετα προβλήματα που περιλαμβάνουν πολύπλοκες συναρτήσεις, όπως π.χ. στην όραση υπολογιστών. Στην περίπτωση αυτή, έχει αποδειχθεί ότι η χρήση ειδικών νευρωνικών δικτύων με πολλά επίπεδα δίνει πολύ καλύτερα αποτελέσματα. Τα νευρωνικά δίκτυα αυτά ονομάζονται βαθιά νευρωνικά δίκτυα (deep neural networks), ενώ το πεδίο της μηχανικής μάθησης που ασχολείται τέτοιες δομές βαθιάς αρχιτεκτονικής ονομάζεται βαθιά μηχανική μάθηση (deep learning, [4]).

Ο βασικός λόγος που οι ρηχές (shallow) αρχιτεκτονικές αποτυγχάνουν στην περίπτωση τέτοιων πολύπλοκων προβλημάτων, είναι επειδή εφαρμόζονται και επεξεργάζονται απευθείας στα δεδομένα εισόδου, δηλ. προσπαθούν να κάνουν την ταξινόμηση με βάση κυρίως αυτά που βλέπουν στην είσοδο. Αυτό είναι λάθος, διότι στην περίπτωση π.χ. της αναγνώρισης αντικειμένων από εικόνα (ένα τυπικό πρόβλημα της αρχιτεκτονικής υπολογιστών), ένα ρηχό νευρωνικό δίκτυο προσπαθεί να κατατάξει την εικόνα κυρίως με βάση τα πίξελ της, αφού αυτά είναι τα πρωτογενή δεδομένα που λαμβάνει το δίκτυο.

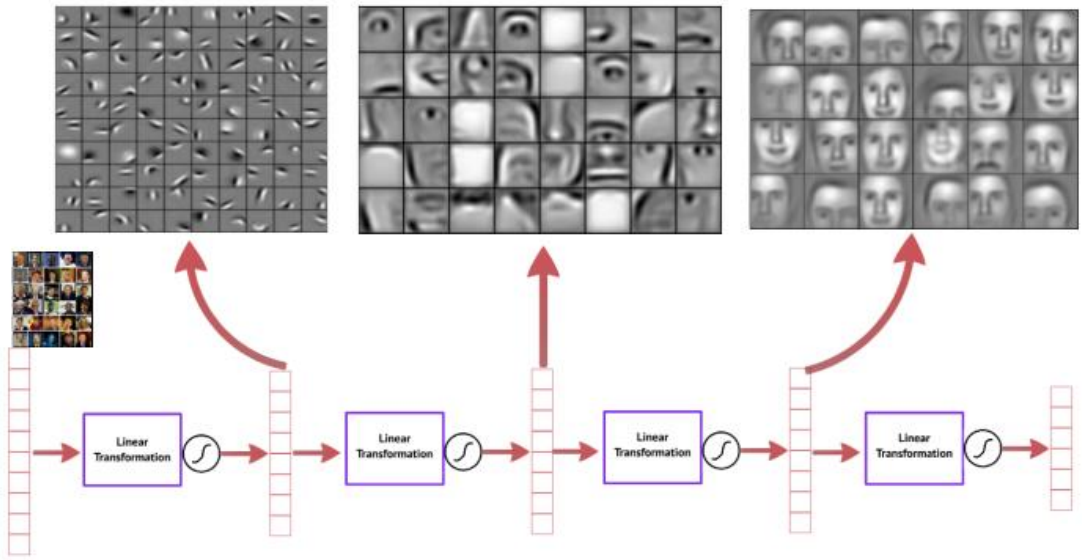
Αντίθετα, οι βαθιές αρχιτεκτονικές έχουν την εξής δυνατότητα: Τα διάφορα επίπεδα μίας βαθιάς αρχιτεκτονικής έχουν την ικανότητα να μετατρέπουν μία αναπαράσταση χαμηλού επιπέδου (π.χ. πίξελ), σε μία αναπαράσταση υψηλότερου επιπέδου (π.χ. γωνίες). Δηλαδή, αν φτιάξουμε ένα δίκτυο με κατάλληλη αρχιτεκτονική, τα διάφορα επίπεδά του αρχίζουν να εξάγουν σιγά σιγά αναπαραστάσεις υψηλότερου επιπέδου το ένα από το άλλο. Τα χαρακτηριστικά αυτά είναι γενικά περισσότερο διαχωρίσιμα και ουσιώδη για την ταξινόμηση απ' ότι τα χαρακτηριστικά χαμηλού επιπέδου, οπότε έναν ανώτερο επίπεδο μπορεί να τα κατατάξει στη συνέχεια πιο εύκολα. Η όλη μέθοδος φαίνεται στο παρακάτω σχήμα:




*Σχήμα 1.3.1: Μία αρχιτεκτονική βαθιάς μάθησης. Το χαρακτηριστικό των αρχιτεκτονικών αυτών είναι ότι κάθε επίπεδο μετατρέπει την είσοδό του σε μία αναπαράσταση υψηλότερου επιπέδου. Στο τέλος, η αναπαράσταση υψηλού επιπέδου που προκύπτει οδηγείται σε έναν συνήθη ταξινιμητή (π.χ. ένα απλό νευρωνικό δίκτυο, ένα SVM, κλπ.) για να ταξινομηθεί.*

Όπως βλέπουμε απ' το παραπάνω σχήμα, μία βαθιά αρχιτεκτονική αποτελείται από ένα σύνολο πολλών επιπέδων, το καθένα απ' τα οποία μετατρέπει τα δεδομένα που λαμβάνει σε μία αναπαράσταση υψηλότερου επιπέδου. Τέλος, η τελική αναπαράσταση οδηγείται σε έναν απλό ταξινομητή (π.χ. ένα απλό νευρωνικό ή ένα SVM) για να ταξινομηθεί. Για την κατανόηση του πως γίνεται να μετατραπούν τα δεδομένα σε αναπαραστάσεις υψηλότερης τάξεως χρησιμοποιούνται εργαλεία απ' τα μαθηματικά και τη στατιστική, όπως π.χ. η θεωρία των αραιών αναπαραστάσεων (sparse coding), κλπ.

Παρακάτω μπορούμε να δούμε ένα συγκεκριμένο παράδειγμα μίας βαθιάς αρχιτεκτονικής, η οποία αποτελείται από ένα βαθύ νευρωνικό δίκτυο με στόχο την αναγνώριση προσώπου. Όπως βλέπουμε απ' το σχήμα, με είσοδο απλές εικόνες, το πρώτο επίπεδο αναγνωρίζει γωνίες (δηλαδή στην ουσία εκφράζει την είσοδο ως έναν «γραμμικό συνδυασμό γωνιών»). Στη συνέχεια, με δεδομένες τις γωνίες, το δεύτερο επίπεδο τις συνδυάζει και αναγνωρίζει πιο πολύπλοκα χαρακτηριστικά, όπως π.χ. μάτια, μύτες, κλπ. Τέλος, με αυτά ως είσοδο, το τρίτο επίπεδο αναγνωρίζει αναπαραστάσεις του προσώπου, δηλ. η είσοδος έχει πλέον εκφραστεί ως ένας γραμμικός συνδυασμός κάποιας βάσης προσώπων (eigenfaces). Η τελική αυτή αναπαράσταση μπορεί να οδηγηθεί σε ένα SVM λόγου χάρη, το οποίο θα μας δώσει την ταυτότητα του χρήστη.




*Σχήμα 1.3.2: Ένα σχηματικό παράδειγμα του τρόπου λειτουργίας μίας βαθιάς αρχιτεκτονικής για αναγνώριση εικόνας.*

Εδώ πρέπει να αναφερθεί ότι, ανάλογα με το πρόβλημα που έχουμε να λύσουμε, οι διάφορες υπομονάδες της βαθιάς αρχιτεκτονικής μπορούν να εκτελούν και άλλες λειτουργίες, όπως π.χ. κανονικοποίηση της εισόδου, φιλτράρισμα, μη γραμμικούς μετασχηματισμούς για αραίωση ή πύκνωση στο χώρο των χαρακτηριστικών, κ.α.

Η εκπαίδευση των βαθιών αρχιτεκτονικών γίνεται συνήθως με τους ακόλουθους τρόπους:

- Επιβλεπόμενη μάθηση: Εδώ πρόκειται για τη συνηθισμένη μέθοδο εκπαίδευσης που έχουμε και στα απλά νευρωνικά δίκτυα. Η εκπαίδευση γίνεται επιβλεπόμενα, με κάποια παραλλαγή του αλγορίθμου back – propagation, π.χ. στη στοχαστική εκδοχή του (stochastic back propagation), για λόγους ταχύτητας. Φυσικά, για να πετύχουμε ικανοποιητική εκπαίδευση θα πρέπει να έχουμε πάρα πολλά δεδομένα, λόγω των πολλών παραμέτρων της αρχιτεκτονικής.

- Μη επιβλεπόμενη μάθηση με επιβλεπόμενο ταξινομητή στην έξοδο: Η μέθοδος εκπαιδεύει κάθε επίπεδο διαδοχικά χωρίς επίβλεψη, ενώ εκπαιδεύει έναν επιβλεπόμενο ταξινομητή στην έξοδο. Στην ουσία, ο ταξινομητής μαθαίνει τις αναπαραστάσεις του δικτύου. Η μέθοδος αυτή είναι χρήσιμη όταν είναι διαθέσιμα πολύ λίγα κατηγοριοποιημένα δείγματα.

- Μη επιβλεπόμενη μάθηση με επιβλεπόμενο εξομαλυντή: Η μέθοδος αυτή εκπαιδεύει το κάθε επίπεδο χωρίς επίβλεψη, προσθέτει έναν ταξινομητή στην έξοδο, και επανεκπαιδεύει όλο το σύστημα με επίβλεψη. Η μέθοδος αυτή είναι χρήσιμη όταν το σύνολο των labels είναι μικρό.

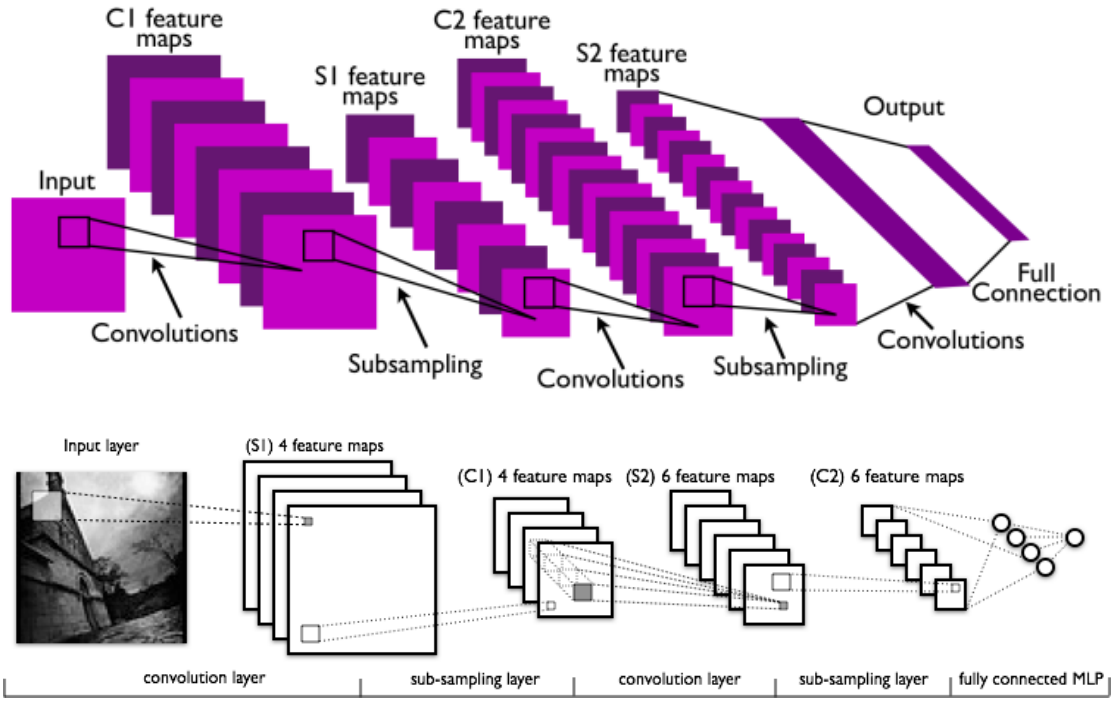
Τα τελευταία χρόνια, οι μεθοδολογίες της βαθιάς μάθησης έχουν οδηγήσει σε πολύ μεγάλη πρόοδο στον τομέα της μηχανικής μάθησης σε πολλούς τομείς, από την

αναγνώριση εικόνας και ομιλίας, μέχρι την αυτόματη οδήγηση, την τεχνητή νοημοσύνη σε παιχνίδια (π.χ. το Alpha Go της Google), καθώς και σε άλλους τομείς. Για το λόγο αυτό, το πεδίο του deep learning είναι σήμερα ένα από τα πεδία με τη μεγαλύτερη έρευνα στο χώρο της πληροφορικής. Φυσικά, αυτό κατέστη δυνατό λόγω της αύξησης της ισχύος των επεξεργαστών, π.χ. λόγω του νόμου του Moore, αλλά ιδιαίτερα και των μονάδων επεξεργασίας γραφικών (GPUs), αφού η δυνατότητά τους να εκτελούν υπολογισμούς με παράλληλο τρόπο τις καθιστά ιδανικές για την εκπαίδευση μεγάλων δικτύων με εκατομμύρια παραμέτρους!

## 1.4 Συνελικτικά νευρωνικά δίκτυα

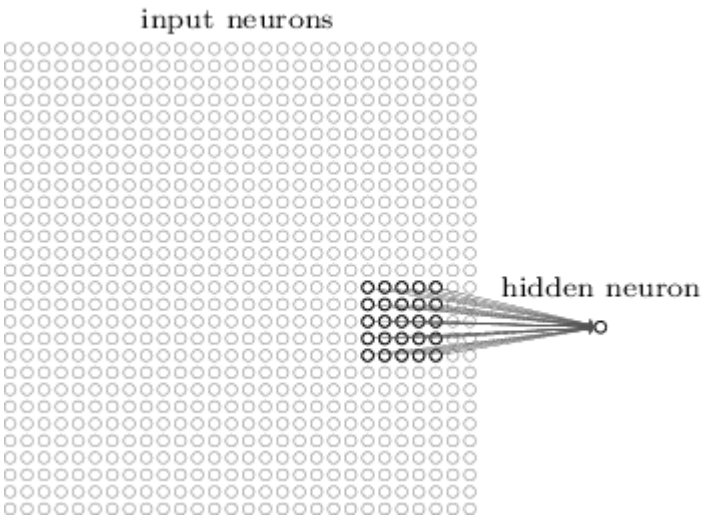
Τα συνελικτικά νευρωνικά δίκτυα (convolutional neural networks – CNN), είναι μία σχετικά καινούργια αρχιτεκτονική νευρωνικών δικτύων εμπρόσθιας τροφοδότησης, η οποία προτάθηκε για πρώτη φορά απ' τον Y. Le Cun το 1989 ([4]). Τα συνελικτικά νευρωνικά δίκτυα αντλούν ιδέες από τη βιολογία και τη νευροεπιστήμη, και συγκεκριμένα από εργασίες που αφορούν τη λειτουργία της οπτικής αντίληψης της γάτας των Hubel και Wiesel, η οποία περιλαμβάνει πολλαπλά επίπεδα επεξεργασίας και διάφορες αναπαραστάσεις. Ο λόγος γι' αυτό είναι ότι τα δίκτυα αυτά είχαν αναπτυχθεί αρχικά κυρίως για τη χρήση τους σε προβλήματα αναγνώρισης εικόνας.
Γενικά, έναν συνελικτικό νευρωνικό δίκτυο είναι κατάλληλο για περιπτώσεις δισδιάστατων δεδομένων εισόδου, όπου τα κοντινά δεδομένα έχουν μία πιο ισχυρή συσχέτιση μεταξύ τους. Στην αναγνώριση εικόνας έχουμε ακριβώς αυτήν την περίπτωση, αφού τα γειτονικά pixel της εικόνας έχουν άμεση σχέση μεταξύ τους, με την έννοια του όχι σχηματίζουν κάποια ανώτερα χαρακτηριστικά, τα οποία είναι σημαντικά για την αναγνώριση (π.χ. γραμμές ή γωνίες). Για το λόγο αυτό, τα βαθιά νευρωνικά δίκτυα έχουν εφαρμοστεί εκτενώς τα τελευταία χρόνια σε προβλήματα αναγνώρισης εικόνας (π.χ. αναγνώριση χειρόγραφων ψηφίων, αναγνώριση πινακίδων, αυτόματη οδήγηση, κλπ.) με αξιοσημείωτη επιτυχία. Υλοποιήσεις των δικτύων αυτών χρησιμοποιούνται σε πολλές εφαρμογές μεγάλων εταιριών, όπως π.χ. της Google.

Ένα βαθύ συνελικτικό δίκτυο περιλαμβάνει κυρίως δύο ειδών επίπεδα: επίπεδα συνέλιξης (convolution layers), και επίπεδα υποδειγματοληψίας – συγκέντρωσης (subsampling – polling layers). Τέλος, στην έξοδο βρίσκεται συνήθως ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης, το οποίο υπολογίζει την έξοδο. Η όλη αρχιτεκτονική του δικτύου φαίνεται παρακάτω:




*Σχήμα 1.4.1: Η αρχιτεκτονική ενός συνελικτικού νευρωνικού δικτύου.*

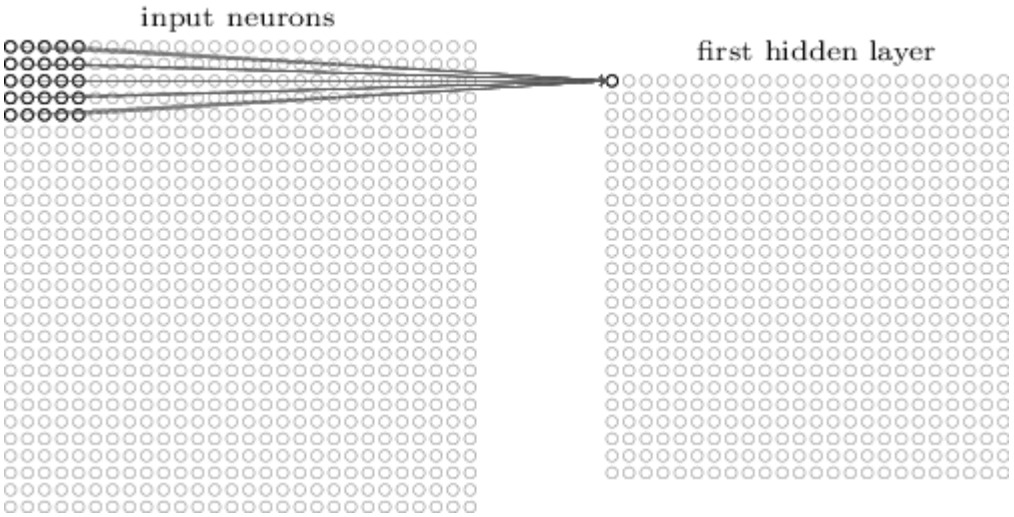
Ας δούμε αναλυτικά τι κάνει το κάθε επίπεδο. Αρχικά, η είσοδος του δικτύου είναι μία δισδιάστατη εικόνα ή πίνακας, όπως φαίνεται στο παραπάνω σχήμα. Στη συνέχεια ακολουθεί ένα επίπεδο συνέλιξης. Το επίπεδο αυτό είναι ένα σύνολο νευρώνων, όπως στα συνήθη νευρωνικά δίκτυα, με μία διαφορά: ο κάθε νευρώνας δεν λαμβάνει ως είσοδο όλα τα pixel της εικόνας, αλλά μόνο τα pixel μίας μικρής περιοχής (π.χ. 5x5 pixel), όπως φαίνεται στο παρακάτω σχήμα:




*Σχήμα 1.4.1: Στο επίπεδο συνέλιξης, ένας νευρώνας δεν συνδέεται με ολόκληρη την εικόνα, αλλά με μία μικρή περιοχή.*

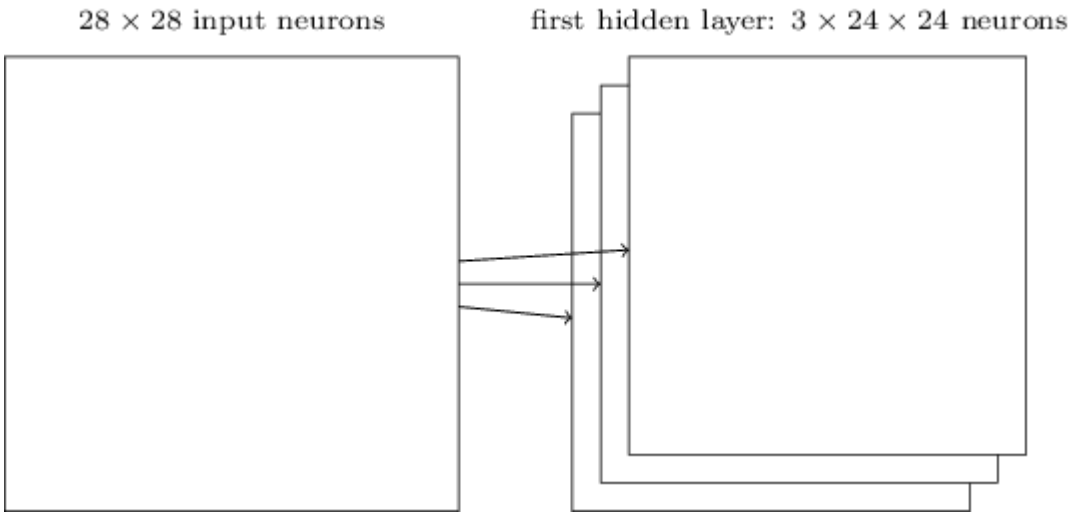
Ο λόγος για αυτό είναι, ότι σε μία δισδιάστατη εικόνα, αναμένουμε τα pixel που είναι γειτονικά μεταξύ τους να έχουν μεγάλη σχέση, ενώ αντίθετα, τα απομακρυσμένα pixel θα έχουν μικρή σχέση. Συνεπώς, για να αναγνωρίσουμε τα

χαρακτηριστικά μίας εικόνας, αρκεί να την εξετάσουμε τοπικά. Αυτή είναι μία απ' τις βασικές ιδέες των συνελικτικών νευρωνικών δικτύων. Από εδώ προέρχεται και η ονομασία, αφού η περιοχή εισόδου του κρυφού νευρώνα μπορεί να θεωρηθεί ότι προκύπτει απ' τη συνέλιξη της εικόνας εισόδου με ένα κατάλληλο παράθυρο.
Συνεπώς, κάθε περιοχή της εικόνας οδηγείται ως είσοδος σε έναν κρυφό νευρώνα του συνελικτικού επιπέδου, όπως φαίνεται παρακάτω:




*Σχήμα 1.4.3: Κάθε περιοχή της εικόνας εισόδου οδηγείται σε έναν κρυφό νευρώνα.*

Ο κρυφός νευρώνας αναμένεται να αναγνωρίσει στην περιοχή αυτή κάποιο χαρακτηριστικό χαμηλού επιπέδου, π.χ. μία γραμμή. Όμως, μία γραμμή θα μπορούσε να βρίσκεται και σε κάποιο άλλο σημείο της εικόνας, η οποία να συνδέεται με έναν άλλο νευρώνα. Από τη στιγμή που θέλουμε να αναγνωρίσουμε χαρακτηριστικά σε εικόνες, θα πρέπει να μπορούμε να πετύχουμε αναγνώριση της γραμμής σε οποιοδήποτε σημείο και αν βρίσκεται. Για το λόγο αυτό, σε ένα επίπεδο συνέλιξης, απαιτούμε τα βάρη όλων των περιοχών να είναι ίσα μεταξύ τους, έτσι ώστε το ίδιο χαρακτηριστικό να μπορεί να ανιχνευτεί παντού στην εικόνα. Επίσης, λόγω της τοπικότητας των χαρακτηριστικών, αναμένουμε ότι τα διάφορα pixel μέσα σε μία περιοχή θα είναι λίγο πολύ ισοδύναμα. Για το λόγο αυτό, θέτουμε και στα βάρη κάθε περιοχής ίσες τιμές (επίσης, με τον τρόπο αυτό πετυχαίνουμε την ανεξαρτησία των τοπικών χαρακτηριστικών από τις στροφές). Η απαίτηση αυτή πρέπει να διατηρείται και κατά την εκπαίδευση, πράγμα που επιτυγχάνεται αλλάζοντας λίγο τον αλγόριθμο back – propagation.
Παρόλα αυτά, στην παραπάνω προσέγγιση, όλοι οι νευρώνες μας αναγνωρίζουν μόνο ένα χαρακτηριστικό (π.χ. γραμμές). Μία εικόνα όμως μπορεί να έχει και άλλα χαρακτηριστικά, όπως π.χ. γωνίες. Για το λόγο αυτό, βάζουμε επιπλέον επίπεδα με κρυφούς νευρώνες, οι οποίοι θα εξειδικεύονται στην αναγνώριση ενός δεύτερου, τρίτου, κλπ., τοπικού χαρακτηριστικού. Συνεπώς, το συνολικό επίπεδο συνέλιξης θα έχει τελικά την εξής μορφή:



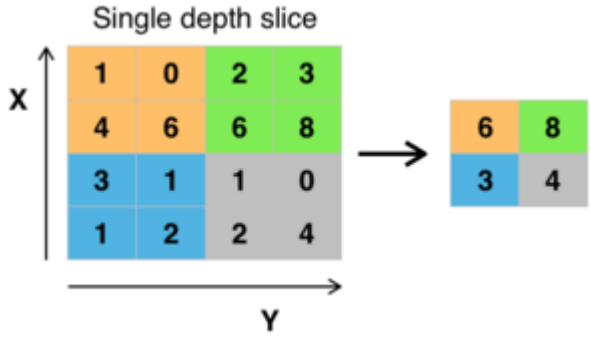

*Σχήμα 1.4.4: Το συνολικό επίπεδο συνέλιξης του δικτύου.*

Οι διάφοροι πίνακες κρυφών νευρώνων σε ένα επίπεδο συνέλιξης ονομάζονται χάρτες χαρακτηριστικών (feature maps), διότι ο καθένας ειδικεύεται σε ένα συγκεκριμένο χαρακτηριστικό.

Στη συνέχεια, μετά από ένα convolutional layer ακολουθεί ένα pooling layer. Τα επίπεδα συγκέντρωσης είναι στην ουσία επίπεδα φιλτραρίσματος. Μία μικρή περιοχή χαρακτηριστικών φιλτράρεται, έτσι ώστε να απομονωθεί η ουσιώδης πληροφορία που προσδιορίζει ένα τοπικό χαρακτηριστικό. Οι νευρώνες που κάνουν την υποδειγματοληψία ονομάζονται μονάδες ReLu (Rectified Linear Units). Η πιο συνηθισμένη συνάρτηση που χρησιμοποιούν οι μονάδες ReLu είναι η συνάρτηση μεγίστου, δηλ. η έξοδος μίας μονάδας ReLu δίνεται απ' τη σχέση:

$$Z = \max(0, A), \tag{1.4.1}$$

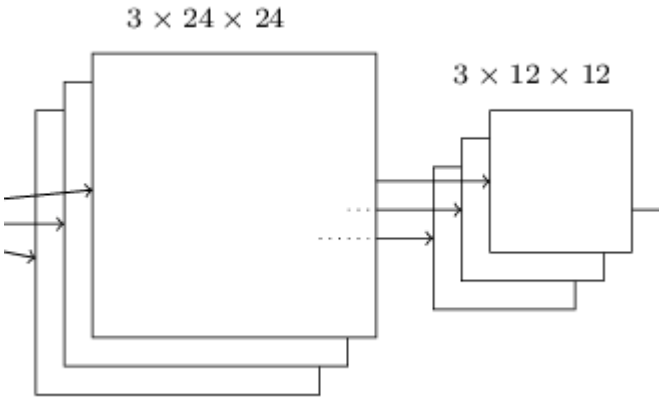
όπου $A$ είναι ο πίνακας εισόδου (μία συνήθης τιμή είναι το 2x2), και $Z$ η τιμή εξόδου. Η μέθοδος αυτή ονομάζεται max pooling – ένα παράδειγμα φαίνεται στο παρακάτω σχήμα:




*Σχήμα 1.4.5: Φιλτράρισμα max pooling.*

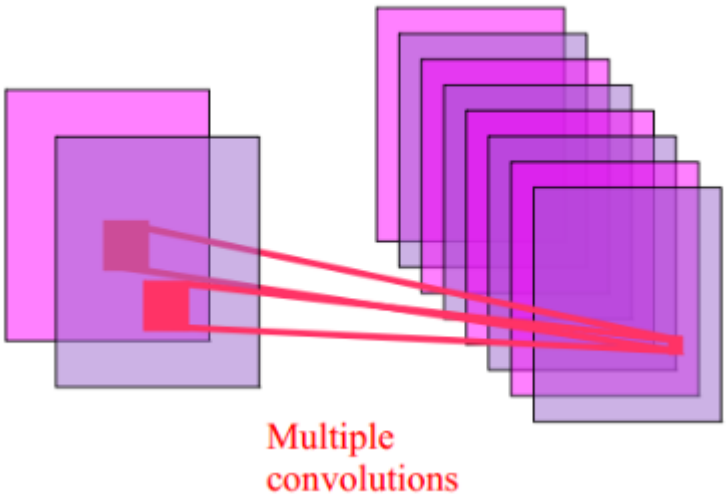
Η διαίσθηση πίσω απ' αυτή τη διαδικασία είναι ότι, αν θεωρήσουμε ότι το μέγιστο της περιοχής αντιπροσωπεύει ένα χαρακτηριστικό, η θέση του μέσα στην περιοχή

δεν έχει τόση σημασία, όσο η θέση του σε σχέση με τα άλλα χαρακτηριστικά. Συνολικά, το επίπεδο pooling φαίνεται παρακάτω:



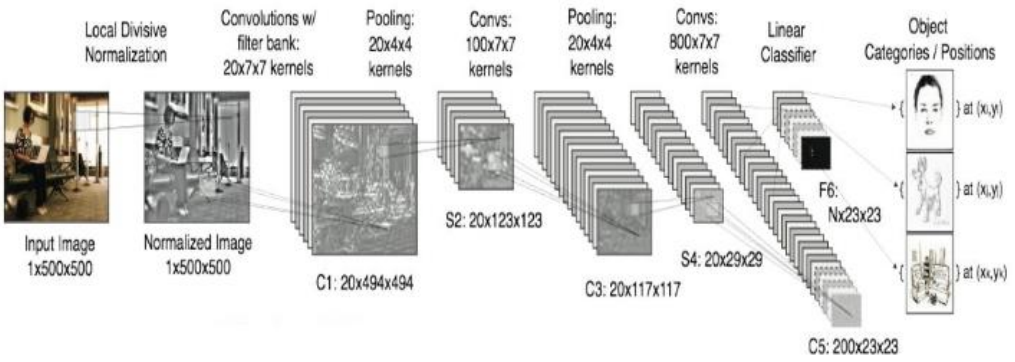

*Σχήμα 1.4.6: Το επίπεδο pooling ενός CNN.*

Στη συνέχεια, μετά από ένα επίπεδο pooling ακολουθεί και πάλι ένα επίπεδο συνέλιξης, έτσι ώστε να αναγνωριστούν τα χαρακτηριστικά ανώτερης τάξεως της εικόνας. Εδώ, λόγω των πολλών επιπέδων, η συνέλιξη είναι πολλαπλή, όπως στο παρακάτω σχήμα:




*Σχήμα 1.4.7: Ένα πολλαπλό συνελικτικό επίπεδο ενός CNN.*

Σε ένα τυπικό βαθύ CNN, τα επίπεδα συνέλιξης και συνένωσης επαναλαμβάνονται αρκετές φορές (τουλάχιστον δύο και πάνω), έτσι ώστε το δίκτυο να εξάγει τα χαρακτηριστικά ανώτερης τάξεως. Τέλος, μετά από το τελευταίο επίπεδο pooling ακολουθεί ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο, το οποίο κάνει την τελική ταξινόμηση. Η όλη διαδικασία φαίνεται παρακάτω:




*Σχήμα 1.4.8: Συνολική δομή ενός συνελικτικού δικτύου για ταξινόμηση εικόνων.*

Τα βαθιά συνελικτικά δίκτυα έχουν αναγνωριστεί ως μία από τις καλύτερες μεθόδους σε ένα μεγάλο σύνολο εφαρμογών όρασης υπολογιστών, όπως π.χ. η αναγνώριση χειρόγραφων ψηφίων, αριθμών σπιτιών στο StreetView, σημάτων κυκλοφορίας αναγνώρισης αντικειμένων στη βάση ImageNet, ενώ έχουν κερδίσει σε μία πληθώρα διαγωνισμών μηχανικής μάθησης και όρασης υπολογιστών.

## Η προσαρμογή των βαθιών νευρωνικών δικτύων (fine tuning)

Ένα ακόμη χαρακτηριστικό των βαθιών νευρωνικών δικτύων (και των νευρωνικών δικτύων γενικότερα), το οποίο θα μας χρησιμεύσει πολύ στα επόμενα, είναι η ικανότητα τους στο να προσαρμόζουν σχετικά εύκολα νέα γνώση. Ας υποθέσουμε για παράδειγμα ότι έχουμε εκπαιδεύσει ένα βαθύ νευρωνικό δίκτυο στο να αναγνωρίζει κάποια αντικείμενα (π.χ., το δίκτυο Alex Net έχει τη δυνατότητα να αναγνωρίζει αντικείμενα 1000 κατηγοριών). Ας υποθέσουμε ότι σε κάποια στιγμή φτάνουν στην είσοδο κάποια δεδομένα λίγο διαφορετικού τύπου, π.χ. κάποια νέα αντικείμενα. Χρειάζεται να επανεκπαιδεύσουμε όλο το δίκτυ απ’ την αρχή;
Η απάντηση είναι όχι! Αυτό μπορεί να γίνει αρκετά γρηγορότερα απ’ ότι η κυρίως εκπαίδευση με μία τεχνική που ονομάζεται fine – tuning.
Αυτό που κάνει το fine – tuning στην ουσία, είναι ότι εκπαιδεύει το δίκτυο χρησιμοποιώντας τα νέα δεδομένα, και αρχικοποιώντας τα βάρη στην παρούσα τιμή τους, και όχι σε τυχαίες τιμές, όπως γίνεται κατά τη συνήθη διαδικασία της εκπαίδευσης. Επειδή τα διαθέσιμα δεδομένα στο fine tuning είναι λίγα, ο σκοπός δεν είναι να γίνει επανεκπαίδευση του δικτύου – το βασικό σκεπτικό πίσω απ’ το fine tuning έχει ως εξής: Όπως είδαμε στο κεφάλαιο 1, σε ένα βαθύ νευρωνικό δίκτυο, τα χαμηλά επίπεδα, είναι επίπεδα convolution και pooling, τα οποία αυτό που κάνουν είναι να εξάγουν χαρακτηριστικά χαμηλότερου επιπέδου, δηλ. να μετατρέπουν τα raw pixel σε μία ανώτερη και πιο χρήσιμη αναπαράσταση. Στη συνέχεια, αυτά δίνονται ως είσοδοι στα ανώτερα επίπεδα, τα οποία είναι απλά πολυεπίπεδα perceptron, τα οποία εκτελούν την ταξινόμηση. Με βάση αυτό, το σκεπτικό πίσω απ’ το fine tuning είναι το εξής: δίνοντας στο δίκτυο ένα παρόμοιο, αλλά λίγο διαφορετικό σύνολο δεδομένων απ’ αυτό στο οποίο εκπαιδεύτηκε, τα χαρακτηριστικά χαμηλότερου επιπέδου που αναγνωρίζονται δεν αλλάζουν και πολύ: ένα επίπεδο convolution που ανιχνεύει λ.χ. γωνίες, θα συνεχίσει κατά βάση

να κάνει το ίδιο. Συνεπώς, δεν υπάρχει μεγάλη ανάγκη να επανεκπαιδεύσουμε τα κατώτερα επίπεδα του δικτύου. Αντίθετα, τα ανώτερα επίπεδα χρειάζονται επανεκπαίδευση, αφού τα ίδια χαρακτηριστικά χαμηλότερου επιπέδου μπορεί τώρα να αντιστοιχούν σε μία άλλη κλάση, οπότε ο MLP ταξινομητής στην κορυφή του δικτύου πρέπει να μάθει τα νέα δεδομένα. Συνεπώς, αυτό που κάνουμε στο fine tuning είναι το εξής: αρχίζοντας απ' τα τρέχοντα βάρη του δικτύου, συνεχίζουμε την εκπαίδευση στα νέα δεδομένα, αλλάζοντας όμως τους ρυθμούς μάθησης του δικτύου: στα χαμηλότερα επίπεδα ο ρυθμός μάθησης γίνεται πολύ μικρός, έτσι ώστε αυτά να τροποποιηθούν μόνο λίγο, αφού είναι κατά βάση έτοιμα, ενώ ο ρυθμός μάθησης των ανώτερων επιπέδων γίνεται αρκετά μεγάλος, έτσι ώστε τα επίπεδα αυτά να μάθουν γρήγορα τη νέα ταξινόμηση. Με τον τρόπο αυτό, μπορούμε να προσαρμόζουμε εύκολα βαθιά μοντέλα μάθησης σε νέα, παρόμοια δεδομένα, χρησιμοποιώντας μόνο λίγα σχετικά παραδείγματα εκπαίδευσης.

# 2 Μηχανική μάθηση σε μεταβαλλόμενα περιβάλλοντα

## 2.1 Εισαγωγή

Στις μέρες μας, η ανάπτυξη του Internet και της τεχνολογίας έχει αρχίσει να δημιουργεί έναν τεράστιο όγκο δεδομένων. Επίσης, νέες τεχνολογίες όπως το Internet of Things (IOT) αναμένεται να αυξήσουν κατακόρυφα τα δεδομένα αυτά. Η εκρηκτική αύξηση των δεδομένων, μαζί με την αύξηση της υπολογιστικής δύναμης των επεξεργαστών, έδωσε το έναυσμα για την μεγάλη έρευνα και πρόοδο στον τομέα της μηχανικής μάθησης που συντελείται στις μέρες μας, αφού προκύπτει η ανάγκη επεξεργασίας των δεδομένων αυτών για διάφορους λόγους. Επιπλέον, η δυνατότητα της χρησιμοποίησης των μεθόδων του machine learning στη ρομποτική και στην αυτοματοποίηση της παραγωγής έχουν δώσει επιπλέον έναυσμα στον τομέα αυτό.

Λόγω των παραπάνω, έχουν δημιουργηθεί με το πέρασμα του χρόνου εκατοντάδες παραδείγματα και μέθοδοι μηχανικής μάθησης. Από απλές μεθόδους, όπως π.χ. τα SVMs και τα νευρωνικά δίκτυα, έχουμε περάσει σε πιο εξελιγμένες τεχνικές όπως το deep learning, τα μαρκοβιανά μοντέλα, το data mining, κλπ.

Οι πιο συνηθισμένοι αλγόριθμοι μηχανικής μάθησης (νευρωνικά δίκτυα, SVM, Random forests, κλπ.) λειτουργούν με τον εξής τρόπο: αρχικά εκπαιδεύονται σε ένα γνωστό σύνολο δεδομένων (train data) και στη συνέχεια εφαρμόζονται στα υπόλοιπα δεδομένα που θέλουμε να ταξινομήσουμε (test data). Η διαδικασία αυτή κάνει μία θεμελιώδη παραδοχή: ότι η στατιστική κατανομή των test data είναι η ίδια με τη στατιστική κατανομή των train data. Με τον τρόπο αυτό, ένας αλγόριθμος μηχανικής μάθησης μαθαίνει τις ιδιότητες των δεδομένων απ’ τα test data, και τις εφαρμόζει μετά στα προς ταξινόμηση δεδομένα.

Υπάρχουν όμως κάποιες περιπτώσεις, όπου η παραπάνω υπόθεση είναι λανθασμένη. Σε μία αρκετά μεγάλη ποικιλία φαινομένων, η στατιστική κατανομή των δεδομένων μεταβάλλεται με το χρόνο. Μερικά παραδείγματα είναι οι κλιματολογικές συνθήκες, δημογραφικά δεδομένα, εισηγητικά συστήματα όπου οι προτιμήσεις του χρήστη αλλάζουν όσο αυτός μεγαλώνει, ρομπότ που αλληλεπιδρούν με το περιβάλλον, κλπ. Εδώ, ένα στατικό μοντέλο που εκπαιδεύτηκε σε κάποια χρονική στιγμή στο παρελθόν, θα αρχίσει σταδιακά να αποτυγχάνει, και το σφάλμα στην πρόβλεψη διαρκώς θα μεγαλώνει. Σε αυτές τις περιπτώσεις χρειαζόμαστε κάποιες μεθόδους, οι οποίες να μπορούν να ανιχνεύουν τις στατιστικές μεταβολές των δεδομένων, και να τροποποιούν ανάλογα τον ταξινομητή, προσθέτοντας σε αυτόν τη νέα γνώση. Ο κλάδος αυτός της μηχανικής μάθησης ονομάζεται adaptive learning, ή learning in nonstationary environments, ενώ το φαινόμενο της στατιστικής μεταβολής των φαινομένων ονομάζεται concept drift, όπως αναφέρθηκε και στην περίληψη.

Μέχρι πριν λίγα χρόνια, ο κλάδος αυτός της μηχανικής μάθησης δεν είχε λάβει μεγάλη προσοχή απ' την ερευνητική κοινότητα, και το πρόβλημα αντιμετωπιζόταν με απλές τεχνικές, π.χ. με παραθύρωση των δεδομένων, έτσι ώστε να κρατάμε τα $n$ πιο πρόσφατα δείγματα, κλπ. Ένας άλλος απλοϊκός τρόπος είναι η αποθήκευση όλων των παρατηρούμενων δειγμάτων σε μία βάση, βάζοντας ως παράμετρο και το χρόνο, και η εξαγωγή ενός μοντέλου συναρτήσει και του χρόνου. Όμως, η εκρηκτική αύξηση των δεδομένων σήμερα (big data) κάνει μία τέτοια προσέγγιση δύσκολη, αφού ο απαιτούμενος αποθηκευτικός χώρος θα ήταν απαγορευτικός. Συνεπώς, θα πρέπει να μπορούμε να ανιχνεύουμε την αλλαγή των στατιστικών παραμέτρων και του μοντέλου κοιτάζοντας μόνο ένα μέρος των τελευταίων δεδομένων που έχουμε. Επίσης, ένα ρομπότ που αλληλεπιδρά με το περιβάλλον του, ή ένα ενσωματωμένο σύστημα δεν μπορεί να αποθηκεύει μεγάλο όγκο δεδομένων. Άλλωστε, ο χρόνος μπορεί να μην είναι καν η παράμετρος που προκαλεί τη μεταβολή! Επομένως, η ανάγκη του adaptive learning, δηλαδή το να μπορούν τα μοντέλα να προσαρμόζονται στις αλλαγές του περιβάλλοντος γίνεται όλο και περισσότερο αναγκαία. Για το λόγο αυτό, έχει συμβεί άλλωστε μία σημαντική αύξηση της έρευνας στο πεδίο αυτό τα τελευταία χρόνια. Άλλωστε, αυτό είναι σε τελική ανάλυση η νοημοσύνη: το να μπορεί ένα σύστημα να μαθαίνει και να προσαρμόζεται στο περιβάλλον του. Ένα σύστημα που απλά ακολουθεί προκαθορισμένους κανόνες δεν μπορεί να χαρακτηριστεί ευφυές.
Για να δείξουμε στον αναγνώστη την αξία του πεδίου αυτού, παραθέτουμε στην επόμενη παράγραφο μερικές σημαντικές εφαρμογές των ιδεών αυτών.

## 2.2 Μερικές εφαρμογές

Το φαινόμενο του concept drift εμφανίζεται σε μία πληθώρα εφαρμογών ([5]-[8]) . Παρακάτω παρουσιάζουμε μερικές απ' αυτές:

- Εισηγητικά συστήματα (Recommendation systems): Τα εισηγητικά συστήματα προτείνουν στους χρήστες προϊόντα και υπηρεσίες που είναι πιθανό να τους ενδιαφέρουν. Οι προτάσεις αυτές βασίζονται στο ιστορικό αγορών ή αναζητήσεων του χρήστη. Όμως, τα ενδιαφέροντα του χρήστη είναι πιθανό να αλλάξουν με την πάροδο του χρόνου λόγω ποικίλων παραγόντων, όπως π.χ. οι προσωπικές του ανάγκες, οι νέες τάσεις της μόδας, η τρέχουσα οικονομική και επαγγελματική του κατάσταση, η ηλικία, κλπ. Συνεπώς, η πιθανότητα του να κατασκευάσουμε ένα αρχικό μοντέλο για τον χρήστη, το οποίο θα παραμείνει αξιόπιστο και στο μέλλον, είναι μάλλον μη ρεαλιστική. Επομένως, τα εισηγητικά συστήματα λειτουργούν σε μεταβαλλόμενα περιβάλλοντα, και για το λόγο αυτό, το μοντέλο του συστήματος θα πρέπει να μπορεί να προσαρμόζεται στα μεταβαλλόμενα ενδιαφέροντα του χρήστη.

- Ανίχνευση ιών / spam (intrusion / spam detection): Ας υποθέσουμε ότι μία εταιρία λογισμικού θέλει να κατασκευάσει ένα πρόγραμμα αυτόματης αναγνώρισης μηνυμάτων spam ηλεκτρονικού ταχυδρομείου. Ας υποθέσουμε

ότι η εταιρία χρησιμοποιεί έναν αλγόριθμο μηχανικής μάθησης, και τον εκπαιδεύει σε ένα διαθέσιμο σύνολο ηλεκτρονικών μηνυμάτων. Έστω ότι στη συνέχεια το προϊόν βγαίνει στην αγορά. Θα είναι το λογισμικό αυτό αξιόπιστο μετά από, λ.χ., πέντε χρόνια; Η απάντηση είναι μάλλον όχι, αφού τα χαρακτηριστικά των spam και των επιθέσεων θα έχουν αλλάξει από τότε. Επομένως, και στην περίπτωση αυτή έχουμε concept drift, και θα πρέπει ο αλγόριθμος του συστήματος να μπορεί να προσαρμόζεται στις τρέχουσες συνθήκες. Μάλιστα, τα spam και οι επιθέσεις εξειδικεύονται για να αντιμετωπίζουν τα υπάρχοντα συστήματα ασφαλείας, συνεπώς χρειαζόμαστε αντίστοιχα έξυπνους αλγόριθμους που να μπορούν να κάνουν κάτι αντίστοιχο! Για περισσότερες λεπτομέρειες, δείτε την [12].

- Πρόβλεψη ενεργειακών αναγκών: Η πρόβλεψη των ενεργειακών αναγκών είναι ένα απ' τα πιο σημαντικά έργα για την αποδοτική λειτουργία του ενεργειακού δικτύου. Γενικά, παλαιότερα σύνολα δεδομένων για να μπορέσει κανείς να χτίσει ένα μοντέλο πρόβλεψης είναι διαθέσιμα, αλλά γενικά η πρόβλεψη των ενεργειακών αναγκών είναι ένα μη σταθερό – μεταβαλλόμενο πρόβλημα, λόγω μίας ποικιλίας παραγόντων που επηρεάζουν την προσφορά και τη ζήτηση, όμως π.χ. οι κλιματικές αλλαγές κατά τη διάρκεια του έτους. Επίσης, οι αλγόριθμοι ενεργειακής πρόβλεψης θα πρέπει να μπορούν να αντιμετωπίσουν και μακροχρόνιες μεταβολές, λόγω π.χ. της αύξησης του πληθυσμού, ή της εξάπλωσης των ηλιακών πάνελ, τα οποία προσφέρουν ενέργεια στο δίκτυο.

- Πρόβλεψη οικονομικών φαινομένων: Όπως και στην προηγούμενη περίπτωση των ενεργειακών αναγκών, η πρόβλεψη των οικονομικών δεδομένων είναι ένα πρόβλημα μεταβαλλόμενου περιβάλλοντος λόγω ποικίλων παραγόντων, όπως π.χ. των τάσεων στην κατανάλωση, των οικονομικών κρίσεων, κλπ. Οπότε, οι adaptive αλγόριθμοι είναι και εδώ απαραίτητοι.

- Ρομποτική: Τα περισσότερα ρομπότ σήμερα αλληλεπιδρούν κατά κάποιο τρόπο με το περιβάλλον τους. Τα περισσότερα από αυτά εκτελούν απλές βιομηχανικές εργασίες, επομένως στην περίπτωση αυτή οι παραπάνω μέθοδοι δεν είναι απαραίτητες. Στις περιπτώσεις όμως που ένα ρομπότ καλείται να αντιμετωπίσει ένα περιβάλλον του οποίου οι μεταβλητές αλλάζουν, τότε η χρήση adaptive αλγορίθμων μηχανικής μάθησης είναι απαραίτητη. Ένα κλασσικό παράδειγμα είναι η αυτόματη οδήγηση, γύρω απ' την οποία γίνεται μεγάλη έρευνα τα τελευταία χρόνια. Μία ημερομηνία ορόσημο στον τομέα αυτό είναι το 2005, όπου το όχημα της ομάδας του Stanford, Stanley, κατάφερε να πλοηγηθεί αυτόνομα σε μία διαδρομή ανωμάλου εδάφους, και κέρδισε την πρώτη θέση στον διαγωνισμό της DARPA. Ένα απ' τα μέρη του συστήματος πλοήγησης ήταν ένας ταξινομητής που κατέτασσε την εικόνα του δρόμου σε drivable και non – drivable. Ο σκοπός γι' αυτό ήταν ότι το Stanley θα έπρεπε να αποφεύγει τις non – drivable περιοχές του δρόμου, και να μειώνει ταχύτητα όταν τις πλησίαζε. Στην ταξινόμηση αυτή εμφανίζονταν πολλοί νέοι και αστάθμητοι

παράγοντες, όπως ο φωτισμός, η σκόνη, η κατάσταση της κάμερας, κλπ. Για το λόγο αυτό ήταν απαραίτητη η χρήση ενός adaptive μοντέλου, το οποίο θα προσαρμόζεται στις συνθήκες που παρουσιάζονται. Για το σκοπό αυτό, οι ερευνητές χρησιμοποίησαν ένα adaptive μοντέλο μείξης Γκαουσσιανών, όπου οι σταδιακές μεταβολές μοντελοποιούταν με σταδιακή μεταβολή των συντελεστών των Gaussian, ενώ στις απότομες μεταβολές οι Gaussian αντικαθιστώταν με νέες ([5], [13]).

Όπως βλέπουμε λοιπόν, υπάρχει μία μεγάλη πληθώρα εφαρμογών όπου το περιβάλλον και οι παράμετροι του προβλήματος μεταβάλλονται με το χρόνο. Επομένως, η χρήση, καθώς και η θεωρητική κατανόηση του adaptive learning καθίστανται αναγκαίες.

## 2.3 Η διατύπωση του προβλήματος

Μέχρι τώρα περιγράψαμε σε γενικές γραμμές το πρόβλημα της μάθησης σε μεταβαλλόμενα περιβάλλοντα, και παρουσιάσαμε μερικές ενδεικτικές εφαρμογές. Στην παράγραφο αυτή θα ορίσουμε το πρόβλημα με πιο αυστηρό μαθηματικό τρόπο.

Έστω $P: \boldsymbol{X} \to y$ μία διαδικασία - φαινόμενο, η οποία αντιστοιχεί στην $n$-διάστατη μεταβλητή $x$ την έξοδο $y$. Έστω επίσης $\boldsymbol{x}_i$, $i = 1, \dots, m$ ένα σύνολο $n$-διάστατων διανυσμάτων, και έστω $y_i$, $i = 1, \dots, m$ ένα σύνολο «εξόδων» που συνδέεται με τα διανύσματα αυτά. Έστω επίσης $f: \boldsymbol{X} \to y$ η συνάρτηση – κανόνας που συνδέει τα $\boldsymbol{x}$ με τα $y$, δηλαδή θα έχουμε $f(\boldsymbol{x}) = y$ για κάθε $\boldsymbol{x}$. Ως γνωστόν, το πρόβλημα της μηχανικής μάθησης είναι το εξής: δεδομένου του συνόλου $(\boldsymbol{x}_i, y_i)$, $i = 1, \dots, m$ (σύνολο εκπαίδευσης) θα πρέπει να βρούμε μία συνάρτηση $\hat{f}$, η οποία να προσεγγίζει όσο το δυνατόν καλύτερα την άγνωστη συνάρτηση $f$, δηλ. η $\hat{f}$ θα πρέπει να ελαχιστοποιεί ένα κριτήριο σφάλματος $E(f, \hat{f})$ που συνδέεται με την $f$.
Εάν η έξοδος $y$ είναι διακριτή, τότε το πρόβλημα ονομάζεται ταξινόμηση. Αντίθετα, αν το $y$ είναι πραγματικός αριθμός, τότε το πρόβλημα ονομάζεται παλινδρόμηση (regression).

Έστω τώρα $p_t(\boldsymbol{x}, y)$ η από κοινού κατανομή πιθανότητας των $\boldsymbol{x}, y$, και έστωσαν $p_t(y|\boldsymbol{x})$ και $p_t(\boldsymbol{x})$ η εκ των υστέρων πιθανότητα και η σ.π.π. (συνάρτηση πυκνότητας πιθανότητας) της κατανομής. Η παράμετρος $t$ υποδηλώνει το χρόνο, αφού στην γενική περίπτωση οι πιθανότητες αυτές μεταβάλλονται με το χρόνο. Στους συνηθισμένους αλγορίθμους μηχανικής μάθησης υποθέτουμε ότι οι παραπάνω κατανομές πιθανότητας είναι ανεξάρτητες του $t$ – αυτό όμως δεν ισχύει στην περίπτωση του μεταβαλλόμενου περιβάλλοντος.

Επιπλέον, τα δεδομένα μπορεί να εισέρχονται στον ταξινομητή είναι ένα τη φορά, είτε κατά ομάδες. Στην πρώτη περίπτωση, μόνο ένα instance $\boldsymbol{S}_t = (\boldsymbol{x}_i, y_i)$ δίνεται στον ταξινομητή σε κάθε κύκλο, ενώ στην δεύτερη περίπτωση ο αλγόριθμος λαμβάνει ένα σύνολο δεσμίδων – πακέτων δεδομένων της μορφής $\boldsymbol{S}_t =$

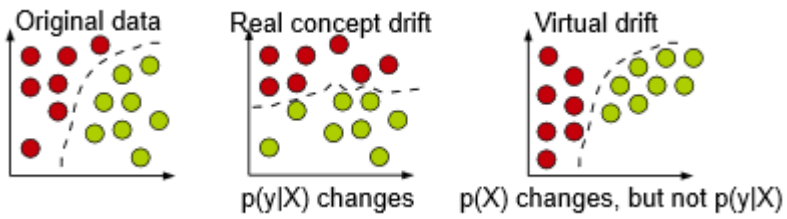
$\{(\boldsymbol{x}_i{}^1, y_i{}^1), \ldots, (\boldsymbol{x}_i{}^n, y_i{}^n)\}$ σε κάθε κύκλο λειτουργίας (batch setting). Προφανώς, όταν $n = 1$, οι δύο περιπτώσεις ταυτίζονται.

Ας επιστρέψουμε τώρα στο πρόβλημα του concept drift. Ως προς τον τρόπο που μεταβάλλονται οι διάφορες κατανομές πιθανότητας, το πρόβλημα διακρίνεται στις ακόλουθες κατηγορίες ([5], [6]):

- Real Drift: Στην περίπτωση αυτή μεταβάλλονται τόσο η εκ των υστέρων πιθανότητα$p_t(y|\boldsymbol{x})$, όσο και σ.π.π. $p_t(\boldsymbol{x})$ της κατανομής. Η μεταβολή των δύο μεγεθών με το χρόνο είναι κατά κανόνα ασυσχέτιστη η μία με την άλλη.

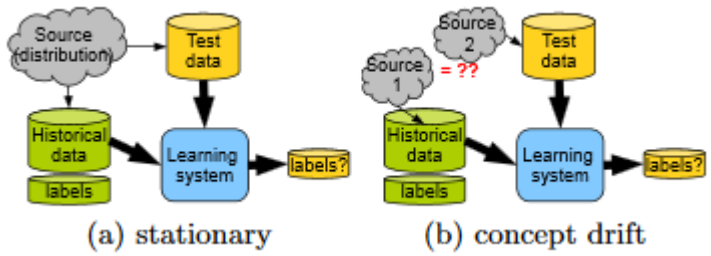
- Virtual Drift: Στην περίπτωση αυτή, η κατανομή $p_t(\boldsymbol{x})$ μεταβάλλεται με το χρόνο, αλλά η posterior πιθανότητα $p_t(y|\boldsymbol{x})$ μένει σταθερή. Στην περίπτωση αυτή, το φαινόμενο ονομάζεται covariant shift.

Οι δύο παραπάνω κατηγορίες φαίνονται σχηματικά παρακάτω:




*Σχήμα 2.3.1: Real και Virtual concept drift.*

Σε κάθε περίπτωση, το concept drift έχει ως αποτέλεσμα ότι η κατανομή πιθανότητας $p_{te}(\boldsymbol{x})$ των δεδομένων προς ταξινόμηση (test sample) θα διαφέρει απ' την κατανομή $p_{tr}(\boldsymbol{x})$ των train δεδομένων (δεδομένα εκπαίδευσης) με τα οποία εκπαιδεύσαμε τον αλγόριθμο μάθησης, δηλαδή θα έχουμε $p_{te}(\boldsymbol{x}) \neq p_{tr}(\boldsymbol{x})$, πράγμα που έρχεται σε αντίθεση με την βασική υπόθεση των συνηθισμένων αλγορίθμων μηχανικής μάθησης, οι οποίοι προϋποθέτουν ότι θα ισχύει $p_{te}(\boldsymbol{x}) = p_{tr}(\boldsymbol{x})$. Επιπλέον, η κατανομή $p_{te}(\boldsymbol{x})$ θα εξαρτάται επίσης απ' το χρόνο, δηλ. θα είναι $p_{te}(\boldsymbol{x}) = p_{te}(\boldsymbol{x}, t)$, ενώ θα μεταβάλλεται και η $p(y|\boldsymbol{x})$.
Επομένως, σε ένα πρόβλημα ύπαρξης concept drift μπορούμε να θεωρήσουμε ότι τα train data και τα test data ανήκουν σε διαφορετικές κατανομές, ενώ σε ένα stationary πρόβλημα οι δύο κατανομές είναι ίδιες. Η διαφορά μεταξύ των δύο συστημάτων μάθησης φαίνεται σχηματικά παρακάτω:




(a) stationary (b) concept drift

*Σχήμα 2.3.2: Δύο συστήματα μάθησης. Το (a) είναι ένα stationary σύστημα: η συνήθης συνθήκη της μηχανικής μάθησης, ότι δηλαδή τα train και test data προέρχονται απ' την ίδια στατιστική κατανομή, ικανοποιείται. Αντίθετα, στο (b) έχουμε concept drift, και τα train και test data προέρχονται από διαφορετικές κατανομές πιθανότητας.*

Εκτός από τα παραπάνω, το concept drift ταξινομείται και με βάση την ταχύτητα που συμβαίνουν οι διάφορες μεταβολές.

- Για παράδειγμα, το concept drift μπορεί να είναι απότομο (abrupt), οδηγώντας σε μία απότομη μεταβολή των δεδομένων. Ένα παράδειγμα τέτοιου φαινομένου είναι για παράδειγμα ένας αισθητήρας, ο οποίος λόγω κάποιων συνθηκών (θόρυβος, κλπ.) αρχίζει να παράγει δεδομένα με διαφορετική κατανομή απ' ότι πριν. Τέτοια φαινόμενα αναφέρονται ως abrupt concept drift ή concept change.

- Απ' την άλλη μεριά, το concept drift μπορεί να είναι βαθμιαίο (gradual), όπου στην περίπτωση αυτή η μεταβολή των κατανομών γίνεται σταδιακά. Ένα παράδειγμα είναι ένας αισθητήρας ο οποίος επηρεάζεται σταδιακά απ' την αύξηση της θερμοκρασίας. Το φαινόμενο αυτό ονομάζεται gradual concept drift.

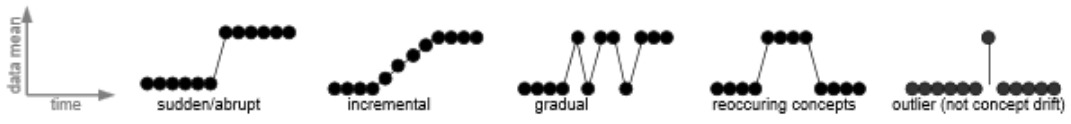
Επιπλέον, το drifts εκτός από abrupt και gradual μπορεί να χαρακτηριστεί και ως:

- Μόνιμο (Permanent): Η επίδραση της μεταβολής δεν περιορίζεται στο χρόνο – το drift συμβαίνει συνεχώς.

- Παροδικό (Transient): Μετά την πάροδο ενός συγκεκριμένου χρονικού διαστήματος, οι κατανομές των δεδομένων σταματάνε να μεταβάλλονται.

Τέλος, το concept drift μπορεί να είναι κυκλικό ή επαναλαμβανόμενο, δηλαδή η ροή των δεδομένων να παρουσιάζει μία περιοδική συμπεριφορά. Το φαινόμενο αυτό ονομάζεται recurrent concepts. Σε τέτοιες περιπτώσεις, η ικανότητα ενός adaptive αλγορίθμου να αντιλαμβάνεται την περιοδικότητα αυτή και να την χρησιμοποιεί στις προβλέψεις είναι ιδιαίτερα σημαντική.

Οι έννοιες αυτές αποτυπώνονται στο παρακάτω σχήμα:




*Σχήμα 2.3.3: Διάφορες κατηγορίες concept drift.*

## 2.4 Αλγόριθμοι μάθησης σε μεταβολή πλαισίου-γενικά

Σχεδιάζοντας έναν αλγόριθμο μάθησης για μεταβαλλόμενα περιβάλλοντα, υπάρχουν πολλοί παράγοντες που πρέπει να λάβουμε υπόψιν. Αρχικά, θυμηθείτε απ' την προηγούμενη παράγραφο ότι η διαδικασία $P$ παράγει μία ακολουθία δεδομένων $\boldsymbol{S}_t$, $t = 1,2,3, \ldots$, τα οποία θεωρούμε ότι προέρχονται από δυνητικά διαφορετικές στατιστικές κατανομές μεταξύ τους. Εάν η ακολουθία των δεδομένων αυτών γίνει αρκετά μεγάλη, είναι μη ρεαλιστικό το να υποθέτουμε ότι όλα τα δεδομένα θα είναι διαθέσιμα κάθε χρονική στιγμή. Η παρατήρηση αυτή ισχύει ιδιαίτερα στις εφαρμογές big data, όπου ο όγκος των δεδομένων είναι υπερβολικά μεγάλος. Συνεπώς, μία πιο ρεαλιστική προσέγγιση είναι το να υποθέσουμε ότι τα δεδομένα $\boldsymbol{S}_t$ είναι διαθέσιμα μόνο – ή κυρίως – κατά τη στιγμή που δίνονται στον αγλόριθμο για πρώτη φορά. Αυτός ο τρόπος μάθησης ονομάζεται one pass learning ή incremental learning.

Επιπλέον, οι περισσότεροι αλγόριθμοι concept drift υποθέτουν ότι οι προβλέψεις τους θα μπορούν να επαληθευτούν και να αξιολογηθούν από τα labels του $\boldsymbol{S}_t$, τα οποία θα φτάσουν μαζί με τα νέα δεδομένα $\boldsymbol{S}_{t+1}$. Αυτή η προσέγγιση επιτρέπει στον αλγόριθμο να υπολογίζει ένα μέτρο σφάλματος σε κάθε χρονικό βήμα, και ονομάζεται test-then-train scenario – η αξιολόγηση του προηγούμενου dataset γίνεται πριν την εκπαίδευση με το νέο dataset. Έτσι, ανάλογα με την ακρίβεια της ταξινόμησης στο προηγούμενοι dataset, ο αλγόριθμος μπορεί και προσαρμόζει τη συμπεριφορά του στα επόμενα dataset – για παράδειγμα, αν οι επιδόσεις του στο παρόν dataset ήταν αρκετά χαμηλές, ο αλγόριθμος υποθέτει ότι έχουμε concept drift και προβαίνει στις απαραίτητες ενέργειες.
Εάν τα labels δεν είναι διαθέσιμα αμέσως μόλις φτάνει το επόμενο batch δεδομένων αλλά αργότερα, τότε έχουμε ένα σενάριο γνωστό ως verification latency. Στην ακραία περίπτωση όπου τα labels δεν γίνονται ποτέ γνωστά μετά το στάδιο της αρχικοποίησης, τότε έχουμε μη επιβλεπόμενη μάθηση, και τα περιβάλλοντα αυτά ονομάζονται initially labeled environments.

Τέλος, υπάρχουν περιπτώσεις όπου το concept drift είναι φαινομενικό και όχι πραγματικό, και προκαλείται από άγνωστες ή μη παρατηρήσιμες παραμέτρους. Το φαινόμενο αυτό ονομάζεται hidden context ή unknown unknown ([11]). Στην περίπτωση αυτή, υπάρχει μία στατική, κείμενη από κάτω διαδικασία, η οποία είναι κρυφή για το σύστημα μάθησης. Θεωρητικά, στην περίπτωση αυτή, η γνώση του hidden context θα απάλειφε την μεταβλητότητα. Παρόλα αυτά, στις περισσότερες περιπτώσεις η διαδικασία αυτή δεν είναι γνωστή, οπότε οι αλγόριθμοι που χρησιμοποιούμε βασίζονται και πάλι στις μεθοδολογίες του concept drift για να λειτουργήσουν.

Όλα τα παραπάνω πρέπει να λαμβάνονται σοβαρά υπόψιν κατά τη σχεδίαση ενός αλγορίθμου μάθησης σε μεταβαλλόμενα περιβάλλοντα.

## 2.5 Άλλα προβλήματα που συσχετίζονται με τη μάθηση σε μεταβαλλόμενα περιβάλλοντα

Η μάθηση σε μεταβαλλόμενα περιβάλλοντα μπορεί να ιδωθεί ως ένα πλαίσιο (framework), μέσα στο οποίο περιλαμβάνονται μία πληθώρα από έννοιες της μηχανικής μάθησης, καθώς και πολλά πεδία προβλημάτων και εφαρμογής ([5] – [7], [9], [10]). Αυτά φαίνονται σχηματικά στο Σχ. 2.5.1:


*Σχ. 2.5.1: Σχηματική αναπαράσταση της σχέσης του concept drift με σχετικά πεδία της μηχανικής μάθησης καθώς και με εφαρμογές.*

Ας μελετήσουμε λίγο περισσότερο το παραπάνω σχήμα. Αρχικά, σε μία συγκεκριμένη εφαρμογή, θα πρέπει πρώτα να επιλεγεί ο τύπος της μάθησης, δηλαδή αν θα είναι supervised, unsupervised, ή semi-supervised, καθώς και ο τρόπος που εισέρχονται τα δεδομένα, δηλ. σε incremental τρόπο ή online. Καθένας από αυτούς τους τύπους μάθησης (learning modalities) παραδοσιακά θα υπέθετε ότι τα δεδομένα για το training και το testing θα προέρχονταν από την ίδια, άγνωστη, στατιστική κατανομή. Στην περίπτωση του concept drift όμως, χρειαζόμαστε αποδοτικούς μηχανισμούς ανίχνευσης του drift, έτσι ώστε να εντοπίσουμε μεταβολές στα datasets, είτε σε incremental είτε σε online αλγορίθμους μάθησης (αυτές οι μέθοδοι θα αναλυθούν στα επόμενα). Συνεπώς, οι παραπάνω τύποι μάθησης δεν εξαρτώνται από το αν έχουμε stationary ή όχι περιβάλλοντα, αλλά περισσότερο από τις συνθήκες του προβλήματος. Απ' την άλλη

μεριά όμως, η λειτουργία των αλγορίθμων ανίχνευσης του concept drift εξαρτάται ουσιωδώς από τον τύπο της μάθησης, οπότε θα πρέπει να επιλέγεται ο κατάλληλος αλγόριθμος σε κάθε περίπτωση.

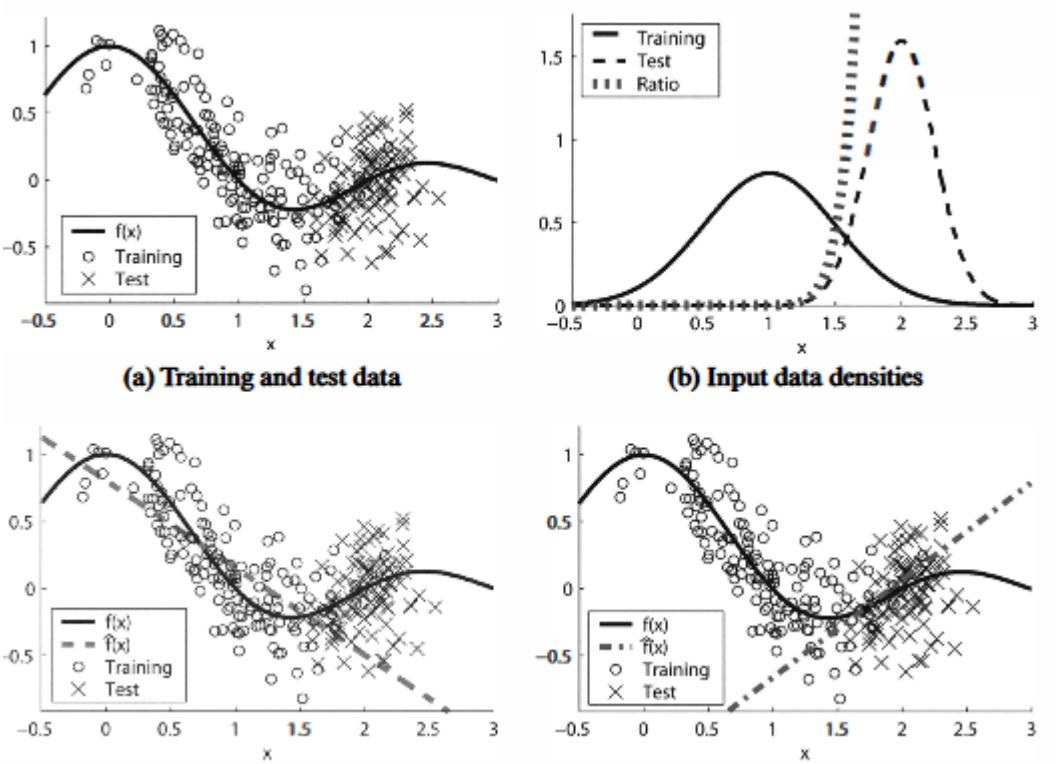
Ας πάμε τώρα στο πεδίο του knowledge shift (μετατόπιση γνώσης). Εδώ, τα πεδία του covariate shift, domain adaptation και transfer learning χαρακτηρίζονται όλα από κάποια μετατόπιση μεταξύ των κατανομών πιθανότητας των training και testing δεδομένων, αλλά οι μεταβολές αυτές θεωρείται ότι συμβαίνουν σε συγκεκριμένα χρονικά διαστήματα, παρά με έναν συνεχή τρόπο. Για παράδειγμα, το covariate shift περιγράφει, όπως είδαμε πριν, περιγράφει μία μεταβολή στις κατανομές πιθανότητας των test δεδομένων, χωρίς όμως να μεταβάλλεται και η labeling function, δηλαδή υποθέτει ότι ισχύει γενικά $p_t(y|\boldsymbol{x}) = p_{t+1}(y|\boldsymbol{x})$, ενώ θα είναι γενικά $p_t(\boldsymbol{x}) \neq p_{t+1}(\boldsymbol{x})$, όπου τα $p_t$ και $p_{t+1}$ συμβολίζουν τις πιθανοτηκές κατανομές των δεδομένων εκπαίδευσης και ελέγχου.
Το transfer learning απ' την άλλη μεριά, αντιμετωπίζει το θέμα όπου τα δεδομένα εκπαίδευσης, καθώς και τα μελλοντικά δεδομένα θα πρέπει να προέρχονται απ' τον ίδιο χώρο χαρακτηριστικών (feature space), και να ακολουθούν την ίδια κατανομή. Τέλος, στο domain adaptation, τα training και test δεδομένα προέρχονται από διαφορετικά, αλλά σχετικά μεταξύ τους πεδία – για παράδειγμα, σε ένα εισηγητικό σύστημα ταινιών, δεδομένων κάποιων training data που προέρχονται από κωμωδίες, θέλουμε να προβλέψουμε τα ενδιαφέροντα του χρήστη σε test data που προέρχονται απ' την κατηγορία κωμωδίες. Σε αυτά τα προβλήματα, υπάρχει επίσης η περίπτωση τα dataset να είναι non-stationary, καθώς οι στατιστικές κατανομές μεταβάλλονται απ' τα train data στα test data. Παρόλα αυτά, σε αντίθεση με τις προηγούμενες περιπτώσεις όπου έχουμε streaming data (ροές δεδομένων), εδώ δεν υπάρχει η έννοια της συνεχούς μεταβολής της κατανομής με την πάροδο του χρόνου. Αντίθετα, τα train και test data μπορεί να θεωρηθεί ότι παρέχονται στις χρονικές στιγμές $t = 1$ και $t = 2$, και ακολουθούν τις κατανομές $p_1(x)$ και $p_2(x)$, χωρίς αναφορά σε μεταγενέστερους χρόνους.

Τέλος, οι πιο γενικές περιπτώσεις μεταβαλλόμενου περιβάλλοντος προέρχονται από εφαρμογές όπως η πρόβλεψη χρονοσειρών (time series), η ταξινόμηση αναρτήσεων twitter, ή τα genomics. Τέλος, οι εφαρμογές big data αποτελούν επίσης ένα μεγάλο πεδίο εφαρμογής των μεθόδων της μηχανικής μάθησης σε μεταβαλλόμενα περιβάλλοντα. Όλα αυτά φαίνονται καθαρά στο πάνω αριστερά μέρος του σχήματος 2.5.1.

## 2.6 Ένα απλό παράδειγμα

Για να κατανοήσουμε καλύτερα τα παραπάνω, θα παρουσιάσουμε εδώ ένα απλό παράδειγμα. Συγκεκριμένα, ας υποθέσουμε ότι θέλουμε να μάθουμε μία συνάρτηση $f(x)$ από ένα σύνολο δειγμάτων $\{(x_i^{tr}, y_i^{tr})\}_{i=1}^{n_{tr}}$. Ο στόχος μας εδώ είναι, μέσω των διαθέσιμων δειγμάτων να βρούμε μία καλή προσεγγιστική συνάρτηση $\hat{f}(x)$, μέσω της οποίας θα μπορούμε στη συνέχειας να υπολογίσουμε την έξοδο $y^{te}$ για μία άγνωστη είσοδο $x^{te}$, υπολογίζοντας το $\hat{f}(x^{te})$.

Εδώ θα θεωρήσουμε μία περίπτωση covariate shift, όπου, σύμφωνα με τα προηγούμενα, τα train και test σημεία ακολουθούν διαφορετικές κατανομές μεταξύ τους, αλλά η target function $f(x)$ είναι η ίδια και στις δύο περιπτώσεις. Στο παράδειγμά μας, το οποίο φαίνεται στο σχήμα 2.6.1, τα training samples βρίσκονται στην αριστερή πλευρά του γραφήματος, ενώ τα test samples βρίσκονται στα δεξιά. Έχουμε δηλαδή να λύσουμε ένα πρόβλημα extrapolation, όπου τα δείγματα βρίσκονται έξω απ' την περιοχή εκπαίδευσης (σημειώστε ότι τα σημεία αυτά δεν είναι διαθέσιμα για εκπαίδευση, απλά παριστάνονται εδώ για καλύτερη εποπτεία). Οι πυκνότητες πιθανότητας των training και test points, $p_{tr}(x)$ και $p_{te}(x)$, απεικονίζονται επίσης στο σχήμα 2.6.1, και όπως βλέπουμε διαφέρουν μεταξύ τους.




*Σχ. 2.6.2: Ένα πρόβλημα regression με covariate shift. Στο (a) βλέπουμε τη συνάρτηση στόχου, $f(x)$, τα δεδομένα εκπαίδευσης (o), και τα δεδομένα testing (x). Στο (b) βλέπουμε τις κατανομές πιθανότητας των train και test data, καθώς και το λόγο τους. Στο (c) βλέπουμε τη συνάρτηση $\hat{f}(x)$ που παίρνουμε χρησιμοποιώντας την συνήθη μέθοδο των ελαχίστων τετραγώνων, ενώ στο (d) βλέπουμε την συνάρτηση $\hat{f}(x)$ που προκύπτει χρησιμοποιώντας τη μέθοδο των importance-weighted least squares.*

Στη συνέχεια, για να λύσουμε το πρόβλημα, θα μοντελοποιήσουμε την $\hat{f}(x)$ ως μία γραμμική συνάρτηση, δηλαδή θα είναι

$$\hat{f}(x) = \theta_1 x + \theta_2, \tag{2.6.1}$$

για πραγματικές παραμέτρους $\theta_1, \theta_2 \in \mathbb{R}$, οι οποίες θα υπολογιστούν με τη μέθοδο των ελαχίστων τετραγώνων, ελαχιστοποιώντας την ποσότητα

$$\min_{\theta_1,\theta_2}\left[\sum_{i=1}^{n_{tr}}\left(\hat{f}(x_i^{tr})-y_i^{tr}\right)^2\right], \tag{2.6.2}$$

ως προς τις παραμέτρους $\theta_1, \theta_2$ (δηλαδή ελαχιστοποιώντας το άθροισμα των τετραγώνων).

Εκτελώντας τους υπολογισμούς βρίσκουμε μία συνάρτηση $\hat{f}(x)$ η οποία διέρχεται μέσα απ' τα σημεία εκπαίδευσης, όπως φαίνεται στο σχήμα 2.6.1c, και τα προσεγγίζει σε σχετικά ικανοποιητικό βαθμό. Παρόλα αυτά, η συνάρτηση αυτή δεν είναι χρήσιμη για τον υπολογισμό των test δεδομένων που βρίσκονται στο δεξί μέρος του γραφήματος, όπως είναι εμφανές απ' το σχήμα.

Ας δούμε για ποιο λόγο συμβαίνει αυτό. Διαισθητικά, τα training samples που είναι μακριά απ' τα test points δίνουν λιγότερες πληροφορίες για την πρόβλεψη των test σημείων στο δεξί μέρος του γραφήματος. Με βάση αυτή την παρατήρηση, υποθέτουμε ότι αγνοώντας σε ένα βαθμό αυτά τα σημεία, και λαμβάνοντας υπόψιν περισσότερο τα σημεία που είναι πιο κοντά στην test περιοχή θα πάρουμε γενικά καλύτερα αποτελέσματα. Γενικά, η ιδέα της προσαρμογής του covariate shift (covariate shift adaptation) είναι το να επιλέγουμε τα σημαντικότερα σημεία εκπαίδευσης με έναν συστηματικό τρόπο, λαμβάνοντας υπόψιν την σπουδαιότητα του κάθε σημείου ως προς την πρόβλεψη των τιμών εξόδου. Συγκεκριμένα, ως μέτρο σημαντικότητας χρησιμοποιούμε τον λόγο των κατανομών πιθανότητας των train και test δεδομένων, δηλ. το λόγο

$$\frac{p_{te}(x_i^{tr})}{p_{tr}(x_i^{tr})}, \tag{2.6.3}$$

και θεωρούμε τον παραπάνω λόγο ως βάρος του δείγματος $x_i^{tr}$ στη μέθοδο ελαχίστων τετραγώνων:

$$\min_{\theta_1,\theta_2}\left[\sum_{i=1}^{n_{tr}}\frac{p_{te}(x_i^{tr})}{p_{tr}(x_i^{tr})}\left(\hat{f}(x_i^{tr})-y_i^{tr}\right)^2\right]. \tag{2.6.4}$$

Με το τρόπο αυτό παίρνουμε τη συνάρτηση $\hat{f}(x)$ που φαίνεται στο σχήμα 2.6.1d, η οποία περιγράφει τα test samples πολύ καλύτερα απ' ότι πριν (σημειώστε ότι τα test samples δεν χρησιμοποιούνται στην εκπαίδευση). Με την παραπάνω μέθοδο, η επίδραση των σημείων εκπαίδευσης που βρίσκονται αρκετά αριστερά (π.χ. για $x < 1.2$) γίνεται αυτομάτων πολύ μικρή, ενώ λαμβάνονται περισσότερο υπόψιν τα πιο κεντρικά σημεία, τα οποία ταιριάζουν καλύτερα στα δεδομένα εξόδου.

Από το παραπάνω παράδειγμα βλέπουμε την ουσία του covariance shift, καθώς και έναν απλό αλγόριθμο που αντιμετωπίζει το πρόβλημα. Φυσικά, ο αλγόριθμος αυτός είναι πολύ απλός, ενώ υπάρχουν πολλά ακόμα ζητήματα που πρέπει να αντιμετωπιστούν (όπως π.χ. ο υπολογισμός των κατανομών πιθανότητας απ' τα δείγματα), αλλά μας δείχνει με αρκετά απλό τρόπο την όλη φύση του προβλήματος.

Τέλος, πρέπει να σημειώσουμε ότι η εισαγωγή ενός βάρους σημαντικότητας στα δεδομένα (importance weights) παίζει σημαντικό λόγο στο covariance shift adaptation.
Το παραπάνω παράδειγμα ανήκει στην κατηγορία του covariate shift, η οποία είναι μία υποπερίπτωση του concept drift. Για τα προβλήματα αυτά έχουν αναπτυχθεί παρόμοιες μέθοδοι με την παραπάνω, όπου τα βάρη χρησιμοποιούνται ώστε ο αλγόριθμος ταξινόμησης να προσαρμοστεί καλύτερα στο σύνολο ελέγχου (test set). Με την ίδια λογική όπως παραπάνω, μπορούμε να κατασκευάσουμε σταθμισμένα SVM, σταθμισμένους πυρήνες, κλπ. ([2]). Στη γενικότερη βέβαια περίπτωση του concept drift οι κατανομές μεταβάλλονται τυχαία και κάθε στιγμή, οπότε είναι αρκετά δυσκολότερο να αναπτύξει κανείς παρόμοιες προσεγγίσεις. Οι μέθοδοι που έχουν αναπτυχθεί για τη γενική περίπτωση του concept drift αναλύονται στις επόμενες ενότητες.

## 2.7 Μαθαίνοντας σε μεταβαλλόμενα περιβάλλοντα: οι ενεργές και παθητικές μέθοδοι

Στην προηγούμενη παράγραφο είδαμε έναν απλό αλγόριθμο για regression μίας συνάρτησης σε περιβάλλον covariance shift. Όπως είδαμε στις προηγούμενες παραγράφους, το covariance shift είναι μία υποπερίπτωση του concept drift, όπου οι πιθανότητες κατανομής των train και test δεδομένων είναι διαφορετικές μεταξύ τους, ενώ η εκ των υστέρων πιθανότητα $p(y|\boldsymbol{x})$ παραμένει σταθερή. Στην γενική περίπτωση του concept drift οι κατανομές πιθανότητας, καθώς και η εκ των υστέρων πιθανότητα, μεταβάλλονται με το χρόνο. Επομένως, δεν μπορούμε να χρησιμοποιήσουμε μεθόδους όπως αυτή της προηγούμενης παραγράφου, και απαιτούνται γενικότεροι αλγόριθμοι.

Στη γενική περίπτωση του concept drift, οι adaptive αλγόριθμοι που χρησιμοποιούνται ανήκουν κατά βάση σε δύο κατηγορίες: active (ενεργητικοί) ή passive (παθητικοί) (δείτε τις αναφορές [19], [22]). Οι αλγόριθμοι που ακολουθούν την active προσέγγιση έχουν ως στόχο το να ανιχνεύσουν το concept drift, ενώ οι passive αλγόριθμοι ενημερώνουν το μοντέλο ταξινόμησης κάθε φορά που λαμβάνουν νέα δεδομένα, ασχέτως του αν έχει συμβεί concept drift ή όχι – δηλ. οι passive αλγόριθμοι μαθαίνουν συνεχώς. Και στις δύο προσεγγίσεις, ο στόχος είναι η ενημέρωση του ευφυούς συστήματος, έτσι ώστε αυτό να προβλέπει με ακρίβεια τα νέα δεδομένα σε κάθε στιγμή. Όμως, αυτό γίνεται με διαφορετικούς μηχανισμούς σε κάθε μία απ' τις δύο προσεγγίσεις (active ή passive).

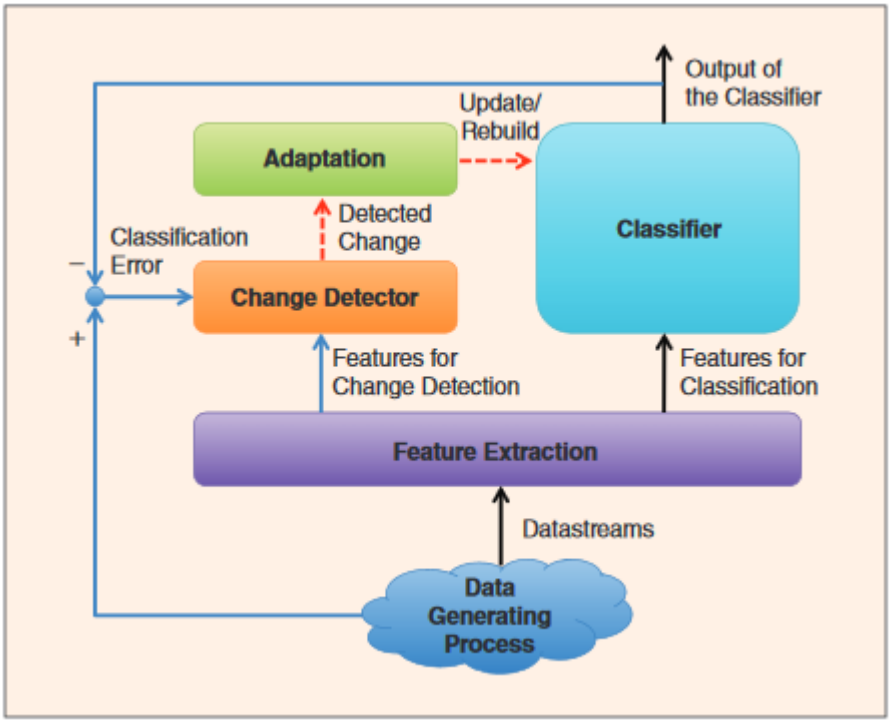
Και οι δύο μέθοδοι μπορούν να λύσουν ένα συγκεκριμένο πρόβλημα μηχανικής μάθησης, αλλά, ανάλογα με το πρόβλημα, οι επιδόσεις της active και της passive μεθόδου θα διαφέρουν. Για το λόγο αυτό, πριν επιλέξουμε κάποιον συγκεκριμένο adaptive αλγόριθμο θα πρέπει πρώτα να λάβουμε υπόψιν τις παραμέτρους του προβλήματος, όπως π.χ. τα drift rates, τον τρόπο που παράγονται τα δεδομένα (online ή batch), καθώς και άλλες παραμέτρους, όπως οι διαθέσιμοι υπολογιστικοί

πόροι (π.χ. ενσωματωμένα συστήματα ή υπολογιστές υψηλών επιδόσεων), καθώς και υποθέσεις για την στατιστική κατανομή των δεδομένων. Σε γενικές γραμμές, οι παθητικές μέθοδοι φαίνεται να είναι αρκετά αποδοτικοί σε προβλήματα που χαρακτηρίζονται από βαθμιαία drift (gradual drifts), ή και επαναλαμβανόμενα πρότυπα (recurring concepts) [19]. Αν και αυτό μπορεί να γίνει επίσης και με active αλγόριθμους, η ανίχνευση μεταβολών με βαθμιαίο drift είναι παραταύτα πιο δύσκολη. Αντίθετα, οι ενεργές μέθοδοι λειτουργούν καλύτερα σε περιπτώσεις όπου τα drift είναι απότομα. Επιπλέον, οι παθητικές μέθοδοι είναι γενικά καλύτερες στην περίπτωση που έχουμε δέσμες δεδομένων (batch learning), ενώ οι ενεργές μέθοδοι λειτουργούν συνήθως καλύτερα στις περιπτώσεις που τα δεδομένα έρχονται σειριακά (online) ([22], [24]).

Στις ενότητες που ακολουθούν θα αναλύσουμε τις δύο μεθόδους, και θα παρουσιάσουμε τους κυριότερους αλγορίθμους.

## 2.8 Ενεργές μέθοδοι: ανίχνευση μεταβολής και προσαρμογή

Η βασική αρχιτεκτονική ενός active αλγόριθμου για μάθηση σε μεταβαλλόμενα περιβάλλοντα φαίνεται στο παρακάτω σχήμα:




*Σχ. 2.8.1: Η βασική αρχιτεκτονική μίας active μεθόδου για ταξινόμηση σε μεταβαλλόμενα περιβάλλοντα. Αρχικά, όπως συνηθίζεται στα έξυπνα συστήματα, έχουμε μία μονάδα εξαγωγής χαρακτηριστικών, η οποία εξάγει τα χαρακτηριστικά των δεδομένων που θα χρησιμεύσουν αργότερα για την ταξινόμηση. Στη συνέχεια, υπάρχει ένας ανιχνευτής μεταβολών, ο οποίος εξετάζει τα διανύσματα των χαρακτηριστικών, ή, σε κάποιες περιπτώσεις, και το σφάλμα της ταξινόμησης τη*

*δεδομένη στιγμή. Μόλις ανιχνευτεί μία μεταβολή, η διαδικασία της adaptation ενεργοποιείται, η οποία ενημερώνει ή επανεκπαιδεύει τον ταξινομητή.*

Όπως βλέπουμε απ' το παραπάνω σχήμα, η όλη μέθοδος βασίζεται σε έναν μηχανισμό ανίχνευσης μεταβολής, ο οποίος πυροδοτείται όταν ανιχνεύσει μία μεταβολή στην κατανομή των δεδομένων, καθώς και σε έναν μηχανισμό προσαρμογής (adaptation), ο οποίος ενεργοποιείται όταν ανιχνευτεί κάποια μεταβολή, και ενημερώνει ή επανεκπαιδεύει τον ταξινομητή ([20]).
Αναλυτικότερα, ο στόχος του change detector είναι να ενημερώσει το σύστημα ότι ανιχνεύτηκε μία μεταβολή στη διαδικασία παραγωγής των δεδομένων $P$, σε κάποια συγκεκριμένη χρονική στιγμή. Αυτό το επιτυγχάνει παρακολουθώντας κάποια χαρακτηριστικά τα οποία εξάγονται απ' τα δεδομένα για το σκοπό αυτό (π.χ. στατιστικές παράμετροι), ενώ σε κάποιες περιπτώσεις ο ανιχνευτής παρακολουθεί και το τρέχον σφάλμα της ταξινόμησης, το οποίο αποτιμάται από τα labeled δεδομένα που εισέρχονται στον ανιχνευτή. Η ανάλυση των χαρακτηριστικών ελέγχει τη στασιμότητα της κατανομής $p_i(\boldsymbol{x})$, ενώ η ανάλυση του σφάλματος ταξινόμησης μας δίνει πληροφορίες για τη μεταβολή της πιθανότητας $p_i(y|\boldsymbol{x})$. Η φάση προσαρμογής (adaptation phase), η οποία ενημερώνει ή ξαναδημιουργεί το μοντέλο ταξινόμησης, ενεργοποιείται μόνο έναν ανιχνευτεί μία μεταβολή. Οι adaptive στρατηγικές που ακολουθούν το μηχανισμό αυτό είναι επίσης γνωστές ως "detect & react" ([25]): μόλις εντοπιστεί μία μεταβολή, ο ταξινομητής απορρίπτει την προηγούμενη γνώση και προσαρμόζεται στο νέο περιβάλλον.
Οι γνωστότεροι αλγόριθμοι ανίχνευσης μεταβολής και προσαρμογής παρουσιάζονται παρακάτω.

### 2.8.1 Ανίχνευση μεταβολής

Οι αλγόριθμοι ανίχνευσης μεταβολών σπάνια λαμβάνουν ως είσοδο τα αρχικά δεδομένα. Αντιθέτως, η ανίχνευση μεταβολών γίνεται συνήθως εξετάζοντας ανεξάρτητα και όμοια κατανεμημένα χαρακτηριστικά, τα οποία εξάγονται απ' τα δεδομένα εισόδου, όπως π.χ. η μέση τιμή του δείγματος, η διακύμανση, και/ή το σφάλμα ταξινόμησης.

Οι περισσότεροι αλγόριθμοι ανίχνευσης μεταβολών μπορούν να ταξινομηθούν σε τέσσερις μεγάλες κατηγορίες: Έλεγχος υποθέσεως (Hypothesis Tests), μέθοδοι μεταβολής σημείου (Change-Point Methods), ακολουθιακός έλεγχος υπόθεσης (Sequential Hypothesis Tests), και ανίχνευση μεταβολής (Change Detection). Οι αλγόριθμοι στις τέσσερις αυτές κατηγορίες έχουν την ικανότητα να ανιχνεύουν μεταβολές χρησιμοποιώντας καθιερωμένες, θεωρητικά θεμελιωμένες στατιστικές τεχνικές. Οι διαφορές έγκεινται κυρίως στον τρόπο που επεξεργάζονται τα δεδομένα.

#### Hypothesis Tests (HT)

Ο σκοπός των ελέγχων υποθέσεως είναι το να εξετάσουν την ισχύ ή όχι μίας υπόθεσης, με βάση κάποιο μέτρο εμπιστοσύνης. Για παράδειγμα, ας θεωρήσουμε

ότι διαθέτουμε δύο σύνολα δειγμάτων, τα οποία έχουν ίδια μέση τιμή μεταξύ τους. Τα δείγματα αυτά αντλήθηκαν από δύο διαφορετικές κατανομές με ίδια μέση τιμή, ή από την ίδια κατανομή; Η απάντηση ερωτημάτων του τύπου αυτού είναι το αντικείμενο του ελέγχου υποθέσεως στη στατιστική. Η συνήθης διαδικασία είναι να κάνουμε μία υπόθεση (π.χ., ότι τα δύο σύνολα δειγμάτων προέρχονται απ' την ίδια κατανομή), και στη συνέχεια να εξετάσουμε αν η υπόθεση αυτή είναι αληθής ή εσφαλμένη, με τη βοήθεια κάποιων στατιστικών μέτρων. Συνήθως, το αποτέλεσμα αυτών είναι μία πιθανότητα ισχύος της υπόθεσης, η οποία αν είναι αρκετά μεγάλη, τότε θεωρούμε ότι η υπόθεση είναι αληθής.
Αυτές οι στατιστικές τεχνικές εφαρμόζονται σε ακολουθίες δεδομένων σταθερού μεγέθους (δηλ. δεν έχουμε ακολουθιακή ανάλυση των δεδομένων), και κάνουν έναν έλεγχο υποθέσεως για την ανίχνευση μεταβολών. Π.χ., ένας απλοϊκός τρόπος είναι να χωρίσουμε τα δεδομένα σε δύο ομάδες, και να εξετάσουμε αν προέρχονται απ' την ίδια κατανομή (υπόθεση) – αν όχι (η υπόθεση απορρίφθηκε), τότε έχουμε μία μεταβολή.
Μερικά παραδείγματα αλγορίθμων που χρησιμοποιούν έλεγχο υποθέσεως για την ανίχνευση μεταβολών μπορούν να βρεθούν στις εργασίες [28] και [38]. Συγκεκριμένα, στην [28] προτείνεται η χρήση της κανονικοποιημένης απόστασης Kolmogorov – Smirnov, η οποία αποτιμά τις διαφορές μεταξύ των συναρτήσεων κατανομής που εκτιμώνται η μία στα δείγματα εκπαίδευσης και η άλλη σε ένα παράθυρο πρόσφατων δεδομένων. Αντίθετα, στο [38] προτείνεται η χρήση του στατιστικού τεστ των ίσων αναλογιών για την εξέταση μεταβολών στο σφάλμα ταξινόμησης.

## Change Point Methods (CPMs)

Παρόμοια με τα hypothesis tests, οι μέθοδοι μεταβολής σημείου εφαρμόζονται σε ακολουθίες δεδομένων σταθερού μήκους. Οι στατιστικές τεχνικές αυτού του είδους, οι οποίες παρουσιάζονται στην εργασία [39], έχουν ως στόχο το να διαπιστώσουν αν μία δεδομένη ακολουθία περιέχει ένα σημείο μεταβολής, το οποίο ορίζεται ως ένα σημείο στο οποίο η διαδικασία παραγωγής των δεδομένων αλλάζει τη στατιστική συμπεριφορά της, ή όχι. Αυτό επιτυγχάνεται ελέγχοντας όλες τις δυνατές διαμερίσεις της διαθέσιμης ακολουθίας δεδομένων. Το κύριο χαρακτηριστικό αυτής της κατηγορίας στατιστικών τεχνικών είναι η ικανότητά τους στο να αντιμετωπίζουν από κοινό, τόσο το πρόβλημα της ανίχνευσης μίας μεταβολής, όσο και της εκτίμησης της χρονικής στιγμής που η μεταβολή αυτή συνέβη. Απ' την άλλη μεριά, το βασικό μειονέκτημα αυτών των τεχνικών είναι η μεγάλη υπολογιστική πολυπλοκότητα, που προκύπτει απ' την εξέταση όλων των διαμερίσεων των δεδομένων, όπως αναφέραμε προηγουμένως. Αυτό καθιστά τη χρήση των αλγορίθμων αυτών σε σενάρια ακολουθιακής ροής των δεδομένων (streaming data) απαγορευτική. Προσεγγιστικές λύσεις που έχουν σχεδιαστεί για να λειτουργούν ακολουθιακά έχουν προταθεί πρόσφατα στην βιβλιογραφία (π.χ. στο [40]), αλλά η πολυπλοκότητά τους παραμένει ακόμη σημαντικό εμπόδιο.

## Sequential Hypothesis Tests (SHT)

Σε αντίθεση με τα hypothesis tests και τα change – point methods, τα οποία εφαρμόζονται σε σταθερές ακολουθίες δεδομένων, οι μέθοδοι ακολουθιακού

ελέγχου υπόθεσης (sequential hypothesis tests) έχουν τη δυνατότητα να εξετάζουν ακολουθιακά τα δεδομένα που φτάνουν στην είσοδο, ένα προς ένα, μέχρι να ληφθεί η απόφαση της αποδοχής ή απόρριψης της υπόθεσης (π.χ. ότι η κατανομή δεν έχει μεταβληθεί). Με άλλα λόγια, αυτές οι στατιστικές μέθοδοι αναλύουν τα δεδομένα που έρχονται ακολουθιακά, μέχρι να αποκτήσουν αρκετή στατιστική εμπιστοσύνη ώστε να αποφανθούν αν έχει συμβεί ή όχι μεταβολή. Τα δείγματα που λαμβάνονται μετά την απόφαση δεν λαμβάνονται υπόψιν. Μερικά παραδείγματα των τεχνικών αυτών είναι ο ακολουθιακός έλεγχος λόγου πιθανότητας (sequential probability ratio test), ή και ο έλεγχος επαναλαμβανόμενης σημαντικότητας (repeated significance test) ([41]). Το βασικό μειονέκτημα των μεθόδων SHT προέρχεται από την απαίτηση του να λαμβάνεται μία απόφαση σχετικά με τη μηδενική υπόθεση (δηλ. μεταβολή ή όχι μεταβολή) μόλις έχει αποκτηθεί σημαντική στατιστική εμπιστοσύνη. Στην πραγματικότητα, μετά την απόφαση, το SHT σταματάει να αναλύει το data stream (μόλις η απόφαση ληφθεί, δεν χρειάζεται να αναλυθούν άλλα δεδομένα), και αυτό είναι ένα σοβαρό μειονέκτημα στην ακολουθιακή ανάλυση, όπου ο στόχος είναι του να συνεχίζεται η λειτουργία του αλγορίθμου και μετά την ανίχνευση της μεταβολής πλαισίου, ούτως ώστε να ανιχνεύσουμε και πιθανές επόμενες μεταβολές στο μέλλον.

## Change Detection Tests (CDT)

Η ανάγκη των αλγορίθμων ανίχνευσης μεταβολών να λειτουργούν με πλήρως ακολουθιακό τρόπο αντιμετωπίζεται από τους αλγόριθμους ανίχνευσης μεταβολών (change detection tests), οι οποίοι είναι ειδικά σχεδιασμένοι για να αναλύουν ακολουθιακά τη στατιστική συμπεριφορά των δεδομένων. Οι μέθοδοι αυτές χαρακτηρίζονται γενικά από μειωμένη υπολογιστική πολυπλοκότητα (αυτό είναι αναγκαίο, αφού πρέπει να αναλύουν ακολουθιακά τη ροή των δεδομένων), αλλά δεν μπορούν να εγγυηθούν τον έλεγχο των ψευδών θετικών αποτελεσμάτων (false positive rates ή false alarms), πράγμα που σημαίνει ανιχνεύουμε μία μεταβολή που δεν υπάρχει στην πραγματικότητα. Αντίθετα, οι μέθοδοι HT, CPM και SHT δεν αντιμετωπίζουν το πρόβλημα αυτό.

Ο απλούστερος αλγόριθμος CDT βασίζεται σε ένα κατώφλι: μία μεταβολή ανιχνεύεται κάθε φορά που η τιμή ενός χαρακτηριστικού των δεδομένων, ή το σφάλμα της ταξινόμησης ξεπερνά ένα συγκεκριμένο κατώφλι. Για παράδειγμα, στην εργασία [23] προτείνεται ένα σταθερό κατώφλι που βασίζεται στο όριο Hoeffding (ανισότητα Hoeffding), το οποίο εφαρμόζεται στη διαφορά μεταξύ των μέσων τιμών δύο ομάδων δειγμάτων που προέρχονται από δύο μη επικαλυπτόμενα παράθυρα δεδομένων. Αντίθετα, μία διαφορετική μέθοδος προτείνεται στο [35], όπου η ανίχνευση μεταβολών πυροδοτείται από τη σύγκριση του σφάλματος επικύρωσης (validation error) που υπολογίζεται στο πιο πρόσφατο παράθυρο δεδομένων, με το σφάλμα επικύρωσης που υπολογίζεται σε ένα παράθυρο τυχαία επιλεγμένο από τα προηγούμενα δεδομένα.

Ένας διαφορετικός μηχανισμός κατωφλίωσης που βασίζεται στο σφάλμα ταξινόμησης προτείνεται στο [34], όπου το κατώφλι είναι μία συνάρτηση της διακύμανσης της διαφοράς μεταξύ των ποσοστών σφάλματος των δεδομένων εκπαίδευσης και επικύρωσης. Αντίθετα, ένας μηχανισμός κατωφλίωσης που

βασίζεται στην ανάλυση του Bernoulli Exponential Weighted Moving Average (EWMA) των σφαλμάτων μπορεί να εισαχθεί από τον τελευταίο προστιθέμενο ταξινομητή, όπως περιγράφεται στην [36], όπου το κατώφλι είναι μία συνάρτηση του ποσοστού του σφάλματος του τελευταίου ταξινομητή που προστέθηκε, καθώς και μίας παραμέτρου ευαισθησίας οριζόμενης απ' το χρήστη. Απ' την άλλη μεριά, στην [22] προτείνεται ένας αλγόριθμος που ανιχνεύει μία μεταβολή όταν το σφάλμα ταξινόμησης ξεπεράσει ένα κατώφλι, που είναι συνάρτηση της τυπικής απόκλισης της συσχετιζόμενης κατανομής Bernoulli. Αυτός ο μηχανισμός έχει επεκταθεί στην [37], η οποία βασίζεται στην ανάλυση της απόστασης μεταξύ δύο σφαλμάτων ταξινόμησης (δηλ. του τρέχοντος και της ελάχιστης τιμής), αντί για το ποσοστό των σφαλμάτων. Αυτή η βασιζόμενη στην απόσταση σύγκριση επιτρέπει στον προτεινόμενο μηχανισμό να βελτιώνει την επίδοση της ανίχνευσης σε περιπτώσεις βραδέως concept drift. Ένας άλλος μηχανισμός concept change, ο οποίος στοχεύει στην αξιολόγηση μεταβολών στην αναμενόμενη τιμή του σφάλματος ταξινόμησης μεταξύ ενός παραθύρου αναφοράς και ενός κυλιόμενου παραθύρου ανίχνευσης προτείνεται στην [42], όπου το κατώφλι βασίζεται στα όρια Bernstein. Επίσης, ένα πιο αποτελεσματικό κατώφλι ανίχνευσης συνδυασμένο με έναν μηχανισμό τυχαίας δειγματοληψίας για την αποθήκευση δειγμάτων στο παράθυρο ανίχνευσης παρουσιάζεται στην [43]. Ομοίως, δύο μηχανισμοί κινούμενου μέσου όρου (moving average), όπου το κατώφλι βασίζεται στα όρια Hoefffinger, προτείνονται στην [44].
Η χρήση της απόστασης Hellinger για τη μέτρηση της απόκλισης μεταξύ της εκτίμησης της κατανομής από πακέτα των πρόσφατων δεδομένων, καθώς και κάποιων δεδομένων αναφοράς προτείνεται στην [27], όπου το (προσαρμοζόμενο) κατώφλι βασίζεται στην κατανομή t. Στη ίδια γραμμή, μία οικογένεια από μέτρα απόστασης μεταξύ κατανομών (τα οποία βασίζονται σε συγκρίσεις μεταξύ παραθύρων δεδομένων), καθώς και ένας αλγόριθμος που βασίζεται σε κατωφλίωση για τον έλεγχο των μεταβολών, τόσο σε διακριτές, όσο και σε συνεχείς κατανομές, προτείνεται στην [26].

Παρόλο που οι μέθοδοι που βασίζονται σε κάποιο κατώφλι είναι αρκετά απλές στη σχεδίαση και την υλοποίηση, το κύριο μειονέκτημά τους είναι στο να καθοριστεί το κατώφλι στη φάση της σχεδίασης (χωρίς να προϋποθέτουμε κάποια επιπλέον γνώση για την κατανομή και τις πιθανές μεταβολές): πολύ χαμηλές τιμές μπορεί να προκαλέσουν λανθασμένες ενδείξεις ανίχνευσης μεταβολών, ενώ οι μεγάλες τιμές μπορεί να προκαλέσουν απώλεια ανίχνευσης των μεταβολών.

Για το λόγο αυτό, στην εργασία [30] προτείνεται μία διαφορετική προσέγγιση ενός προσαρμοζόμενου (adaptive) ανιχνευτή, βασιζόμενου στο στατιστικό τεστ Cumulative SUM (τεστ σωρευτικού αθροίσματος – CUSUM) για την παρακολούθηση της στατικότητας (stationarity) της μέσης τιμής των δειγμάτων των δεδομένων στη διάρκεια του χρόνου. Εδώ, ένας λογαριθμικός λόγος πιθανοφανειών (log – likehood ratio) μεταξύ δύο αυτόματα εκτιμώμενων συναρτήσεων πυκνότητας πιθανότητας (pdf) – συγκεκριμένα, της μηδενικής και μίας εναλλακτικής – αποτιμάται ακολουθιακά στη διάρκεια του χρόνου για τον εντοπισμό μεταβολών στην διαδικασία παραγωγής δεδομένων. Μία επέκταση του adaptive CUSUM, που βασίζεται στις μεθόδους της υπολογιστικής νοημοσύνης, η οποία παρακολουθεί

μεταβολές σε διάφορες στατιστικές ροπές των δεδομένων, καθώς και σε κάποιες εσωτερικές μεταβλητές, προτείνεται στην [31]. Αντίθετα, ένας CDT που βασίζεται στην τομή διαστημάτων εμπιστοσύνης (Intersection of Confidence Intervals – ICI) και στις παραλλαγές της, προτείνεται στις εργασίες [32], [33] και [45]. Αυτοί οι CDT είναι ιδιαίτερα αποδοτικοί όταν τα χαρακτηριστικά των δεδομένων παράγονται από μία Γκαουσσιανή κατανομή με σταθερή διακύμανση. Επίσης, οι ICI CDT συνοδεύονται συνήθως από μία διαδικασία εκλέπτυνσης (refinement procedure), η οποία παρέχει μία εκτίμηση της χρονικής στιγμής που έλαβε χώρα η μεταβολή, αφού αυτή ανιχνευτεί. Η ικανότητα αυτή του αλγορίθμου είναι ουσιώδης για την προσαρμογή των Just-in-Time-Adaptive Classifiers, οι οποίοι περιγράφονται στα επόμενα.

Επίσης, εδώ πρέπει να αναφέρουμε ότι οι έλεγχοι υποθέσεως μπορούν να συνδυαστούν με τους αλγορίθμους CDT για την επιβεβαίωση ή όχι των μεταβολών που αυτοί εντοπίζουν. Οι μηχανισμοί ανίχνευσης μεταβολών που ακολουθούν τη διαδικασία αυτή είναι γνωστοί με τον όρο «Ιεραρχικοί CDT», και είναι τυπικά σε θέση να παρέχουν μία μείωση των ψευδών ανιχνεύσεων concept drift, χωρίς να αυξάνουν την καθυστέρηση της ανίχνευσης των μεταβολών ([46]). Επίσης, οι μέθοδοι CPM μπορούν επίσης να συνδυαστούν με τους CDT σε μία ιεραρχική προσέγγιση. Για παράδειγμα, η συνδυασμένη χρήση ενός επιπέδου ανίχνευσης μεταβολών βασισμένου στη μέθοδο ICI CDT, μαζί με ένα επίπεδο επιβεβαίωσης των μεταβολών βασισμένο σε μεθόδους CPM προτείνεται στην [47].

### 2.8.2 Προσαρμογή (adaptation)

Μόλις εντοπιστεί μία μεταβολή, ο ταξινομητής χρειάζεται να προσαρμοστεί στη μεταβολή αυτή, μαθαίνοντας τη νέα διαθέσιμη πληροφορία, και απορρίπτοντας την προηγούμενη γνώση. Η δυσκολία στο να γίνει αυτό έγκειται στη σχεδίαση ενός προσαρμοζόμενου μηχανισμού ο οποίος να έχει τη δυνατότητα να ξεχωρίζει τα δείγματα των δεδομένων σε ενημερωμένα και παρωχημένα. Οι υπάρχουσες μέθοδοι για active ταξινομητές μπορούν να χωριστούν σε τρεις κύριες κατηγορίες: παραθύρωση (windowing), στάθμιση (weighting), και τυχαία δειγματοληψία (random sampling).

Η παραθύρωση είναι η πιο συνήθης και απλή μέθοδος. Εδώ, μόλις εντοπιστεί μία μεταβολή, εφαρμόζεται ένα κυλιόμενο παράθυρο πάνω στα τελευταία δείγματα, το οποίο περιλαμβάνει μόνο το ενημερωμένο κομμάτι των δεδομένων (π.χ., από το σημείο της μεταβολής μέχρι το τελευταίο που έφτασε όταν την εντοπίσαμε), ενώ τα προηγούμενα δείγματα που βρίσκονται εκτός του παραθύρου θεωρούνται απαρχαιωμένα. Στη συνέχεια, όλα τα δείγματα που περιέχονται στο παράθυρο χρησιμοποιούνται για την επανεκπαίδευση του ταξινομητή (ή και του CDT, αν χρειάζεται), ενώ τα παλαιότερα δεδομένα απλώς αγνοούνται. Η κατάλληλη επιλογή του μήκους του παραθύρου είναι ένα σημαντικό ζήτημα, και μπορεί να υπολογιστεί με διάφορους τρόπους. Στην [21] λ.χ., προτείνεται η χρήση του αναμενόμενου λόγου της μεταβολής για το σκοπό αυτό. Αντίθετα, οι εργασίες [22] – [24], [29], [33], [34] και [45] προτείνουν τη χρήση προσαρμοζόμενων (adaptive) μεθόδων. Ένας

προσαρμοζόμενος μηχανισμός καθορισμού του μήκους παραθύρου, ο οποίος βασίζεται στην ανάλυση της μέσης τιμής κάποιων υποπαραθύρων που εφαρμόζονται πάνω στα τελευταία δείγματα δεδομένων προτείνεται στην [23]: το παράθυρο μεγαλώνει σε στατικές συνθήκες, και μικραίνει εάν ανιχνευθούν μεταβολές. Αντίθετα, στην [22] προτείνεται ένας μηχανισμός ανίχνευσης που χρησιμοποιεί δύο κατώφλια – ένα κατώφλι προειδοποίησης, καθώς και ένα κατώφλι ανίχνευσης. Εδώ, το μήκος του παραθύρου προσαρμόζεται ώστε να περιέχει όλα τα δείγματα που προκύπτουν απ' τη στιγμή που ένα χαρακτηριστικό (π.χ. κάποιο στατιστικό μέγεθος) ξεπερνά το κατώφλι προειδοποίησης, μέχρι τη στιγμή που το χαρακτηριστικό αυτό ξεπέρασε το κατώφλι ανίχνευσης.

Μία νέα γενιά προσαρμοζόμενων ταξινομητών, οι ονομαζόμενοι Just-In-Time (JIT) ταξινομητές, σχεδιασμένοι για να λειτουργούν σε μεταβαλλόμενα περιβάλλοντα, προτείνεται στις εργασίες [24], [33] και [45]. Οι αλγόριθμοι αυτοί βασίζονται σε ένα προσαρμοζόμενο παράθυρο (adaptive window), του οποίου το μήκος υπολογίζεται με την ICI διαδικασία εκλέπτυνσης που περιγράφηκε παραπάνω. Οι αλγόριθμοι αυτοί προτείνουν τη χρήση δύο CDT, έτσι ώστε να παρακολουθούν από κοινού, τόσο την κατανομή των δεδομένων εισόδου, όσο και το σφάλμα ταξινόμησης. Επιπλέον, οι JIT ταξινομητές έχουν την ικανότητα να ενσωματώνουν τυχόν επιβλεπόμενα (labeled) δεδομένα που έρχονται στην είσοδο, έτσι ώστε να βελτιώνουν την απόδοση της ταξινόμησης με το χρόνο. Επιπλέον, πρόσφατα παρουσιάστηκε ένας JIT ταξινομητής ειδικά σχεδιασμένος για να λειτουργεί σε συνθήκες βαθμιαίου concept drift στην [48]. Εδώ, ο CDT έχει ως στόχο το να ανιχνεύσει μεταβολές στην πολυωνυμική προσέγγιση (polynomial trend – ένα πολυώνυμο που διέρχεται απ' τα σημεία των δεδομένων) της αναμενόμενης τιμής των δεδομένων. Μόλις εντοπιστεί μία μεταβολή, μία μέθοδος παραθύρωσης προσαρμοζόμενου μήκους, η οποία βασίζεται σε μία εκτίμηση της δυναμικής του drift, χρησιμοποιείται για την τροποποίηση του μήκους του παραθύρου.

Ένας ψευδοκώδικας για την οικογένεια των JIT προσαρμοζόμενων ταξινομητών δίνεται στο σχήμα 2.8.2. Μία αρχική ακολουθία δεδομένων εκπαίδευσης $S_{T_0}$ χρησιμοποιείται για την αρχικοποίηση, τόσο του ταξινομητή, όσο και του ICI CDT (γραμμή 1 στον κώδικα). Μετά την φάση της εκπαίδευσης, κάθε φορά που λαμβάνουμε ένα νέο δείγμα $x_i$ στην είσοδο (μαζί με την κλάση $y_i$, όποτε είναι διαθέσιμη – επιβλεπόμενο δείγμα), ο CDT παρακολουθεί τη στατικότητα της διαδικασίας $P$ (διαδικασία παραγωγής δεδομένων) (γραμμή 5). Εάν ανιχνεύσει μία μεταβολή στη χρονική στιγμή $T$, η διαδικασία ICI θα παράγει μία εκτίμηση $\hat{T}$ για το χρόνο στον οποίο η μεταβολή συνέβη (γραμμή 7). Τη στιγμή αυτή, όλα τα δείγματα που λάβαμε πριν τη χρονική στιγμή $\hat{T}$ θεωρούνται ότι ανήκουν σε προηγούμενη γνώση, και συνεπώς απορρίπτονται. Αντίθετα, τα δείγματα που λάβαμε μεταξύ των χρονικών στιγμών $\hat{T}$ και $T$, τα οποία θεωρούμε ότι αναπαριστούν την τρέχουσα, ενημερωμένη γνώση, αντιπροσωπεύουν το προσαρμοζόμενο παράθυρο, και χρησιμοποιούνται για την επανεκπαίδευση, τόσο του ταξινομητή, όσο και του CDT (γραμμή 8). Απ' την άλλη μεριά, σε στάσιμες συνθήκες (χωρίς concept drift), τα επιβλεπόμενα δείγματα $(x_i, y_i)$ ενσωματώνονται στη βάση γνώσης του ταξινομητή, με στόχο τη βελτίωση, αν αυτό είναι δυνατό, της ακρίβειας της ταξινόμησης (γραμμή 11).

**Input**: A Training Sequence $\mathcal{S}_{T_0} := \{(\mathbf{x}_i, y_i) : i \in \{1, ..., T_0\}\}$;

1: Configure the classifier and the ICI-based CDT on $\mathcal{S}_{T_0}$;
2: $i = T_0 + 1$;
3: **while** (1) **do**
4: Input receive new data $\mathbf{x}_i$ (with supervised information $y_i$ whenever available);
5: **if** (ICI-based CDT detects a variation in the statistical distribution of inputs or in the classification error) **then**
6: Let $T$ be the time of detection;
7: Activate the ICI-based refinement procedure to provide an estimate $\hat{T}$ (the time the change started);
8: Characterize the new $\mathcal{S}_{\hat{T}}$ as the set of samples acquired between $\hat{T}$ and $T$;
9: Configure the classifier and the CDT on $\mathcal{S}_{\hat{T}}$;
10: **else**
11: Integrate the available information $(\mathbf{x}_i, y_i)$ in the knowledge base of the classifier;
12: **end if**
13: Predict the output $\hat{y}_i$ of the input samples $\mathbf{x}_i$ (whenever $y_i$ is not available);
14: **end while**

*Σχ. 2.8.2: Ψευδοκώδικας ενός JIT ταξινομητή για μάθηση σε μεταβαλλόμενα περιβάλλοντα.*

Επίσης, στην [34] προτείνεται μία υβριδική σταθερή – προσαρμοζόμενη (fixed – adaptive) προσέγγιση, στην οποία ο αλγόριθμος μάθησης αρχικά εκπαιδεύεται σε ένα παράθυρο δεδομένων σταθερού μήκους, και στη συνέχεια ένας προσαρμοζόμενος μηχανισμός τροποποιεί το μήκος του παραθύρου.

Σε αντίθεση με τις τεχνικές παραθύρωσης, οι οποίες επιλέγουν ένα υποσύνολο των δειγμάτων απ' τη ροή των δεδομένων, οι μέθοδοι στάθμισης λαμβάνουν υπόψιν όλα τα διαθέσιμα δείγματα, αλλά τους προσδίδουν ένα κατάλληλα επιλεγμένο βάρος, το οποίο εξαρτάται από την παλαιότητά τους, ή τη συνάφειά τους σε συνάρτηση με την ακρίβεια της ταξινόμησης των τελευταίων πακέτων επιβλεπόμενων δεδομένων ([49], [50]). Ένας μηχανισμός στάθμισης "βαθμιαίας λήθης" (gradual – forgetting) προτείνεται στην [50], όπου τα βάρη των δειγμάτων φθίνουν γραμμικά με το χρόνο (τα πρόσφατα δεδομένα έχουν μεγαλύτερα βάρη από τα πιο παλιά). Ομοίως, ένας άλλος βασισμένος στο χρόνο μηχανισμός στάθμισης παρουσιάζεται στην [51]. Εκεί, ένα σύνολο συναρτήσεων μείωσης των βαρών (οι οποίες περιλαμβάνουν από εκθετικές συναρτήσεις μέχρι πολυώνυμα) παρουσιάζονται και συγκρίνονται ως προς την απόδοση. Αντίθετα, στην [49] χρησιμοποιείται μία διαφορετική προσέγγιση, στην οποία τα βάρη εξαρτώνται από κάποιους δείκτες μεταβολής, οι οποίοι μετράνε τη μεταβολή της διαδικασίας παραγωγής δεδομένων στη διάρκεια του χρόνου (σε συνάρτηση με ένα σύνολο εκπαίδευσης αναφοράς). Επιπλέον, όπως προτείνεται στην [52], τα δείγματα μπορούν επίσης να σταθμιστούν ανάλογα με την ακρίβεια ή το σφάλμα της ταξινόμησης, υπολογιζόμενο στην τελευταία δεσμίδα επιβλεπόμενων δεδομένων. Παρόλα αυτά, το βασικό μειονέκτημα των αλγορίθμων στάθμισης είναι η ανάγκη

του να αποθηκεύουν στη μνήμη όλα τα προηγούμενα δεδομένα, κάτι που είναι δύσκολο να γίνει σε κάποιες περιπτώσεις, όπως π.χ. σε εφαρμογές big data.

Η μέθοδος της δειγματοληψίας προσφέρει μία εναλλακτική στις προηγούμενες μεθόδους της παραθύρωσης και της στάθμισης. Συγκεκριμένα, το reservoir sampling ([53]) είναι μία γνωστή τεχνική τυχαίας δειγματοληψίας που έχει τη δυνατότητα να επιλέγει ένα αντιπροσωπευτικό υποσύνολο δειγμάτων (χωρίς επανατοποθέτηση) από μία ροή δεδομένων. Η βάση του reservoir sampling είναι ως εξής: ένα δείγμα $(\boldsymbol{x}_i, y_i)$ που λαμβάνεται τη χρονική στιγμή $t$ αποθηκεύεται σε μία "δεξαμενή" (reservoir) με μία πιθανότητα $p = k/t$, όπου το $k$ είναι η χωρητικότητα της δεξαμενής, που ορίζεται απ' το χρήστη. Εάν ένα νέο δείγμα πάει να εισαχθεί, ενώ η χωρητικότητα της δεξαμενής έχει εξαντληθεί, ένα δείγμα επιλέγεται τυχαία από τη δεξαμενή και αφαιρείται. Αποδεικνύεται εύκολα ότι με τη μέθοδο αυτή, όλα τα δείγματα στη δεξαμενή είναι ισοπίθανα, και επιλέγονται με πιθανότητα ίση με $k/l$, όπου $l$ το μήκος του συνόλου των δεδομένων, το οποίο είναι άγνωστο. Έτσι, έχουμε δημιουργήσει ένα αντιπροσωπευτικό δείγμα των δεδομένων. Ένα παράδειγμα της χρήσης της μεθόδου αυτής στο πλαίσιο των ροών δεδομένων μπορεί να βρεθεί στην [54], ενώ μία χρήση του reservoir sampling για ανίχνευση μεταβολών περιγράφεται στην [55].

Μία διαφορετική προσέγγιση από τις παραπάνω που αναφέραμε, είναι η χρήση ενός συνόλου ταξινομητών (ensemble of classifiers). Η προσέγγιση αυτή χρησιμοποιείται κυρίως σε παθητικούς αλγόριθμους ανίχνευσης μεταβολών, όπως θα δούμε παρακάτω – παρόλα αυτά, στη βιβλιογραφία έχουν αναπτυχθεί και κάποιες τέτοιες προσεγγίσεις για active αλγόριθμους. Για παράδειγμα, στην [36] προτείνεται ένας αλγόριθμος, ο οποίος δημιουργεί έναν νέο ταξινομητή και τον προσθέτει στο σύνολο, κάθε φορά που ανιχνεύεται κάποια μεταβολή (η ανίχνευση βασίζεται στην ανάλυση του σφάλματος ταξινόμησης). Επίσης, αλγόριθμοι JIT έχουν προταθεί σε συνδυασμό με ensembles ταξινομητών ([33]).

## 2.9 Παθητικές μέθοδοι

Στην ενότητα αυτή θα περιγράψουμε τη δεύτερη μεγάλη κατηγορία adaptive αλγορίθμων, τους παθητικούς (passive) αλγόριθμους.
Όπως υποδηλώνει και η ονομασία, οι παθητικοί αλγόριθμοι δεν έχουν ως στόχο να ανιχνεύσουν απευθείας τις μεταβολές – αντίθετα, δέχονται ότι οι κατανομές των δεδομένων μπορεί να αλλάξουν οποιαδήποτε στιγμή, με οποιοδήποτε ρυθμό. Για να αντιμετωπίσουν αυτή την αβεβαιότητα όσον αφορά τη μεταβολή, οι παθητικοί μέθοδοι κάνουν συνεχώς προσαρμογή, κάθε φορά που έρχονται δεδομένα στην είσοδό τους. Αυτή η συνεχής προσαρμογή επιτρέπει στους παθητικούς αλγορίθμους να διατηρούν ένα σωστά ενημερωμένο μοντέλο ταξινόμησης κάθε στιγμή, αποφεύγοντας με αυτό τον τρόπο τα συνήθη μειονεκτήματα των ενεργών μεθόδων, δηλ. της μη ανίχνευσης κάποιων drifts, ή της ανίχνευσης λάθος μεταβολών.

Υπάρχουν δύο βασικές κατηγορίες passive αλγορίθμων: αυτοί που βασίζονται στην προσαρμογή ενός μόνο ταξινομητή, και αυτοί που διαθέτουν σύνολα ταξινομητών, στο οποίο προσθέτουν ή αφαιρούν ταξινομητές, ή τροποποιούν τους ήδη υπάρχοντες.

**Μέθοδοι ενός ταξινομητή**

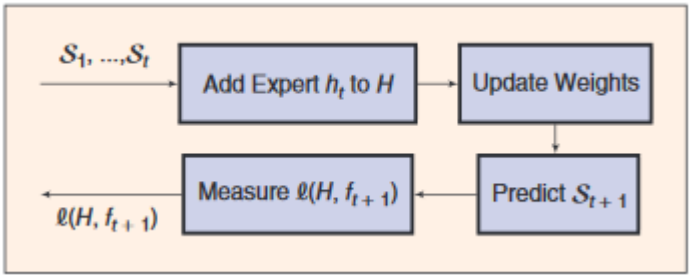
Οι προσεγγίσεις που βασίζονται σε έναν μόνο ταξινομητή παρέχουν γενικά ένα χαμηλότερο υπολογιστικό κόστος, απ' ότι τα ensemble συστήματα, πράγμα που κάνει τις μεθόδους αυτές μία ελκυστική λύση για μεγάλες ροές δεδομένων.
Οι πιο κοινοί ταξινομητές που χρησιμοποιούνται για ανάλυση – εξόρυξη σε μία ροή δεδομένων είναι τα δέντρα απόφασης (decision trees), με το very-fast decision tree (VFDT) να είναι μία απ' τις πιο δημοφιλείς μεθόδους ([56]). Με βάση αυτό, ένας αλγόριθμος ανίχνευσης μεταβολών VFDT (CVFDT) προτάθηκε στην [57], ο οποίος χρησιμοποιεί ένα προσαρμοζόμενο κυλιόμενο παράθυρο για εκπαίδευση. Επίσης, ο αλγόριθμος CVFDT επεκτάθηκε έτσι ώστε να εξετάζει πολλαπλές επιλογές σε κάθε κόμβο κάθε φορά που ένας κόμβος πάει να διαχωριστεί. Αντίθετα με αυτές τις προσεγγίσεις, μία άλλη μέθοδος είναι τα Online Information Networks (OLIN), που είναι μία προσέγγιση που βασίζεται στην ασαφή λογική, και εφαρμόζει επίσης ένα κυλιόμενο παράθυρο πάνω στα δεδομένα εκπαίδευσης ([58]). Επιπλέον, προσφάτως τα νευρωνικά δίκτυα έχουν επίσης αποκτήσει κάποια δημοφιλία όσον αφορά την προσαρμοζόμενη μάθηση. Για παράδειγμα, μία πρόσφατη εργασία περιγράφει μία online extreme learning machine (ELM), συνδυασμένη με ένα χρονικά μεταβαλλόμενο νευρωνικό δίκτυο για μάθηση σε μεταβαλλόμενα περιβάλλοντα.

**Μέθοδοι ομάδας ταξινομητών (ensemble)**

Ανάμεσα σε όλους τους παθητικούς αλγόριθμους για μάθηση σε μεταβαλλόμενα περιβάλλοντα, τα ensemble μοντέλα φαίνεται να είναι τα πιο δημοφιλή. Υπάρχουν αρκετοί λόγοι γι' αυτό. Αρχικά, τα συστήματα αυτά έχουν μία φυσική προσαρμογή σε προβλήματα μεταβαλλόμενων δεδομένων, και προσφέρουν επίσης κάποια πλεονεκτήματα: Πρώτον, τείνουν να είναι περισσότερο ακριβή από τα συστήματα ενός μόνο ταξινομητή, λόγω της μείωσης της διασποράς του σφάλματος. Δεύτερον, έχουν μεγαλύτερη ευελιξία στο να ενσωματώνουν νέα δεδομένα στη βάση γνώσης τους, απλώς προσθέτοντας νέους ταξινομητές στο σύνολο. Τρίτον, διαθέτουν μηχανισμούς έτσι ώστε να "ξεχνάνε" παλαιότερη γνώση, απλώς αφαιρώντας τους αντίστοιχους παλαιούς ταξινομητές απ' το ΄σύνολο ([59], [60]). Οι δύο τελευταίες παρατηρήσεις μπορούν να συγχωνευτούν στο λεγόμενο δίλημμα σταθερότητας – πλαστικότητας (stability – plasticity dilemma, [61]), σύμφωνα με το οποίο ένας ταξινομητής μπορεί είτε να διατηρεί την παλαιότερη γνώση του, είτε να μαθαίνει καινούρια, αλλά δεν μπορεί να κάνει και τα δύο εξίσου καλά. Αυτό σημαίνει ότι ένας ταξινομητής δεν γίνεται και να διατηρεί την παλιά του γνώση και να μαθαίνει καινούρια απεριόριστα καλά – το ένα θα είναι εις βάρος του άλλου. Στην περίπτωση αυτή, τα συστήματα πολλών ταξινομητών παρέχουν μία καλή ισορροπία σε αυτό το φάσμα σταθερότητας – πλαστικότητας, λόγω της ικανότητάς τους να προσθαφαιρούν ταξινομητές απ' το σύνολο (δείτε το σχήμα 2.9.1). Το χαρακτηριστικό αυτό κάνει τα ensemble συστήματα κατάλληλα για μάθηση σε μεταβαλλόμενα περιβάλλοντα, αφού το drift μπορεί να επηρεάζει μόνο ένα

υποσύνολο της βάσης γνώσης, αφήνοντας την υπόλοιπη, προηγούμενη γνώση ανέπαφη για μελλοντική χρήση – τα νέα δεδομένα προσθέτουν απλώς τους δικούς τους ταξινομητές στην ομάδα, αφήνοντας ανέπαφους τους παλαιούς. Επίσης, τα ensemble μπορούν να ανανεώνουν συνεχώς τα βάρη των ταξινομητών τους με στρατηγικό τρόπο – μετρώντας την απόδοση των μεμονωμένων ταξινομητών πάνω στα πρόσφατα δεδομένα, μπορούν να αλλάζουν κατάλληλα τα βάρη έτσι ώστε να παρέχουν μικρότερο σφάλμα σε σχέση με έναν μεμονωμένο ταξινομητή.

Τα πλεονεκτήματα των ομαδικών συστημάτων μάθησης σε μεταβαλλόμενα περιβάλλοντα έχουν αποδειχθεί κατά ένα μέρος και θεωρητικά, δείχνοντας ότι τα συστήματα αυτά μπορούν και παράγουν πιο σταθερά αποτελέσματα απ' ότι τα συστήματα ενός ταξινομητή ([62], [63]). Επίσης, θεωρητικά έχουν δειχθεί και άλλα πλεονεκτήματα των ομαδικών συστημάτων, και συγκεκριμένα σχετικά με την λεγόμενη ποικιλότητά (diversity) τους. Η ποικιλότητα ενός ταξινομητή είναι ένα μέτρο που σχετίζεται με το πόσο ομοιόμορφα κατανεμημένα είναι τα λάθη που κάνει ο ταξινομητής στο σύνολο των δεδομένων – όσο περισσότερο ομοιόμορφα κατανέμονται τα σφάλματα του ταξινομητή στα δεδομένα, τόσο μεγαλύτερη η ποικιλότητα. Και αυτό είναι χρήσιμο, διότι μία πρόσφατη έρευνα έδειξε ότι τα συστήματα ομάδας, είτε μεγάλης είτε μικρής ποικιλότητας έχουν πλεονεκτήματα στην αναγνώριση διαφορετικών βαθμών μεταβολών σε ροές δεδομένων ([64], [65]).




*Σχ. 2.9.1: Γενικό διάγραμμα ενός ensemble συστήματος ταξινόμησης σε μεταβαλλόμενο περιβάλλον. Τα δεδομένα λαμβάνονται σε δεσμίδες $\boldsymbol{S}_t$ με την πάροδο του χρόνου. Με τα δεδομένα αυτά εκπαιδεύεται ένας νέος ταξινομητής $h_t$, ο οποίος προστίθεται στο ensemble $H$. Στη συνέχεια, τα unlabeled δεδομένα της δεσμίδας $\boldsymbol{S}_{t+1}$ ταξινομούνται από το ensemble. Επίσης, μία συνάρτηση σφάλματος υπολογίζεται όταν φτάνουν τα labels του $\boldsymbol{S}_{t+1}$. Με βάση την τιμή της, ανανεώνονται τα βάρη για τους διάφορους ταξινομητές.*

Στη συνέχεια θα δούμε περιληπτικά μερικούς απ' τους πιο γνωστούς παθητικούς αλγόριθμους.

**Παθητικοί αλγόριθμοι**

Ένας απ' τους πιο παλαιούς και απλούς παθητικούς αλγόριθμους για μάθηση σε μεταβαλλόμενα περιβάλλονται είναι ο αλγόριθμος SEA (streaming ensemble algorithm, [66]). Η λειτουργία του αλγορίθμου SEA είναι αρκετά απλή: κάθε φορά απλώς προστίθεται ένας νέος ταξινομητής όταν φτάνει ένα νέο πακέτο δεδομένων. Μόλις το ensemble φτάσει ένα προκαθορισμένο μέγεθος, ο SEA αρχίζει να αφαιρεί ταξινομητές απ' το σύνολο, βασιζόμενος σε ένα μέτρο ποιότητας για τους ταξινομητές, όπως για παράδειγμα η ακρίβεια των προβλέψεων κάθε μεμονωμένου ταξινομητή σε σύγκριση με την απόδοση του συνόλου, ή η ηλικία του ταξινομητή

(αν ένας ταξινομητής είναι αρκετά παλιός, θεωρείται ότι δεν είναι πλέον ενημερωμένος, οπότε αφαιρείται). Αυτή η στρατηγική του SEA του επιτρέπει να μειώνει την επίδραση του διλήμματος στατικότητας – πλαστικότητας, αφού μπορεί και προσθαφαιρεί ταξινομητές κατά βούληση. Επίσης, υπάρχουν και άλλες παρόμοιες προσεγγίσεις στη βιβλιογραφία, που ακολουθούν τη φιλοσοφία «της αφαίρεσης του χειρότερου», π.χ. η εργασία [67].

Κάποιες άλλες γνωστές προσεγγίσεις για συστήματα παθητικής μάθησης περιλαμβάνουν κάποια έξυπνη τροποποίηση ενός συνηθισμένου αλγορίθμου μηχανικής μάθησης. Για παράδειγμα, οι μέθοδοι bagging και boosting αποτελούν τη βάση του αλγόριθμου online nonstationary boosting algorithm (ONSBoost, [68]), ο οποίος προσθέτει μία περίοδο ανανέωσης (update period) στον γνωστό αλγόριθμο online boosting ([69]), με στόχο την αφαίρεση ταξινομητών με κακή επίδοση. Επίσης, έχουν αναπτυχθεί διάφορες επεκτάσεις των online bagging και boosting, κάποιες απ’ τις οποίες ενσωματώνουν ενεργές και παθητικές τεχνικές για την ανίχνευση απότομων και βαθμιαίων μεταβολών ([70], [71]).
Μία άλλη μέθοδος είναι η δυναμική σταθμισμένη πλειοψηφία (dynamic weighted majority) - (DWM, [72]), η οποία είναι μία επέκταση του αλγορίθμου σταθμισμένης πλειοψηφίας (WM, [73]), όπου ο WM επεκτείνεται σε ροές δεδομένων με concept drift, και χρησιμοποιεί μία περίοδο ανανέωσης για να προσθέτει – αφαιρεί ταξινομητές απ’ το σύνολο. Αυτό φαίνεται εκ πρώτης όψεως αρκετά όμοιο με το ONSBoost, αλλά ο DWM επιτρέπει στο μέγεθος του ensemble να είναι προσαρμοζόμενο, ενώ αντίθετα στον ONSBoost το μέγεθος του ensemble είναι σταθερό. Εκτός αυτού, άλλες προσεγγίσεις, όπως ο αλγόριθμος accuracy updated ensemble (AUE), ακολουθούν μία παρόμοια μεθοδολογία όσον αφορά τον τρόπο που εξετάζουν πότε θα προσθέσουν ή θα αφαιρέσουν ταξινομητές από ένα ensemble σταθερού μεγέθους ([74]). Τέλος, ο γνωστός αλγόριθμος random forest έχει επίσης επεκταθεί για μη - στάσιμα δεδομένα, όπως περιγράφεται στην [75].

Ένας άλλος γνωστός παθητικός αλγόριθμος για μάθηση σε μεταβαλλόμενα περιβάλλονται είναι ο Learn$^{++}$.NSE (το ακρώνυμο NSE προέρχεται απ’ το non – stationary environments) [19], του οποίου ο ψευδικώδικας φαίνεται στο παρακάτω σχήμα:

**Input:** Datasets $\mathcal{S}_t := \{(\mathbf{x}_i, y_i) : i \in [N_t]\}$, supervised learning algorithm BASE, and parameters $a$ & $b$.
**Initialize:** $h_1 = \text{BASE}(\mathcal{S}_1)$ and $W_1^1 = 1$.

1: **for** $t = 2,3, \ldots$ **do**

2: Compute loss of the existing ensemble

$$E_t = \frac{1}{N_t} \sum_{j=1}^{N_t} 1_{H_{t-1}(\mathbf{x}_j) \neq y_j}, \tag{1}$$

where $1_\tau$ evaluates to 1 if $\tau =$ `True` otherwise it is 0.

3: Update instance weights

$$D_t(j) = \frac{1}{Z_t} \begin{cases} E_t & H_{t-1}(\mathbf{x}_j) = y_j \\ 1 & \text{otherwise} \end{cases}, \tag{2}$$

where $Z_t$ is a normalization constant.

4: $h_t = \text{BASE}(\mathcal{S}_t)$

5: Evaluate existing classifiers with new data

$$\varepsilon_k^t = \sum_{j=1}^{N_t} D_t(j) 1_{h_k(\mathbf{x}_j) \neq y_j}. \tag{3}$$

Set $\beta_k^t = \varepsilon_k^t / (1 - \varepsilon_k^t)$.

6: Compute time-adjusted loss

$$\varphi_k^t = \frac{1}{Z_t'} \frac{1}{1 + \exp(-a(t - k - b))}, \tag{4}$$

$$\rho_k^t = \sum_{j=0}^{t-k} \varphi_k^{t-j} \beta_k^{t-j}. \tag{5}$$

7: Update classifier voting weights: $W_k^t = \log \frac{1}{\rho_k^t}$.

8: **end for**

**Output:** Learn$^{++}$.NSE's prediction on $\mathbf{x}$

$$H_t(\mathbf{x}) = \arg\max_{\omega \in \Omega} \sum_{k=1}^{t} W_k^t 1_{h_k(\mathbf{x}) = \omega}. \tag{6}$$

*Σχ. 2.9.2: Ψευδοκώδικας του αλγορίθμου Learn$^{++}$.NSE.*

Ο αλγόριθμος Learn$^{++}$.NSE διατηρεί ένα ensemble το οποίο εφαρμόζει μία προσαρμοζόμενη από το χρόνο συνάρτηση σφάλματος, με στόχο το να ευνοήσει τους ταξινομητές που απέδωσαν καλά σε πρόσφατους χρόνους, και όχι μόνο στο τελευταίο πακέτο δεδομένων. Ένα από τα πλεονεκτήματα αυτής της προσέγγισης είναι ότι επιτρέπει σε έναν ταξινομητή που είχε χαμηλή απόδοση σε προηγούμενες

χρονικές στιγμές – και άρα είχε λάβει μικρό ή καθόλου βάρος ψήφου (voting weight) – να επανενεργοποιηθεί και να αποκτήσει μεγαλύτερο βάρος ψήφου εάν γίνει ξανά αποδοτικός στα τρέχοντα δεδομένα (αυτό μπορεί να συμβεί για διάφορους λόγους, π.χ. λόγω περιοδικότητας των μεταβολών πλαισίου).
Ας δούμε αναλυτικά τον ψευδοκώδικα. Αρχικά, ο αλγόριθμος λαμβάνει πακέτα δεδομένων $\boldsymbol{S}_t$ τα οποία προέρχονται από διαφορετικές ή από μεταβαλλόμενες κατανομές πιθανότητας $p_t(\boldsymbol{x}, y)$. Σε κάθε χρονική στιγμή, ο αλγόριθμος Learn$^{++}$.NSE μετράει το σφάλμα του παρόντος ensemble πάνω στα πιο πρόσφατα δεδομένα $\boldsymbol{S}_t$ (γραμμή 2 και σχέση (1)). Ομοίως με τον αλγόριθμο Adaboost ([76]), ο Learn$^{++}$.NSE κρατάει ένα σύνολο βαρών πάνω στα δεδομένα (εδώ ο αναγνώστης πρέπει να προσέξει ότι αυτά τα βάρη δεν είναι τα βάρη ψήφισης των ταξινομητών), τέτοια ώστε ένα μεγάλο βάρος υποδεικνύει ότι τα δείγματα αυτά είναι πιο δύσκολο να ταξινομηθούν από τα υπόλοιπα, τα οποία έχουν χαμηλότερο βάρος (γραμμή 3 και σχέση (2)). Στα πλαίσια του concept drift, ένα πακέτο απ' την νέα κατανομή που προκύπτει είναι δύσκολο να ταξινομηθεί με το υπάρχον ensemble. Αντίθετα όμως απ' τον Adaboost, ο Learn$^{++}$.NSE όταν δημιουργεί έναν νέο ταξινομητή (γραμμή 4) δεν ελαχιστοποιεί τη συνάρτηση σφάλματος πάνω στο $\boldsymbol{S}_t$ σύμφωνα με τα βάρη, αλλά χρησιμοποιεί τη χρονοεξαρτημένη συνάρτηση σφάλματος (σχέσεις (3), (4) και (5)), δίνοντας έτσι στην απόδοση στην παρούσα χρονική στιγμή μεγαλύτερο βάρος απ' ότι στην απόδοση κατά το παρελθόν. Συγκεκριμένα, σε αντίθεση με άλλες προσεγγίσεις που χρησιμοποιούν το σφάλμα στα πιο πρόσφατα δεδομένα ([66]), ο Learn$^{++}$.NSE εφαρμόζει μία σιγμοειδή μέση τιμή (sigmoidal averaging) (γραμμή 6) στο ιστορικό σφάλματος (loss history) των ταξινομητών, το οποίο προωθεί τους ταξινομητές που αποδίδουν καλά σε πρόσφατους χρόνους. Αυτή η χρονοεξαρτώμενη συνάρτηση σφάλματος είναι ένα απ' τα κύρια πλεονεκτήματα του Learn$^{++}$.NSE, το οποίο του επιτρέπει στο ensemble να είναι περισσότερο ενημερωμένο στα πρόσφατα δεδομένα. Αυτή η προσέγγιση έχει συγκριθεί εμπειρικά με άλλες, όπως π.χ. η SEA, και η σύγκριση έδειξε ότι είναι αρκετά αποτελεσματική στο να δίνει σταθερότητα στον αλγόριθμο, λόγω της δυνατότητάς του να επαναφέρει πληροφορίες που έμαθε στο παρελθόν. Σε αντίθεση με τους περισσότερους ensemble αλγόριθμους, ο Learn$^{++}$.NSE δεν απορρίπτει τους παλιούς ταξινομητές, απλώς τους δίνει ένα προσαρμοζόμενο βάρος ψήφου, το οποίο τους επιτρέπει να ξαναενεργοποιηθούν αργότερα, σε περίπτωση φαινομένων περιοδικού ή επαναλαμβανόμενου drift. Μία σύγκριση των παραδοσιακών μηχανισμών απόδοσης των βαρών, που εξαρτώνται απ' το χρόνο, ή την ακρίβεια ταξινόμησης (με σταθερό μέγεθος ensemble), με την παρούσα χρονοεξαρτώμενη μέθοδο έδειξε ότι η τελευταία είναι αποδοτικότερη απ' τις άλλες δύο ως προς την ακρίβεια της ταξινόμησης ([77]).

Τέλος, ensemble προσεγγίσεις έχουν εφαρμοστεί επίσης και σε άλλες κατηγορίες μη στάσιμης μάθησης, όπως π.χ. το transfer learning και το multi – task learning (δείτε το σχήμα 2.5.1).

## 2.10 Νέες τάσεις και προκλήσεις

Ενώ η μάθηση σε μεταβαλλόμενα δεδομένα είναι από μόνη της μία σημαντική πρόκληση, υπάρχουν κάποιες φορές και επιπλέον περιορισμοί (κάποιοι απ' τους οποίους είναι αυτόνομα προβλήματα της μηχανικής μάθησης), που κάνουν την κατάσταση αρκετά δυσκολότερη. Ένα από αυτά είναι για παράδειγμα το πρόβλημα της ανισορροπίας των κλάσεων (class imbalance). Το πρόβλημα αυτό εμφανίζεται όταν τα δεδομένα των διαφόρων κλάσεων είναι σε μεγάλο βαθμό ανισοκατανεμημένα μεταξύ τους, έτσι ώστε κάποιες κλάσεις να περιέχουν πολύ περισσότερα πρότυπα από άλλες. Στην περίπτωση αυτή, συνήθως το μεγαλύτερο σφάλμα ταξινόμησης το έχει η κλάση με τα λιγότερα πρότυπα. Το πρόβλημα αυτό προκύπτει συχνά σε πρακτικές εφαρμογές, και είναι από μόνο του ένα σημαντικό πρόβλημα της μηχανικής μάθησης ([78] – [80]). Παρόλα αυτά, οι μεθοδολογίες που έχουν αναπτυχθεί για την αντιμετώπιση του class imbalance αναφέρονται συνήθως σε στάσιμες συνθήκες, ενώ η περίπτωση της μεταβαλλόμενης μάθησης με ανισορροπία κλάσεων έχει μελετηθεί λίγο.
Ένας απ' τους πρώτους αλγόριθμους για την από κοινού αντιμετώπιση τόσο του concept drift, όσο και της ανισορροπίας κλάσεων, είναι το ασυσχέτιστο bagging (uncorrelated bagging). Σύμφωνα με τον αλγόριθμο αυτό, χρησιμοποιείται ένα ensemble ταξινομητών, το οποίο εκπαιδεύεται πάνω σε δείγματα της μεγαλύτερης σε πληθυσμό κλάσης που προκύπτουν με υποδειγματοληψία, και στη συνέχεια οι ταξινομητές συνδυάζονται χρησιμοποιώντας έναν μέσο όρο των εξόδων τους ([81], [82]). Επίσης, οι αλγόριθμοι Selective Recursive Approach (SERA) και Recursive Ensemble Approach (REA) είναι παρόμοιας λειτουργίας με το uncorrelated bagging, και χρησιμοποιούν ένα σύστημα σταθμισμένης ψηφοφορίας – παρόλα αυτά, οι μέθοδοι αυτές δεν απαιτούν πρόσβαση σε ιστορικά δεδομένα ([83], [84]).
Ο Learn$^{++}$.CDS (Concept Drift with SMOTE) είναι ένας άλλος πρόσφατος αλγόριθμος που έχει σχεδιαστεί για να μαθαίνει από non – stationary δεδομένα με ανισορροπία κλάσεων, ο οποίος επίσης δεν χρειάζεται πρόσβαση σε ιστορικά δεδομένα ([85], [86]).

Πρόσφατες εργασίες έχουν προσπαθήσει να επεκτείνουν τις ιδέες αυτές σε online αλγορίθμους, όπως π.χ. στις εργασίες [87] και [88]. Η ανάπτυξη όμως ενός πραγματικά online αλγορίθμου για ανίχνευση μεταβολών, ο οποίος να μη χρειάζεται πρόσβαση σε ιστορικά δεδομένα είναι μία μεγάλη πρόκληση, λόγω των δυσκολιών που προκύπτουν από την ανάγκη της μέτρησης των στατιστικών δεδομένων των μικρότερων κλάσεων, χωρίς να παραβιάσουμε την αρχή λειτουργίας ενός online αλγορίθμου, ότι δηλαδή έρχεται μόνο ένα δείγμα τη φορά, και είναι διαθέσιμο μόνο κατά τη στιγμή που έρχεται. Μία πρόσφατη προσπάθεια στην κατεύθυνση αυτή αναφέρεται στην [89], στην οποία αναπτύσσεται ένας online αλγόριθμος για ταξινόμηση δεδομένων από πολλαπλές κλάσεις σε μεταβαλλόμενο περιβάλλον. Παρόλα αυτά, το ζήτημα αυτό παραμένει ακόμα ένα ανοιχτό πεδίο έρευνας ([89]).

Μερικά άλλα πεδία της μηχανικής μάθησης, που έχουν μελετηθεί εκτενώς σε στάσιμες συνθήκες, είναι η ημι – επιβλεπόμενη μάθηση, η μάθηση χωρίς επίβλεψη,

καθώς και αρκετά θέματα της μεταβιβαστικής (transductive), και της ενεργητικής μάθησης. Η εφαρμογή των μεθόδων αυτών σε μη - στάσιμες περιπτώσεις έχει εξεταστεί μόνο πρόσφατα. Στην περίπτωση της ημιεπιβλεπόμενης και της μεταβιβαστικής μάθησης για παράδειγμα χρησιμοποιούνται μη επισημασμένα δεδομένα απ' το σύνολο ελέγχου για τον καθορισμό των παραμέτρων του μοντέλου ([91], [92]). Στην περίπτωση πάλι της μη επιβλεπόμενης μάθησης – συσταδοποίησης (clustering), η μάθηση γίνεται χωρίς τη χρήση επισημασμένων δεδομένων ([93]), ενώ στην ενεργητική μάθηση (η οποία δεν πρέπει να συγχέεται με τις ενεργές μεθόδους για μάθηση σε μεταβαλλόμενα περιβάλλοντα που παρουσιάστηκαν προηγουμένως), ο αλγόριθμος εντοπίζει τα πιο σημαντικά δεδομένα για το πρόβλημα της μάθησης, και ζητά ετικέτες για αυτά τα δεδομένα ([94]).

Μία ιδιαίτερα απαιτητική περίπτωση της ημι – επιβλεπόμενης ή της μη – επιβλεπόμενης μάθησης σε μεταβαλλόμενα περιβάλλοντα είναι το σενάριο (που προκύπτει συχνά στην πράξη), όπου τα επισημασμένα δεδομένα είναι σπάνια, είτε είναι διαθέσιμα μόνο στην αρχή, και ακολουθούνται από μία ροή μη επισημασμένων δεδομένων που προέρχονται από μία μεταβαλλόμενη κατανομή. Αυτές οι περιπτώσεις είναι γνωστές με τον όρο initially labeled nonstationary streaming (ILNS) data, και το πρόβλημα αυτό περιλαμβάνει αρκετές πραγματικές εφαρμογές, όπως η διαχείριση δικτύων ηλεκτρικής ενέργειας, συστήματα τηλεπισκόπησης (remote – sensing), η ασφάλεια υπολογιστών και ο εντοπισμός κακόβουλων εφαρμογών, ή η συλλογή δεδομένων από απομακρυσμένες και επικίνδυνες τοποθεσίες, όπως π.χ. πυρηνικοί σταθμοί, τοξικά περιβάλλοντα, κλπ., όπου η συλλογή μεγάλων ποσοτήτων επισημασμένων δεδομένων μπορεί να είναι δύσκολη ή επικίνδυνη.

Η μάθηση σε ένα περιβάλλον όπου τα labels δεν γίνονται άμεσα γνωστά είναι γνωστή με τον όρο καθυστέρηση επαλήθευσης (verification latency), και στις περιπτώσεις αυτές απαιτείται ένας μηχανισμός που να μεταδίδει την πληροφορία των labels ανάμεσα σε πολλά χρονικά βήματα, στα οποία μεσολαβούν μη επισημασμένα δεδομένα. Οι Zhang et al. πρότειναν μία ensemble προσέγγιση η οποία συνδυάζει ταξινομητές και συσταδοποίηση ([95]), η οποία λειτουργεί ικανοποιητικά όταν τα επισημασμένα δεδομένα είναι διαθέσιμα τουλάχιστον περιοδικά: τα επισημασμένα δεδομένα χρησιμοποιούνται για την εκπαίδευση ενός ταξινομητή, ενώ τα μη επισημασμένα δεδομένα χρησιμοποιούνται για τη δημιουργία συστάδων. Τα νέα δείγματα ταξινομούνται στη συνέχεια με βάση μία ψήφο πλειοψηφίας των ταξινομητών, η οποία περιλαμβάνει αντιστοίχιση ετικετών μεταξύ ταξινομητών και συστάδων. Αντίθετα, μία άλλη προσέγγιση περιλαμβάνει την αναπαράσταση κάθε μεταβαλλόμενης (drifting) κλάσης ως ένα μείγμα υποπληθυσμών (subpopulations), όπου ο καθένας προέρχεται από μία συγκεκριμένη παραμετρική κατανομή. Δεδομένων κάποιων αρχικών επισημασμένων δεδομένων, οι υποπληθυσμοί των μη επισημασμένων δεδομένων μπορούν να ανιχνευτούν και να αντιστοιχηθούν σε αυτούς τους γνωστούς υποπληθυσμούς, όπως δείχνεται στις εργασίες [96] – [98], καθώς και στον αλγόριθμο Arbitrary subPopulation Tracker (APT) στην αναφορά [99].
Οι προαναφερθείσες προσεγγίσεις γενικά υποθέτουν ότι: Πρώτον, ότι τα drift είναι βαθμιαία και μπορούν να προσεγγιστούν μέσω τμηματικά γραμμικών

συναρτήσεων. Δεύτερον, ότι κάθε υποπληθυσμός είναι παρόν στη διαδικασία αρχικοποίησης, και ο πίνακας συνδιασποράς παραμένει σταθερός. Τρίτον, ο ρυθμός του drift (drift rate) παραμένει σταθερός. Συγκεκριμένα, ο αλγόριθμος APT που προαναφέρθηκε περιλαμβάνει μία διαδικασία δύο βημάτων: Πρώτον, μία διαδικασία αναμενόμενης τιμής – μεγιστοποίησης (expectation maximization) χρησιμοποιείται για την εύρεση της βέλτιστης ένα προς έναν αντιστοίχισης μεταξύ των μη επισημασμένων και των επισημασμένων δεδομένων, που έχουν προσαρμοστεί με βάση το drift, υποθέτοντας την τμηματικά γραμμική προσέγγισή του. Στη συνέχεια, ο ταξινομητής ενημερώνεται έτσι ώστε να ανταποκρίνεται στις παραμέτρους πληθυσμού των νέων λαμβανόμενων δεδομένων.

Πρόσφατα παρουσιάστηκε το πλαίσιο COMPOSE (COMPacted Object Sample Extraction), το οποίο μπορεί να χειριστεί δεδομένα πολλών κλάσεων, και περιλαμβάνει επίσης το σενάριο του να προστίθενται και νέες κλάσεις ή υποπληθυσμοί στα δεδομένα, κάνοντας μόνο την υπόθεση του βαθμιαίου drift ([100], [101]). Δεδομένων επισημασμένων δεδομένων μόνο κατά το στάδιο της αρχικοποίησης, ακολουθούμενα από μη επισημασμένα μεταβαλλόμενα δεδομένα, ο αλγόριθμος COMPOSE κάνει επαναληπτικά τα εξής: Αρχικά συνδυάζει τα αρχικά επισημασμένα δεδομένα με τα νέα μη επισημασμένα δεδομένα και εκπαιδεύει έναν ημι – επιβλεπόμενο αλγόριθμο μάθησης (SSL) έτσι ώστε να δώσει labels στα μη επισημασμένα δεδομένα. Στη συνέχεια, για κάθε κλάση, δημιουργεί μία στενή περιβάλλουσα (tight envelope) πάνω στα δεδομένα, χρησιμοποιώντας μία μέθοδο εκτίμησης πυκνότητας πιθανότητας (density) σε περιοχές (multi – modal region), όπως π.χ. τα a – shapes, ή ένα μοντέλο μείξης Γκαουσσιανών. Έπειτα, στο επόμενο βήμα, ο αλγόριθμος «στενεύει» την περιβάλλουσα αυτή, ώστε να εντοπίσει τον πυρήνα της περιοχής στήριξης (core support region) για κάθε κλάση, από τον οποίο μπορούν να προκύψουν labeled δεδομένα, τα ονομαζόμενα core supports. Τα δεδομένα αυτά αποτελούν τα νέα επισημασμένα δεδομένα που θα χρησιμοποιηθούν στην επόμενη επανάληψη, και θα συνδυαστούν με τα νέα μη επισημασμένα δεδομένα, τα οποία θα γίνουν ξανά labeled μέσω του SSL αλγορίθμου, κλπ. Το COMPOSE ενδείκνυται για περιπτώσεις ακραίας υστέρησης επαλήθευσης (verification latency), όπου νέα επισημασμένα δεδομένα δεν φτάνουν ποτέ. Παρόλα αυτά, αν το πρόβλημα παρέχει επιπλέον labeled δεδομένα, ίσως μόνο περιοδικά, αυτά μπορούν να χρησιμοποιηθούν για την ενημέρωση των core supports, και η ύπαρξη των δεδομένων αυτών μπορεί να χαλαρώσει ή να εξαλείψει την απαίτηση του αλγορίθμου για βαθμιαίο drift. Επιπλέον, εάν το πρόβλημα μας επιτρέπει να ζητήσουμε επιπλέον επισημασμένα δεδομένα από έναν χρήστη, π.χ. σε ένα πρόβλημα ενεργητικής μάθησης, τότε το COMPOSE μπορεί εύκολα να συνδυαστεί με έναν αλγόριθμο ενεργητικής μάθησης, έτσι ώστε να εκμεταλλευτεί τη δυνατότητα αυτή ([102]).

## 2.11 Συμπεράσματα και μελλοντική έρευνα

Η μάθηση σε μεταβαλλόμενα περιβάλλοντα αποτελεί μία νέα, υποσχόμενη περιοχή έρευνας στον τομέα της μηχανικής μάθησης και της υπολογιστικής νοημοσύνης,

λόγω της αυξανόμενης σημασίας της στις εφαρμογές του πραγματικού κόσμου, η οποία έχει πάρει νέα ώθηση εξαιτίας παραγόντων όπως τα big data. Στις εφαρμογές αυτές, η χρήση παραδοσιακών μεθόδων, οι οποίες αγνοούν το υποκείμενο drift, είναι καταδικασμένη να αποτύχει, καθιστώντας αναγκαίους νέους αλγορίθμους, οι οποίοι να μπορούν να ανιχνεύουν τις μεταβολές, και να προσαρμόζονται σε αυτές. Στις παραπάνω ενότητες παρουσιάσαμε μία σύντομη ανασκόπηση του πεδίου αυτού, των σχετικών προβλημάτων, και είδαμε περιληπτικά τις κυριότερες μεθόδους της βιβλιογραφίας.
Παρόλα αυτά, παρά την μεγάλη προσπάθεια που έχει ήδη γίνει, υπάρχει ακόμη ένας σημαντικός όγκος ανοιχτών προβλημάτων που πρέπει να αντιμετωπιστούν. Παρακάτω παρουσιάζουμε τα κυριότερα απ' αυτά.

- **Ανάπτυξη ενός θεωρητικού πλαισίου για τη μάθηση:** Το πεδίο της μάθησης σε μεταβαλλόμενα περιβάλλοντα μπορεί να βελτιωθεί σημαντικά από μία εις βάθος θεωρητική ανάλυση ενός γενικού πλαισίου, όπου θεωρητικά άνω όρια της ακρίβειας ταξινόμησης θα μπορούν να υπολογίζονται με βάση τον τύπο και το βαθμό του drift.

- **Αδόμητες και ετερογενείς ροές δεδομένων:** Ένα κεντρικό ζήτημα της μάθησης από big data είναι η ανάγκη της διαχείρισης τεράστιων ποσοτήτων από αδόμητα και ετερογενή δεδομένα, όπως π.χ. κείμενα, εικόνες, γράφοι, κλπ. Επιπλέον, τα δεδομένα που συλλέγονται για τη μάθηση μπορεί να έχουν διαφορετικά χαρακτηριστικά, όπως π.χ. πολυδιάστατα, πολλών κλάσεων, πολυκλιμακωτά, καθώς και χωρικές συσχετίσεις (π.χ. εικόνες). Επομένως, στα πλαίσια της έρευνας για το concept drift θα ήταν χρήσιμη η ανάπτυξη μεθόδων και στρατηγικών για τη διαχείριση τέτοιων δεδομένων.

- **Ακριβής ορισμός του περιορισμένου – βαθμιαίου drift (limited – gradual drift):** Το περιορισμένο – βαθμιαίο drift είναι μία βασική προϋπόθεση πολλών αλγορίθμων μάθησης σε μεταβαλλόμενα περιβάλλοντα, και ιδιαίτερα στις περιπτώσεις της ημιεπιβλεπόμενης και της μη επιβλεπόμενης μάθησης. Παρόλα αυτά, ο ορισμός του τι ακριβώς είναι το περιορισμένο drift είναι ασαφής. Όχι μόνο δεν διαθέτουμε μεθόδους που να αντιμετωπίζουν τις περιπτώσεις εκείνες όπου η υπόθεση αυτή παραβιάζεται, αλλά δεν έχουμε ούτε καν έναν μαθηματικά ακριβή ορισμό του τι είναι το περιορισμένο drift. Η ύπαρξη ενός ακριβούς μαθηματικού ορισμού θα επέτρεπε στους ερευνητές να κατανοήσουνε σε μεγαλύτερο βάθος τους περιορισμούς της μάθησης σε μεταβαλλόμενα περιβάλλοντα.

- **Παροδικό (transient) concept drift και περιορισμένα δεδομένα:** Το πρόβλημα αυτό αναφέρεται στην περίπτωση όπου το concept drift είναι παροδικό, και τα δείγματα που συσχετίζονται με τη μεταβολή είναι πολύ λίγα. Αυτή η κατάσταση αποτελεί μία ιδιαίτερη πρόκληση, καθώς η εκτίμηση των χαρακτηριστικών που θα χρησιμοποιηθούν για την ανάλυση του change detection γίνεται πάνω σε ένα πολύ μικρό δείγμα δεδομένων,

πράγμα που μειώνει την ακρίβεια της εκτίμησης των παραμέτρων του μεταβαλλόμενου περιβάλλοντος.

## 2.12 Μέθοδοι αναλλοίωτης μεταβολής (covariant shift)

Όπως είδαμε προηγουμένως, το covariant shift είναι μία υποπερίπτωση του concept drift, στην οποία έχουμε την εξής κατάσταση: έχουμε δύο σύνολα δεδομένων, ένα σύνολο εκπαίδευσης, και ένα σύνολο ελέγχου, ενώ η μάθηση δεν είναι ακολουθιακή (online). Σε αντίθεση με τη συνήθη μηχανική μάθηση, οι κατανομές πιθανότητας $p_{tr}(x)$ και $p_{te}(x)$ διαφέρουν μεταξύ τους, ενώ είναι ανεξάρτητες του χρόνου (αφού η μάθηση δεν είναι online). Επιπλέον, η υπό συνθήκη πιθανότητα $p(y|x)$ είναι και για τα δύο σύνολα δεδομένων σταθερή, και ανεξάρτητη του χρόνου. Αυτό που εκφράζει η συνθήκη αυτή στην ουσία είναι ότι η συνάρτηση αντιστοίχισης, $y = f(x)$ παραμένει σταθερή για τα δύο σύνολα δεδομένων (αντίθετα, στη γενική περίπτωση μπορεί να μεταβάλλεται). Στη γενική περίπτωση του concept drift, που δεν έχουμε κάποιον περιορισμό, εφαρμόζονται οι γενικές active ή passive μέθοδοι, οι οποίες στην ουσία αναγνωρίζουν τις μεταβολές και ενημερώνουν τους ταξινομητές. Στην περίπτωση του covariant shift όμως, λόγω των αυστηρότερων περιορισμών του προβλήματος, μπορούμε να κατασκευάσουμε κάποιες «πιο μαθηματικές» μεθόδους για την εκπαίδευση των ταξινομητών. Μερικές απ' τις κυριότερες μεθόδους παρουσιάζονται παρακάτω.

Η κύρια ιδέα πίσω απ' τις παραπάνω μεθόδους είναι η χρήση του λεγόμενου βάρους σημαντικότητας (importance weight). Για να εξηγήσουμε την ιδέα αυτή, ας θεωρήσουμε ένα πρόβλημα μηχανικής μάθησης, στο οποίο έχουμε να μάθουμε μία συνάρτηση $f(x)$ δεδομένου ενός συνόλου εκπαίδευσης $\{(x_i^{tr}, y_i^{tr})\}_{i=1}^{n_{tr}}$. Για να το κάνουμε αυτό, επιλέγουμε ένα μοντέλο $\hat{f}(x;\theta)$, και ο στόχος μας είναι να υπολογίσουμε την παράμετρο $\theta$ έτσι ώστε η $\hat{f}(x;\theta)$ να προσεγγίζει την $f(x)$ όσο το δυνατόν καλύτερα. Ο συνήθης τρόπος να υπολογιστεί το $\theta$ είναι μέσω της μεθόδου ERM (empirical risk minimization), όπου η παράμετρος $\theta$ δίνεται απ' την παρακάτω σχέση:

$$\hat{\theta} = \operatorname{argmin}\left(\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}} \operatorname{loss}\left(x_i^{tr}, y_i^{tr}, \hat{f}(x_i^{tr};\theta)\right)\right), \qquad (2.12.1)$$

όπου η συνάρτηση $\operatorname{loss}\left(x, y, \hat{f}(x;\theta)\right)$ είναι μία συνάρτηση κόστους (loss function), η οποία υπολογίζει το πόσο καλά προσεγγίζει η $\hat{f}(x;\theta)$ το σημείο $y$ – για παράδειγμα, στη μέθοδο των ελαχίστων τετραγώνων που παρουσιάστηκε προηγουμένως, η συνάρτηση κόστους είναι η $\operatorname{loss}\left(x, y, \hat{f}(x;\theta)\right) = \left(y - \hat{f}(x;\theta)\right)^2$, η οποία ισούται απλώς με το τετράγωνο της απόστασης της τιμής του μοντέλου $\hat{f}(x;\theta)$ απ' την πραγματική τιμή $y$. Επομένως, αυτό που κάνει η μέθοδος ERM δεν είναι τίποτε άλλο, απ' το να βρίσκει το $\hat{\theta}$ που ελαχιστοποιεί τη συνάρτηση κόστους για όλα τα σημεία εκπαίδευσης. Φυσικά,

αναάλογα με το πρόβλημα, η κατάλληλη συνάρτηση κόστους μπορεί να διαφέρει (σε ένα πρόβλημα ταξινόμησης λ.χ., το να επιλέξουμε ως συνάρτηση κόστους το τετράγωνο της απόστασης δεν θα είχε ιδιαίτερο νόημα).
Στην περίπτωση της συνήθους μηχανικής μάθησης, όπου $p_{tr}(x) = p_{te}(x)$, αποδεικνύεται ότι η μέθοδος ERM είναι συνεπής: αυτό σημαίνει ότι όταν ο αριθμός των δειγμάτων γίνει αρκετά μεγάλος (τείνει στο άπειρο), η παράμετρος $\hat{\theta}$ που υπολογίζει η μέθοδος ισούται με την παραγματική βέλτιστη παράμετρο $\theta^*$ του μοντέλου, η οποία ορίζεται με την παρακάτω σχέση:

$$\theta^* = \operatorname{argmin}\left\{\mathbb{E}_{x^{te}}\mathbb{E}_{y^{te}}\left(\operatorname{loss}\left(x^{te}, y^{te}, \hat{f}(x^{te};\theta)\right)\right)\right\}. \tag{2.12.2}$$

Επομένως, το $\theta^*$ είναι η παράμετρος που ελαχιστοποιεί το λεγόμενο σφάλμα γενίκευσης, το οποίο ισούται με την αναμενόμενη τιμή της συνάρτησης κόστους πάνω στο συνολο ελέγχου (test set). Στην περίπτωση όμως του covariant shift, όπου $p_{tr}(x) \neq p_{te}(x)$, η μέθοδος δεν είναι πλέον συνεπής, αφού αποδεικνύεται εύκολα ότι $\hat{\theta}_{ERM} \neq \theta^*$.

Ο λόγος που η παραπάνω μέθοδος είναι πλέον μη συνεπής προέρχεται απ' το γεγονός του ότι η κατανομή των δεδομένων εκπαίδευσης είναι τώρα διαφορετική απ΄ την κατανομή των δεδομένων ελέγχου. Για να λυθεί το πρόβλημα αυτό χρησιμοποιείται μέθοδος της στάθμισης σημαντικότητας (importance weighting), η οποία χρησιμοποιείται για να αντισταθμίσει τη διαφορά μεταξύ των δύο κατανομών. Για να εισάγουμε τη μέθοδο αυτή, θεωρούμε την παρακάτω σχέση, η οποία υπολογίζει την αναμενόμενη τιμή μίας συνάρτησης $g$ πάνω στο σύνολο ελέγχου:

$$\mathbb{E}_{x^{te}}[g(x^{te})] = \int g(x)p_{te}(x)dx =$$

$$= \int g(x)\frac{p_{te}(x)}{p_{tr}(x)}p_{tr}(x)dx = \mathbb{E}_{x^{tr}}\left[g(x^{tr})\frac{p_{te}(x^{tr})}{p_{tr}(x^{tr})}\right]. \tag{2.12.3}$$

Βλέπουμε λοιπόν ότι ο υπολογισμός της αναμενόμενης τιμής της $g$ στο σύνολο ελέγχου μπορεί να αναχθεί σε έναν υπολογισμό που γίνεται πάνω στο σύνολο εκπαίδευσης, με την προϋπόθεση ότι η συνάρτηση $g(x)$ θα αντικατασταθεί με τη συνάρτηση $g(x)\frac{p_{te}(x)}{p_{tr}(x)}$. Ο παράγοντας

$$h(x) = \frac{p_{te}(x)}{p_{tr}(x)}, \tag{2.12.4}$$

ο οποίος πολλαπλασιάζει την $g(x)$ ονομάζεται σημαντικότητα, και η τιμή του μας δίνει το βάρος με το οποίο πρέπει να πολλαπλασιάσουμε την $g(x)$ έτσι ώστε να αντισταθμίσουμε τη διαφορά μεταξύ των δύο κατανομών, και να πάρουμε τη σωστή μέση τιμή χρησιμοποιώντας μόνο το σύνολο εκπαίδευσης. Αυτή είναι η βάση της μεθόδου της στάθμευσης σηαντικότητας, την οποία θα χρησιμοποιήσουμε για να τροποποιήσουμε τους αλγορίθμους μηχανικής μάθησης έτσι ώστε να δίνουν

σωστά αποτελέσματα και στην περίπτωση του covariant shift. Συγκεκριμένα, εφαρμόζοντας τη μέθοδο αυτή στο ERM, παίρνουμε τη σταθμισμένη μέθοδο ERM (importance weighted ERM – IWERM), η οποία εκφράζεται από τη σχέση:

$$\hat{\theta} = \text{argmin}\left(\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}}\frac{p_{te}(x_i^{tr})}{p_{tr}(x_i^{tr})}\text{loss}\left(x_i^{tr}, y_i^{tr}, \hat{f}(x_i^{tr};\theta)\right)\right). \qquad (2.12.5)$$

Μπορούμε να δείξουμε ότι η μέθοδος τώρα θα είναι συνεπής, δηλαδή θα ισχύει $\hat{\theta}_{IWERM} = \theta^*$.
Παρόλα αυτά, αποδεικνύεται επίσης ότι η μέθοδος αυτή είναι και ασταθής. Μία μέθοδος για να το λύσουμε αυτό είναι να αντικαταστήσουμε την παραπάνω σχέση με την ακόλουθη:

$$\hat{\theta} = \text{argmin}\left(\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}}\left(\frac{p_{te}(x_i^{tr})}{p_{tr}(x_i^{tr})}\right)^{\gamma}\text{loss}\left(x_i^{tr}, y_i^{tr}, \hat{f}(x_i^{tr};\theta)\right)\right), \qquad (2.12.5)$$

όπου το $\gamma$ είναι μία παράμετρος που ανήκει στο διάστημα $[0,1]$. Το $\gamma = 0$ αντιστοχεί στη συνήθη μέθοδο, η οποία είναι μη συνεπής, ενώ το $\gamma = 1$ αντιστοιχεί στην σταθμισμένη μέθοδο, η οποία είναι συνεπής αλλά ασταθής. Επομένως, στην ουσία το $\gamma$ εξομαλύνει τον παράγοντα σημαντικότητας, και πετυχαίνει μία ισορροπία μεταξύ συνέπειας και ευστάθειας. Χονδρικά, η επιλογή του $\gamma$ εξαρτάται απ' τον αριθμό των δειγμάτων. Όταν το $n_{tr}$ είναι μεγάλο, η μεροληψία (bias) κυριαρχεί της διακύμανσης (variance), οπότε στην περίπτωση αυτή συνήθως επιλέγουμε το $\gamma$ να είναι μεγάλο, διότι με τον τρόπο αυτό αυξάνουμε τις παραμέτρους του μοντέλου (δίνοντας συντελεστές στους όρους της 2.12.5), και αυξάνοντας το βάρος των δειγμάτων που δεν ακολουθούν την κατανομή του συνόλου εκπαίδευσης. Αντίθετα, όταν έχουμε λίγα δείγματα εκπαίδευσης, το variance κυριαρχεί συνήθως του bias, οπότε στην περίπτωση αυτή επιλέγουμε ένα πιο μικρό $\gamma$, έτσι ώστε να μειώσουμε το variance του μοντέλου. Τέλος, ανάλογα με το μοντέλο που χρησιμοποιούμε, υπάρχουν διάφορες μέθοδοι για την σωστή επιλογή του $\gamma$, οι οποίες περιγράφονται αναλυτικά στο [2]. Η πιο στοιχειώδης είναι η χρήση της cross – validation για διάφορες εύλογες τιμές στο διάστημα $[0,1]$. Τέλος, στην περίπτωση που χρησιμοποιούμε regularization (συστηματικοποίηση), η παραπάνω σχέση τροποποιείται ως εξής:

$$\hat{\theta} = \text{argmin}\left(\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}}\left(\frac{p_{te}(x_i^{tr})}{p_{tr}(x_i^{tr})}\right)^{\gamma}\text{loss}\left(x_i^{tr}, y_i^{tr}, \hat{f}(x_i^{tr};\theta)\right) + R(\theta)\right), \quad (2.12.6)$$

όπου η $R(\theta)$ είναι η συνάρτηση συστηματικοποίησης (regularization function).

Ας δούμε τώρα πως τροποποιούνται διάφοροι γνωστοί αλγόριθμοι μηχανικής μάθησης με βάση τα παραπάνω. Αρχικά, για τη μέθοδο των ελαχίστων τετραγώνων, η σχέση γίνεται:

$$\hat{\theta} = \text{argmin}\left(\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}}\left(\frac{p_{te}(x_i^{tr})}{p_{tr}(x_i^{tr})}\right)^{\gamma}(y_{tr} - \theta \cdot x_{tr})^2\right), \quad (2.12.7)$$

λόγω της συνάρτησης κόστους στην περίπτωση αυτή.
Στην περίπτωση της λογιστικής παλινδρόμησης (logistic regression), η (2.12.5) γίνεται:

$$\hat{\theta} = \text{argmin}\left(\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}}\left(\frac{p_{te}(x_i^{tr})}{p_{tr}(x_i^{tr})}\right)^{\gamma}\log\left(1 + exp\left(-y_i^{tr}\hat{f}(x_i^{tr};\theta)\right)\right)\right), \quad (2.12.8)$$

όπου ο τελευταίος όρος είναι η συνάρτηση κόστους της logistic regression.
Στην περίπτωση που χρησιμοποιούμε μοντέλα με πυρήνες (kernels), η μέθοδος είναι ακριβώς η ίδια, με τη διαφορά ότι τώρα η συνάρτηση – μοντέλο γράφεται στη μορφή $\hat{f}(x^{tr};\theta) = \sum_{i=1}^{n^{tr}}\theta_i K(x, x_i^{tr})$, όπου $K(x, x_i^{tr})$ η συνάρτηση πυρήνα.
Τέλος, στην περίπτωση των SVM, η συνάρτηση κόστους μπορεί να γραφτεί στη μορφή

$$\hat{\theta} = \text{argmin}\left(\sum_{i=1}^{n_{tr}}\max\left(0, 1 - y_i^{tr}\hat{f}(x_i^{tr};\theta)\right)\right), \quad (2.12.9)$$

οπότε η adaptive εκδοχή είναι η παρακάτω:

$$\hat{\theta} = \text{argmin}\left(\sum_{i=1}^{n_{tr}}\left(\frac{p_{te}(x_i^{tr})}{p_{tr}(x_i^{tr})}\right)^{\gamma}\max\left(0, 1 - y_i^{tr}\hat{f}(x_i^{tr};\theta)\right)\right). \quad (2.12.10)$$

Τέλος, ένα πράγμα που δεν αναφέραμε είναι το πώς υπολογίζονται οι παράγοντες σημαντικότητας $\frac{p_{te}(x_i^{tr})}{p_{tr}(x_i^{tr})}$. Αυτοί μπορούν να εκτιμηθούν με συνήθεις στατιστικές τεχνικές, όπως περιγράφεται αναλυτικά στο [2]. Αυτό ολοκληρώνει τη σύντομη παρουσίαση των κυριότερων μεθόδων στην περίπτωση του covariant shift.

# 3 Οι ιεραρχικοί ανιχνευτές μεταβολών

## 3.1 Ενεργές μέθοδοι

Όπως είδαμε στο προηγούμενο κεφάλαιο, οι active μέθοδοι για μάθηση σε περιβάλλοντα με concept drift, βασίζονται στην αλληλεπίδραση του ταξινομητή με έναν ανιχνευτή μεταβολής (change detector), ο οποίος πυροδοτείται όταν ανιχνεύσει μία μεταβολή στη στατιστική κατανομή των δεδομένων εισόδου. Η ανίχνευση αυτή γίνεται μελετώντας κάποια στατιστικά χαρακτηριστικά που εξάγονται απ' τα δεδομένα εισόδου, τα οποία αναμένεται να μένουν γενικά σταθερά όταν η κατανομή των δεδομένων είναι σταθερή, αλλά αναμένεται να αλλάξουν όταν υπάρξει κάποια μεταβολή. Οι συνηθέστερες μέθοδοι για την ανίχνευση αυτών των μεταβολών είναι, όπως είδαμε πριν, οι μέθοδοι σημείου μεταβολής (Change – Point Methods – CPM), καθώς και οι μέθοδοι ελέγχου ανίχνευσης μεταβολής (Change – Detection Tests – CDT). Συνηθέστερα δε χρησιμοποιούνται οι μέθοδοι CDT, λόγω της μειωμένης τους υπολογιστικής πολυπλοκότητας, όπως είδαμε και προηγουμένως. Στην περίπτωση που ανιχνευτεί μία μεταβολή, η εφαρμογή προσαρμόζεται – επανεκπαιδεύεται στα νέα δεδομένα.
Ένα παράδειγμα είναι το σύστημα στο σχήμα 3.1.1, όπου μία ομάδα αισθητήρων μετρούν κάποια δεδομένα, τα οποία οδηγούνται σε μία εφαρμογή, καθώς και σε έναν ανιχνευτή μεταβολών. Όταν ανιχνευτεί μία μεταβολή, αυτό υποδηλώνει δύο πράγματα: είτε ότι έχει γίνει κάποια βλάβη στον αισθητήρα, είτε ότι κάτι άλλαξε στα δεδομένα που λαμβάνουμε. Σε κάθε περίπτωση η εφαρμογή θα πρέπει, μόλις λάβει το σήμα απ' τον ανιχνευτή μεταβολών, να προσαρμοστεί στη νέα κατάσταση, πρώτα π.χ. ελέγχοντας και καλιμπράροντας κατάλληλα τους αισθητήρες, είτε, σε περίπτωση που η παρατηρούμενη μεταβολή οφείλεται στα δεδομένα, να κάνει τις κατάλληλες ενέργειες, ανάλογα με την εκάστοτε εφαρμογή (επανεκπαίδευση, ειδοποίηση του χρήστη, κλπ.).

Στην περίπτωσή μας, μας ενδιαφέρει κυρίως η μηχανική μάθηση, οπότε η εφαρμογή θα είναι κάποιο έξυπνο σύστημα ή κάποιος ταξινομητής, τα οποία θα πρέπει να μάθουν τα νέα δεδομένα που εντοπίστηκαν. Επίσης, ανάλογα με την περίπτωση, χρειάζεται και η επανεκπαίδευση του ανιχνευτή. Η προσέγγιση αυτή της μάθησης σε μεταβαλλόμενα περιβάλλοντα ονομάζεται, όπως είδαμε και στο προηγούμενο κεφάλαιο, «ανίχνευσης και δράσης» - "detect and react", και είναι το βασικό παράδειγμα των ενεργών μεθόδων μάθησης σε concept drift.

## 3.2 Ορισμός του προβλήματος

Πριν προχωρήσουμε, είναι σημαντικό να ορίσουμε το πρόβλημα του change detection με ακρίβεια. Ως προς αυτό, ας θεωρήσουμε μία δοσμένη ακολουθία δεδομένων της μορφής

$$X = \{x(t), t = 1,2, \dots, n\}. \tag{3.2.1}$$

Θα λέμε ότι η διαδικασία $X$ έχει ένα σημείο μεταβολής κατά τη χρονική στιγμή – δείγμα $\tau < n$, εάν οι δύο υποακολουθίες

$$A_\tau = \{x(t), t = 1,2, \dots, \tau\}, \tag{3.2.2}$$
$$B_\tau = \{x(t), t = \tau + 1, \dots, \tau\} \tag{3.2.3}$$

είναι δύο ανεξάρτητες και όμοια κατανεμημένες ακολουθίες δειγμάτων δύο διαφορετικών άγνωστων τυχαίων μεταβλητών με κατανομές $F_0$ και $F_1$. Συνεπώς, ο ορισμός του προβλήματος μπορεί να γραφτεί ως εξής:

$$\tau o\ \tau\ \epsilon\acute{\iota}\nu\alpha\iota\ \sigma\eta\mu\epsilon\acute{\iota}o\ \mu\epsilon\tau\alpha\beta o\lambda\acute{\eta}\varsigma\ \Leftrightarrow x(t) \sim \begin{cases} F_0, & \gamma\iota\alpha\ t < \tau \\ F_1, & \gamma\iota\alpha\ t \geq \tau \end{cases} \tag{3.2.4}$$

Με τον τρόπο αυτό, το πρόβλημα της εύρεσης του σημείου μεταβολής ανάγεται σε ένα ισοδύναμο πρόβλημα, στο οποίο πρέπει να εξετάσουμε αν τα σύνολα $A_\tau$ και $B_\tau$ παράγονται από μία ίδια κατανομή («όχι μεταβολή»), ή από δύο διαφορετικές («μεταβολή»).

Η στατιστική μέθοδος για να το διαπιστώσουμε αυτό είναι μέσω του ελέγχου υποθέσεως. Συγκεκριμένα, διατυπώνουμε δύο υποθέσεις, την μηδενική $H_0$ και την εναλλακτική $H_1$, οι οποίες ορίζονται όπως παρακάτω:

$$H_0 : x(t) \sim F_0\ \forall t, \tag{3.2.5}$$

$$H_1 : x(t) \sim \begin{cases} F_0, & \gamma\iota\alpha\ t < \tau \\ F_1, & \gamma\iota\alpha\ t \geq \tau \end{cases}. \tag{3.2.6}$$

Δηλαδή, σύμφωνα με τη μηδενική υπόθεση, η $x(t)$ είναι στάσιμη, ενώ αντίθετα η $H_1$ υποθέτει ότι έχουμε μεταβολή στο σημείο $\tau = t$.
Για να ελέγξουμε τις παραπάνω υποθέσεις, χρειαζόμαστε έναν στατιστικό έλεγχο υποθέσεως $T$, οποίος θα αποδίδει μία τιμή ανομοιότητας μεταξύ των συνόλων $A_\tau$ και $B_\tau$, χρησιμοποιώντας κάποια στατιστικά μέτρα, και έστω

$$T_\tau = T(A_\tau, B_\tau) \tag{3.2.7}$$

η τιμή αυτή. Τότε, σύμφωνα με μία τυπική μέθοδο ελέγχου υποθέσεως, η υπόθεση $H_0$ θα απορρίπτεται εάν η τιμή του $T_\tau$ υπερβαίνει ένα κατώφλι $h_{a,n}$, το οποίο εξαρτάται από έναν βαθμό εμπιστοσύνης $a$, καθώς και από τον αριθμό των δειγμάτων $n$ – φυσικά το $h_{a,n}$ είναι διαφορετικό, αναλόγως με την συγκεκριμένη

μέθοδο ελέγχου υποθέσεως που χρησιμοποιούμε. Στην περίπτωση που απορριφθεί η $H_0$, τότε μπορούμε να υποστηρίκουμε ότι οι ακολουθίες $A_\tau$ και $B_\tau$ υλοποιούνται από διαφορετικές μεταξύ τους κατανομές πιθανότητας, και συνεπώς η διαδικασία $X$ είναι non – stationary και έχει ένα σημείο μεταβολής στο $t = \tau$.

**Παράδειγμα:** Ας θεωρήσουμε ότι τα δεδομένα στα σύνολα $A_\tau$ και $B_\tau$ ακολουθούν γκαουσσιανές κατανομές με ίδια διακύμανση, και θέλουμε να ελέγξουμε αν έχουν ή όχι την ίδια αναμενόμενη τιμή. Για να το κάνουμε αυτό, θα χρησιμοποιήσουμε τη μέθοδο t-test. Το μέτρο συνάφειας στη μέθοδο αυτή είναι το παρακάτω:

$$T_\tau = \sqrt{\frac{\tau(n-\tau)}{n}} \cdot \frac{\overline{A_\tau} - \overline{B_\tau}}{S_\tau}, \tag{3.2.8}$$

όπου τα $\overline{A_\tau}$, $\overline{B_\tau}$ είναι οι μέσοι όροι των συνόλων $A_\tau$, $B_\tau$, και η $S_\tau$ είναι η συνδιασμένη διακύμανση (pooled variance) των δειγμάτων των δύο συνόλων. Το κατώφλι $h_{a,n}$ δίνεται απ' τους τύπους για την κατανομή Student $n-2$ βαθμών ελευθερίας. Αν $T_\tau \leq h_{a,n}$, τότε, στο συγκεκριμένο διάστημα εμπιστοσύνης $(a)$ οι αναμενόμενες τιμές θα είναι ίσες, και η μηδενική υπόθεση γίνεται δεκτή. Διαφορετικά, γίνεται δεκτή η εναλλακτική υπόθεση, και οι αναμενόμενες τιμές είναι διαφορετικές.

Στις μεθόδους CPM, τα δεδομένα είναι εξ αρχής δοσμένα (δεν έχουμε ακολουθιακή λειτουργία). Επομένως, οι αλγόριθμοι αυτοί λειτουργούν ως εξής: δίνοντας τιμές από 2 έως $n-1$ στο $\tau$, εξετάζουν την τιμή $T_\tau = T(A_\tau, B_\tau)$ για όλες τις διαμερίσεις του συνόλου $X$, και βρίσκουν το $\tau$ στο οποίο έχουμε μέγιστο:

$$M = \underset{\tau=2,\dots,n-1}{\operatorname{argmax}} T_\tau. \tag{3.2.9}$$

Στη συνέχεια εξετάζουν αν το $M$ ξεπερνά το κατώφλι $h_{a,n}$. Αν ναι, τότε το $\tau = t(M)$ είναι το σημείο μεταβολής, διαφορετικά η διαδικασία θεωρείται στατική (stationary).

Αντίθετα, οι μέθοδοι CDT λειτουργούν ακολουθιακά, οπότε εδώ ο αλγόριθμος κάνει αρχικά μία εκτίμηση για το αν υπάρχει μεταβολή ή όχι, και στη συνέχεια, ελέγχει με κάποιον έλεγχο υπόθεσης αν πράγματι είναι έτσι ή όχι, και κάνει και μία εκτίμηση για το σημείο $\tau$ της μεταβολής.

## 3.3 Τεστ ανίχνευσης μεταβολής (CDT)

Όπως γνωρίζουμε απ' το προηγούμενο κεφάλαιο, τα τεστ ανίχνευσης μεταβολής (Change Detection Tests – CDT) χρησιμοποιούνται για την ανίχνευση στατιστικών μεταβολών σε μία ακολουθία δεδομένων. Το σημαντικό πλεονέκτημά τους για τις active μεθόδους μάθησης είναι ότι μπορούν να λειτουργούν με online τρόπο, δηλ. ακουλουθικά κάθε φορά που φτάνει στην είσοδο κάποιο νέο δείγμα, σε αντίθεση

με τις μεθόδους CPM, οι οποίες απαιτούν κατά κανόνα ένα σταθερό σύνολο δεδομένων.
Στη βιβλιογραφία υπάρχει μία πληθώρα αναφορών για concept drift detectors, οι οποίοι βασίζονται σε κάποιους στατιστικούς ελέγχους υποθέσεως (statistical hypothesis tests), οι κυριότεροι από τους οποίους παρουσιάζονται συνοπτικά στους πίνακες 3.3.1 και 3.3.2.
Συνήθως, η πληθώρα των μεθόδων αυτών είναι παραμετρικές, δηλ. προϋποθέτουν κάποια προηγούμενη γνώση σχετικά με την συνάρτηση πυκνότητας πιθανότητας της διαδικασίας παραγωγής δεδομένων, καθώς και κάποιες φορές και για τον τύπο και τη δομή του concept drift. Μερικές κλασσικές μέθοδοι αυτής της κατηγορίας είναι το t-test του Student και το f-test του Fisher, τα οποία παρακολουθούν μεταβολές που σχετίζονται με τη μέση τιμή και την διακύμανση των χαρακτηριστικών αντίστοιχα ([107]). Μία σύνοψη των παραμετρικών μεθόδων δίνεται στον Πίνακα 3.1.1, που φαίνεται παρακάτω. Στον πίνακα αυτό παρουσιάζονται οι ονομασίες των μεθόδων, το είδος τους (στατιστικός έλεγχος υπόθεσης, ακολουθιακός έλεγχος υπόθεσης, κλπ.), ο τύπος του drift που αναγνωρίζουν (βαθμιαίο / απότομο), το στατιστικό χαρακτηριστικό που εξετάζουν για να εντοπίσουν τις μεταβολές (μέση τιμή, διακύμανση, κλπ.), καθώς και η διάσταση των δεδομένων εισόδου που υποστηρίζεται (μονοδιάστατα – 1D, πολυδιάστατα – ND).

| Ονομασία | Είδος | Τύπος drift που αναγνωρίζεται | Στατιστικό χαρακτηριστικό που εξετάζεται | Διάσταση δεδομένων |
|---|---|---|---|---|
| Z-test | Έλεγχος στατιστικής υποθέσεως | Απότομο | Μέση τιμή | 1D |
| t-test | Έλεγχος στατιστικής υποθέσεως | Απότομο | Μέση τιμή | 1D |
| F-test | Έλεγχος στατιστικής υποθέσεως | Απότομο | Διακύμανση | 1D |
| Hotelling T-square | Έλεγχος στατιστικής υποθέσεως | Απότομο | Μέση τιμή | ND |
| SPRT | Ακολουθιακός έλεγχος υποθέσεως | Απότομο | Pdf | 1D |
| CUSUM | Ακολουθιακό change-point detection | Απότομο | Pdf | ND |
| Parametric CPM | Ακολουθιακό change-point detection | Απότομο | Εξαρτάται απ' την περίπτωση | 1D/ND |

*Πίνακας 3.3.1: Παραμετρικές στατιστικές μέθοδοι για change detection.*

Εκτός από τα παραπάνω, υπάρχουν και μη παραμετρικές μέθοδοι, οι οποίες χρειάζονται μόνο ασθενείς υποθέσεις, οι οποίες συνήθως ικανοποιούνται από τις εφαρμογές. Για παράδειγμα, το Mann-Whitney-U-test και το Wilcoxon test είναι μη παραμετρικοί έλεγχοι σχεδιασμένοι να ανιχνεύουν έναν μεμονωμένο σημείο μεταβολής, και συνεπώς δεν μπορούν να υποστηρίξουν ακολουθιακή χρήση. Αντίθετα, οι έλεγχοι Mann-Kendall και CUSUM είναι δύο μέθοδοι κατάλληλες για ακολουθιακή χρήση, όπως και ο πρόσφατος κανόνας ICI και τα ιεραρχικά test. Με βάση τις μεθόδους αυτές θα οικοδομήσουμε, στη συνέχεια του κεφαλαίου, τον change detector που θα χρησιμοποιήσουμε. Τέλος, μία σύνοψη των μη παραμετρικών μεθόδων φαίνεται στον παρακάτω πίνακα.

| Ονομασία | Είδος | Τύπος drift που αναγνωρίζεται | Στατιστικό χαρακτηριστικό που εξετάζεται | Διάσταση δεδομένων |
|---|---|---|---|---|
| Mann-Whitney U test | Έλεγχος στατιστικής υποθέσεως | Απότομο | Διάμεσος | 1D |
| Kolmogorov-Smirnov test | Έλεγχος στατιστικής υποθέσεως | Απότομο | Pdf | 1D |
| Mann Whitney Wilcoxon test | Έλεγχος στατιστικής υποθέσεως | Απότομο | Pdf | 1D |
| Kruskal-Wallis test | Έλεγχος στατιστικής υποθέσεως | Απότομο | Διάμεσος | 1D |
| Pearson's chi-squared test | Έλεγχος στατιστικής υποθέσεως | Απότομο | Pdf | 1D |
| Distribution-Free CUSUM | Ακολουθιακό change-point detection | Απότομο | Διάμεσος | 1D |
| Mann Kendall | Ακολουθιακό change-point detection | Απότομο | Διάμεσος | 1D |
| Multi-chart detection algorithm | Ακολουθιακό change-point detection | Απότομο | Διάμεσος | 1D/ND |
| CI-CUSUM | Ακολουθιακό change-point detection | Απότομο, βαθμιαίο | Pdf, ροπές δειγμάτων | 1D/ND |
| ICI change detection test | Ακολουθιακό change-point detection | Απότομο, βαθμιαίο | Μέση τιμή και διακύμανση | 1D |
| Hierarchical change detection test | Ακολουθιακό change-point detection | Απότομο, βαθμιαίο | Μέση τιμή και διακύμανση | 1D |

| | | | | |
|---|---|---|---|---|
| Shiryaev-Robert Extension | Ακολουθιακό change-point detection | Απότομο | Διάμεσος | 1D |
| Mood | Έλεγχος στατιστικής υποθέσεως | Απότομο | Διασπορά | 1D |
| Lepage | Έλεγχος στατιστικής υποθέσεως | Απότομο | Θέση και Διασπορά | 1D |
| Nonparametric CPM | Ακολουθιακό change-point detection | Απότομο | Εξαρτάται απ' τη στατιστική που χρησιμοποιείται | 1D |

*Πίνακας 3.3.2: Μη παραμετρικές στατιστικές μέθοδοι για change detection.*

## 3.4 Το στατιστικό τεστ CUSUM

Στους δύο πίνακες της προηγούμενης ενότητας είδαμε διάφορα στατιστικά τεστ, τόσο παραμετρικά όσο και μη παραμετρικά. Γενικά, στην πράξη δεν μπορούμε να γνωρίζουμε εκ των προτέρων τη μορφή της κατανομής που θα έχουν τα δεδομένα μας. Συνεπώς, για εφαρμογές ανίχνευσης μεταβολών προτιμώνται συνήθως οι μη παραμετρικές εφαρμογές.

Γενικά, τα πιο πολύπλοκα και αποδοτικά μη παραμετρικά στατιστικά τεστ χρειάζονται μία αρχική φάση ρύθμισης παραμέτρων (configuration phase), έτσι ώστε να υπολογίσουν τις παραμέτρους κατά τη φάση εκτέλεσης, χωρίς να χρειάζεται να είναι γνωστές εκ των προτέρων. Το γνωστό στατιστικό τεστ του συσσωρευτικού αθροίσματος (Cumulative SUM – CUSUM, [107]) είναι μία ακολουθιακή τεχνική που έχει σχεδιαστεί για ανίχνευση μεταβολών, η οποία εγγυάται μία αρκετά καλή ακρίβεια στον προσδιορισμό των μεταβολών, με την προϋπόθεση ότι κάποιες πληροφορίες σχετικά με το concept drift και την κατανομή των δεδομένων θα είναι διαθέσιμες εκ των προτέρων. Πάνω σε αυτό θα προσπαθήσουμε να χτίσουμε δύο παραλλαγές της μεθόδου αυτής, επεκτείνοντας το απλό CUSUM test έτσι ώστε να χαλαρώσουμε περισσότερο τους περιορισμούς αυτούς. Η πρώτη προσέγγιση, η οποία ονομάζεται adaptive CUSUM, επεκτείνει το CUSUM έτσι ώστε να επιτρέπει στον αλγόριθμο να καθορίζει αυτόματα τις ζητούμενες παραμέτρους κατά τη φάση αρχικοποίησης. Το adaptive CUSUM εντοπίζει τις μεταβολές στην κατανομή με βάση την ανάλυση της μέσης τιμής και της διακύμανσης κάποιων χαρακτηριστικών των δεδομένων, και την εξέλιξή τους στο χρόνο. Αντίθετα, το δεύτερο τεστ, το οποίο ονομάζεται Computational Intelligence CUSUM (CI – CUSUM), επεκτείνει το πρώτο εξετάζοντας ένα μεγαλύτερο σύνολο στατιστικών χαρακτηριστικών, έτσι ώστε να βελτιώσει την απόδοση στον εντοπισμό των μεταβολών.

## Ο προσαρμοζόμενος CUSUM CDT

Έστω $X = \{x(t), t = 1, \dots, N\}$, $x(t) \in \mathbb{R}$ μία ακολουθία δεδομένων, που προέρχονται από μία διαδικασία παραγωγής δεδομένων (data generating process), η οποία ακολουθεί μία κατανομή πιθανότητας $f_\theta(x)$, η οποία υποθέτουμε ότι είναι άγνωστη, ενώ το διάνυσμα $\theta \in \mathbb{R}^n$ είναι ένα σύνολο παραμέτρων.

Ας υποθέσουμε τώρα ότι η στοχαστική διαδικασία αλλάζει τη στατιστική συμπεριφορά της σε μία άγνωστη χρονική στιγμή $T^0$. Αυτό συνήθως μοντελοποιείται θεωρώντας ότι τη στιγμή $T^0$ έχουμε μία μετάβαση από το σύνολο παραμέτρων $\theta_0$ στο σύνολο $\theta_1$, όπου $f_{\theta_0}(x)$ είναι η παλιά κατανομή, και $f_{\theta_1}(x)$ η νέα. Όπως και στο κλασσικό CUSUM, υπολογίζουμε την ασυμφωνία μεταξύ των δύο σ.π.π. (pdf) κατά τη χρονική στιγμή $t$, υπολογίζοντας τον λογάριθμο του λόγου πιθανοφάνειας

$$s(t) = \ln \frac{f_{\theta_1}\big(x(t)\big)}{f_{\theta_0}\big(x(t)\big)}, t = 1,2, \dots, N, \tag{3.4.1}$$

καθώς και το συσσωρευτικό άθροισμα

$$S(t) = \sum_{\tau=1}^{t} s(\tau). \tag{3.4.2}$$

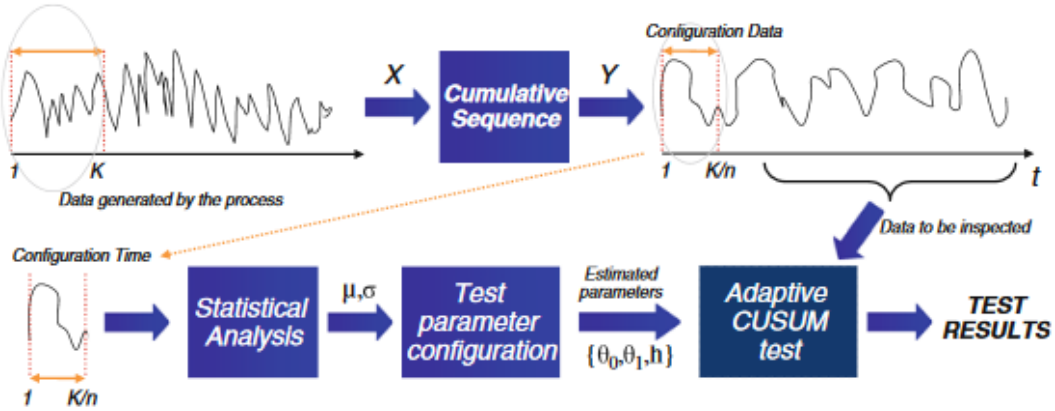
Το CUSUM ανιχνεύει μία μεταβολή στη διαδικασία $X$ κατά τη στιγμή $\hat{T}$, όταν το $g(t) = S(t) - m(t)$, δηλ. η διαφορά μεταξύ της τιμής του συσσωρευτικού αθροίσματος και της τρέχουσας ελάχιστης τιμής του, $m(t) = \min_{\tau=1,\dots,t} S(\tau)$, ξεπεράσει ένα δεδομένο κατώφλι $h$. Δηλαδή, το $\hat{T}$ είναι η ελάχιστη χρονική στιγμή για την οποία ισχύει $g(t) \geq h$.

Από τα παραπάνω φαίνεται ένα σημαντικό πλεονέκτημα του τεστ CUSUM: το CUSUM υποστηρίζει την ακολουθιακή λειτουργία με έναν απλό και φυσικό τρόπο – κάθε φορά που φτάνει ένα νέο δείγμα, ενσωματώνεται άμεσα στα $S(t)$ και $m(t)$, χωρίς να χρειάζεται να τα υπολογίσουμε ξανά απ' την αρχή. Αυτό κάνει το CUSUM ένα ιδανικό τεστ για ανίχνευση μεταβολών, και για το λόγο αυτό θα το χρησιμοποιήσουμε και ως βάση για τα επόμενα.

Όπως βλέπουμε από τα παραπάνω, το κλασσικό CUSUM προϋποθέτει ότι οι παράμετροι $\theta_0$ και $\theta_1$, το $h$, καθώς και η συνάρτηση $f$ θα είναι εκ των προτέρων διαθέσιμες. Η απαίτηση αυτή είναι όμως δύσκολο να ικανοποιηθεί στην πράξη. Παρόλα αυτά, μπορούμε να παρακάμψουμε αυτή τη δυσκολία με τον ακόλουθο τρόπο: Αρχικά θα δημιουργήσουμε την ακολουθία $Y = \{y(1), y(2), \dots\}$, όπου το $y(s)$, $s = 1,2, \dots$ ισούται με την μέση τιμή ενός δείγματος της $X$, το οποίο υπολογίζεται πάνω σε ένα κυλιόμενο παράθυρο μήκους $n$, δηλαδή ισχύει η σχέση:

$$y(s) = \frac{1}{n} \sum_{t=s(n-1)+1}^{sn} x(t). \tag{3.4.3}$$

Και τώρα έρχεται η θεωρία πιθανοτήτων να δώσει τη λύση: σύμφωνα με το κεντρικό οριακό θεώρημα (central limit theorem), η $Y$ μπορεί να προσσεγιστεί με μία Γκαουσσιανή κατανομή, με την προϋπόθεση βέβαια ότι το $n$ είναι αρκετά μεγάλο. Επομένως, το κλασσικό CUSUM μπορεί τώρα να εφαρμοστεί στην ακολουθία $Y$. Τα πρώτα $K$ δείγματα της ακολουθίας $X$ θα χρησιμοποιηθούν για να παράξουμε το σύνολο εκπαίδευσης της ακολουθίας $Y$ (δηλ. το αρχικό σύνολο που θα μας επιτρέψει να καθορίσουμε τις παραμέτρους), το οποίο θα έχε μήκος $K/n$ (το $K$ επιλέγεται ως πολλαπλάσιο του $n$). Οι παράμετροι $\theta_0$ που χαρακτηρίζουν την Γκαουσσιανή κατανομή είναι η μέση τιμή και η διακύμανση, δηλ. είναι $\theta = [\mu, \sigma^2]$, τα οποία εκτιμώνται από το σύνολο εκπαίδευσης της $Y$. Οι παράμετροι $\theta_1$ απ' την άλλη μερία, καθώς και το κατώφλι, υπολογίζονται από ένα δοσμένο διάστημα εμπιστοσύνης, με βάση τους τύπους για το CUSUM. Η όλη διαδικασία απεικονίζεται συνοπτικά στο παρακάτω σχήμα.

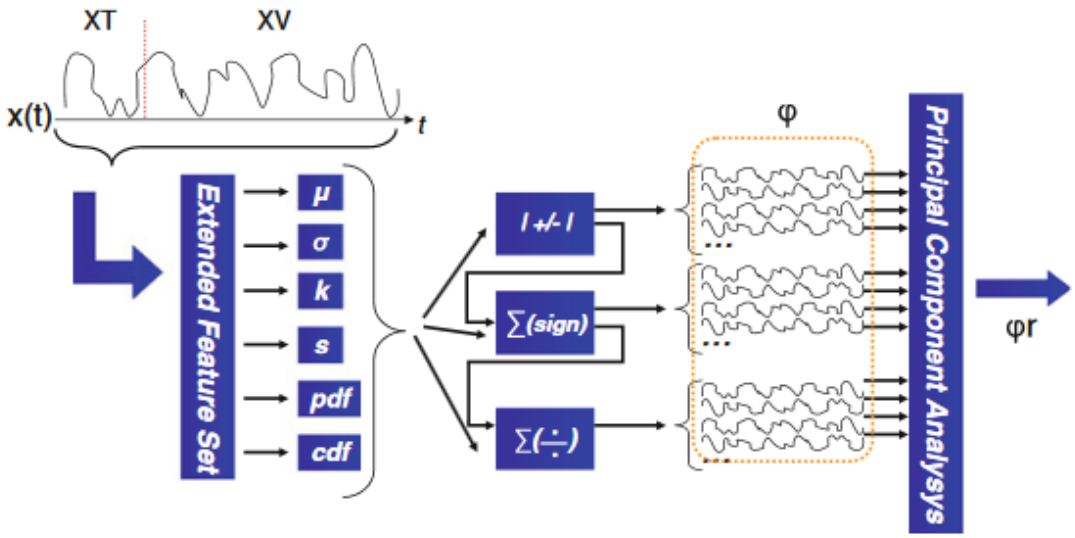



*Σχήμα 3.4.1: Η βασική λειτουργία του adaptive CUSUM. Αρχικά, τα δεδομένα $X$ υφίστανται παραθύρωση καθώς φτάνουν στην είσοδο. Όταν $n$ δεδομένα γίνουν διαθέσιμα, το παράθυρο γεμίζει, οπότε υπολογίζεται η μέση τιμή, και παράγεται ένα δείγμα $y(s)$. Η κατανομή των $y(s)$ είναι κατά προσέγγιση Γκαουσσιανή, λόγω του κεντρικού οριακού θεωρήματος. Συνεπώς, το βασικό CUSUM test μπορεί να εφαρμοστεί, με παραμέτρους $\theta = [\mu, \sigma^2]$. Οι απαιτούμενες παράμετροι $\theta_0$, $\theta_1$, καθώς και το κατώφλι υπολογίζονται απ' το σύνολο εκπαίδευσης.*

Σημειώστε ότι η παραπάνω διαδικασία μπορεί να εφαρμοστεί και στην περίπτωση που τα δεδομένα $x(t)$ είναι πολυδιάστατα: στην περίπτωση αυτή, η απλή μέση τιμή και διασπορά αντικαθιστώνται με τους αντίστοιχους πίνακες μέσης τιμής και συνδιακύμανσης, όπως θα δούμε παρακάτω.

## Ο CI-CUSUM CDT

Το CI – CUSUM είναι μία επέκταση του adaptive CUSUM, η οποία είναι αρκετά ισχυρότερη τόσο απ' το απλό CUSUM, όσο και απ' το adaptive CUSUM, διότι εξάγει και παρακολουθεί αρκετά περισσότερα χαρακτηριστικά των δεδομένων, και αυτό του δίνει μία επιπλέον ευαισθησία στην ανίχνευση μεταβολών. Η μέθοδος φαίνεται συνοπτικά στο παρακάτω σχήμα.




*Σχήμα 3.4.2: Το τμήμα εξαγωγής χαρακτηριστικών της μεθόδου CI – CUSUM. Εδώ, ένα μεγάλο σετ χαρακτηριστικών εξάγεται απ' το σήμα εισόδου, και συντίθεται στο διάνυσμα χαρακτηριστικών φ. Τα χαρακτηριστικά που εξάγονται από την τρέχουσα ακολουθία δεδομένων XV συγκρίνεται με αυτά που αντλήθηκαν από το σύνολο εκπαίδευσης XT. Στη συνέχεια, μία μονάδα PCA παράγει ένα ελαττωμένο διάνυσμα χαρακτηριστικών $\varphi_r$, που ελέγχεται στη συνέχεια για ανίχνευση μεταβολών..*

Όπως βλέπουμε στο παραπάνω σχήμα, ο αλγόριθμος εξάγει ένα σύνολο χαρακτηριστικών $\varphi$, τα οποια επιλέγονται έτσι ώστε να είναι ευαίσθητα στο concept drift. Στη συνέχεια, τα τρέχοντα χαρακτηριστικά συγκρίνονται με τα χαρακτηριστικά του συνόλου εκπαίδευσης, $XT$, έτσι ώστε να εκτιμήσουμε την ανομοιότητα μεταξύ των δύο κατανομών.

Το παραπάνω σύνολο χαρακτηριστικών περιλαμβάνουν, εκτός απ' τη μέση τιμή και τη διακύμανση, και άλλα χαρακτηριστικά, όπως οι δείκτες κύρτωσης (kurt) και σκέβρωσης (skew), τα οποία μετράνε το κατά πόσον η κατανομή είναι συγκεντρωμένη γύρω απ' τη μέση τιμή, καθώς και τον βαθμό ασσυμετρίας αντίστοιχα, καθώς και άλλα χαρακτηριστικά που σχετίζονται με την σ.π.π. και την συνάρτηση κατανομής (cdf) του σήματος. Στη συνέχεια, το τρέχον χαρακτηριστικό συγκρίνεται με αυτό που υπολογίστηκε στο σύνολο εκπαίδευσης, με στόχο την μέτρηση της απόκλισης ανάμεσα στα δύο. Για παράδειγμα, το χαρακτηριστικό $\varphi_1(t) = |\mu_0 - \mu_V|$ στοχεύει στη μέτρηση της απόκλισης των μέσων όρων των δύο κατανομών – το $\mu_0$ είναι η μέση τιμή υπολογισμένη στο σύνολο εκπαίδευσης, ενώ ο δείκτης $V$ αναφέρεται στο σύνολο ελέγχου, δηλαδή στην τρέχουσα ακολουθία των δεδομένων μέχρι τώρα (εξαιρώντας το σύνολο εκπαίδευσης). Μερικά βασικά χαρακτηριστικά, που προτείνονται στο [103], είναι τα παρακάτω:

$$\varphi_1(t) = |\mu_0 - \mu_V|, \varphi_2(t) = |\sigma_0 - \sigma_V|,$$

$$\varphi_3(t) = |kurt_0 - kurt_V|, \varphi_4(t) = |skew_0 - skew_V|,$$

$$\varphi_5(t) = \int_X |pdf_0(x) - pdf_V(x)|dx\,, \varphi_6(t) = \int_X |cdf_0(x) - cdf_V(x)|dx,$$

$$\varphi_{7\le j\le 12}(t) = \sum_{v=1}^{t-1} sgn\left(\varphi_{j-6,v+1} - \varphi_{j-6,v}\right),$$

$$\varphi_{13\le j\le 24}(t) = \sum_{v=1}^{t-1}\left(\frac{\varphi_{j-12,v+1}}{\varphi_{j-12,v}}\right). \quad (3.4.4)$$

Συγκεκριμένα, τα $\varphi_1, \varphi_2$ μετράνει την ανομοιότητα των μέσων τιμών και της διακύμανσης, τα $\varphi_3$, $\varphi_4$ του kurt και skew, ενώ τα $\varphi_5$, $\varphi_6$ μετράνε την απόκλιση της τρέχουσας συνάρτησης pdf και cdf απ' αυτές που εκτιμήθηκαν στο σύνολο εκπαίδευσης. Τέλος, τα χαρακτηριστικά $\varphi_7$ έως $\varphi_{12}$ παρακολουθούν τις μεταβολές του προσήμου σε διαδοχικά χαρακτηριστικά, ενώ τα $\varphi_{13}$ έως $\varphi_{24}$ παρακολουθούν το συσσωρευτικό άθροισμα των λόγων διαδοχικών χαρακτηριστικών.

Στη συνέχεια, για λόγους μείωσης της πολυπλοκότητας του χώρου των χαρακτηριστικών, είναι χρήσιμο να εκτελέσουμε μία ανάλυση κύριων συνιστωσών (PCA) στο διάνυσμα χαρακτηριστικών $\varphi$, η οποία θα μας δώσει ένα μειωμένο διάνυσμα $\varphi_r$. Αφού η σ.π.π. του $\varphi_r$ είναι προφανώς άγνωστη, θα εργαστούμε στη συνέχεια όπως και στην περίπτωση του adaptive CUSUM. Συγκεκριμένα, παίρνουμε τη μέση τιμή των $\varphi_r$ πάνω σε μη επικαλυπτόμενα παράθυρα, και χρησιμοποιούμε το κεντρικό οριακό θεώρημα, οπότε παίρνουμε μία προσεγγιστική πολυδιάστατη Γκαουσσιανή κατανομή για τη νέα μεταβλητή $\varphi'$, η οποία χαρακτηρίζεται από μία μέση τιμή $M$ και έναν πίνακα συνδιακύμανσης $C$. Αρχικά υπολογίζονται η μέση τιμή $M_0$ και ο πίνακας συνδιακύμανσης $C_0$ της $\varphi'$ στο σύνολο εκπαίδευσης, από τα οποία παίρνουμε την παράμετρο αναφοράς $\theta_0 = [M_0, C_0]$. Στη συνέχεια, υπολογίζουμε την παράμετρο $\theta_1 = [M_1, C_1]$ του τρέχοντος συνόλου εκπαίδευσης, και εφαρμόζεται η διαδικασία του adaptive CUSUM για τον έλεγχο υπόθεσης. Συνεπώς, το CI – CUSUM είναι πλέον αρχικοποιημένο, και εξετάζει εάν το σύνολο χαρακτηριστικών $\varphi'$ ανήκει στην πολυδιάστατη κανονική κατανομή $N(M_0, C_0)$ ή όχι, ελέγχοντας την απόκλιση μεταξύ των δύο κατανομών $N(M_0, C_0)$ και $N(M_1, C_1)$ μέσω του συσσωρευτικού αθροίσματος,

$$s(t) = \ln\frac{N_{M_1,C_1}\left(\varphi'(n)\right)}{N_{M_0,C_0}\left(\varphi'(n)\right)}, n = 1,2,\dots,t, \quad (3.4.5)$$

ακριβώς όπως και στην προηγούμενη περίπτωση. Τώρα, το adaptive CUSUM μπορεί να εφαρμοστεί, και είτε να ανιχνεύσει μία μεταβολή, είτε να αποφανθεί ότι δεν υπάρχει κάποια.

## 3.5 Τα τεστ Τομής Διαστημάτων Εμπιστοσύνης

Τα τεστ τομής διαστημάτων εμπιστοσύνης (Intersection of Confidence Intervals – ICI) και οι παραλλαγές του ανιχνεύουν το concept drift που υπεισέρχεται σε μία ροή δεδομένων παρακολουθώντας την εξέλιξη κατάλληλων χαρακτηριστικών που εξάγονται απ' τα δεδομένα εισόδου. Τα χαρακτηριστικά αυτά θα πρέπει να είναι

ανεξάρτητα και όμοια κατανεμημένα (i.i.d.), και να ακολουθούν την Γκαουσσιανή κατανομή, τουλάχιστον μέχρι να συμβεί το drift. Οι προϋποθέσεις αυτές εκ πρώτης όψεως φαίνονται πολύ ισχυρές και μοιάζουν να είναι αρκετά μακριά απ' τις πραγματικές εφαρμογές, ειδικά π.χ. η απαίτηση i.i.d. Παρόλα αυτά αυτό δεν ισχύει σε αρκετές πραγματικές εφαρμογές, αν γίνουν κατάλληλοι μετασχηματισμοί.
Για παράδειγμα, η μέθοδος μπορεί να χρησιμοποιηθεί για την εξέταση διαφορών που σχετίζονται με την ανομοιότητα μεταξύ ενός μοντέλου που περιγράφει τα δεδομένα και των πραγματικών δεδομένων. Όταν το τεστ εντοπίζει μία μεταβολή, τότε έχουμε concept drift. Αυτά θα αναλυθούν εκτενέστερα παρακάτω.

## Ο ICI CDT

Στην περίπτωση του ICI – CDT, εξάγουμε και πάλι χαρακτηριστικά απ' τα δεδομένα με παραθύρωση των δειγμάτων σε ξένες μεταξύ τους υποακολουθίες, που αποτελούνται το καθένα από $n$ δείγματα. Για κάθε υποακολουθία, υπολογίζουμε τη μέση τιμη και τη διακύμανση, τα οποία ακολουθούν την Γκαουσσιανή κατανομή λόγω του κεντρικού οριακού θεωρήματος για την μέση τιμή, και ενός ειδικού μετασχηματισμού για την διακύμανση. Συγκεκριμένα, για την υπακολουθία νο. $s$, τα εξαγώμενα χαρακτηριστικά είναι τα εξής:

$$M(s) = \frac{1}{n} \sum_{t=(s-1)n+1}^{ns} x(t), \tag{3.5.1}$$

$$V(s) = \left( \frac{1}{n-1} \sum_{t=(s-1)n+1}^{ns} \left( x(t) - M(s) \right)^2 \right)^{h_0}. \tag{3.5.2}$$

Η παράμετρος $h_0$ είναι ένας εκθέτης που κάνει τη διακύμανση να μπορεί να προσεγγιστεί από μία κανονική κατανομή, όπως αποδεικνύεται στο [104]. Επιπλέον, το $h_0$ μπορεί να εκτιμηθεί από τα δείγματα του συνόλου εκπαίδευσης $O_{T_0}$ .
Ο ICI – CDT αρχικοποιείται σε δύο ακολουθίες χαρακτηριστικών, τις $\{M(s), s = 1, \dots S_0\}$ και $\{V(s), s = 1, \dots S_0\}$, τα οποία εξάγονται απ' το $O_{T_0}$ , με $S_0 = T_0/n$.
Στη συνέχεια υπολογίζουμε τις μέσες τιμές ${\mu_{S_0}}^M$, ${\mu_{S_0}}^V$, και τις τυπικές αποκλίσεις ${\sigma_{S_0}}^M$, ${\sigma_{S_0}}^V$ των χαρακτηριστικών πάνω στο σύνολο εκπαίδευσης, σύμφωνα με τις σχέσεις:

$$\mu_{S_0}^M = \frac{1}{S_0} \sum_{s=1}^{S_0} M(s) \,, \sigma_{S_0}^M = \sqrt{\frac{1}{S_0 - 1} \sum_{s=1}^{S_0} \left( M(s) - \mu_{S_0}^M \right)^2}, \tag{3.5.3}$$

$$\mu_{S_0}^V = \frac{1}{S_0} \sum_{s=1}^{S_0} V(s) \,, \sigma_{S_0}^V = \sqrt{\frac{1}{S_0 - 1} \sum_{s=1}^{S_0} \left( V(s) - \mu_{S_0}^M \right)^2}. \tag{3.5.4}$$

Οι παραπάνω εκτιμήσεις ορίζουν κάποια διαστήματα εμπιστοσύνης για τη μέση τιμή και την τυπική απόκλιση τα οποία, σε στάσιμες συνθήκες, ορίζονται ως εξής:

$$\mathcal{I}_{S_0}^M = \left[\mu_{S_0}^M - \Gamma\sigma_{S_0}^M, \mu_{S_0}^M + \Gamma\sigma_{S_0}^M\right], \tag{3.5.5}$$

$$\mathcal{I}_{S_0}^V = \left[\mu_{S_0}^V - \Gamma\sigma_{S_0}^V, \mu_{S_0}^V + \Gamma\sigma_{S_0}^V\right], \tag{3.5.6}$$

όπου το $\Gamma$ είναι μία παράμετρος που ελέγχει το μήκος του διαστήματος εμπιστοσύνης και κατ' επέκταση και την πιθανότητα του ότι ένα χαρακτηριστικά ανήκει στο διάστημα. Σε στάσιμες συνθήκες.

Μόλις η φάση της εκπαίδευσης τελειώσει, ο CDT έχει αρχικοποιηθεί, και μπορεί να χρησιμοποιηθεί για να ανιχνεύσει μεταβολές στη ροή των δεδομένων. Κάθε φορά που $n$ δεδομένα γίνονται διαθέσιμα, δημιουργείται μία νέα υπακολουθία $s$, απ' την οποία εξάγωνται τα χαρακτηριστικά $M(s)$ και $V(s)$, και επαναπροσδιορίζονται τα διαστήματα εμπιστοσύνης $\mathcal{I}_{S_0}^M$ και $\mathcal{I}_{S_0}^V$.

Στο σημείο αυτό, ο κανόνας τομής διαστημάτων εμπιστοσύνης ([105]) μπορεί να εφαρμοστεί. Ο κανόνας αυτός ελέγχει αν ένα νέο χαρακτηριστικό αντιπροσωπεύει ένα δείγμα της υπάρχουσας Γκαουσσιανής κατανομής. Αν όχι, τότε θεωρούμε ότι ανιχνεύτηκε concept drift.

Αναλυτικά, αυτό γίνεται ως εξής: κάθε φορά που έχουμε $n$ δείγματα (ένα νέο $M(s)$ και $V(s)$) υπολογίονται τα νέα διαστήματα εμπιστοσύνης, σύμφωνα με τις παραπάνω εξισώσεις. Στη συνέχεια παίρνουμε την τομή όλων των προηγούμενων διαστημάτων εμπιστοσύνης με το τρέχον. Αν αυτό μας δώσει ένα κενό σύνολο, τότε ο ICI – DCT ανιχνεύει μία μεταβολή. Συγκεκριμένα, σύμφωνα με τον ακριβή κανόνα, ανιχνεύουμε μία μεταβολή στην ακολουθία $\hat{s}$ αν ισχύει η συνθήκη

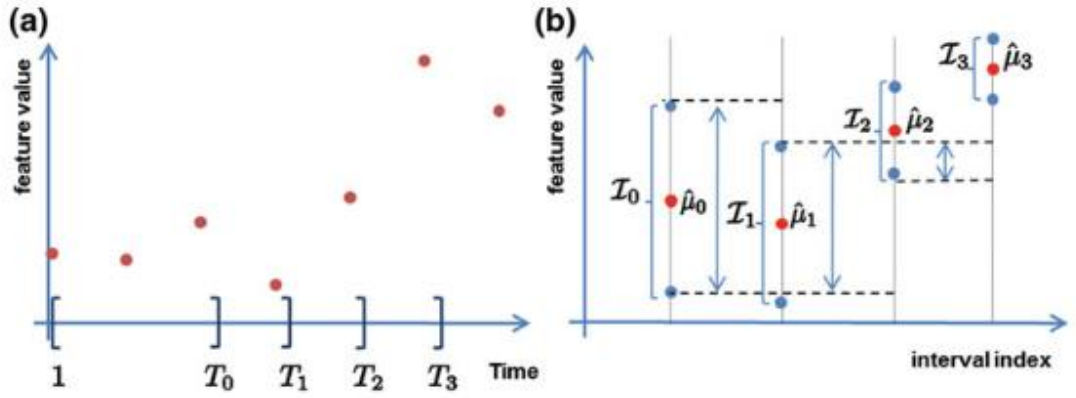
$$\bigcap_{s<\hat{s}} \mathcal{I}_S^M \neq \emptyset \; και \bigcap_{s\leq\hat{s}} \mathcal{I}_S^M \neq \emptyset \tag{3.5.7}$$

ή η συνθήκη

$$\bigcap_{s<\hat{s}} \mathcal{I}_S^V \neq \emptyset \; και \bigcap_{s\leq\hat{s}} \mathcal{I}_S^V \neq \emptyset, \tag{3.5.8}$$

και η χρονική στιγμή της μεταβολής είναι η $\hat{T} = n\hat{s}$, η οποία αντιστοιχεί στο τελευταίο δείγμα της υπακολουθίας $\hat{s}$.

Όπως βλέπουμε λοιπόν, στη μέθοδο αυτή το concept drift συνδέεται με τα χαρακτηριστικά αυτά που δίνουν ένα κενό σύνολο στην τομή των διαστημάτων εμπιστοσύνης. Επιπλέον, για να μειώσουμε την υπολογιστική πολυπλοκότητα, οι μέσες τιμές και οι τομές των διαστημάτων υπολογίζονται αυξητικά (incrementally). Όλα αυτά φαίνονται εποπτικά στο παρακάτω σχήμα:




*Σχήμα 3.5.1: Ένα παράδειγμα του κανόνα ICI, όπως χρησιμοποιείται για ανίχνευση μεταβολών.*

Όλη η διαδικασία συνοψίζεται στον αλγόριθμο 3.5.1, που φαίνεται παρακάτω. Επίσης, όπως σημειώνεται στην [45], ο αλγόριθμος αυτός είναι αρκετά αποδοτικός, αλλά έχει την τάση να παράγει λανθασμένες θετικές ενδείξεις όσο περνάει ο χρόνος.

Αλγόριθμος 3.5.1: Η βασική διαδικασία ICI - CDT

Υπολόγισε τα $\{M(s), s = 1, \dots, S_0\}$, όπου $S_0 = T_0/n$;
$\mu_{S_0}^M = \frac{1}{S_0}\sum_{s=1}^{S_0} M(s)$;
$\sigma^M = \sqrt{\frac{1}{S_0-1}\sum_{s=1}^{S_0}\left(M(s) - \mu_{S_0}^M\right)^2}$, $\sigma_{S_0}^M = \sigma^M/\sqrt{S_0}$;
Όρισε το $\mathcal{I}_{S_0}^M = \left[\mu_{S_0}^M - \Gamma\sigma_{S_0}^M, \mu_{S_0}^M + \Gamma\sigma_{S_0}^M\right]$;
Υπολόγισε το $h_0$;
Υπολόγισε τα $\{V(s), s = 1, \dots, S_0\}$;
$\mu_{S_0}^V = \frac{1}{S_0}\sum_{s=1}^{S_0} V(s)$;
$\sigma^V = \sqrt{\frac{1}{S_0-1}\sum_{s=1}^{S_0}\left(V(s) - \mu_{S_0}^M\right)^2}$, $\sigma_{S_0}^V = \sigma^V/\sqrt{S_0}$;
Όρισε το $\mathcal{I}_{S_0}^V = \left[\mu_{S_0}^V - \Gamma\sigma_{S_0}^V, \mu_{S_0}^V + \Gamma\sigma_{S_0}^V\right]$;
Θέσε $s = S_0$;
**while** ($\mathcal{I}_S^M \neq \emptyset$ και $\mathcal{I}_S^V \neq \emptyset$)
  θέσε $s = s + 1$;
  Περίμενε για $n$ δείγματα, μέχρι να γεμίσει το παράθυρο;
  Υπολόγισε τα $M(s)$ και $V(s)$ απ' τα δεδομένα του παραθύρου;
  Υπολόγισε τα $\mu_s^M = \frac{(s-1)\mu_{s-1}^M + M(s)}{s}$ και $\sigma_s^M = \sigma^M/\sqrt{s}$;
  Υπολόγισε τα $\mu_s^V = \frac{(s-1)\mu_{s-1}^V + V(s)}{s}$ και $\sigma_s^V = \sigma^V/\sqrt{s}$;
  $\mathcal{I}_s^M = \left[\mu_s^M - \Gamma\sigma_{S_0}^M, \mu_{S_0}^M + \Gamma\sigma_{S_0}^M\right] \cap \mathcal{I}_{s-1}^M$;
  $\mathcal{I}_s^V = \left[\mu_s^V - \Gamma\sigma_{S_0}^V, \mu_{S_0}^V + \Gamma\sigma_{S_0}^V\right] \cap \mathcal{I}_{s-1}^V$;
**end**
Ανίχνευση concept drift στο $s = \hat{s}$ και χρονικό διάστημα $[(\hat{s}-1)n, \hat{s}n]$;
Χρονική στιγμή drift στο $\hat{T} = n\hat{s}$;

*Αλγόριθμος 3.5.1: Βασικός κανόνας ICI – CDT.*

Παρόλο που το πρόβλημα των λανθασμένων θετικών ενδείξεων μπορεί να γίνει ανεκτό σε αρκετές “detect and react” προσεγγίσεις, είναι σημαντικό το να σχεδιάσουμε ένα τεστ το οποίο δεν θα παράγει λανθασμένα θετικά όσο περνάει ο χρόνος. Το πρόβλημα αυτό μπορεί να λυθεί θεωρώντας ένα δεύτερο τεστ, πάνω από το βασικό ICI – CDT, το οποίο θα επαληθεύει εάν η ένδειξη του ICI – CDT είναι όντως concept drift, ή μία λανθασμένη θετική ένδειξη. Λόγω της ιεραρχικής αυτή δομής, το τεστ αυτό ονομάζεται ιεραρχικό CDT, και αναλύεται στην επόμενη ενότητα.
Τέλος, από τα παραπάνω είναι φανερό ότι η μέθοδος ICI – CDT είναι μονοδιάστατη, και δεν έχει κάποια προφανή επέκταση σε πολυδιάστατα δεδομένα.

## Ο ιεραρχικός CDT

Ο ιεραρχικός CDT (Hierarchical CDT) είναι μία επέκταση του βασικού ICI – CDT, έτσι ώστε να αποφεύγονται οι λανθασμένες θετικές ανιχνεύσεις του τελευταίου με την πάροδο του χρόνου.
Ο H – CDT είναι ένα ιεραρχικό ακολουθιακό τεστ change detection, το οποίο αποτελείται από δύο επίπεδα. Το πρώτο επίπεδο αποτελείται από το βασικό ICI – CDT τεστ, ενώ το δεύτερο επίπεδο περιλαμβάνει ένα στατιστικό τεστ το οποίο είτε επιβεβαιώνει είτε απορρίπτει την μεταβολή που ανιχνεύτηκε. Συγκεκριμένα, ο ICI – CDT λειτουργεί σειριακά, όπως είδαμε πριν, και μόλις ανιχνεύσει μία μεταβολή στην ακολουθία $x(t)$ τη χρονική στιγμή $\hat{T}$, ενεργοποιεί το δεύτερο επίπεδο έτσι ώστε να επικυρώσει ή όχι τη μεταβολή, ελέγχοντας αν τα σύνολα των δεδομένων πριν και μετά τη μεταβολή είναι συνεπή με την υπόθεση της μεταβολής (έλεγχος υποθέσεως).
Για το σκοπό της επικύρωσης της μεταβολής, χρειάζεται να συγκεντρώσουμε ένα επιπλέον σύνολο $O_{\hat{T}} = \{x(t), t = \hat{T}, \dots, \hat{T} + N\}$ από $N$ ακόμη δείγματα που παράγονται μετά τη χρονική στιγμή $\hat{T}$, τα οποία θεωρούμε ότι παράχθηκαν με βάση τη νέα κατανομή, δηλαδή μετά τη μεταβολή. Φυσικά, υπάρχει πάντα και η πιθανότητα η μεταβολή που ανιχνεύτηκε να είναι λανθασμένη, οπότε τα δεδομένα του συνόλου $O_{\hat{T}}$ θα είναι στην περίπτωση αυτή συμβατά με τα δεδομένα του συνόλου εκπαίδευσης $O_{T_0}$, στ οποίο αρχικοποιήθηκε ο αλγόριθμος (δηλ. τα δύο σύνολα δεδομένων θα έχουν την ίδια σ.π.π.).
Εδώ πρέπει να σημειώσουμε ότι αν η εκτίμηση του $\hat{T}$ είναι αρκετά ακριβής, θα μπορούσαμε στη θέση του $O_{T_0}$ να πάρουμε ολόκληρο το σύνολο $\{x(t), t < \hat{T}\}$ των προηγούμενων δεδομένων. Ο λόγος γι’ αυτό είναι ότι, στην περίπτωση που η εκτίμηση του χρόνου $\hat{T}$ είναι αρκετά καλή, τότε τα δεδομένα πριν το concept drift είναι κατά πλειοψηφία άχρηστα και μπορούν να απορριφθούν. Επίσης, η ακρίβεια που παίρνουμε είναι μεγαλύτερη, αφού συγκρίνουμε άμεσα τα σύνολα πριν και μετά τη μεταβολή, ενώ στην δεύτερη περίπτωση η σύγκριση γίνεται με βάση ένα «μακρινό» σύνολο εκπαίδευσης. Το μειονέκτημα βέβαια, απ’ την άλλη πλευρά, είναι η αύξηση της υπολογιστικής πολυπλοκότητας, αφού θα πρέπει να θεωρήσουμε ως σύνολο εκπαίδευσης το προηγούμενο σύνολο και να βρούμε ξανά τις παραμέτρους.
Η στατιστική συνάφεια μεταξύ των συνόλων $O_{T_0}$ και $O_{\hat{T}}$ μπορεί να εκτιμηθεί εύκολα με κάποιο στατιστικό τεστ, όπως π.χ. το τεστ Kolmogorov – Smirnov, ή με κάποιο

άλλο τεστ απ' αυτά που παρουσιάστηκαν στους προηγούμενους πίνακες. Παρόλα αυτά, αυτά τα τεστ έχουν συνήθως μία σημαντική υπολογιστική πολυπλοκότητα, η οποία μειώνει την ταχύτητα, η οποία είναι συνήθως ένας σημαντικός παράγοντας στην επεξεργασία ροών δεδομένων. Παρόλα αυτά, όπως θα δούμε παρακάτω, το πρόβλημα της σύγκρισης των κατανομών των συνόλων $O_{T_0}$ και $O_{\hat{T}}$ μπορεί να απλοποιηθεί σημαντικά, και να αναχθεί σε ένα πρόβλημα σύγκρισης των αναμενόμενων τιμών των χαρακτηριστικών $M(s)$ και $V(s)$ των δύο συνόλων, μέσω ενός τεστ Hotelling.

Συγκεκριμένα, το Hotelling test είναι ένα πολυδιάστατο τεστ ελέγχου υποθέσεως, το οποίο εφαρμόζεται με στόχο τη σύγκριση των τιμών των παραπάνω χαρακτηριστικών, τα οποία τοποθετούνται σε δισδιάστατα διανύσματα $F = [M(s), V(s)]$. Αυτά τα διανύσματα χαρακτηριστικών εξάγωνται απ' το σύνολο εκπαίδευσης $O_{T_0}$, καθώς και το σύνολο $O_{\hat{T}}$. Για το καθένα απ' τα δύο σύνολα διανυσμάτων υπολογίζονται οι μέσες τιμές, $\bar{F}(O_{T_0})$ και $\bar{F}(O_{\hat{T}})$, καθώς και ο από κοινού πίνακας συνδιακύμανσης $S_{T_0,\hat{T}}$. Στη συνέχεια, δομείται η μηδενική υπόθεση του τεστ, $H_0$, η οποία είναι η παρακάτω:

$$H_0 : \bar{F}(O_{T_0}) - \bar{F}(O_{\hat{T}}) = \underline{0}, \tag{3.5.9}$$

όπου το $\underline{0}$ είναι ένα διδιάστατο διάνυσμα που περιέχει μηδενικά στοιχεία. Κατόπιν αυτού, το Hotelling $T^2$ test μπορεί να εφαρμοστεί, ώστε να απορρίψει ή όχι τη μηδενική υπόθεση, με βάση ένα δοσμένο βαθμό εμπιστοσύνης$a$. Εάν, η υπόθεση απορριθφεί, τότε το τεστ ανιχνεύει μία μεταβολή, οπότε επιβεβαιώνει την ένδειξη που έδωσε το ICI – CDT. Αντίθετα, αν η μηδενική υπόθεση γίνει δεκτή, αυτό σημαίνει ότι η ένδειξη που μας έδωσε η μονάδα ICI – CDT είναι λανθασμένη, οπότε απορρίπτουμε τη μεταβολή και επανεκπαιδεύουμε τον αλγόριθμο πάνω στο σύνολο $O_{T_0}$. OI δοκιμές δείχνουν ότι η προσέγγιση αυτή είναι αρκετά αποτελεσματική στην ανίχνευση των λανθασμένων θετικών σημάτων.

Επομένως, το H – CDT είναι ένα adaptive test, το οποίο αντιδρά στις λανθασμένες ενδείξεις του ICI – CDT. Επιπλέον, η μειωμένη υπολογιστική του πολυπλοκότητα το καθιστά ιδανικό για εφαρμογές ταχύτητας, ή σε περιπτώσεις όπου έχουμε χαμηλή υπολογιστική ισχύ, όπως π.χ. σε ενσωματωμένα συστήματα. Το τεστ αυτό περιγράφεται συνοπτικά στον παρακάτω αλγόριθμο.

Αλγόριθμος 3.5.2: Το τεστ H - CDT

```
1    Εκπαίδευσε τον ICI – CDT στο O_{T_0};
2    while(1) do
3      Υπολόγισε τα χαρακτηριστικά M(s) και V(s) από τη ροή δεδομένων;
4      if (ο ICI – CDT ανιχνεύσει μία μεταβολή στα χαρακτηριστικά) AND ...
5      (το Hotteling τεστ επιβεβαιώσει τη μεταβολή)
6      then
7        ConceptDrift = true;
8        Επανεκπαίδευσή τον ICI – CDT στο σύνολο O_{T_0} = O_{T̂};
9      else
10       λανθασμένο θετικό σήμα: επανεκπαίδευσε τον ICI – CDT στο O_{T_0};
11     end
```

12 **end**

*Αλγόριθμος 3.5.2:Το ιεραρχικό change detection test H – CDT. Αρχικά, το τεστ εκπαιδεύεται στο σύνολο $O_{T_0}$. Στη συνέχεια, μόλις ο ICI – CDT ανιχνεύσει κάποια μεταβολή, το Hotelling test ενεργοποιείται. Εάν το Hotteling test επιβεβαιώσει την υπόθεση του concept drift, τότε ο H – CDT ανιχνεύει μία μεταβολή. Αντίθετα, αν η υπόθεση απορριφθεί, δεν έχουμε concept drift, και ο ICI – CDT επανεκπαιδεύεται πάνω στο αρχικό σύνολο εκπαίδευσης.*

Ο παραπάνω αλγόριθμος μπορεί να λειτουργήσει και ακολουθιακά, ανιχνεύοντας concept drift το ένα μετά το άλλο, κάθε φορά που εμφανίζονται. Για να γίνει αυτό, το μόνο που χρειάζεται είναι, εκτός απ' την επανεκπαίδευση του ICI – CDT πάνω στο νέο σύνολο δεδομένων $O_{T_0} = O_{\hat{T}}$, θα πρέπει να θέσουμε το σύνολο αυτό ως νέο σύνολο αναφοράς στο Hotteling τεστ. Πράγματι, αν ανιχνευτεί κάποιο drift, το νέο σύνολο δεδομένων $O_{\hat{T}}$ περιέχει δείγματα που συνδέονται με την νέα κατάσταση της διαδικασίας παραγωγής δεδομένων. Οπότε, με αναφορά το νέο αυτό σύνολο, μπορούμε να συνεχίσουμε να ανιχνεύουμε concept drifts, και έτσι μπορούμε να χρησιμοποιήσουμε τη μέθοδο αυτή σε έναν ενεργό αλγόριθμο μάθησης. Η διαδικασία αυτή περιγράφεται συνοπτικά παρακάτω:

Αλγόριθμος 3.5.3: Το τεστ H – CDT στο πλαίσιο ενός active αλγορίθμου μάθησης

```
1   Εκπαίδευσε τον ICI – CDT στο O_{T_0};
2   while(1) do
3     Υπολόγισε τα χαρακτηριστικά M(s) και V(s) από τη ροή δεδομένων;
4     if (ο ICI – CDT ανιχνεύσει μία μεταβολή στα χαρακτηριστικά) AND ...
5     (το Hotteling τεστ επιβεβαιώσει τη μεταβολή)
6     then
7       ConceptDrift = true;
8       Διαχείριση του drift απ' την εφαρμογή (ανανέωση ταξινομητή, κλπ.);
9       Επανεκπαίδευσε τον ICI – CDT στο σύνολο O_{T_0} = O_{T̂};
10      Όρισε το O_{T_0} = O_{T̂} ως νέο σύνολο αναφοράς για το Hotelling test;
11    else
12      λανθασμένο θετικό σήμα: επανεκπαίδευσε τον ICI – CDT στο O_{T_0};
13    end
14  end
```

*Αλγόριθμος 3.5.3: Ο H – CDT στα πλαίσια ενός ενεργού αλγορίθμου μάθησης. Όταν ανιχνευτεί ένα concept drift, η εφαρμογή ενημερώνεται και ο H –CDT επανεκπαιδεύεται πάνω στο νέο σύνολο.*

## Μία βελτιωμένη εκτίμηση του χρόνου της μεταβολής

Όπως είδαμε προηγουμένως, ο H – CDT είναι ένας αρκετά αποτελεσματικός αλγόριθμος για τον ορθό εντοπισμό του concept drift. Παρόλα αυτά έχει ένα σημαντικό μειονέκτημα, που είναι ότι χρειάζεται να περιμένει επιπλέον για $N$ δείγματα μετά τη στιγμή $\hat{T}$ για να προχωρήσει στη φάση της επιβεβαίωσης. Αυτό δεν είναι ελκυστικό σε ένα σενάριο online μάθησης. Επιπλέον, ο χρόνος $\hat{T}$ είναι απλώς ο χρόνος που ο αλγόριθμός μας ανίχνευσε το concept drift, και όχι ο πραγματικός χρόνος που αυτό συνέβη (αυτό είναι απόλυτα λογικά, αφού χρειάζεται

ένας αριθμός δειγμάτων για να αποφανθούμε ότι οι στατιστικές παράμετροι έχουν αλλάξει!). Στην ενότητα αυτή, αυτό που θέλουμε να κάνουμε είναι να βρούμε μία εκτίμηση για τον πραγματικό χρόνο $T°$ στον οποίο συνέβη το drift, έτσι ώστε να μπορέσουμε να εκμεταλευτούμε και τα δείγματα μεταξύ των χρονικών στιγμών $T°$ και $\hat{T}$, ούτως ώστε, απ' τη μία μεριά να επιταχύνουμε τη διαδικασία της επιβεβαίωσης του drift χωρίς να χρειάζεται να περιμένουμε για όλα να $N$ επιπλέον δείγματα, αφετέρου δε να μπορούμε να χρησιμοποιήσουμε τα επιπλέον αυτά δείγματα για την επανεκπαίδευση – ενημέρωση του ταξινομητή, ώστε να κερδίσουμε σε ακρίβεια.
Επομένως, μία βελτιωμένη εκτίμηση του πραγματικού χρόνου του drift, $\bar{t}$, η οποία θα ικανοποιεί προφανώς την ανισότητα $T° \leq \bar{t} \leq \hat{T}$, μπορεί να χρησιμοποιηθεί έτσι ώστε να ορίσουμε το σύνολο $O_{\hat{T}}$ ως εξής,

$$O_{\hat{T}} = \{x(t), t = \bar{t}, \dots, \hat{T}\}, \tag{3.5.10}$$

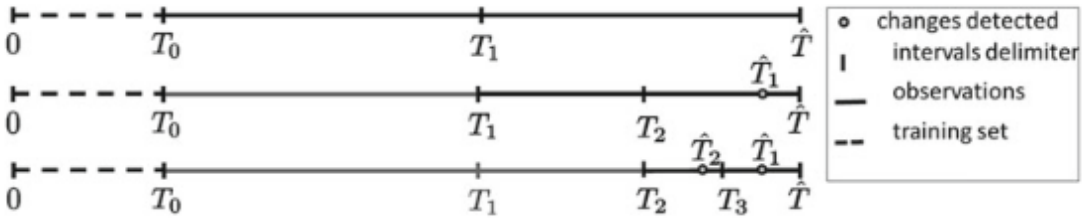
και να κάνουμε την επιβεβαίωση ή όχι του concept drift πάνω σε αυτό το σύνολο, χωρίς να χρειάζεται να περιμένουμε επιπλέον $N$ δείγματα. Φυσικά, αν τα δείγματα του $O_{\hat{T}}$ είναι πολύ λίγα, αναγκαστικά θα περιμένουμε λίγο ακόμη μέχρι ένας ικανοποιητικός αριθμός $N$ δειγμάτων να συγκεντρωθεί – συνήθως όμως ο χρόνος αναμονής είναι είτε μηδενικός είτε πολύ μικρότερος απ' ότι προηγουμένως.
Η κύρια ιδέα της εκτίμησης αυτής του $\bar{t}$ είναι ότι αποδεικνύεται ότι ο αλγόριθμος ICI – CDT έχει την τάση να παρουσιάζει μία δομική καθυστέρηση του χρόνου ανίχνευσης καθώς περνάει ο χρόνος. ([45]). Τη δομική αυτή αδυναμία μπορούμε να την εκμεταλλευτούμε για την κατασκευή μίας αναδρομικής διαδικασίας η οποία, αρχίζοντας απ' το $\hat{T}$, θα μπορεί να παρέχει μία καλύτερη εκτίμηση $\bar{t}$. Η διαδικασία αυτή περιγράφεται στον παρακάτω αλγόριθμο:

Αλγόριθμος 3.5.4: Η αναδρομική διαδικασία εκλέπτυνσης του χρόνου $\hat{T}$

Δώσε σαν είσοδο το $\hat{T}$;
Υπολόγισε το $T_1 = T_0 + (\hat{T} - T_0)/\lambda$;
i=1;
continue = true;
**while** (continue = true) **do**
  Εφάρμοσε το ICI – CDT στο dataset $[0, T_0] \cup [T_i, \hat{T}]$, με έξοδο το $\hat{T}_i$;
  Υπολόγισε το $T_{i+1} = T_i + (\hat{T} - T_i)/\lambda$;
  Όρισε το $T_{min} = \min(\hat{T}_j), j = 1, \dots, i$;
  **if** $(T_{min} < T_{i+1})$ **then**
    continue = false;
  **end**
  $i = i + 1$;
**end**
Όρισε το $\bar{t} = T_{min}$;

*Αλγόριθμος 3.5.4: Η διαδικασία εκτίμησης του χρόνου $\bar{t}$.*

Η λειτουργία του παραπάνω αλγορίθμου έχει ως εξής: Αρχικά, δεδομένης της χρονικής στιγμής $\hat{T}$ όπου ο ICI – CDT ανίχνευσε ένα drift, χωρίζουμε το διάστημα $[T_0, \hat{T}]$ σε δύο υποδιαστήματα $[T_0, T_1]$ και $[T_1, \hat{T}]$, όπου το $T_1 = T_0 + (\hat{T} - T_0)/\lambda$ υπολογίζεται με βάση μία παράμετρο $\lambda > 1$ που ορίζεται από το χρήστη (γραμμή 2 στον κώδικα). Στη συνέχεια, εφαρμόζουμε τον κανόνα ICI – CDT στο dataset $[0, T_0] \cup [T_1, \hat{T}]$, το οποίο μας δίνει έναν νέο χρόνο ανίχνευσης $\hat{T}_1$. Το $\hat{T}_1$ είναι μία καλύτερη εκτίμηση του χρόνου του drift, αφού το τεστ εφαρμόζεται πάνω σε μία μικρότερη ακολουθία απ' αυτή που μας έδωσε ως έξοδο το $\hat{T}$. Στη συνέχεια, το διάστημα $[T_1, \hat{T}]$ χωρίζεται με τη σειρά του σε δύο υποδιαστήματα $[T_1, T_2]$ και $[T_2, \hat{T}]$, με $T_2 = T_1 + (\hat{T} - T_1)/\lambda$. Εάν $T_2 > \hat{T}_1$, η διαδικασία σταματά, και δίνει ως έξοδο το $\bar{t} = \hat{T}_1$. Διαφορετικά, η διαδικασία επαναλαμβάνεται επαναληπτικά: στην $i$-οστή επανάληψη, ο ICI – CDT εφαρμόζεται στο διάστημα $[0, T_0] \cup [T_i, \hat{T}]$, παρέχοντας μία εκτίμηση $\hat{T}_i$ (γραμμή 6), και στη συνέχεια το διάστημα $[T_i, \hat{T}]$ χωρίζεται στο σημείο $T_{i+1} = T_i + (\hat{T} - T_i)/\lambda$ (γραμμή 7). Η διαδικασία σταματάει όταν το $T_{i+1}$ γίνει μεγαλύτερο απ' το $T_{min}$, που είναι ο ελάχιστος χρόνος μεταβολής που έχει ανιχνευτεί μέχρι τη στιγμή αυτή (γραμμή 8). Τέλος, επιτρέφεται το $T_{min}$, το οποίο είναι η καλύτερη εκτίμηση του $T°$ που έχει κάνει η διαδικασία, και τελικά έχουμε $\bar{t} = T_{min}$. Η όλη μέθοδος οπτικοποιείται στο παρακάτω σχήμα:




Σχήμα 3.5.2: Οπτική απεικόνιση του αλγορίθμου 3.5.4.

Η εκτίμηση $\bar{t}$ που παράγει ο αλγόριθμος ICI – CDT τον κάνει ιδιαίτερα χρήσιμο σε πλαίσια active – detect and react μάθησης, αφού μας παρέχει ένα σύνολο $O_{\hat{T}}$ το οποίο περιέχει δείγματα που σχετίζονται με την νέα κατάσταση της κατανομής μετά το drift. Τα δεδομένα αυτά μπορούν τώρα να χρησιμοποιηθούν και για την ενημέρωση – επανεκπαίδευση και του ταξινομητή – εφαρμογής, εκτός απ' την επανεκπαίδευση του ICI – CDT. Επιπλέον, ο Η – CDT αποδεικνύεται ότι απαιτεί λιγότερη υπολογιστική πολυπλοκότητα απ' ότι το CI – CUSUM ([106]), και άρα είναι ταχύτερος και πιο κατάλληλος σε περιπτώσεις μειωμένης επεξεργαστικής ισχύος, όπως π.χ. σε ενσωματωμένα συστήματα. Επιπλέον, ένα άλλο πλεονέκτημα του αλγορίθμου αυτού είναι η αποφυγή των λανθασμένων θετικών ενδείξεων, αφού στις περιπτώσεις αυτές πρώτον επανεκπαιδεύουμε τον ταξινομητή χωρίς λόγο (υπολογισμοί που θα μπορούσαμε να αποφύγουμε), ενώ υπάρχει η περίπτωση να απορρίψουμε ένα μεγάλο σύνολο εκπαίδευσης $O_{T_0}$ για ένα όμοια κατανεμημένο αλλά αρκετά μικρότερο σύνολο $O_{\hat{T}}$, χάνοντας πληροφορία. Επομένως, η ικανότητα του αλγορίθμου αυτού να αναγνωρίζει τις λάθος θετικές ενδείξεις είναι σημαντική.
Επίσης, ένα άλλο θέμα με το οποίο δεν ασχοληθήκαμε μέχρι τώρα είναι η περίπτωση που έχουμε βαθμιαίες μεταβολές. Στην περίπτωση αυτή αναμένουμε ότι το concept drift δε θα ανιχνευτεί αμέσως, αλλά κατά πάσα πιθανότητα αργότερα, όταν η επίδραση που θα έχει προκαλέσει στα δεδομένα θα έχει γίνει αρκετά μεγάλη ώστε να ανιχνεύεται απ' τις στατιστικές μεθόδους. Παρόλα αυτά, έχουμε να

πληρώσουμε το κόστος της καθυστέρησης στην αναγνώριση. Επίσης, δεδομένων των CDT που έχουμε θεωρήσει, υπάρχει η περίπτωση όπου ένα αργό concept drift προκαλεί μία σειρά από ανιχνεύσεις, κάτι που είναι γενικά ανεπιθύμητο.
Για το λόγο αυτό, αν και για το βαθμιαίο concept drift οι παθητικοί αλγόριθμοι είναι γενικά πιο αποτελεσματικοί, έχουν προταθεί μέθοδοι στη βιβλιογραφία για την ενεργό αντιμετώπιση του βαθμιαίου concept drift, κάνοντας κάποιες παραδοχές για τον τύπο της μεταβολής, με τη συνηθέστερη να είναι ότι αυτή μπορεί να προσεγγιστεί με πολυωνυμικές συναρτήσεις κάποιου σταθερού βαθμού ([48]). Περισσότερα για το θέμα αυτό θα αναφέρουμε σε επόμενη παράγραφο.

## 3.6 Το Just – in – Time πλαίσιο μάθησης

Η δυνατότητα της χρήσης ενός change detector σε ένα ακολουθιακό πλαίσιο, όπου νέα δεδομένα έρχονται σε κάθε χρονική στιγμή, μας επιτρέπει να σχεδιάσουμε, ως γνωστόν, εφαρμογές ενεργούς μάθησης. Η λειτουργία έχει ως εξής: ο CDT ελέγχει τη ροή των δεδομένων και αναγνωρίζει τα concept drift, και στη συνέχεια, ο αλγόριθμος προσαρμόζει τον ταξινομητή στα νέα δεδομένα. Ο τύπος αυτός της μάθησης ονομάζεται Just in Time (JIT), το οποίο δηλώνει ότι η αναγνώριση του concept drift και η προσαρμογή του ταξινομητή και της εφαρμογής γίνονται τη στιγμή που ανιχνεύεται η μεταβολή (just in time), σε αντίθεση με τις παθητικές μεθόδους, όπου αυτό γίνεται συνέχεια. Επιπλέον, ένα πλεονέκτημα της just in time προσέγγισης είναι ότι, σε περίπτωση που δεν έχουμε concept drift, τα επιβλεπόμενα δεδομένα που είναι διαθέσιμα προστίθενται στη γνώση του ταξινομητή.
Μία περιγραφή υψηλού επιπέδου του JIT πλαισίου, η οποία παρουσιάστηκε περιληπτικά και στο προηγούμενο κεφάλαιο, φαίνεται παρακάτω. Σημειώστε εδώ ότι το πλαίσιο αυτό είναι πολύ γενικό και μπορεί να λειτουργήσει με οποιονδήποτε CDT ή ταξινομητή. Η επιλογή των συγκεκριμένων CDT και ταξινομητών θα πρέπει να γίνει σε συνάρτηση απ' τη μία μεριά της ακρίβειας που επιθυμούμε, και απ' την άλλη μεριά της υπολογιστικής πολυπλοκότητας.

Αλγόριθμος 3.6.1: ο JIT adaptive ταξινομητής

```
1    Αρχικοποίησε – εκπαίδευσε τον ταξινομητή και τον CDT;
2    while (true) do
3      Πάρε ένα νέο δείγμα απ' την είσοδο;
4      if (ο CDT ανιχνεύει concept drift) then
5        Βρες το νέο σύνολο δεδομένων;
6        Προσάρμοσε τον JIT ταξινομητή και τον CDT στα νέα δεδομένα;
7      else
8        Ενσωμάτωσε τυχόν επιβλεπόμενα δεδομένα στον JIT;
9      end
10     Ταξινόμησε το τρέχον δείγμα;
11   end
```

*Αλγόριθμος 3.6.1: Ο JIT προσαρμοζόμενος ταξινομητής. Εδώ, τα νέα δεδομένα ενσωματώνονται στον ταξινομητή όταν φτάνουν στην είσοδο, μέχρι να ανιχνευτεί concept drift. Όταν ο CDT ανιχνεύσει concept drift, το νέο σύνολο δεδομένων $O_{\hat{T}}$ χρησιμοποιείται για την επανεκπαίδευση του ταξινομητή.*

Στα επόμενα θα παρουσιάσουμε σε ένα παράδειγμα τον JIT για απότομα concept drift, και στη συνέχεια, θα δούμε περιληπτικά και την περίπτωση του βαθμιαίου concept drift. Ως CDT θα χρησιμοποιήσουμε τον ICI – CDT, αν και οποιοσδήποτε CDT θα μπορούσε να εφαρμοστεί.

## Το πρόβλημα

Ας θεωρήσουμε, χάριν απλότητας, ένα πρόβλημα ταξινόμησης δύο κλάσεων. Συγκεκριμένα, έστω $x \in X \subset \mathbb{R}^d$ μία i.i.r. μεταβλητή, και $y \in \{\omega_1, \omega_2\}$ οι αντίστοιχες κλάσεις. Η σ.π.π. της κατανομής τη χρονική στιγμή $t$,

$$p(x|t) = p(\omega_1|t)p(x|\omega_1, t) + p(\omega_2|t)p(x|\omega_2, t), \tag{3.6.1}$$

εξαρτάται απ' τις σ.π.π. των εξόδων $p(\omega_1|t)$ και $p(\omega_2|t) = 1 - p(\omega_1|t)$, καθώς και τις υπό συνθήκη κατανομές πιθανότητας $p(x|\omega_1, t)$ και $p(x|\omega_2, t)$. Γενικά, οι κατανομές αυτές είναι άγνωστες.

Έστω τώρα $O_T = \{x(t), t = 1, \dots, T\}$ η ακολουθία των δεδομένων κατά τη στιγμή $T$, και έστω $D_T = \{(x(t), y(t)), t \in I_T\}$ η βάση γνώσης του ταξινομητή τη χρονική στιγμή $T$, η οποία περιέχει τα supervised ζεύγη $(x(t), y(t))$, όπου το $y(t)$ είναι το label του $x(t)$, ενώ το $I_T$ είναι ένα σύνολο που περιέχει τους χρόνους άφιξης των επιβλεπόμενων δεδομένων μέχρι τη στιγμή $T$. Επιπλέον, υποθέτουμε ότι υπάρχει μία χρονική στιγμή $T_0$, πριν απ' την οποία τα δεδομένα είναι stationary, δηλαδή η κατανομή τους παραμένει σταθερή. Τότε, το σύνολο $O_{T_0}$ χρησιμεύει στην εκπαίδευση του CDT, ενώ το $D_0 = \{(x(t), y(t)), t \in I_0\}$ είναι η αρχική βάση γνώσης του ταξινομητή, πάνω στην οποία εκπαιδεύεται αρχικά (το $I_0$ είναι το σύνολο των διαθέσιμων επιβλεπόμενων δειγμάτων κατά τη στιγμή $T_0$). Ας υποθέσουμε τώρα ότι κατά τη χρονική στιγμή $T^\circ > T_0$ εμφανίζεται μία μεταβολή στη στατιστική κατανομή των δεδομένων $x(t)$ – η κατανομή μετά τη μεταβολή είναι φυσικά άγνωστη. Στο πλαίσιο JIT, ο CDT παρακολουθεί τη διαδικασία ελέγχοντας τα δεδομένα $O_T$, και, σε κάποιες παραλλαγές, και την διαθέσιμη επιβλεπόμενη πληροφορία.

## Ο ταξινομητής

Όπως βλέπουμε στον αλγόριθμο 3.6.1, ο JIT ταξινομητής εισέρχεται σε μία φάση προσαρμογής κάθε φορά που ο CDT εντοπίζει ένα concept drift. Αντίθετα, σε στάσιμες συνθήκες ενσωματώνει, σε κάποιες περιπτώσεις, τα διαθέσιμα επιβλεπόμενα δεδομένα στη βάση γνώσης του.

Μετά από ένα concept drift επομένως, ο ταξινομητής πρέπει να επανεκπαιδευτεί στα νέα δεδομένα που υπάρχουν απ' το drift και μετά. Τα δεδομένα αυτά περιέχονται στο σύνολο $D_{s|t>\bar{t}} = D_{s|[\bar{t},\hat{T}]}$, δηλαδή είναι τα δεδομένα του χρονικού διαστήματος $[\bar{t}, \hat{T}]$. Εκτός απ' τον ταξινομητή, με ταδεδομένα αυτά πρέπει να επανεκπαιδευτεί και ο ICI – CDT.

Για την ταξινόμηση μπορεί να χρησιμοποιηθεί κάθε συνεπής ταξινομητής, δηλαδή κάθε ταξινομητής που έχει την ιδιότητα της καθολικής προσέγγισης (universal

function approximation). Τέτοιοι ταξινομητές είναι τα νευρωνικά δίκτυα εμπρόσθιας τροφοδότησης, οι ταξινομητές kNN, τα νευρωνικά δίκτυα ακτινικής βάσης, καθώς και τα SVM και οι κανονικοποιημένοι πυρήνες (kernels), με την προϋπόθεση ότι έχει γίνει σωστή επιλογή των πυρήνων. Ο ταξινομητής που θα επιλέξουμε έχει να κάνει περισσότερο με τις απαιτήσεις της εφαρμογής μας, όσον αφορά την ταχύτητα, το υπολογιστικό κόστος, τις απαιτήσεις μνήμης, κλπ.

Για παράδειγμα, αν θέλουμε χαμηλό υπολογιστικό κόστος, μία καλή επιλογή είναι ο αλγόριθμος kNN. Το πλεονέκτημα του kNN είναι ότι επανεκπαιδεύεται ταχύτατα (απλώς του βάζουμε τα νέα σημεία στη βάση γνώσης) και μπορεί να ενσωματώνει νέα δεδομένα στον ταξινομητή πολύ εύκολα (απλώς προσθέτουμε τα σημεία). Το μειονέκτημα είναι η υψηλή απαίτηση μνήμης (χρειάζεται να αποθηκεύουμε όλα τα σημεία), αυτό όμως μπορεί να βελτιωθεί με διάφορα τεχνάσματα, το πιο απλό εκ των οποίων είναι το να θέσουμε ένα άνω όριο στο μήκος της βάσης γνώσης του ταξινομητή. Αν η βάση γνώσης γέμισε, τα πιο παλιά δεδομένα θα αφαιρούνται χάριν των πιο πρόσφατων, σε μία διαδικασία που μοιάζει με μία ουρά. Εκτός αυτού, υπάρχουν πιο εξελιγμένες τεχνικές, οι οποίες κρατάνε μόνο τα δείγματα που χρειάζονται για τον καθορισμό των ορίων απόφασης (decision boundaries), και απορρίπτουν τα υπόλοιπα (π.χ. η τεχνική Condensed Nearest Neighbor – CNN).

Επίσης, μία καλή επιλογή είναι και τα νευρωνικά δίκτυα, το πλεονέκτημα των οποίων είναι η χαμηλή απαίτηση μνήμης, καθώς και η ύπαρξη πόλων τεχνικών για ενσωμάτωση νέας γνώσης σε αυτά, οι οποίες είναι γενικά γνωστές με τον όρο tuning. Φυσικά, η εκπαίδευση του δικτύου είναι πιο χρονοβόρα, αλλά στη συνέχεια ο υπολογισμός της εξόδου είναι πιο αποδοτικός.

Μετά την επιλογή του CDT και του ταξινομητή, ο JIT αλγόριθμος είναι αρκετά σαφής. Παρακάτω παρουσιάζουμε π.χ. ένα παράδειγμα στο οποίο χρησιμοποιείται ο H – CDT ως change detector.

Αλγόριθμος 3.6.2: Ένας JIT ταξινομητής βασισμένος στον H - CDT

Θέσε $I_0 = \{1, \dots, T_0\}$, $D_0 = \{(x(t), y(t)), t \in I_0\}$, $O_{T_0}$;
Εκπαίδευσε τον ταξινομητή πάνω στο $D_0$;
Εκπαίδευσε τον ICI – CDT πάνω στο $O_{T_0}$;
Θέσε $D_t = D_0$, $I_t = I_0$, $t = T_0 + 1$;
**while** (1) **do**
  Πάρε το δείγμα $x(t)$ απ' την είσοδο;
  **if** (είναι διαθέσιμη supervised πληροφορία για το $x(t)$) **then**
    $I_t = I_{t-1} \cup \{t\}$;
    $D_t = D_{t-1} \cup \{(x(t), y(t))\}$;
  **else**
    $I_t = I_{t-1}$;
    $D_t = D_{t-1}$;
  **end**
  **if** (ο H – CDT ανιχνεύσει concept drift) **then**
    Έστω $\hat{T}$ η χρονική στιγμή ανίχνευσης του drift;
    Υπολόγισε το $\bar{t}$ απ' τον H – CDT;
    Εκπαίδευσε τον ICI – CDT στο διάστημα $[\bar{t}, \hat{T}]$;
    Εκπαίδευσε το τεστ Hotelling στην ακολουθία χαρακτηριστικών $s|t > \bar{t}$;

19 $I_t = \{t \in T_t, t > \bar{t}\}$;
20 $D_t = \{(x(t), y(t)), t \in I_t\}$;
21 **end**
22 Προσάρμοσε τον ταξινομητή στο σύνολο $D_t$;
23 Ταξινόμησε το δείγμα $x(t)$;
24 $t = t + 1$;
25 **end**

*Αλγόριθμος 3.6.2: Ένας JIT ταξινομητής βασισμένος στον H – CDT.*

Ας δούμε λίγο πιο αναλυτικά τη λειτουργία του παραπάνω αλγορίθμου. Αρχικά, μας δίνεται ένα supervised σύνολο δεδομένων $D_0$, με το οποίο εκπαιδεύουμε τον ταξινομητή μας (θεωρούμε ότι σε αυτό το αρχικό σύνολο δεν υπάρχει concept drift). Επίσης, αρχικοποιούμε το ICI τμήμα του H – CDT στο σύνολο $O_{T_0} = \{x(t), t \in I_0\}$ (γραμμές 2 και 3). Μετά απ' αυτή τη φάση αρχικοποίησης, ο αλγόριθμος λειτουργεί ακολουθιακά (online), και ταξινομεί τα νέα δείγματα, καθώς έρχονται στην είσοδο, προσθέτοντας, όταν είναι διαθέσιμα, τα supervised δεδομένα στη βάση γνώσης του. Στην περίπτωση αυτή, προστίθεται στο $I_t$ η νέα χρονική στιγμή, και το ζεύγος $(x(t), y(t))$ προστίθεται στο $D_t$. Έτσι, στη stationary περίπτωση η ακρίβεια του ταξινομητή αυξάνεται συνεχώς, προσθέτοντας νέα supervised δεδομένα στη βάση γνώσης του. Διαφορετικά, αν το δείγμα δεν είναι labeled, τα $I_t$ και $D_t$ παραμένουν ίδια (γραμμές 11 και 12). Στη συνέχεια, σε περίπτωση που ο H – CDT εντοπίσει κάποια μεταβολή (γραμμή 14), ο αλγόριθμος εξάγει την καλύτερη εκτίμηση του χρόνου drift $\bar{t}$ (γραμμή 16). Στη συνέχεια, ο H – CDT επανεκπαιδεύεται στα χαρακτηριστικά $s$ που συνδέονται με το χρονικό διάστημα $[\bar{t}, \hat{T}]$ (γραμμή 17). Επιπλέον, η πληροφορία για το $\bar{t}$ επιτρέπει στον JIT να απορρίψει τα δεδομένα εκπαίδευσης που έφτασαν πριν τη χρονική στιγμή $\bar{t}$, προσαρμόζοντας κατάλληλα τα $I_t$ και $D_t$ (γραμμές 19 και 20). Τέλος, το $x(t)$ ταξινομείται με βάση τα νέα δεδομένα εκαπίδευσης (γραμμή 23).

## 3.7 Βαθμιαία μεταβολή πλαισίου

Ο παραπάνω αλγόριθμος JIT αποδεικνύεται ότι είναι ασυμπτωτικα βέλτιστος (συγκεκριμένα, στην περίπτωση του kNN, στην [31]), με την έννοια ότι, μετά την ανίχνευση των μεταβολών, η απόδοση του ταξινομητή αυξάνεται κατά τη διάρκεια του χρόνου, χρησιμοποιώντας τα διαθέσιμα επιβλεπόμενα δείγματα. Στην περίπτωση όμως που τα concept drifts που ανιχνεύονται είναι πολύ κοντά χρονικά το ένα με το άλλο, το σύνολο επανεκπαίδευσης είναι πολύ μικρό, και συνεπώς η απόδοση μειώνεται, παρόλο το μηχανισμό προσαρμογής που έχουμε. Σε αυτές τις περιπτώσεις, ακόμη και ένας απλός παθητικός αλγόριθμος που εκπαιδεύεται απλώς με τα τελευταία $N$ δείγματα θα είχε καλύτερη απόδοση.
Η κατάσταση αυτή μπορεί να προκύψει στην περίπτωση που τα δεδομένα μας παρουσιάζουν βαθμιαίο drift, το οποίο ο αλγόριθμος μπορεί να αντιληφθεί ως μία ακολουθία απότομων drifts. Και πάλι, εδώ μια παθητική λύση ίσως να ήταν καλύτερη στην περίπτωση που το concept drift είναι γρήγορο. Παρόλα αυτά, στην

[48] προτείνεται μία επέκταση του JIT πλαισίου, έτσι ώστε να μπορεί να αντιμετωπίσει το βαθμιαίο drift. Η προσέγγιση αυτή περιλαμβάνει δύο στοιχεία: Πρώτον, μία τροποποίηση του βασικού ICI – CDT, η οποία του δίνει τη δυνατότητα να αντιμετωπίζει κατανομές των οποίων η μέση τιμή ακολουθεί μία πολυωνυμική καμπύλη (polynomial trend). Δεύτερον, περιλαμβάνεται επίσης ένας προσαρμοζόμενος ταξινομητής, ο οποίος έχει τη δυνατότητα να διαχειρίζεται το βαθμιαίο concept drift που επηρεάζει την αναμενόμενη τιμή μίας διαδικασίας. Συγκεκριμένα, ο ταξινομητής περιλαμβάνει έναν δείκτη, ο οποίος εκτιμά την δυναμική της εξέλιξης του drift, με στόχο τη βελτίωση της ακρίβειας της ταξινόμησης.

Χονδρικά μιλώντας, η προτεινόμενη επέκταση του ταξινομητή αντιμετωπίζει το βαθμιαίο concept drift εκτιμώντας την τάση του drift, αφαιρώντας τις μεταβολές, και θεωρώντας στη συνέχεια τη διαδικασία ως στάσιμη.
Θα μοντελοποιήσουμε το concept drift με τη βοήθεια των προηγούμενων εξισώσεων για τις κατανομές πιθανότητας. Συγκεκριμένα, το concept drift θα μοντελοποιηθεί ως μία αργά μεταβαλλόμενη στοχαστική διαδικασία, της οποίας η μέση τιμή, $\mathbb{E}[p(x|t)]$, ακολουθεί μία τμηματικά πολυωνυμική συνάρτηση $f_\theta(t)$. Η παραμετρική περιγραφή της $f_\theta(t)$ δίνεται από την $\{(\theta_i, I_i)\}$, όπου το $\theta_i$ είναι ένα διάνυσμα παραμέτρων που χαρακτηρίζει το πολυώνυμο $f_{\theta_i}(t)$, το οποίο ορίζεται στο διάστημα $U_i$, το οποίο είναι ένα τμήμα του χρονικού διαστήματος $[0, t]$, δηλ. μία υπακολουθία διαδοχικών χρονικών στιγμών. Οι μέσες τιμές των υπό συνθήκη συναρτήσεων πιθανότητας μπορούν να γραφτούν στη μορφή:

$$\mathbb{E}[p(x|\omega_1, t)] = f_{\theta_i}(t) + r_{1,i}, \tag{3.7.1}$$

$$\mathbb{E}[p(x|\omega_2, t)] = f_{\theta_i}(t) + r_{2,i}, \tag{3.7.2}$$

όπου το $t \in I_i$, ενώ τα $r_{1,i}$ και $r_{2,i}$ είναι οι μέσες τιμές των δύο κλάσεων σε στάσιμες συνθήκες. Με βάση τα παραπάνω, η διαδικασία παραγωγής δεδομένων γίνεται ως εξής:

$$x(t) = \begin{cases} f_{\theta_i}(t) + g_{1,i}, & y(t) = \omega_1, \\ f_{\theta_i}(t) + g_{2,i}, & y(t) = \omega_2, \end{cases} \tag{3.7.3}$$

όπου τα $g_{1,i}$ και $g_{2,i}$ είναι τυχαίες μεταβλητές, των οποίων οι κατανομές πιθανότητας χαρακτηρίζουν τις κλάσεις σε στάσιμες συνθήκες – συγκεκριμένα, θα έχουμε $\mathbb{E}[g_{1,i}] = r_{1,i}$ και $\mathbb{E}[g_{2,i}] = r_{2,i}$.
Επιπλέον, θεωρούμε ότι οι πιθανότητες $p(x|\omega_1, t)$ και $p(x|\omega_2, t)$ δεν μεταβάλλονται στο εσωτερικό των διαστημάτων που ορίζουν τις τμηματικά πολυωνυμικές συναρτήσεις, επομένως η σ.π.π. του $x(t)$ θα είναι:

$$p(x|t) = p_i(\omega_1)p(x|\omega_1, t) + p_i(\omega_2)p(x|\omega_2, t), \qquad t \in I_i. \tag{3.7.4}$$

Οι σ.π.π. των εισόδων, η υπό συνθήκη κατανομές, και οι κατανομές των εξόδων είναι φυσικά άγνωστες. Επίσης, η τμηματικά πολυωνυμική συνάρτηση μέσα σε κάθε υποδιάστημα $I_i$, δηλ. η $f_{\theta_i}(t)$, είναι επίσης άγνωστη, αλλά κοινή μεταξύ των

δύο κλάσεων, όπως φαίνεται στις παραπάνω εξισώσεις. Επίσης πρέπει να σημειωθεί ότι το πλαίσιο αυτό είναι μία επέκταση παλαιότερων προσεγγίσεων, που υπέθεταν ότι η $f_{\theta_i}(t)$ είναι σταθερή. Με βάση τα παραπάνω, θα περάσουμε τώρα στον ταξινομητή.

## Ο JIT για βαθμιαίο concept drift

Η κύρια ιδέα της παρούσας προσέγγισης είναι το να επεκτείνουμε το μοντέλο που παραδοσιακά υιοθετείται στα προβλήματα ταξινόμησης, επιτρέποντας στη μέση τιμή των υπό συνθήκη σ.π.π. να εξελίσσονται στο χρόνο ως τμηματικά πολυωνυμικές συναρτήσεις, όπως περιγράφηκε στην προηγούμενη παράγραφο. Με αυτό ως υπόθεση, μπορούμε να κατασκευάσουμε έναν CDT, ο οποίος να παρακολουθεί τις μεταβολές στην (πολυωνυμική) τάση (trend) της διαδικασίας, παρά στην μέση τιμή της. Εάν το τεστ δεν ανιχνεύσει μεταβολές, εκτελούμε μία παλινδρόμηση στα δείγματα εισόδου, και χρησιμοποιούμε τους συντελεστές της παλινδρόμησης για να τροποποιήσουμε τη βάση γνώσης ενός προσαρμοζόμενου ταξινομητή. Διαφορετικά, εάν ανιχνευτεί μία μεταβολή, τα παλαιά δείγματα αφαιρούνται απ' τη βάση γνώσης, και το τεστ επαναρχικοποιείται.
Αφού ο ICI – CDT έχει ενδογενώς την ικανότητα να χειρίζεται διαδικασίες που παρουσιάζουν πολυωνυμικές τάσεις, μπορεί να εφαρμοστεί με κάποιες τροποποιήσεις στο επίπεδο των χαρακτηριστικών, σύμφωνα με τις εξισώσεις της προηγούμενης παραγράφου. Μία αναλυτική περιγραφή μπορεί να βρεθεί στο [48].
Στην ίδια εργασία, ο ταξινομητής που χρησιμοποιείται είναι ένας kNN, ο οποίος τροποποιείται σε έναν μικρό βαθμό, έτσι ώστε να προσαρμόζεται στο βαθμιαίο concept drift. Η διαδικασία αυτή φαίνεται στον παρακάτω αλγόριθμο.

Αλγόριθμος 3.7.1: Adaptive kNN για βαθμιαίο concept drift

1 $N = |Z_T|$;
2 $i = 1$;
3 **while** $(i < N)$ **do**
4 $\quad d_i = \left(\left(x(t) - f_{\hat{\theta}(t)}(t)\right) - \left(x(t_i) - f_{\hat{\theta}(t)}(t_i)\right)\right)$;
5 $\quad i = i + 1$;
6 **end**
7 Βρες τους $k$ κοντινότερους γείτονες με βάση τις αποστάσεις $\{d_i\}_{i=1,\ldots,N}$;
8 Ταξινόμησε το $x(t)$ με βάση την κλάση της πλειοψηφίας των $k$ γειτόνων;

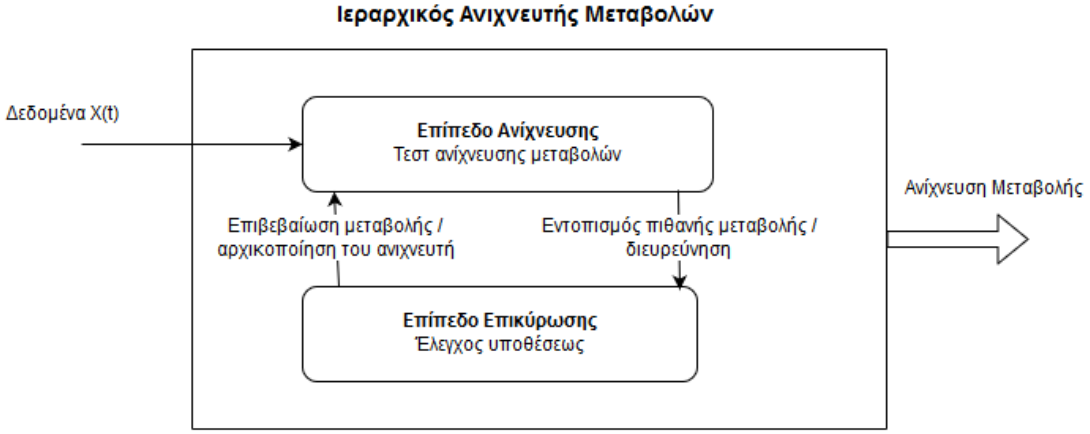
*Αλγόριθμος 3.7.1: Adaptive kNN για βαθμιαίο concept drift.*

Από τον παραπάνω αλγόριθμο βλέπουμε ότι η μόνη αλλαγή του σε σχέση με τον παραδοσιακό kNN είναι ο τρόπος υπολογισμού των αποστάσεων μεταξύ των δειγμάτων και του τρέχοντος (test) δείγματος. Εδώ, το διάνυσμα παραμέτρων $\hat{\theta}(t)$ αντιπροσωπεύει τους συντελεστές της καλύτερης πολυωνυμικής προσέγγισης για τα δεδομένα κατά τη διάρκεια του βαθμιαίου concept drift. Οι συντελεστές αυτοί μπορούν να υπολογιστούν εύκολα από τα δείγματα χρησιμοποιώντας οποιαδήποτε τεχνική παλινδρόμησης.
Η πολυωνυμική προσέγγιση αντιπροσωπεύει το βαθμιαίο drift, και χρησιμοποιείται για τη διόρθωση του κάθε όρου ως συνάρτηση της απόστασης μεταξύ των

δεδομένων και του προσεγγιστικού πολυωνύμου. Συγκεκριμένα, η απόσταση μεταξύ του τρέχοντος δείγματος $x(t)$ και του δείγματος εκπαίδευσης $x(t_i)$ υπολογίζεται αφού αφαιρεθεί απ' αυτήν η τιμή του προσεγγιστικού πολυωνύμου με τους συντελεστές $\hat{\theta}(t)$ στην αντίστοιχη χρονική στιγμή (π.χ., τα $f_{\hat{\theta}(t)}(t)$ και $f_{\hat{\theta}(t)}(t_i)$).
Χρησιμοποιώντας λοιπόν αυτούς τους τροποποιημένους CDT και kNN μπορούμε να εφαρμόσουμε την JIT προσέγγιση και σε περιπτώσεις βαθμιαίου concept drift, ακολουθώντας κατά βάση τον αλγόριθμο 3.6.2. Σημειώστε επίσης ότι η παρούσα επέκταση μπορεί να χρησιμοποιηθεί και στην περίπτωση απουσίας βαθμιαίου drift, αφού στην περίπτωση αυτή, οι συντελεστές μεγαλύτερης τάξης του πολυωνύμου τείνουν στο μηδέν, και ο JIT λειτουργεί σε στάσιμες συνθήκες.

## 3.8 Υλοποίηση και μετρήσεις

Όπως είδαμε στις προηγούμενες παραγράφους, η βασική αρχιτεκτονική ενός ιεραρχικού ανιχνευτή μεταβολών είναι αυτή που φαίνεται στο παρακάτω σχήμα:




*Σχήμα 3.8.1: Η βασική αρχιτεκτονική ενός ιεραρχικού ανιχνευτή μεταβολών (change detector).*

Όπως βλέπουμε στο παραπάνω σχήμα, η αρχιτεκτονική αυτή περιλαμβάνει δύο επίπεδα: ένα επίπεδο ανίχνευσης μεταβολών, το οποίο ανιχνεύει τις μεταβολές με βάση κάποιο τεστ CDT, και ένα επίπεδο ελέγχου – επικύρωσης των μεταβολών, στο οποίο ελέγχονται οι μεταβολές που βρίσκει το πρώτο επίπεδο με βάση κάποιο στατιστικό τεστ υποθέσεως. Με τον τρόπο αυτό, οι ιεραρχικοί ανιχνευτές μεταβολής καταφέρνουν να μειώνουν σημαντικά τον αριθμό των λανθασμένων θετικών ανιχνεύσεων.

Με βάση την αρχιτεκτονική αυτή, καθώς και τη θεωρία που παρουσιάστηκε στο παρόν κεφάλαιο, κατασκευάσαμε τρεις διαφορετικές ιεραρχικές αρχιτεκτονικές ανίχνευσης μεταβολών:

- **Αρχιτεκτονική 1: Ο ιεραρχικός ICI CDT.**
  Η πρώτη αρχιτεκτονική είναι ο ανιχνευτής που βασίζεται στον αλγόριθμο ICI, ο οποίος περιγράφηκε αναλυτικά στο παραπάνω κεφάλαιο. Για την

επικύρωση των μεταβολών χρησιμοποιείται το στατιστικό τεστ Hotelling, ενώ χρησιμοποιείται επίσης και ο κανόνας βελτίωσης του χρόνου ανίχνευσης, που περιγράφηκε στην ενότητα 3.5.

- **Αρχιτεκτονική 2: Ο αλγόριθμος ICI με μέθοδο σημείου μεταβολής (CPM).**
  Όπως προαναφέρθηκε, οι μέθοδοι CPM είναι κατάλληλες στο να βρίσκουν μεταβολές σε στατικά δεδομένα, αλλά δεν υποστηρίζουν ακολουθιακή χρήση. Ο λόγος γι' αυτό είναι ότι στην περίπτωση των ακολουθιακών δεδομένων θα έπρεπε κάθε φορά να χωρίζουμε τα μέχρι τώρα δεδομένα σε όλες τις δυνατές διαμερίσεις, και να εφαρμόζουμε τον έλεγχο υποθέσεως στο καθένα απ' αυτά. Αυτό είναι προφανώς υπολογιστικά δύσκολο.
  Παρόλα αυτά, μπορούμε να συνδυάσουμε μία μέθοδο CPM με ένα τεστ ανίχνευσης μεταβολές με τον εξής τρόπο: Όπως έρχονται σειριακά τα δεδομένα, το τεστ ανίχνευσης μεταβολών (CDT), θα τα εξετάζει κάθε φορά για πιθανές μεταβολές. Μόλις ο CDT εντοπίσει μία μεταβολή, θα μπορούσαμε στη συνέχεια να χρησιμοποιήσουμε μία μέθοδο CPM για την επικύρωση και την εύρεση του ακριβούς χρόνου της μεταβολής. Στην περίπτωση αυτή, ο αλγόριθμος CPM δεν χρησιμοποιείται κάθε φορά, αλλά μόνο όταν το CDT εντοπίσει κάποια μεταβολή, οπότε η υπολογιστικές απαιτήσεις του συστήματος είναι πλέον διαχειρίσιμες. Επιπλέον, το CPM μας προσφέρει αυτομάτως μία βελτιωμένη εκτίμηση του ακριβούς χρόνου της μεταβολής.
  Συνεπώς, στη δεύτερη αρχιτεκτονική υλοποιήσαμε την ιδέα αυτή, και συνδυάσαμε τον βασικό αλγόριθμο ICI – CDT με ένα CPM ως επίπεδο επικύρωσης, το οποίο βασίζεται στο μη παραμετρικό στατιστικό τεστ Mann – Whitney.

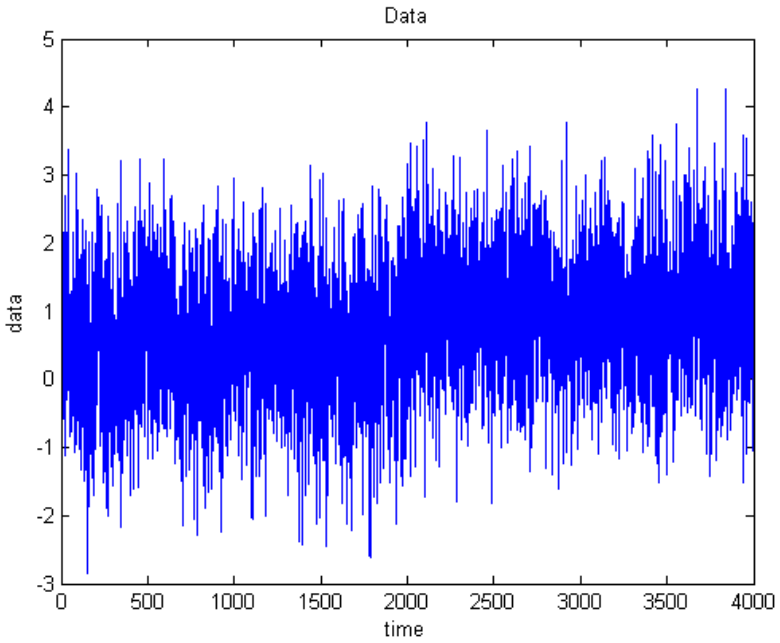
- **Αρχιτεκτονική 3: Ο αλγόριθμος CUSUM με CPM**
  Τέλος, στην Τρίτη αρχιτεκτονική χρησιμοποιήσαμε τον αλγόριθμο adaptive CUSUM για την εύρεση των μεταβολών, σε συνδυασμό με ένα CPM βασισμένο και πάλι στο τεστ Mann – Whitney.

Στη συνέχεια, υλοποιήσαμε τις αρχιτεκτονικές αυτές, και τις εφαρμόσαμε σε μερικές τεχνητές κατανομές δεδομένων (δηλ. τα δεδομένα δεν προέρχονται απ' τον πραγματικό κόσμο, αλλά παράγονται απ' τον υπολογιστή).

**Πείραμα 1**: Αρχικά κατασκευάσαμε μία Γκαουσσιανή κατανομή, της οποίας η μέση τιμή μεταβάλλεται ξαφνικά κατά μισή μονάδα τη χρονική στιγμή $t = 2000$. Ειδικότερα, η στοχαστική διαδικασία $x(t)$ ικανοποιεί τη σχέση:

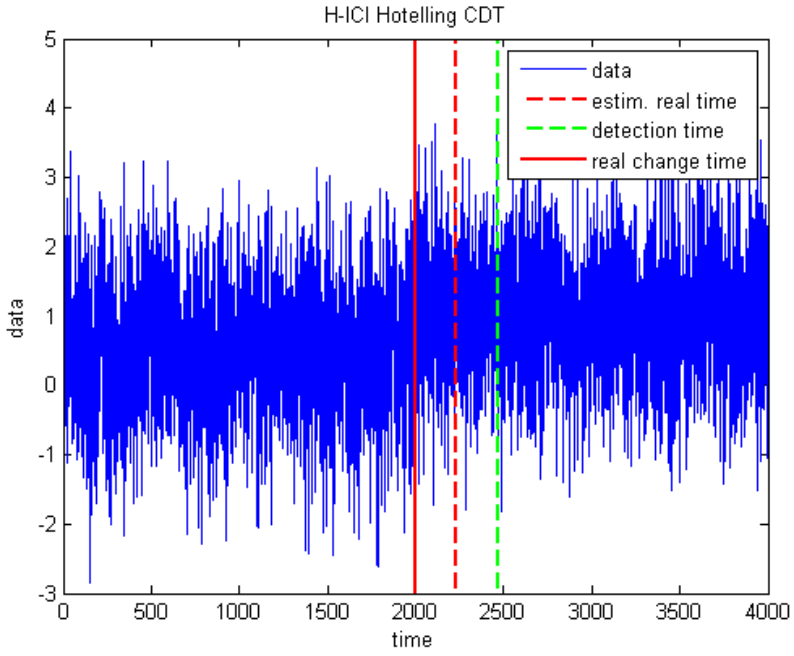
$$P(x(t)) = \begin{cases} N(0.5,1), & t < 2000 \\ N(1.0,1), & t < 2000 \end{cases}. \qquad (3.8.1)$$

Η διαδικασία αυτή φαίνεται παρακάτω:



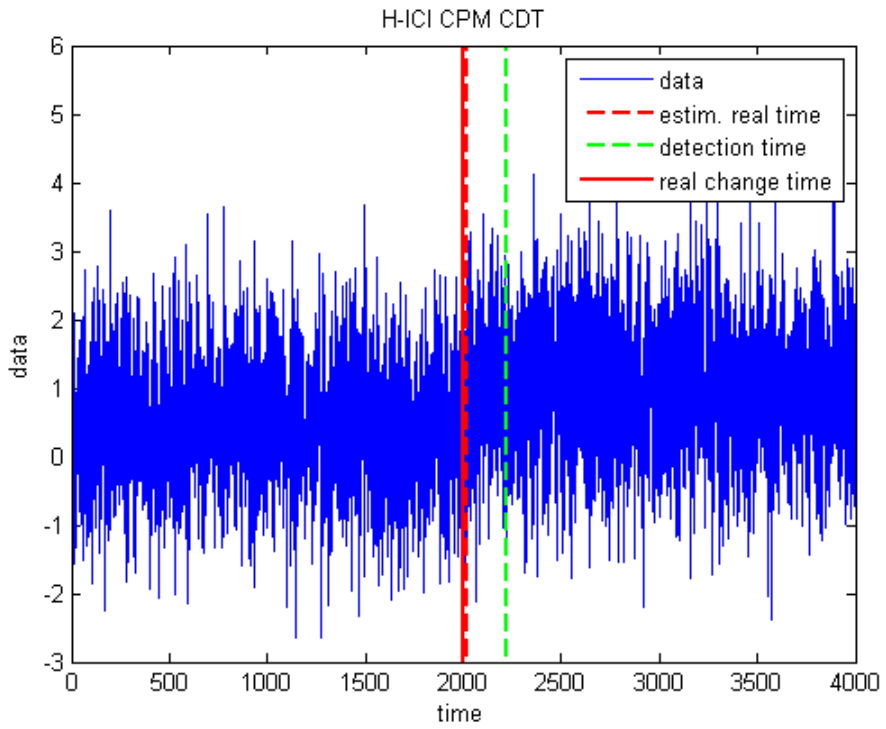

*Σχήμα 3.8.2: Τα δεδομένα του πειράματος 1.*

Στη συνέχεια, εφαρμόσαμε τους τρεις ανιχνευτές μεταβολής.
Ο αλγόριθμος της πρώτης αρχιτεκτονικής (H – ICI Hotelling) έδωσε τα παρακάτω αποτελέσματα:



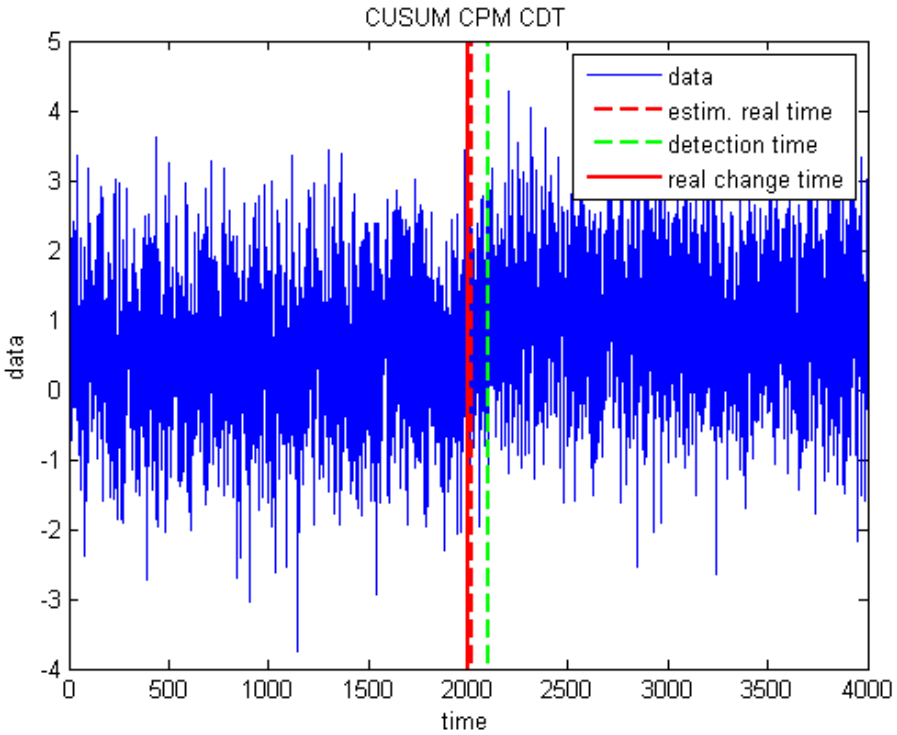

*Σχήμα 3.8.3: Η ανίχνευση του αλγορίθμου H – ICI Hotelling.*

Η δεύτερη αρχιτεκτονική (H – ICI CPM) έδωσε το παρακάτω αποτέλεσμα:




*Σχήμα 3.8.4: Η ανίχνευση του αλγορίθμου H – ICI CPM.*

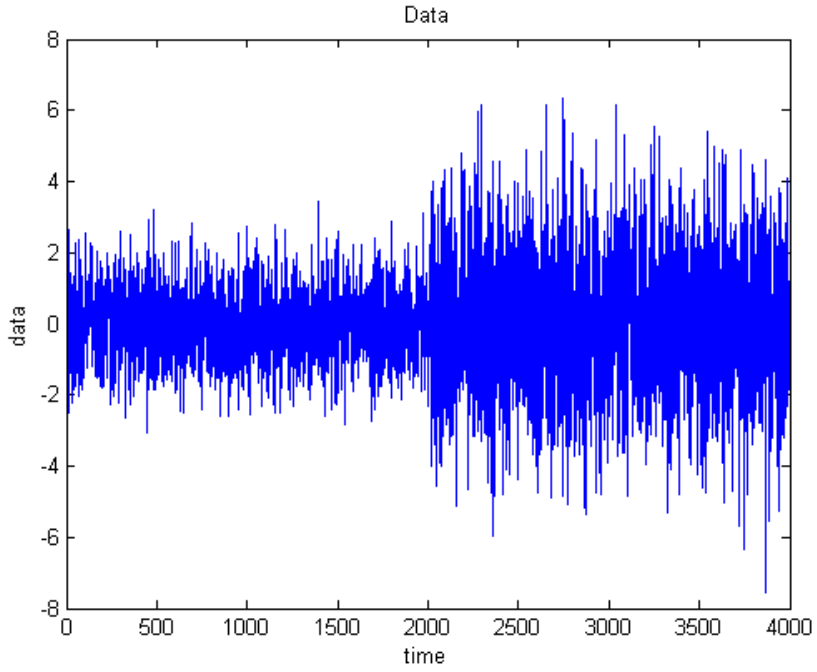
Τέλος, η τρίτη αρχιτεκτονική (CUSUM CPM CDT)




*Σχήμα 3.8.5: Η ανίχνευση του αλγορίθμου CUSUM CPM.*

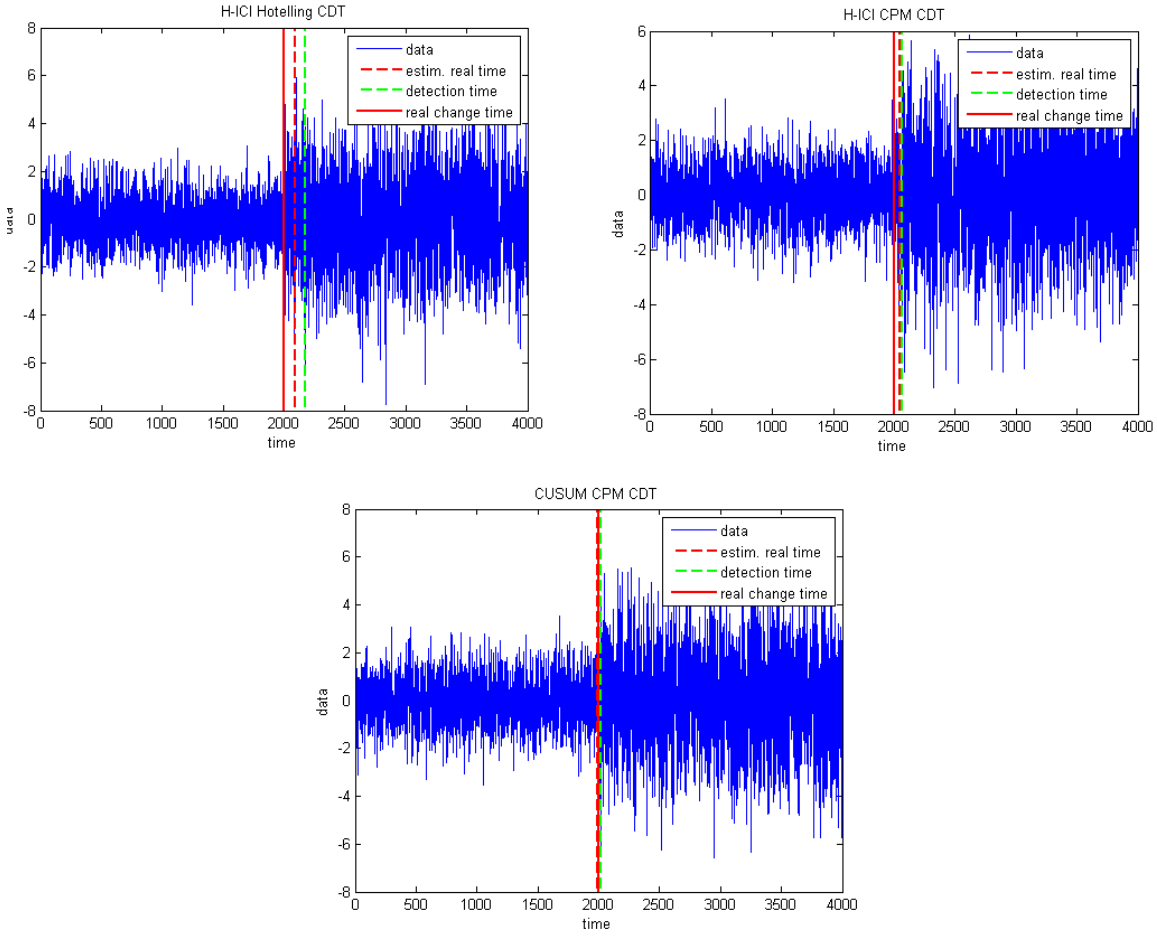
Στα παραπάνω διαγράμματα, η διακεκομμένη πράσινη γραμμή δείχνει τις ανιχνεύσεις μεταβολής που έχουν επικυρωθεί απ' το δεύτερο επίπεδο (οι συνολικές ανιχνεύσεις του πρώτου επιπέδου είναι αρκετά περισσότερες), ενώ η διακεκομμένη κόκκινη γραμμή δείχνει τη βελτιωμένη εκτίμηση του ανιχνευτή για την χρονική στιγμή που συνέβη η μεταβολή. Τέλος, η κόκκινη γραμμή δείχνει τη θέση της πραγματική μεταβολής.
Όπως βλέπουμε, και οι τρεις αλγόριθμοι ήταν σε θέση να εντοπίσουν τη μεταβολή. Βλέπουμε επίσης ότι αλγόριθμος H – ICI Hotelling εντόπισε τη μεταβολή αρκετά αργότερα, απ' τους άλλους, δύο, ενώ η εκτίμησή του για την πραγματική στιγμή της μεταβολής έχει αρκετά μεγαλύτερη απόκλιση απ' ότι αυτές των άλλων δύο αλγορίθμων. Αυτό είναι αναμενόμενο, αφού οι μέθοδοι CPM είναι γενικά ισχυρότερες απ' το απλό στατιστικό τεστ και τη διαδικασία εκλέπτυνσης του αλγορίθμου H – ICI Hotelling. Παρόλα αυτά, ο τελευταίος αλγόριθμος υπερτερεί σημαντικά σε ταχύτητα και μειωμένη υπολογιστική πολυπλοκότητα έναντι των άλλων δύο.

**Πείραμα 2:** Στη συνέχεια κατασκευάσαμε μία δεύτερη κανονική κατανομή, αλλά αυτή τη φορά κρατήσαμε τη μέση τιμή σταθερή και μεταβάλλαμε τη διακύμανση κατά μία μονάδα (από 1 σε 2).
Τα δεδομένα του πειράματος φαίνονται παρακάτω:




*Σχήμα 3.8.6: Τα δεδομένα του πειράματος 2.*

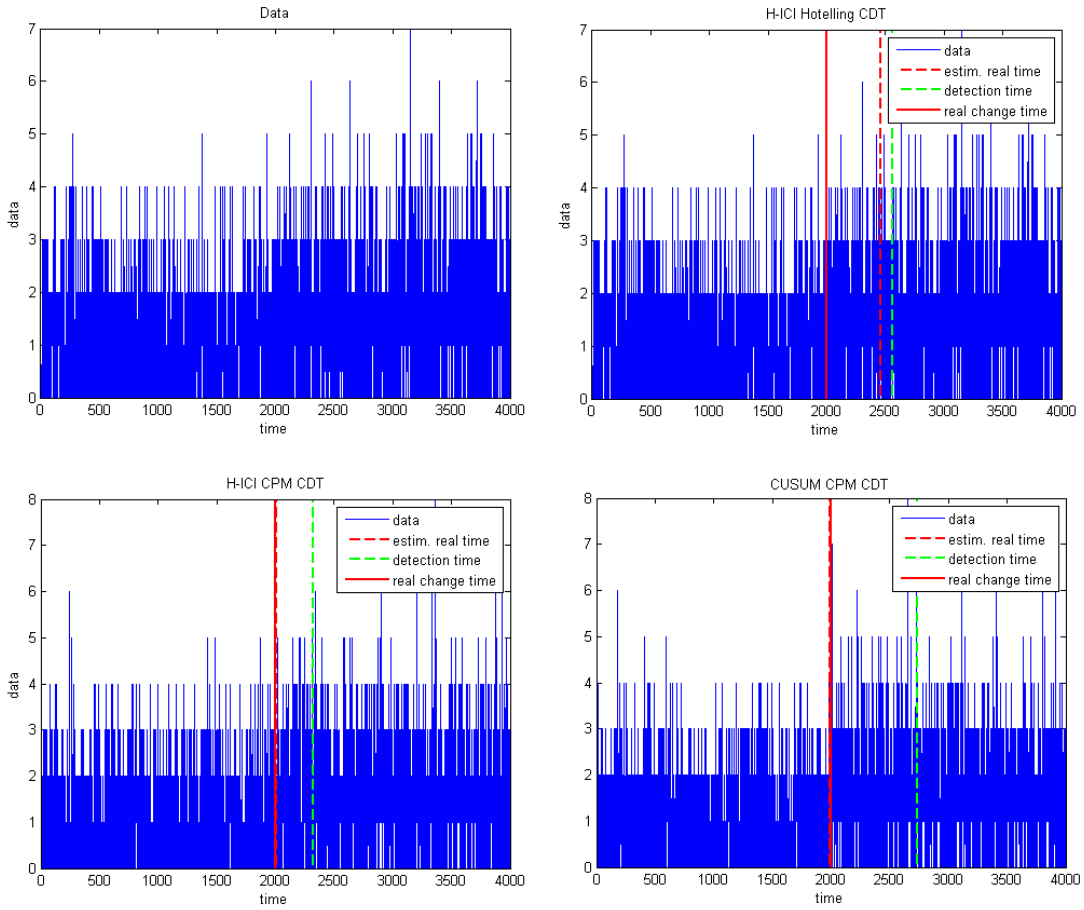
Τα αποτελέσματα της ανίχνευσης ήταν τα εξής:




*Σχήμα 3.8.7: Αποτελέσματα του πειράματος 2.*

Όπως βλέπουμε, η μεταβολή αναγνωρίζεται και πάλι απ' όλους τους ανιχνευτές, ενώ ισχύουν παρόμοια συμπεράσματα, ως προς την απόδοση, με το πείραμα 1.

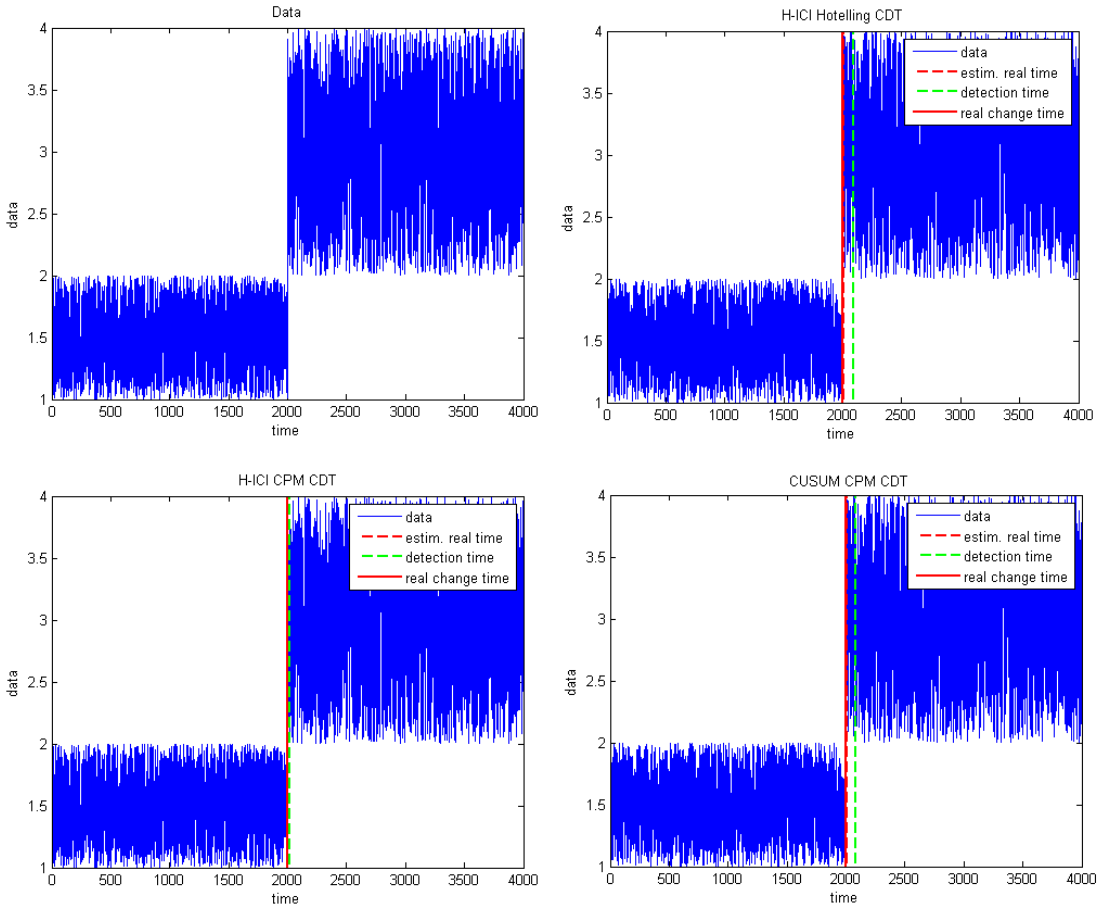
**Πείραμα 3:** Στη συνέχεια δοκιμάζουμε το ίδιο με μία κατανομή Poisson, στην οποία αυξάνουμε ξαφνικά το $\lambda$ κατά 1. Δεδομένα και αποτελέσματα φαίνονται παρακάτω:




*Σχήμα 3.8.8:Δεδομένα και αποτελέσματα του πειράματος 3.*

Όπως βλέπουμε, η κατανομή Poisson δυσκόλεψε λίγο περισσότερο τους ταξινομητές απ' ότι οι απλές Γκαουσσιανές.

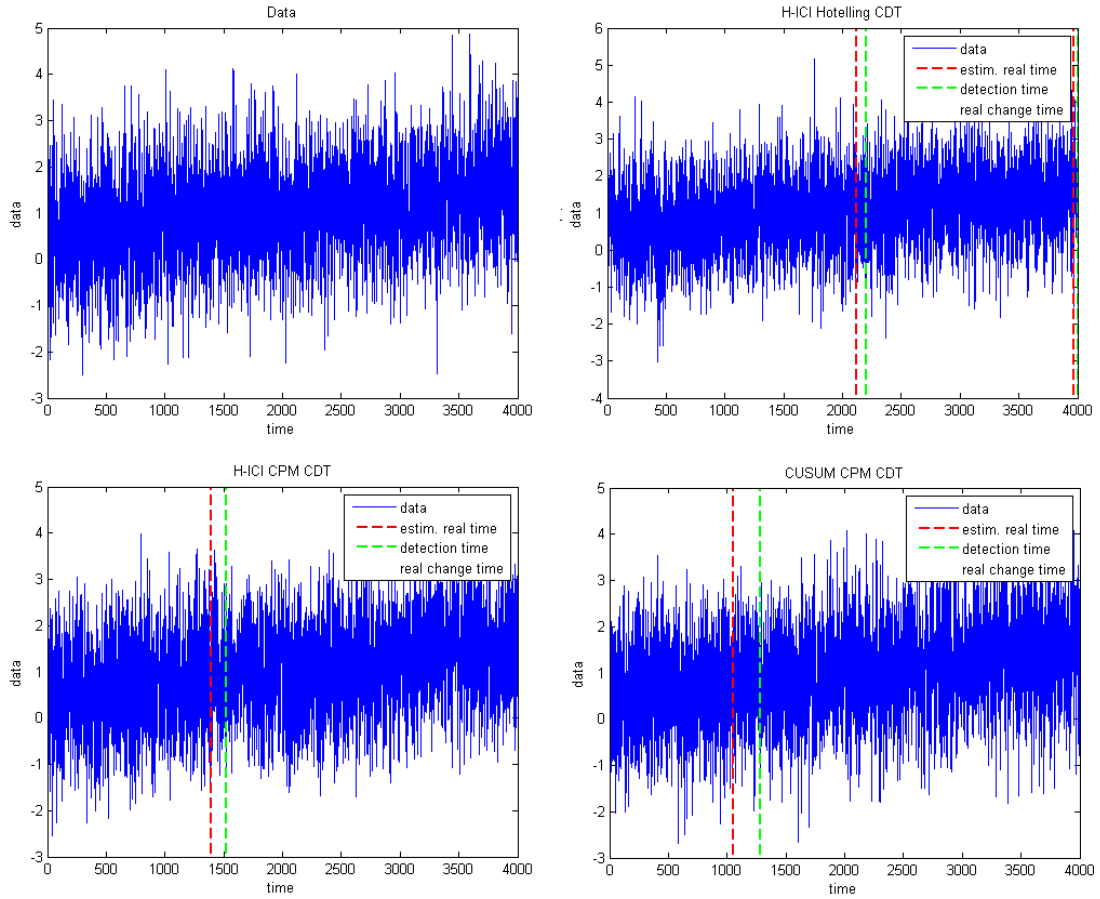
**Πείραμα 4:** Εδώ θέλουμε να δοκιμάσουμε την απόδοση των ανιχνευτών στην περίπτωση που έχουμε μία πολύ απότομη μεταβολή. Για το λόγο αυτό, κατασκευάσαμε μία ομοιόμορφη κατανομή της οποίας το διάστημα μεταβάλλεται ξαφνικά κατά τη στιγμή στιγμή $t = 2000$, απ' το $[1,2]$ στο $[2,4]$. Τα δεδομένα και τα αποτελέσματα φαίνονται παρακάτω:




*Σχήμα 3.8.9: Δεδομένα και αποτελέσματα του πειράματος 4.*

Όπως βλέπουμε, όλοι οι ανιχνευτές ανιχνεύουν τη μεταβολή αυτή με μεγάλη επιτυχία.

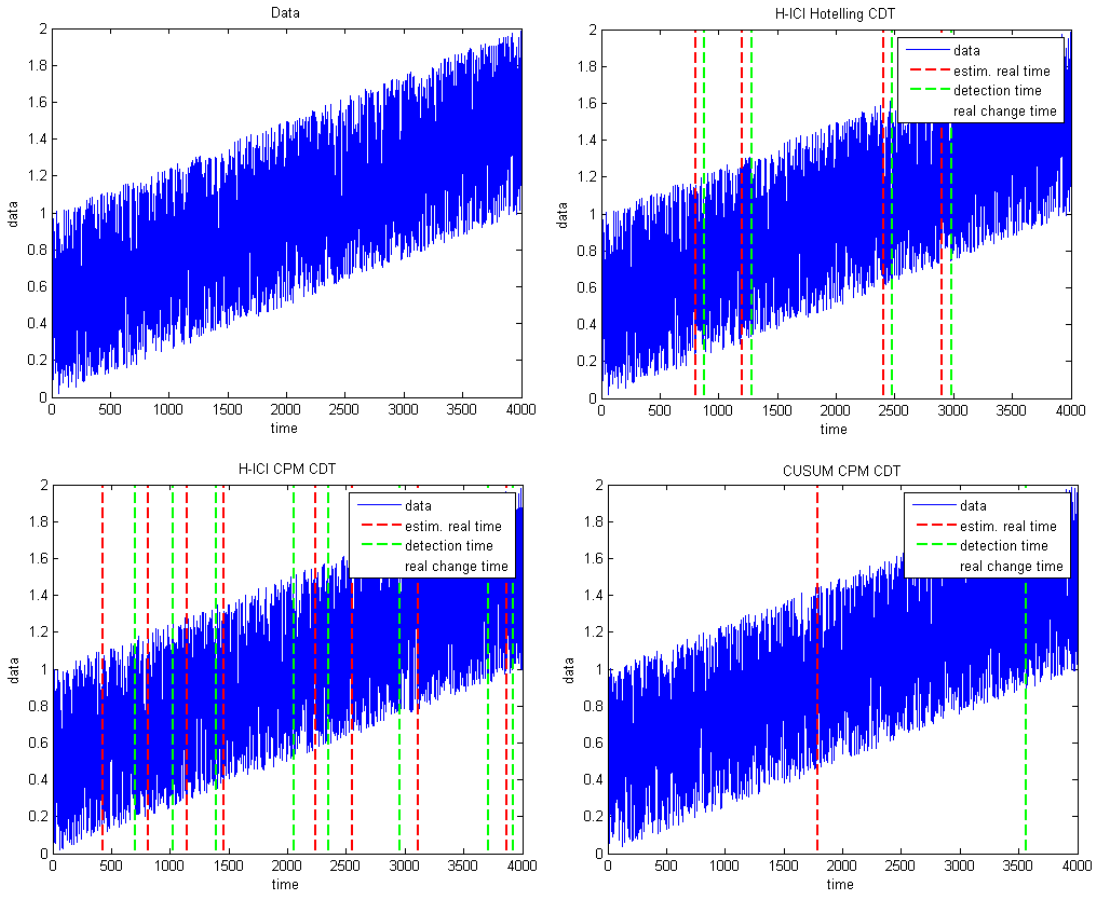
**Πείραμα 5:** Εδώ θέλουμε να εξετάσουμε την απόδοση των αλγορίθμων στην περίπτωση ενός βραδέως μεταβαλλόμενου concept drift. Για το λόγο αυτό, κατασκευάζουμε μία Γκαουσσιανή κατανομή, της οποίας η μέση τιμή μεταβάλλεται όπως στο πείραμα 1, αλλά σταδιακά, απ' την αρχή μέχρι το τέλος του διαστήματος του χρόνου. Τα δεδομένα και τα αποτελέσματα φαίνονται παρακάτω:




*Σχήμα 3.8.10: Δεδομένα και αποτελέσματα του πειράματος 5.*

Όπως βλέπουμε, οι αλγόριθμοι εντοπίζουν το drift κάπου στη μέση του διαστήματος, πράγμα που είναι λογικό (ο H – ICI Hotelling εντοπίζει και μία μεταβολή στο τέλος, αλλά τέτοιες ανιχνεύσεις μπορούν να απορριφθούν).

**Πείραμα 6:** Τέλος, δοκιμάζουμε τους αλγόριθμους σε μία βραδέως μεταβαλλόμενη ομοιόμορφη κατανομή. Δεδομένα και αποτελέσματα φαίνονται παρακάτω:




*Σχήμα 3.8.11: Δεδομένα και αποτελέσματα του πειράματος 5.*

Όπως βλέπουμε, η περίπτωση αυτή δυσκόλεψε τους ανιχνευτές. Ο CUSUM εντόπισε τη μεταβολή αρκετά αργά (αλλά προσδιόρισε αρκετά καλά το χρόνο), ενώ ο H – ICI CPM έκανε υπερβολικά πολλές ανιχνεύσεις. Η συμπεριφορά του H – ICI Hotelling τέλος βρίσκεται κάπου στη μέση.

Όπως βλέπουμε επομένως, όλοι ο ανιχνευτές συμπεριφέρονται αρκετά καλά σε όλες τις κατηγορίες τεχνητών δεδομένων, με εξαίρεση την περίπτωση αργά μεταβαλλόμενων drifts.

# 4 Το συναίσθημα και η ανάλυσή του

Ο στόχος της εργασίας αυτής είναι, όπως αναφέρθηκε και στην εισαγωγή, η εφαρμογή μεθοδολογιών του adaptive learning σε ένα πρόβλημα αναγνώρισης συναισθήματος. Για το λόγο αυτό, θεωρήσαμε σκόπιμο να αναφέρουμε στο κεφάλαιο αυτό κάποια βασικά θέματα σχετικά με το συναίσθημα και την ανάλυσή του. Συγκεκριμένα, θα αναφέρουμε εν συντομία τις βασικές θεωρίες που αφορούν την αναπαράσταση συναισθήματος, τα χαρακτηριστικά του προσώπου που χρησιμοποιούνται για την ανάλυση των εκφράσεων, κ.α.

## 4.1 Βασική θεωρία αναπαράστασης συναισθήματος

Στην ψυχολογία υπάρχει σήμερα μία πληθώρα θεωριών που σχετίζεται με τα συναισθήματα, τη φύση, και την κατηγοριοποίησή τους. Σε αυτές, τα συναισθήματα αναπαριστώνται βασικά με δύο τρόπους: είτε με μία κατηγορική μορφή, στην οποία επικρατούν κυρίως τα 6 βασικά συναισθήματα, είτε σε συνεχή μορφή.
Μία ιδιαίτερα χρήσιμη συνεχής αναπαράσταση, η οποία οφείλεται στους Russel και Whissel ([14]), γίνεται με βάση έναν δισδιάστατο χώρο, του οποίου οι δύο άξονες ορίζονται με βάση την ενεργοποίηση (activation) και την αξιολόγηση (evaluation). Το πλεονέκτημα της αναπαράστασης αυτής είναι ότι μπορεί να απεικονίσει ένα πολύ μεγάλο εύρος συναισθημάτων, και επίσης, με τη βοήθειά της, μπορούμε να δείξουμε τη φύση ενός συναισθήματος χωρίς να χρειάζεται να το προσδιορίσουμε επακριβώς.
Οι δύο παράμετροι του μοντέλου αυτού, ενεργοποίηση και αξιολόγηση, ορίζονται με τον εξής τρόπο:

- Ενεργοποίηση (activation): Σύμφωνα με τις μελέτες της ψυχολογίας, η συναισθηματική κατάσταση ενός ατόμου σχετίζεται με την προδιάθεσή του στο να δρα με συγκεκριμένους τρόπους. Συνεπώς, υπάρχει μία συσχέτιση μεταξύ συναισθήματος, και της επιθυμίας του ατόμου να προβε΄9ι ή όχι σε συγκεκριμένες ενέργειες. Αυτή η παρατήρηση αντανακλάται στην παράμετρο της ενεργοποίησης, η οποία μετρά τη διάθεση του ατόμου να προβεί ή όχι σε κάποιες ενέργειες. Μεγάλη ενεργοποίηση (active) σημαίνει ότι το άτομο επιθυμεί να δράσει με κάποιο τρόπο, και αυτό σχετίζεται με μία συγκεκριμένη κατηγορία συναισθημάτων, όπως π.χ. ο θυμός. Αντίθετα, μικρή ενεργοποίηση (passive) σημαίνει ότι το άτομο δεν επιθυμεί να προβεί σε κάποια ενέργεια, κάτι το οποίο σχετίζεται με μία άλλη κατηγορία συναισθημάτων, όπως π.χ. η θλίψη. Άρα, μπορούμε κατά μία έννοια να εκφράσουμε τα συναισθήματα με βάση την ενεργοποίησή τους, η οποία

αποτελεί την πρώτη παράμετρο του μοντέλου των Russel και Whissel για την έκφραση του συναισθήματος.

- Αξιολόγηση (evaluation): Η δεύτερη παράμετρος που όρισαν οι Russel και Wissel στην προσπάθειά τους να ορίσουν το συναίσθημα είναι η αξιολόγηση. Η παράμετρος αυτή σχετίζεται με το γεγονός, ότι, ανάλογα με τη συναισθηματική κατάστασή του, το άτομο τείνει να εκφράζει θετικές ή αρνητικές απόψεις για τα γεγονότα που συμβαίνουν γύρω του. Η αξιολόγηση εκφράζεται σε θετική (positive) και αρνητική (negative). Θετική αξιολόγηση σημαίνει ότι το άτομο εκφράζεται θετικά για τα πράγματα γύρω του, και η κατάσταση αυτή συνδέεται με κάποια συγκεκριμένα συναισθήματα, όπως π.χ. η χαρά. Αντίθετα, η τάση του ατόμου να εκφράζει αρνητικές απόψεις σχετίζεται με άλλα συναισθήματα, όπως π.χ. η λύπη. Επομένως, μπορούμε να ορίσουμε – διαχωρίσουμε τα συναισθήματα με βάση την παράμετρο αυτή.

Σύμφωνα με την ψυχολογία, ο συνδυασμός των δύο παραπάνω παραμέτρων αρκεί για τον καθορισμό των συναισθημάτων. Επιπλέον, για λόγους εποπτείας, συνηθίζεται τα συναισθήματα να αποτυπώνονται σε ένα σύστημα συντεταγμένων, όπου στους δύο άξονες βρίσκονται οι παραπάνω παράμετροι: ο οριζόντιος άξονας είναι ο άξονας της αξιολόγησης, ενώ ο κάθετος άξονας είναι ο άξονας της ενεργοποίησης. Δίνοντας εμπειρικές τιμές σε αυτές τις δύο παραμέτρους για κάθε συναίσθημα, οι οποίες καθορίζονται με βάση κάποια κριτήρια της ψυχολογίας, μπορούμε να απεικονίσουμε όλα τα συναισθήματα σε αυτό το σύστημα συντεταγμένων, το οποίο φαίνεται σχηματικά παρακάτω:


*Σχήμα 4.1.1: Χώρος αναπαράστασης συναισθήματος, σύμφωνα με τη θεωρία των Russel – Wissel.*

Στην παραπάνω αναπαράσταση υπάρχουν ακόμη δύο βασικά στοιχεία. Πρώτον, από μελέτες έχει φανεί ότι τα διάφορα αντιλαμβανόμενα συναισθήματα δεν είναι ομοιόμορφα κατανεμημένα στο σύστημα αυτό – αντίθετα τα συναισθήματα περιέχονται στο εσωτερικό ενός κύκλου. Το κέντρο του κύκλου αυτού θεωρείται ως κατάσταση ηρεμίας (neutral), όπου δηλαδή το άτομο δεν έχει κάποια συγκεκριμένη συναισθηματική κατάσταση. Ο βαθμός της συναισθηματικής έκφρασης μπορεί να εκφραστεί ως η απόσταση από το κέντρο του κύκλου έως το σημείο που βρίσκεται το συναίσθημα.
Επιπλέον, ένα συμπέρασμα που προκύπτει από την παραπάνω αναπαράσταση είναι ότι τα πιο ισχυρά – έντονα συναισθήματα είναι σε μεγαλύτερο βαθμό διαχωρισμένα μεταξύ τους απ' ότι τα λιγότερο έντονα της ίδιας κατηγορίας (όσον αφορά την ενεργοποίηση – αξιολόγηση). Με βάση αυτό, οι ερευνητές έχουν τοποθετήσει τα βασικά συναισθήματα στην περιφέρεια του κύκλου ([18]).

## 4.2 Χαρακτηριστικά εικόνων προσώπου

Συνήθως, οι περισσότερες τεχνικές για αναγνώριση συναισθήματος από υπολογιστή βασίζονται στην εξαγωγή κάποιων χαρακτηριστικών από την εικόνα του προσώπου. Τα χαρακτηριστικά αυτά είναι γνωστά με τον όρο «Σύστημα Κωδικοποίησης Αντιδράσεων Προσώπου» (Facial Action Coding System – FACS). Τα χαρακτηριστικά FACS σχετίζονται με τις ανατομικές λεπτομέρειες του προσώπου, και βασίζονται στις μονάδες δράσης (action units - AU) του προσώπου, οι οποίες σχετίζονται με την ανατομία και την κίνηση των μυών. Επιπλέον, το μοντέλο FACS αποτελεί βάση για τα μοντέλα σύνθεσης προσώπου και τον καθορισμό των παραμέτρων του προτύπου ψηφιακής αναπαράστασης video MPEG-4.
Τα action units που έχουν οριστεί φαίνονται παρακάτω:

| **AU** | **Περιγραφή** | **Μύες προσώπου** |
|---|---|---|
| 1 | Inner Brow Raiser | Frontalis, pars medialis |
| 2 | Outer Brow Raiser | Frontalis, pars lateralis |
| 4 | Brow Lowerer | Corrugator supercilii, Depressor supercilii |
| 5 | Upper Lid Raiser | Levator palpebrae superioris |
| 6 | Cheek Raiser | Orbicularis oculi, pars orbitalis |
| 7 | Lid Tightener | Orbicularis oculi, pars palpebralis |
| 9 | Nose Wrinkler | Levator labii superioris alaquae nasi |
| 10 | Upper Lip Raiser | Levator labii superioris |
| 11 | Nasolabial Deepener | Levator anguli oris (a.k.a. Caninus) |
| 12 | Lip Corner Puller | Zygomaticus major |
| 13 | Cheek Puffer | Zygomaticus minor |
| 14 | Dimpler | Buccinator |
| 15 | Lip Corner Depressor | Depressor anguli oris (a.k.a. Triangularis) |

| | | |
|---|---|---|
| 16 | Lower Lip Depressor | Depressor labii inferioris |
| 17 | Chin Raiser | Mentalis |
| 18 | Lip Puckerer | Incisivii labii superioris and Incisivii labii inferioris |
| 20 | Lip stretcher | Risorius w/ platysma |
| 22 | Lip Funneler | Orbicularis oris |
| 23 | Lip Tightener | Orbicularis oris |
| 24 | Lip Pressor | Orbicularis oris |
| 25 | Lips part | Depressor labii inferioris or relaxation of Mentalis, or Orbicularis oris |
| 26 | Jaw Drop | Masseter, relaxed Temporalis and internal Pterygoid |
| 27 | Mouth Stretch | Pterygoids, Digastric |
| 28 | Lip Suck | Orbicularis oris |
| 41 | Lid droop | Relaxation of Levator palpebrae superioris |
| 42 | Slit | Orbicularis oculi |
| 43 | Eyes Closed | Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis |
| 44 | Squint | Orbicularis oculi, pars palpebralis |
| 45 | Blink | Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis |
| 46 | Wink | Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis |
| 51 | Head turn left | |
| 52 | Head turn right | |
| 53 | Head up | |
| 54 | Head down | |
| 55 | Head tilt left | |
| 56 | Head tilt right | |
| 57 | Head forward | |
| 58 | Head back | |
| 61 | Eyes turn left | |
| 62 | Eyes turn right | |
| 63 | Eyes up | |
| 64 | Eyes down | |

*Πίνακας 4.1.1: Ορισμός των action units.*

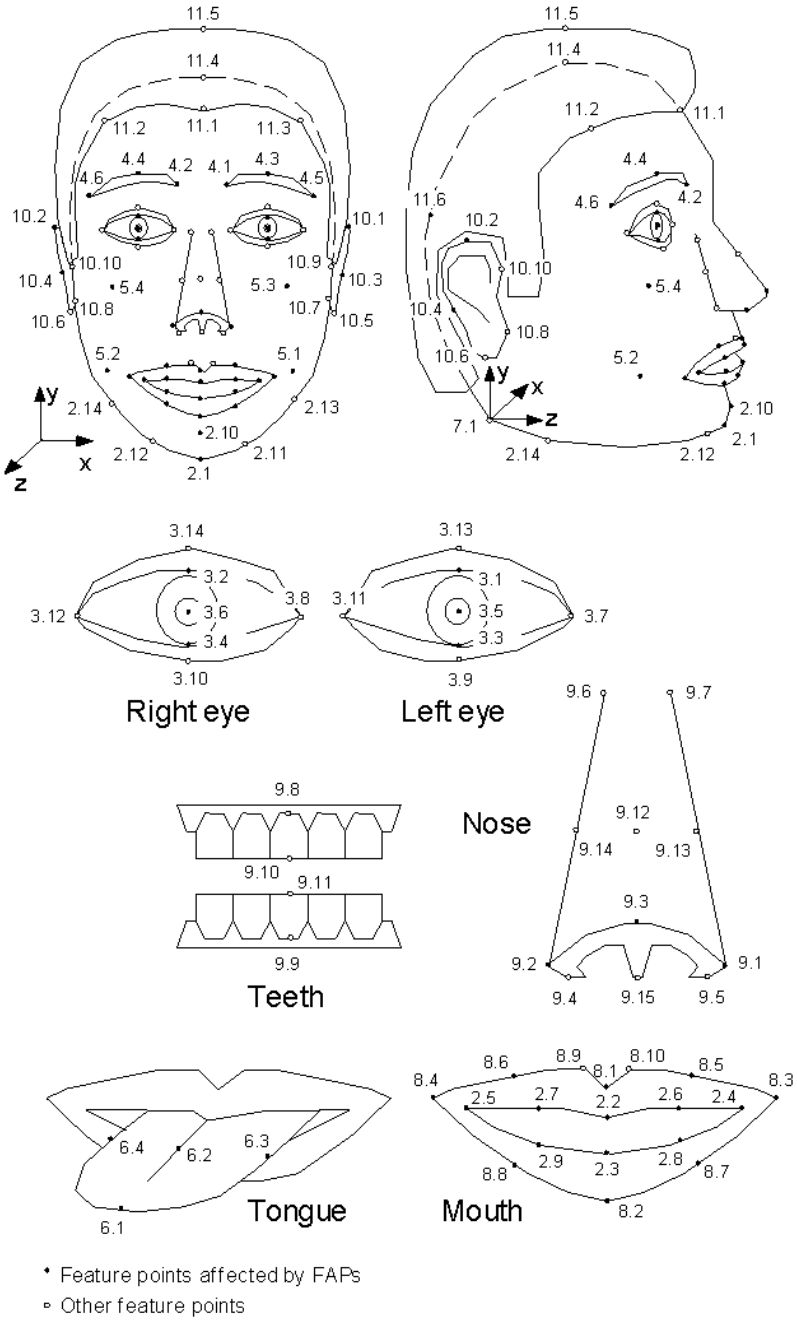
## Εκφράσεις προσώπου και MPEG-4

Στην πρώτη έκδοση του προτύπου MPEG-4 αναπτύχθηκε ένας τρόπος αναπαράστασης και ανάλυσης τοο ανθρώπινου προσώπου (στη συνέχεια, το πρότυπο επεκτάθηκε και για το σώμα, καθώς και για άλλα αντικείμενα). Στο πρότυπο αυτό χρησιμοποιείται ένα σύνολο παραμέτρων για τον προσδιορισμό του σχήματος, του μεγέθους και της υφής του προσώπου, γνωστό ως FDP – Facial Definition Parameters, σε συνδυασμό με ένα ακόμη σύνολο παραμέτρων, που προσδιορίζουν την κίνηση του προσώπου, οι οποίες ονομάζονται FAPs (Facial Animation Parameters) ([16]). Με τη βοήθεια των FDP μπορούμε να προσδιορίσουμε με ακρίβεια την κίνηση συγκεκριμένων χαρακτηριστικών του προσώπου, ενώ με το FAPs μπορούμε να προσδιορίσουμε εκφράσεις και εκφορά λόγου στα διάφορα μοντέλα προσώπων.

Τα FDP περιλαμβάνουν διάφορα πεδία:

- Τα **FeaturePointsCoord**, που είναι τρισδιάστατα χαρακτηριστικά σημεία του προσώπου, τα οποία χρησιμοποιούνται για τη ζυγοστάθμιση του μοντέλου του προσώπου.
- Τα **TextureCoords**, τα οποία αφορούν συντεταγμένες της υφής για τα χαρακτηριστικά σημεία.
- Τα **TextureType**, που περιλαμβάνουν πληροφορίες για τον αποκωδικοποιητή σχετικές με τον τύπο της εικόνας της υφής.
- Τα **FaceDefTables**, που περιγράφουν τη συμπεριφορά των FAPs.
- Το **FaceSceneGraph**, που περιέχει την εικόνα της υφής, ή πληροφορίες για την ιεραρχία του μοντέλου.

Τα Feuter Points (FP) των FDP φαίνονται στην παρακάτω εικόνα:




*Σχήμα 4.1.2: Ο ορισμός των FDP στο πρότυπο MPEG-4.*

Αντίθετα, τα FAPs βασίζονται στη μελέτη των ελάχιστων κινήσεων του προσώπου, και συνδέονται στενά με τις κινήσεις των μυών. Αντιπροσωπεύουν ένα ολοκληρωμένο σύνολο βασικών ενεργειών του προσώπου, επιτρέποντας την απεικόνιση της πλειοψηφίας των φυσιολογικών ανθρώπινων εκφράσεων, ενώ οι υπερβολικές τιμές τους μας επιτρέπουν να ορίσουμε ενέργειες αδύνατες για έναν άνθρωπο, αλλά απαραίτητες, π.χ. στους χαρακτήρες cartoon. Όλες οι παράμετροι που σχετίζονται με μεταφορική κίνηση εκφράζονται μέσω των μονάδων κίνησης των χαρακτηριστικών του προσώπου (FAPU – Facial Animation Parameter Units).

Τα κυριότερα FAPs, όπως έχουν οριστεί στο πρότυπο MPEG-4, περιλαμβάνονται παρακάτω:

| **FAP name** | **Feature for the description** | **Utilized feature** |
|---|---|---|
| Squeeze_l_eyebrow ($F_{37}$) | $D_1 = s(4.5,3.11)$ | $f_1 = D_{1-NEUTRAL} - D_1$ |
| Squeeze_r_eyebrow ($F_{37}$) | $D_2 = s(4.6,3.8)$ | $f_2 = D_{2-NEUTRAL} - D_2$ |
| Lower_t_midlip ($F_4$) | $D_3 = s(9.3,8.1)$ | $f_3 = D_3 - D_{3-NEUTRAL}$ |
| Raise_b_midlip ($F_5$) | $D_4 = s(9.3,8.2)$ | $f_4 = D_{4-NEUTRAL} - D_4$ |
| Raise_l_*I*_eyebrown ($F_{31}$) | $D_5 = s(4.1,3.11)$ | $f_5 = D_5 - D_{5-NEUTRAL}$ |
| Raise_r_*I*_eyebrown ($F_{32}$) | $D_6 = s(4.2,3.8)$ | $f_6 = D_6 - D_{6-NEUTRAL}$ |
| Raise_l_o_eyebrown ($F_{35}$) | $D_7 = s(4.5,3.7)$ | $f_7 = D_7 - D_{7-NEUTRAL}$ |
| Raise_r_o_eyebrown ($F_{36}$) | $D_8 = s(4.6,3.12)$ | $f_8 = D_8 - D_{8-NEUTRAL}$ |
| Raise_l_m_eyebrown ($F_{33}$) | $D_9 = s(4.3,3.7)$ | $f_9 = D_9 - D_{9-NEUTRAL}$ |
| Raise_r_m_eyebrown ($F_{34}$) | $D_{10} = s(4.4,3.12)$ | $f_{10} = D_{10} - D_{10-NEUTRAL}$ |
| Open_jaw ($F_3$) | $D_{11} = s(8.1,8.2)$ | $f_{11} = D_{11} - D_{11-NEUTRAL}$ |
| Close_t_l_eyelid ($F_{19}$) – Close_b_l_eyelid ($F_{21}$) | $D_{12} = s(3.1,3.3)$ | $f_{12} = D_{12} - D_{12-NEUTRAL}$ |
| Close_t_r_eyelid ($F_{20}$) – Close_b_r_eyelid ($F_{22}$) | $D_{13} = s(3.2,3.4)$ | $f_{13} = D_{13} - D_{13-NEUTRAL}$ |
| Stretch_l_cornerlid ($F_6$) (Stretch_l_cornerlid_o) ($F_{13}$)- Stretch_r_cornerlid ($F_3$) (Stretch_r_cornerlid_o) ($F_{54}$) | $D_{14} = s(8.4,8.3)$ | $f_{14} = D_{14} - D_{14-NEUTRAL}$ |
| Squeeze_l_eyebrow ($F_{37}$) AND Squeeze_r_eyebrow ($F_{38}$) | $D_{15} = s(4.6,4.5)$ | $f_{15} = D_{15-NEUTRAL} - D_{15}$ |

*Πίνακας 4.1.2: Ορισμός και υπολογισμός μερικών χαρακτηριστικών FAPs.*

Στο παραπάνω πίνακα, περιλαμβάνονται μερικά χαρακτηριστικά FAPs που σχετίζονται κυρίως με τις περιοχές του στόματος και των ματιών, οι οποίες έχουν ιδιαίτερο ενδιαφέρον στην αναγνώριση συναισθήματος. Εδώ, τα $s(x, y)$ αποτελούν αποστάσεις μεταξύ των σημείων $x$ και $y$, όπου τα $x$ και $y$ είναι χαρακτηριστικά σημεία του προσώπου (FP), τα οποία αναφέρθηκαν προηγομένως. Μέσω των αποστάσεων αυτών μετράμε την τιμή του κάθε FAP – συνήθως τελικά τα FAPs, ανάλογα με τις αποστάσεις αυτές παίρνουν τις τιμές High, Medium και Low ([15], [17]). Τέλος, στην Τρίτη στήλη, τα $D_{i-NEUTRAL}$ υποδηλώνουν τις αποστάσεις του προσώπου όταν αυτό βρίσκεται σε ουδέτερη συναισθηματική – εκφραστική κατάσταση. Τέλος, πρέπει να αναφέρουμε ότι υπάρχει στενή σχέση μεταξύ των FAPs και τω μονάδων δράσης (action units) του βασικού συστήματος κωδικοποίησης δράσης προσώπου (FACS), που αναφέρθηκε προηγουμένως.

## FAPs και αναγνώριση συναισθήματος

Η χρήση των FAPs είναι ένα σημαντικό κομμάτι στην αναγνώριση συναισθήματος ([15]). Οι βασικές τεχνικές αναγνώρισης συναισθηματικής κατάστασης του χρήστη

σε συστήματα αλληλεπίδρασης ανθρώπου – μηχανής, χρησιμοποιούν μεθόδους κατάτμησης και ανίχνευσης αντικειμένων, με στόχο την ανίχνευση των FPs και στη συνέχεια την εξαγωγή των FAPs. Επιπλέον, έχουν αναπτυχθεί κάποια σύνολα κανόνων για τη συσχέτιση μεταξύ FAPs και συναισθήματος. Οι κανόνες αυτοί δεν είναι ίδιοι για κάθε χρήστη, αλλά μπορούν να χρησιμοποιηθούν ως μία καλή βάση γνώσης, και στη συνέχεια να προσαρμοστούν – επεκταθούν κατάλληλα. Ένας ενδεικτικός πίνακας των κανόνων αυτών για τα βασικά συναισθήματα φαίνεται παρακάτω. Τέλος, μία πλήρης λίστα κανόνων μπορεί να βρεθεί στην εργασία [17].

| Κανόνας | FAPs | Τεταρτημόριο |
|---|---|---|
| 2 | F3_M+F4_L+F5_L+[F53+F54]_H +[F19+F21]_H+[F20+F22]_H | (+,+) |
| 7 | F3_L+F4_L+F5_H+[F53+F54]_H+ [F19+F21]_H+[F20+F22]_H+[F37 +F38]_M+F59_H+F60_H | (+,+) |
| 13 | F3_L+F4_M+F5_H+F31_L+F32_L +F33_L+F34_L+F37_H+F38_H+ F59_M+F60_M | (-,+) |
| 16 | F3_L+F4_M+F5_L+F31_L+F32_L +F33_L+F34_L+F37_H+F38_H+ F59_M+F60_M | (-,+) |
| 26 | F3_M+F5_L+[F19+F21]_H+[F20+ F22]_H+F31_H+F32_H+F33_M+ F34_M+F35_M+F36_M+[F37+F3 8]_H | (-,-) |
| 34 | F3_L+F4_L+[F53+F54]_L+F31_M +F32_M+F33_M+F34_M+F35_M +F36_M+F37_M+F38_M | (-,-) |
| 41 | F3_L+F4_M+F31_M+F32_M+F33 _M+F34_M+F35_M+F36_M+F37 _M+F38_M+ F59_M+F60_M | Neutral |

*Πίνακας 4.1.3: Ένα σύνολο κανόνων για ανάλυση συναισθήματος από FAPs.*

# 5 Πειραματικά αποτελέσματα

Στο κεφάλαιο αυτό θέλουμε να χρησιμοποιήσουμε τις active μεθόδους μηχανικής μάθησης που περιγράψαμε στο κεφάλαιο 3, και να δείξουμε με ποιο τρόπο μπορούν να βελτιώσουν την ακρίβεια ενός αλγορίθμου ανάλυσης συναισθήματος από εικόνες, με στόχο την αλληλεπίδραση ανθρώπου – μηχανής.

## 5.1 Οι βάσεις δεδομένων

Για το πειραματικό μέρος της διπλωματικής χρησιμοποιήθηκαν τρεις βάσεις δεδομένων ανάλυσης συναισθήματος από εικόνα.

- Η πρώτη είναι η naturalistic βάση δεδομένων του πανεπιστημίου Queens του Belfast (QUB), που δημιουργήθηκε κατά τη διάρκεια του προγράμματος EC FP5 IST ΕΡΜΗΣ, και επεκτάθηκε περαιτέρω στο EC FP6 IST Humaine Δίκτυο Αριστείας (Network of Excellence), και στο πρόγραμμα EC FP7 IST Semaine.

- Η δεύτερη βάση έχει δημιουργηθεί στο Εργαστήριο Συστημάτων Εικόνων, Βίντεο και Πολυμέσων (IVML) του ΕΜΠ, από τον ερευνητή Κώστα Καρπούζη, ο οποίος είναι ειδικός σε θέματα αλληλεπίδρασης ανθρώπου – μηχανής.

- Η τρίτη βάση δεδομένων προέρχεται από τον διαγωνισμό FER (Facial Expression Recognition) του 2013, του πανεπιστημίου του Montreal στον Καναδά.

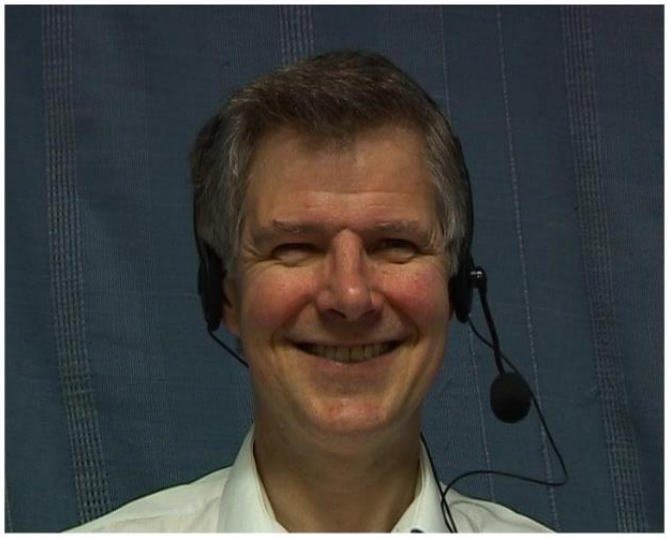
Ας δούμε καταρχήν αναλυτικά τις παραπάνω βάσεις:

### Τεκμηρίωση των βάσεων

**Πρώτη βάση:** Αυτή η βάση δεδομένων αποτελείται από ένα βίντεο διάρκειας περίπου μίας ώρας, στο οποίο ο καθηγητής του πανεπιστημίου Queens του Belfast, Dr. Roddie Cowie, παίρνει μέρος σε μία τηλεδιάσκεψη, κατά την οποία το πρόσωπό του και οι εκφράσεις του φαίνονται ευκρινώς, σε σταθερές συνθήκες φωτισμού. Ο Dr. Roddie Cowie είναι ένας εμπειρογνώμονας στο πεδίο της ανάλυσης συναισθήματος, οπότε οι εκφράσεις στο βίντεο αυτό έχουν επιλεγεί προσεκτικά έτσι ώστε να είναι φυσιολογικές, και να ανταποκρίνονται στην πραγματικότητα. Από τη βάση αυτή, ερευνητές στο εργαστήριο Ψηφιακής Επεξεργασίας Σήματος του ΕΜΠ επέλεξαν ένα σημαντικό αριθμό από αντιπροσωπευτικά καρέ απ’ το βίντεο αυτό, και τα τεκμηρίωσαν, αξιολογώντας τόσο τις παραμέτρους κίνησης (FAPs) του προσώπου, όσο και την συναισθηματική κατάσταση του χρήστη σε αυτά. Τα επιλεγμένα καρέ ήταν τα πιο εκφραστικά ανάμεσα στα καρέ του βίντεο, σύμφωνα με τη γνώμη των ειδικών. Επίσης, μέσω μίας προσέγγισης ομοιότητας των FAPs,

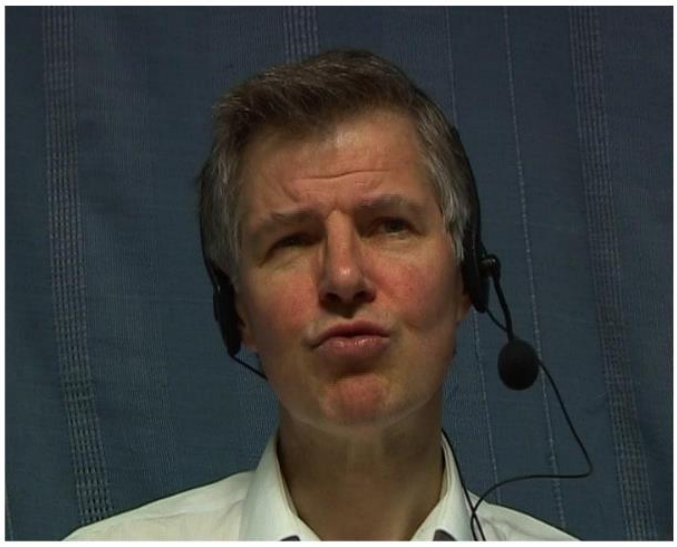
μπορούμε να επεκτείνουμε τα καρέ αυτά σε έναν ευρύτερο σύνολο δεδομένων, έτσι ώστε να πάρουμε ένα πολύ μεγαλύτερο σύνολο εκπαίδευσης. Η επέκταση αυτή βασίζεται στο σύστημα εξαγωγής χαρακτηριστικών του προσώπου του Εργαστηρίου Συστημάτων Εικόνων, Βίντεο και Πολυμέσων (IVML) του ΕΜΠ.
Μερικά ενδεικτικά καρέ της βάσης αυτής, τα οποία αντιστοιχούν σε κάθε συναισθηματική κατάσταση, είναι τα παρακάτω:

*(α)*



*(β)*

*(γ)*



*(δ)*

*(ε)*

*Σχήμα 5.1.1: Μερικά παραδείγματα εικόνων της πρώτης βάσης δεδομένων που χρησιμοποιήθηκε. Τα συναισθήματα είναι: (α) – ουδέτερο, (β) – (+,+), (γ) – (+,-), (δ) – (-,+), (ε) – (-,-).*

**Δεύτερη βάση:** Η βάση αυτή δημιουργήθηκε από το Εργαστήριο Συστημάτων Εικόνων, Βίντεο και Πολυμέσων (IVML) του ΕΜΠ. Στη βάση αυτή απεικονίζεται ο ερευνητής του εργαστηρίου Κώστας Καρπούζης, ο οποίος μετέχει σε μία σύντομη τηλεδιάσκεψη. Υπάρχουν περίπου 2200 καρέ διαθέσιμα, τα σημαντικότερα εκ των οποίων έχουν αξιολογηθεί επίσης ως προς τα FAPs και το συναίσθημα από ειδικούς του εργαστηρίου.
Μερικές χαρακτηριστικές εικόνες φαίνονται παρακάτω:

*(α)*

*(β)*

*(γ)*

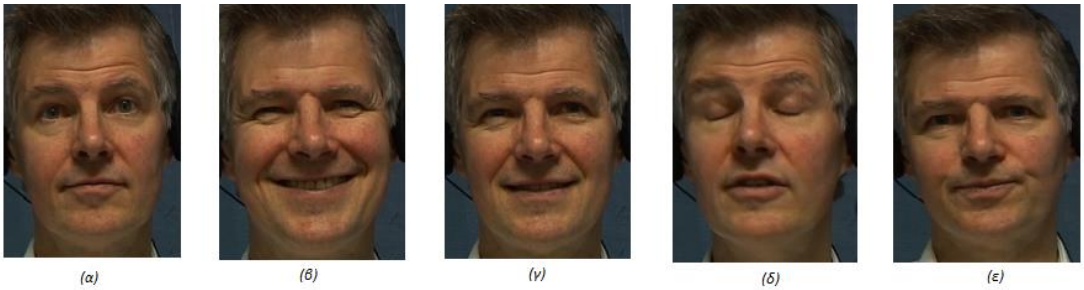
*(δ)*

*Σχήμα 5.1.2: Μερικά παραδείγματα εικόνων της δεύτερης βάσης δεδομένων που χρησιμοποιήθηκε. Τα συναισθήματα είναι: (α) – ουδέτερο, (β) – (+,-), (γ) – (-,+), (δ) – (-,-).*

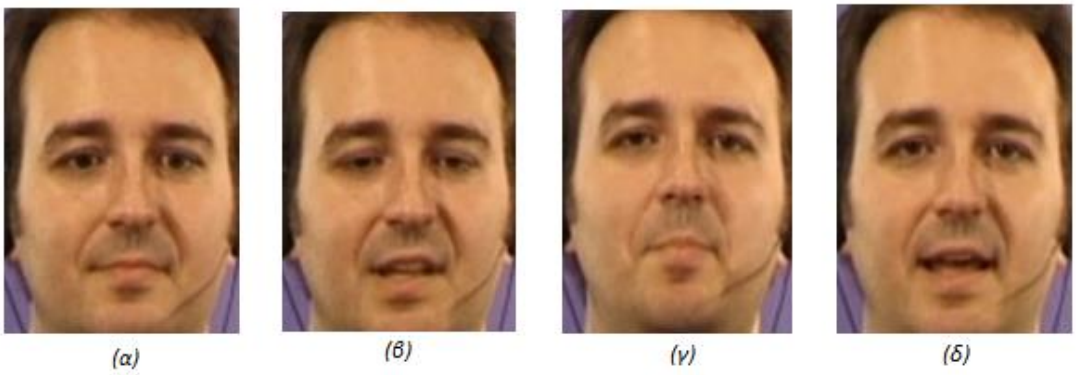
**Τρίτη βάση**: Η βάση αυτή προέρχεται από το διαγωνισμό FER (Facial Expression Recognition) 2013 του πανεπιστημίου Μόντρεαλ στον Καναδά. Περιλαμβάνει περίπου 35.000 ασπρόμαυρα καρέ προσώπων, μικρής διάστασης (48x48), στα οποία έχει γίνει εκτίμηση του συναισθήματος σε έξι κατηγορίες: angry, happy, sad, disquist, surprise, και neutral. Οι κατηγορίες αυτές αντιστοιχήθηκαν στη μορφή activation / valence με τη βοήθεια του κύκλου της Wiessel, ώστε να έχουμε συμβατότητα με τα προηγούμενα. Δυστυχώς, τα FAPs των προσώπων δεν είναι διαθέσιμα.
Μερικές χαρακτηριστικές εικόνες είναι οι παρακάτω:


*Σχήμα 5.1.3: Μερικά παραδείγματα εικόνων της τρίτης βάσης δεδομένων που χρησιμοποιήθηκε. Τα συναισθήματα είναι: (α) – ουδέτερο, (β) – (+,+), (γ) – (-,+), (δ) – (-,-).*

Πριν προχωρήσουμε στα επόμενα, είναι αναγκαίο, τόσο για την ανίχνευση των μεταβολών, όσο και για την αναγνώριση του συναισθήματος, να απομονώσουμε το πρόσωπο του χρήστη, που είναι αυτό που μας ενδιαφέρει, από το υπόλοιπο περιβάλλον. Αυτό έγινε με τη βοήθεια των συναρτήσεων της βιβλιοθήκης της OpenCV. Μερικά ενδεικτικά αποτελέσματα είναι τα παρακάτω:




*Σχήμα 5.1.4: Οι περικομμένες εικόνες του σχήματος 5.1.1.*




*Σχήμα 5.1.5: Οι περικομμένες εικόνες του σχήματος 5.1.2.*

Η βάση FER είναι έτοιμη, και δεν χρειάζεται περαιτέρω επεξεργασία.

## 5.2 Ανιχνεύοντας τις μεταβολές

Ο στόχος αυτής της εργασίας είναι η χρήση μεθόδων της μάθησης σε μεταβαλλόμενα περιβάλλοντα, έτσι ώστε ένας ταξινομητής να μπορεί να εντοπίζει τις μεταβολές στα δεδομένα που λαμβάνει, και να προσαρμόζεται σε αυτές. Εδώ, ο ταξινομητής μας είναι ένας αναλυτής συναισθήματος. Συνεπώς, στην ενότητα αυτή θα θεωρήσουμε ότι ο ταξινομητής μας έχει εκπαιδευτεί σε ένα Α σύνολο δεδομένων, και ο στόχος είναι να αναγνωρίσουμε τις χρονικές στιγμές όπου η

κατανομή των δεδομένων στην είσοδο άλλαξε, δηλ. το σύνολο των δεδομένων μεταβλήθηκε σε ένα σύνολο Β, στατιστικά διαφορετικό απ' το Α.
Για να το ελέγξουμε αυτό, θα φτιάξουμε δύο ομάδες δεδομένων, οι οποίες θα προέρχονται είτε από διαφορετική βάση δεδομένων η κάθε μία, είτε θα τροποποιήσουμε τα δεδομένα μίας βάσης, και γενικά θα κάνουμε αρκετές αλλαγές στα δεδομένα, και θα δούμε αν οι change detector της προηγούμενης ενότητας θα είναι σε θέση να τις αναγνωρίσουν.

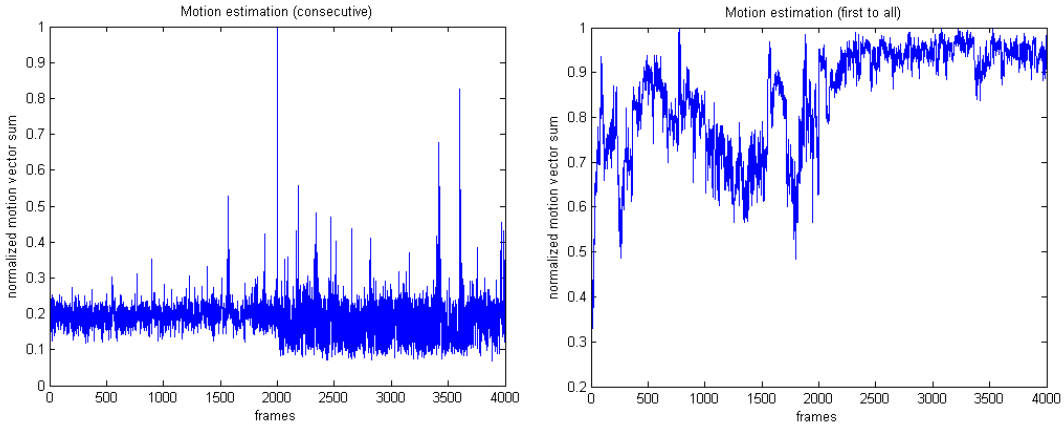
Ο τρόπος που θα σκεφτόταν κανείς εκ πρώτης όψεως θα ήταν να θεωρήσουμε κάθε $N\text{x}M$ εικόνα ως ένα σύνολο μεταβλητών (μία για κάθε pixel), και άρα να θεωρήσουμε τις εικόνες ως τα αποτελέσματα μίας πολυδιάστατης κατανομής $N\text{x}M$ μεταβλητών.
Κάτι τέτοιο όμως θα ήταν καταστροφικό, λόγω της κατάρας της διαστατικότητας (curse of dimensionality): ακόμη και αν σμικρύνουμε δραματικά τις εικόνες (π.χ. σε μία διάσταση 25x25), ο αριθμός των μεταβλητών είναι και πάλι τεράστιος, και άρα θα χρειαζόμασταν τεράστιο όγκο δεδομένων για να ανιχνεύσουμε τις μεταβολές αξιόπιστα.

Για το λόγο αυτό, στην εργασία αυτή δουλέψαμε με έναν άλλο τρόπο. Αντί να κάνουμε την ανίχνευση απευθείας πάνω στα frames, εξήχθησαν από αυτά κάποια μονοδιάστατα σήματα, τα οποία ανιχνεύσαμε στη συνέχεια για μεταβολές. Τα δύο πρώτα σήματα ήταν η μέση τιμή και η διακύμανση των pixel του κάθε καρέ. Εκτός απ' αυτά, κάναμε ανίχνευση κίνησης (motion estimation) ανάμεσα στις εικόνες, και υπολογίσαμε το άθροισμα των διανυσμάτων κίνησης (motion vectors). Στη συνέχεια, το μέτρο αυτού του αθροίσματος το θεωρήσαμε ως μονοδιάστατο σήμα, και το χρησιμοποιήσαμε για την ανίχνευση μεταβολών. Η εκτίμηση κίνησης έγινε με δύο τρόπους: είτε ανάμεσα σε διαδοχικά frames (consecutive), είτε ανάμεσα στο αρχικό frame και σε όλα τα άλλα (first to all). Οπότε, χρησιμοποιήσαμε συνολικά τέσσερα σήματα για την ανίχνευση.
Τέλος, για να αποφύγουμε το φαινόμενο των εσφαλμένων θετικών ανιχνεύσεων, θεωρήσαμε ότι έχουμε μία μεταβολή μόνο όταν τουλάχιστον τρεις απ' τις τέσσερις ανιχνεύσεις ανήκουν σε ένα μικρό παράθυρο (περίπου 100 καρέ) – περισσότερες λεπτομέρειες γι' αυτό θα δούμε παρακάτω. Οι δοκιμές που κάναμε και τα αποτελέσματα που πήραμε φαίνονται παρακάτω.

- **Πείραμα 1**: Εδώ χρησιμοποιήσαμε 2000 διαδοχικά frames της πρώτης βάσης, τα οποία ακολουθούνται από ισάριθμα frames της δεύτερης βάσης, όπως φαίνεται παρακάτω:

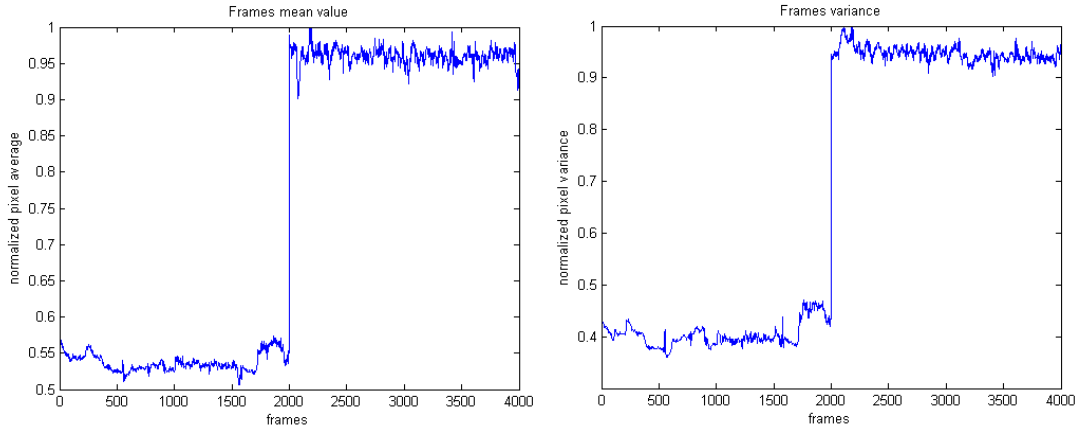

*Σχήμα 5.2.1: Το dataset του πρώτου πειράματος.*

Τα αποτελέσματα της ανίχνευσης κίνησης ήταν τα παρακάτω:



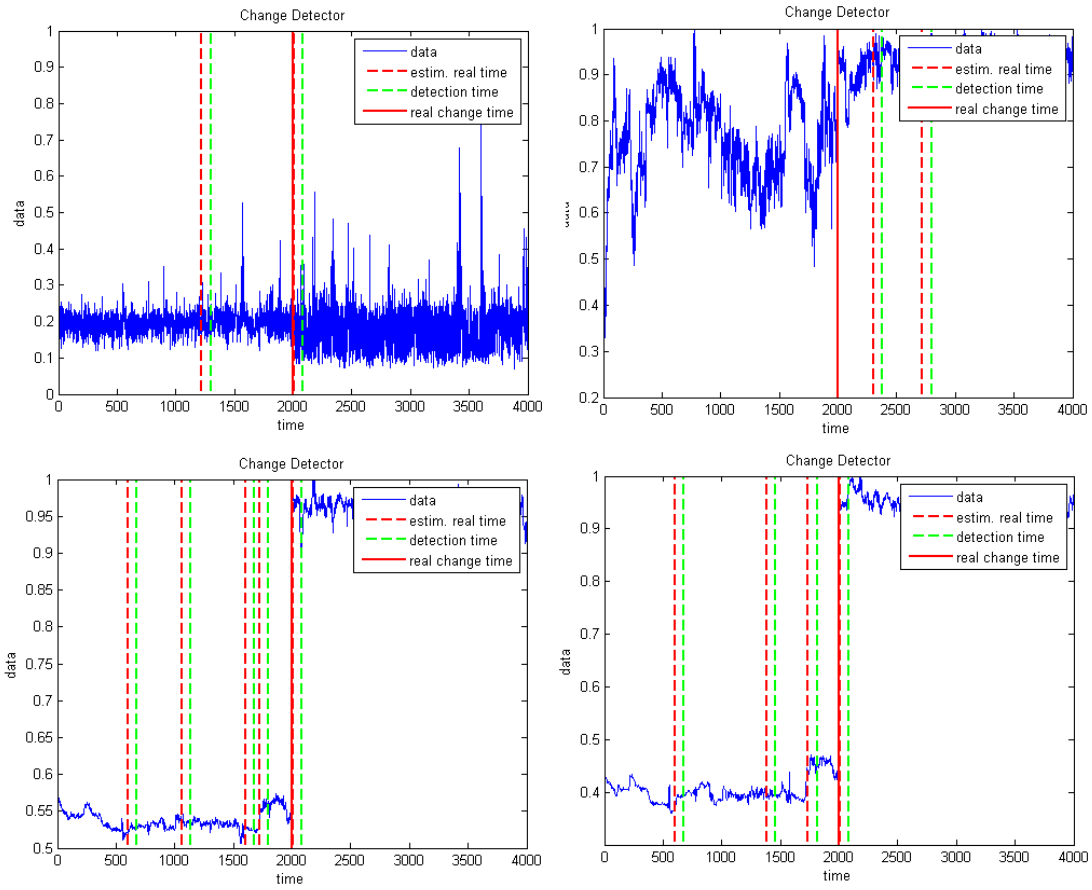

*Σχήμα 5.2.2: Σήματα ανίχνευση κίνησης 1ου πειράματος (διαδοχικά και 1ου καρέ με τα υπόλοιπα, αντίστοιχα).*

Επίσης, τα (κανονικοποιημένα) σήματα της μέσης τιμής και της διακύμανσης των frames ήταν τα ακόλουθα:




*Σχήμα 5.2.3: Σήματα μέσης τιμής και διακύμανσης του 1ου πειράματος.*

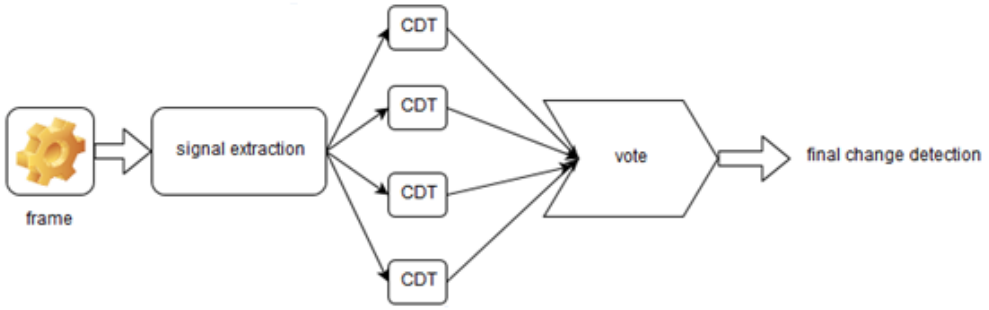
Ως ανιχνευτή μεταβολών στο κεφάλαιο αυτό χρησιμοποιήσαμε τον H – ICI CDT, λόγω της απλότητας και της ταχύτητάς του. Τα αποτελέσματα ανίχνευσης που έδωσε για το κάθε σήμα ήταν τα εξής:




*Σχήμα 5.2.4: Ανίχνευση μεταβολών για το κάθε σήμα του 1ου πειράματος.*

Όπως βλέπουμε, όλα τα σήματα εντοπίζουν την πραγματική μεταβολή των δεδομένων, που βρίσκεται στη θέση $t = 2000$ (το 2ο σήμα την εντοπίζει λίγο αργότερα). Παρόλα αυτά, στο κάθε σήμα εντοπίζονται και ψευδείς θετικές ανιχνεύσεις, πράγμα που είναι λογικό, αφού το κάθε σήμα περιλαμβάνει μόνο μέρος της πληροφορίας των δεδομένων. Παρόλο που οι εσφαλμένες θετικές ανιχνεύσεις θεωρητικά δεν αποτελούν μεγάλο πρόβλημα, θα ήταν επιθυμητό να τις αποφύγουμε. Για το σκοπό αυτό, στην ενότητα αυτή κατασκευάσαμε μία ομάδα (ensemble) ανιχνευτών κίνησης. Συγκεκριμένα, απ' το σήμα εικόνας εξάγονται τα 4 μονοδιάστατα σήματα που περιγράψαμε παραπάνω, και το καθένα απ' αυτά ανιχνεύεται για μεταβολές από έναν ICI – CDT. Στη συνέχεια, οι ανιχνευτές ψηφίζουν: εάν γίνουν τουλάχιστον τρεις ανιχνεύσεις (πλειοψηφία) μέσα σε ένα μικρό παράθυρο δεδομένων (περίπου 100 frames), τότε θεωρούμε ότι η ανίχνευση είναι πραγματική, και επιστρέφουμε τον μικρότερο απ' τους 4 εκτιμώμενους χρόνους πραγματικής μεταβολής των ανιχνευτών. Οπότε, όταν ένα σήμα ανιχνεύσει μεταβολή, ο αλγόριθμος περιμένει ακόμα περίπου 100 frames, και αν μέσα σε αυτά συμβούν τουλάχιστον άλλες δύο μεταβολές, τότε έχουμε ανίχνευση. Διαφορετικά, αν δεν εντοπιστεί πλειοψηφία, ο αλγόριθμος εφαρμόζει ένα τεστ υποθέσεως στα υπόλοιπα σήματα, στα σημεία που εντοπίστηκαν στο παράθυρο (παρόλο που σε κάποιο άλλο σήμα ο CDT μπορεί να μην έχει ανιχνεύσει κάποια μεταβολή, μπορεί παρόλα αυτά η μεταβολή αυτή να υπάρχει και απλώς να αναγνωριστεί αργότερα απ' τον CDT του σήματος). Αν με αυτό τον τρόπο εντοπίσουμε μεταβολές σε τόσα σήματα ώστε να έχουμε πλειοψηφία, τότε εμφανίζουμε και πάλι μία μεταβολή.

Συνολικά, η αρχιτεκτονική του συστήματος αυτού φαίνεται παρακάτω:



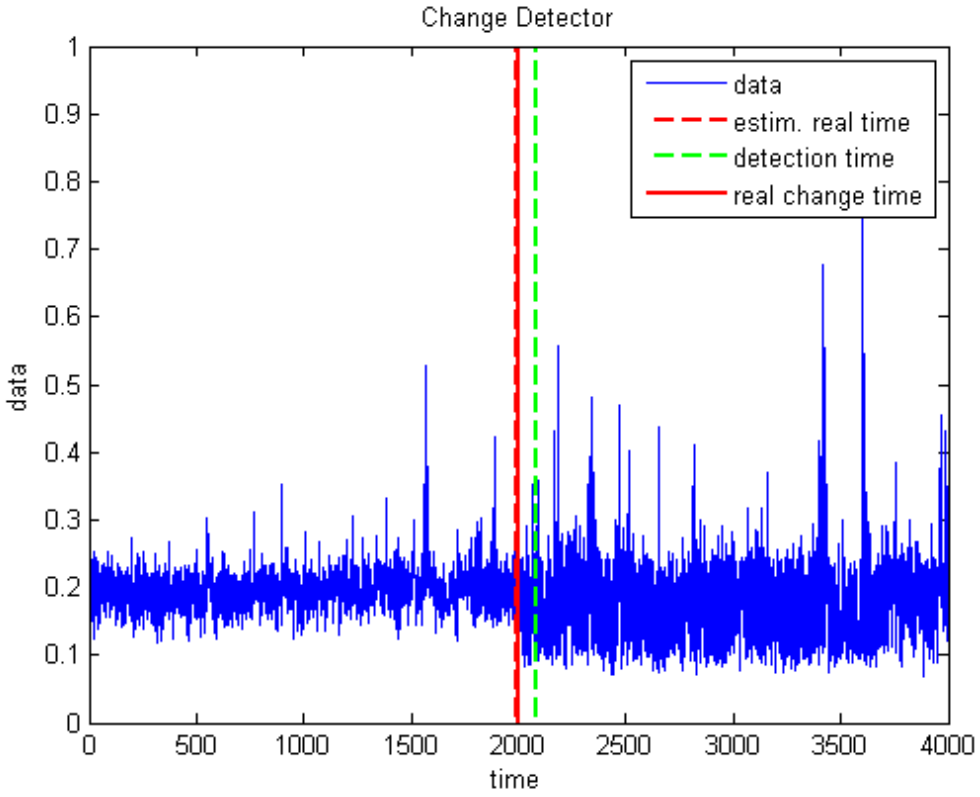

*Σχήμα 5.2.5: Η αρχιτεκτονική του ανιχνευτή μεταβολών των πειραμάτων.*

Παρόλα αυτά, στην περίπτωση του βαθμιαίου concept drift υπάρχει θεωρητικά η πιθανότητα οι ανιχνεύσεις των διάφορων σημάτων να απέχουν περισσότερο μεταξύ τους, ανάλογα με το ρυθμό στον οποίο εμφανίζεται το drift σε κάθε σήμα. Στην περίπτωση αυτή, ο αλγόριθμος θα έπρεπε να αναγνωρίσει την ύπαρξη του drift και να μεταβάλλει το μήκος του παραθύρου σε κάποιο βαθμό. Στην περίπτωσή μας ακολουθήθηκε η εξής προσέγγιση: κάθε φορά που βρίσκονται κάποιες μεταβολές μέσα στο παράθυρο, αλλά όχι η πλειοψηφία, ελέγχουμε τα υπόλοιπα σήματα με το στατιστικό τεστ για μεταβολές, στα σημεία που εντοπίστηκαν απ' τα άλλα σήματα, όπως προαναφέρθηκε. Αν το τεστ δείξει ότι έχουμε πράγματι μεταβολές, και φτάσουμε σε πλειοψηφία, τότε ανιχνεύουμε μία μεταβολή. Διαφορετικά, αν δεν έχουμε μεταβολές, αλλά το p value των στατιστικών τεστ είναι πολύ χαμηλό, αυτό είναι μία ένδειξη ότι είναι πιθανό να υπάρχει κάποια μεταβολή κοντά στην περιοχή αυτή, οπότε τότε αυξάνουμε λίγο το μήκος του παραθύρου ψήφισης. Αυτός είναι ένας απλός αλλά σχετικά αποτελεσματικός τρόπος για να αντιμετωπίσουμε τις καταστάσεις βαθμιαίου drift.

Επιπλέον, σύμφωνα αρχιτεκτονική των ιεραρχικών ανιχνευτών που παρουσιάστηκε στο κεφάλαιο 3, μπορούμε να προσθέσουμε στο σύστημά μας και ένα επίπεδο επικύρωσης (validation layer). Επίσης, το επίπεδο επικύρωσης θα πρέπει, σε αντίθεση με το σύστημα ανίχνευσης που επεξεργάζεται συγκεκριμένα μονοδιάστατα σήματα, να λαμβάνει υπόψιν τα πλήρη δεδομένα, δηλ. τις εικόνες. Αυτό επιτυγχάνεται με τον εξής τρόπο: αρχικά μικραίνουμε αρκετά την διάσταση των εικόνων, έτσι ώστε να μειώσουμε τη διάσταση των δεδομένων μας, και στη συνέχεια εφαρμόζουμε ανάλυση κύριων συνιστωσών στις εικόνες, ώστε να μειώσουμε ακόμα περισσότερο τον αριθμό των μεταβλητών, οπότε τελικά η κάθε εικόνα αναπαρίσταται με μερικές εκατοντάδες μεταβλητές, σε μορφή μονοδιάστατων διανυσμάτων. Στη συνέχεια, παίρνουμε τις μέσες τιμές αυτών των διανυσμάτων αναπαράστασης σε ένα κυλιόμενο παράθυρο σταθερού μήκους. Λόγω του κεντρικού οριακού θεωρήματος, οι τιμές αυτές ακολουθούν κατά προσέγγιση μία Γκαουσσιανή κατανομή, συνεπώς, όταν ανιχνευτεί μία μεταβολή, μπορούμε να την επικυρώσουμε εφαρμόζοντας ένα τεστ Hotelling. Εδώ, αυτό που χρειάζεται να προσέξουμε είναι ότι μέσω της τεχνικής PCA χάνεται αναπόφευκτα κάποια πληροφορία απ' τα δεδομένα, η οποία είναι τόσο μεγαλύτερη, όσο

λιγότερες κύριες συνιστώσες επιλέξουμε να κρατήσουμε. Επομένως, αν θέλουμε να μπορούμε να ανιχνεύουμε αρκετά μικρές μεταβολές, θα πρέπει να αυξήσουμε τον αριθμό των κύριων συνιστωσών που χρησιμοποιούμε.
Τα παραπάνω ολοκληρώνουν την περιγραφή του συστήματος ανίχνευσης που χρησιμοποιούμε.

Η συνολική ανίχνευση που προκύπτει φαίνεται παρακάτω:



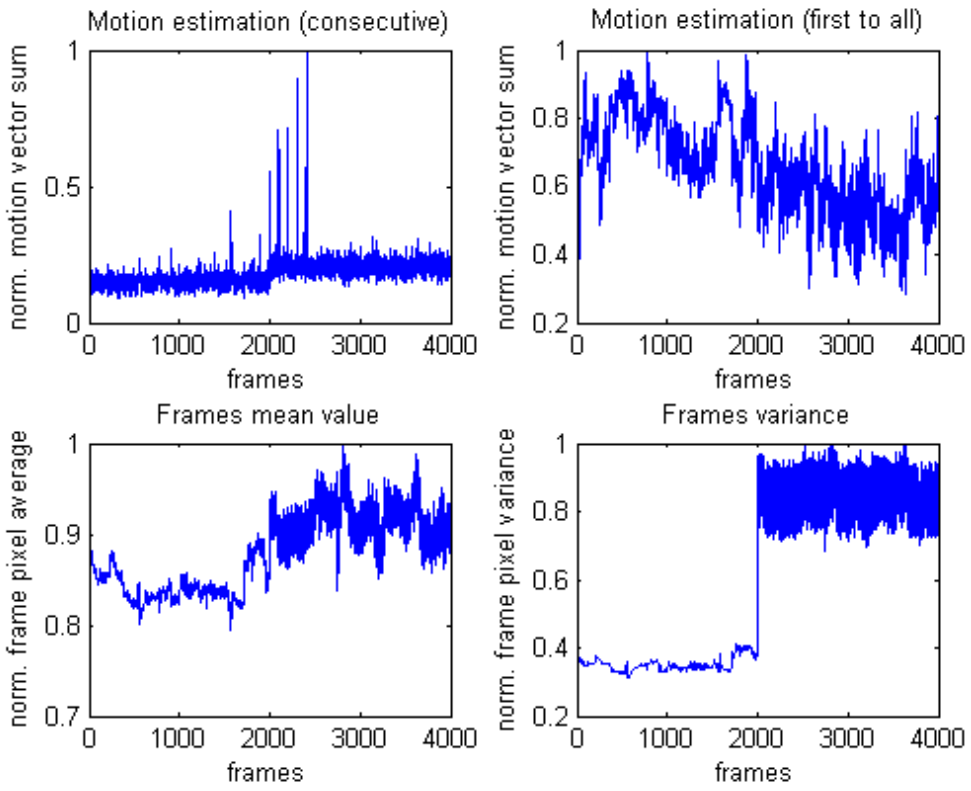

*Σχήμα 5.2.6: Ανίχνευση μεταβολών του 1ου πειράματος.*

Όπως βλέπουμε, ο ανιχνευτής εντόπισε σωστά τη μεταβολή (για λόγους ευκρίνειας, έχουμε απεικονίσει τη μεταβολή πάνω στο πρώτο σήμα).

- **Πείραμα 2**: Εδώ χρησιμοποιήσαμε 2000 διαδοχικά frames της πρώτης βάσης, τα οποία ακολουθούνται από ισάριθμα frames της ίδιας βάσης, στα οποία προσθέσαμε τυχαίο Γκαουσσιανό θόρυβο, όπως φαίνεται παρακάτω:

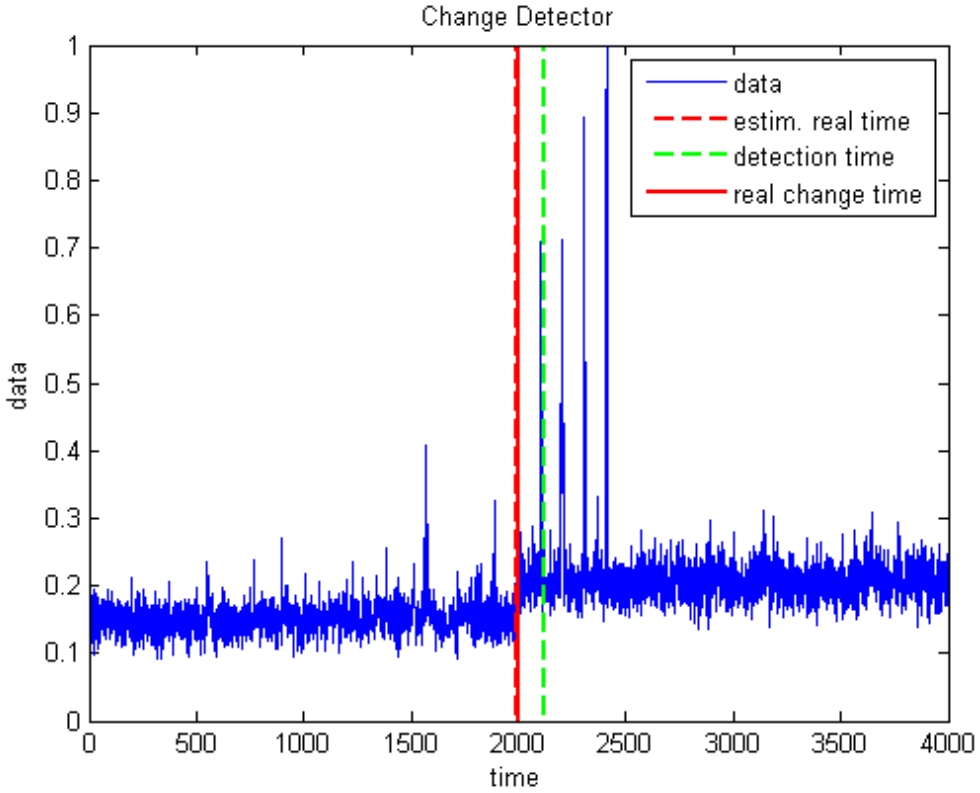

*Σχήμα 5.2.7: Το dataset του δεύτερου πειράματος.*

Τα σήματα του 2ου πειράματος ήταν τα παρακάτω:




*Σχήμα 5.2.8: Τα σήματα του 2ου πειράματος.*

Ο ανιχνευτής μεταβολών έδωσε τα εξής αποτελέσματα:




*Σχήμα 5.2.9: Ανίχνευση μεταβολών του 2ου πειράματος.*

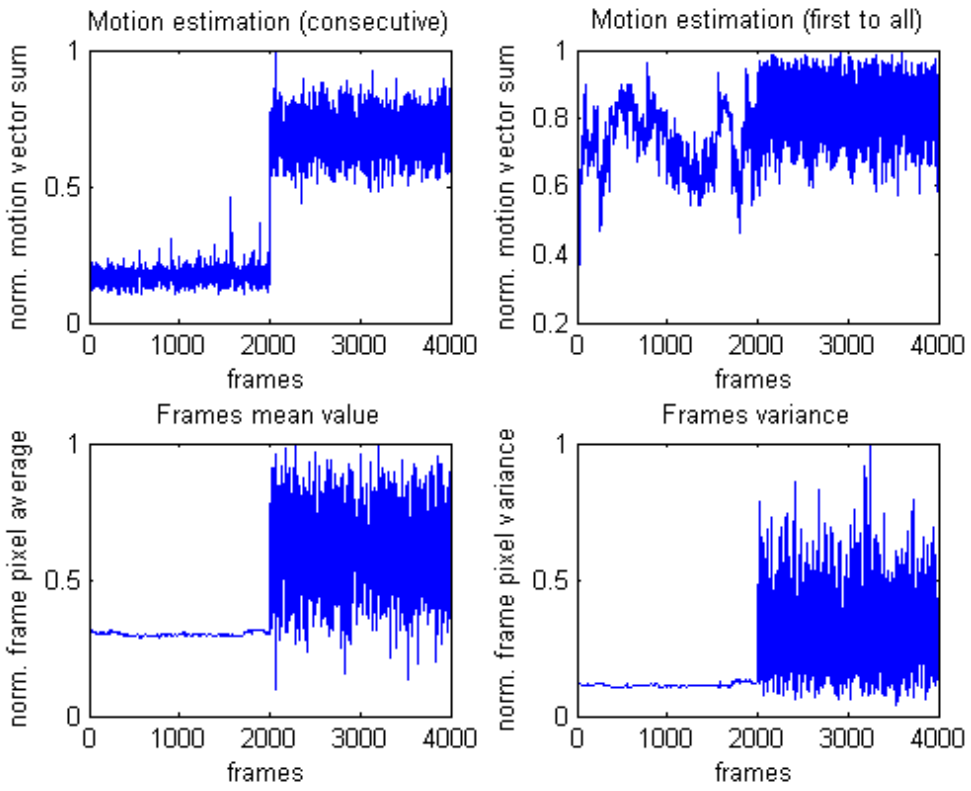
Όπως βλέπουμε, η μεταβολή εντοπίστηκε και πάλι αρκετά καλά.

- **Πείραμα 3**: Εδώ χρησιμοποιήσαμε 2000 διαδοχικά frames της πρώτης βάσης, τα οποία ακολουθούνται από ισάριθμα frames της τρίτης βάσης, τα οποία περιέχουν διαφορετικά πρόσωπα μεταξύ τους:


*Σχήμα 5.2.10: Το dataset του 3ου πειράματος.*

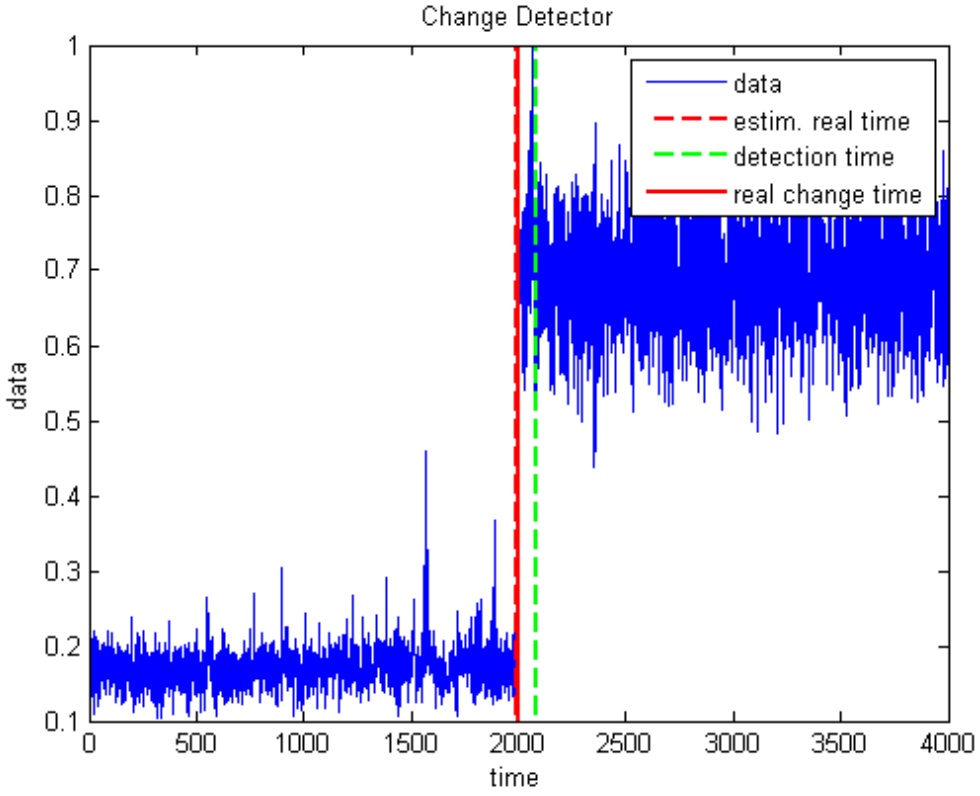
Στο πείραμα αυτό μετατρέψαμε και τα πρώτα 2000 frames σε ασπρόμαυρα, για λόγους ομοιομορφίας.

Τα σήματα που προέκυψαν ήταν τα παρακάτω:




*Σχήμα 5.2.11: Τα σήματα του 3ου πειράματος.*

Ο ανιχνευτής μεταβολών έδωσε τα εξής αποτελέσματα:



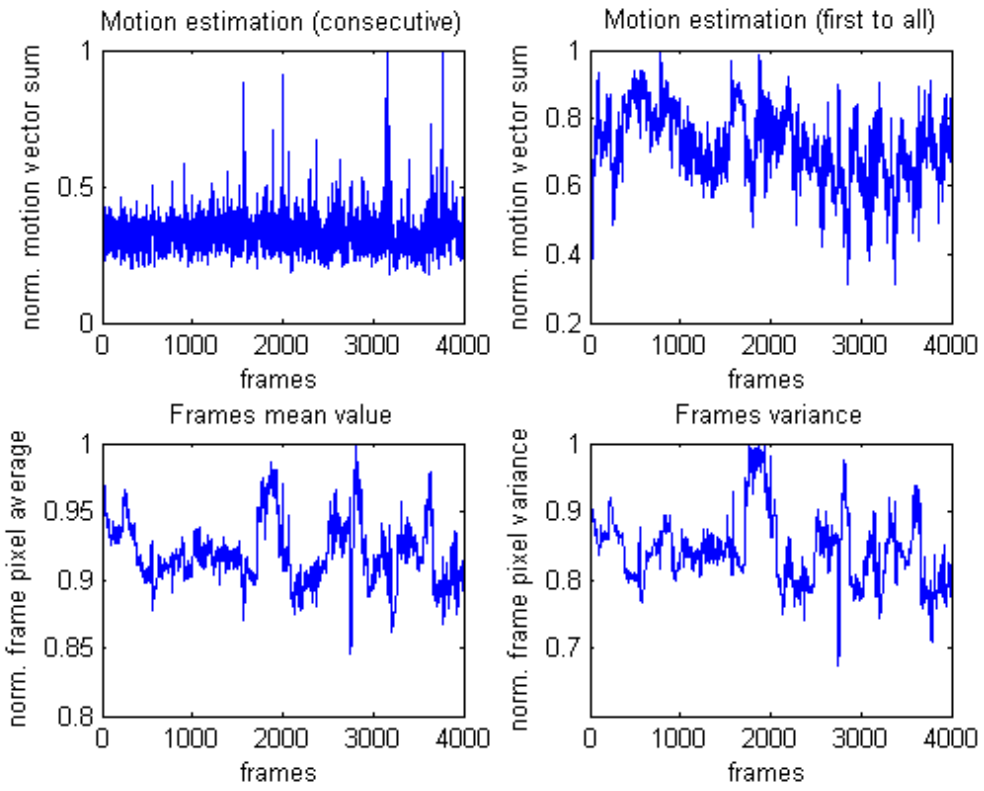
*Σχήμα 5.2.12: Ανίχνευση μεταβολών του 3ου πειράματος.*

Όπως βλέπουμε, η μεταβολή εντοπίστηκε και πάλι. Αυτό ήταν αναμενόμενο, αφού η μεταβολή αυτή είναι προφανής! Το πρώτο κομμάτι του dataset αποτελείται από μία συνεχή ακολουθία βίντεο, ενώ το δεύτερο από τυχαία καρέ, οπότε τα διανύσματα της κίνησης αυξάνονται προφανώς κατακόρυφα. Επιπλέον, η διακύμανση της μέσης τιμής και της διακύμανσης αυξάνεται επίσης σε μεγάλο βαθμό.
Εδώ πρέπει να σημειωθεί ότι αν όλα τα δεδομένα μας ήταν τυχαία εντελώς πρόσωπα δεν θα μπορούσαμε να κάνουμε κάποια ανίχνευση, παρά μόνο να τα πρόσωπα αυτά είχαν κάποιο εγγενώς διαφορετικό στατιστικό χαρακτηριστικό. Παρόλα αυτά, σε μία τέτοια περίπτωση δεν θα είχε νόημα η ανίχνευση των μεταβολών έτσι και αλλιώς.

- **Πείραμα 4**: Εδώ χρησιμοποιήσαμε 2000 διαδοχικά frames της πρώτης βάσης, τα οποία ακολουθούνται από ισάριθμα frames της ίδιας βάσης, στα οποία φιλτραρίστηκαν με Γκαουσσιανό φίλτρο (μ=4, σ=4), όπως φαίνεται παρακάτω:

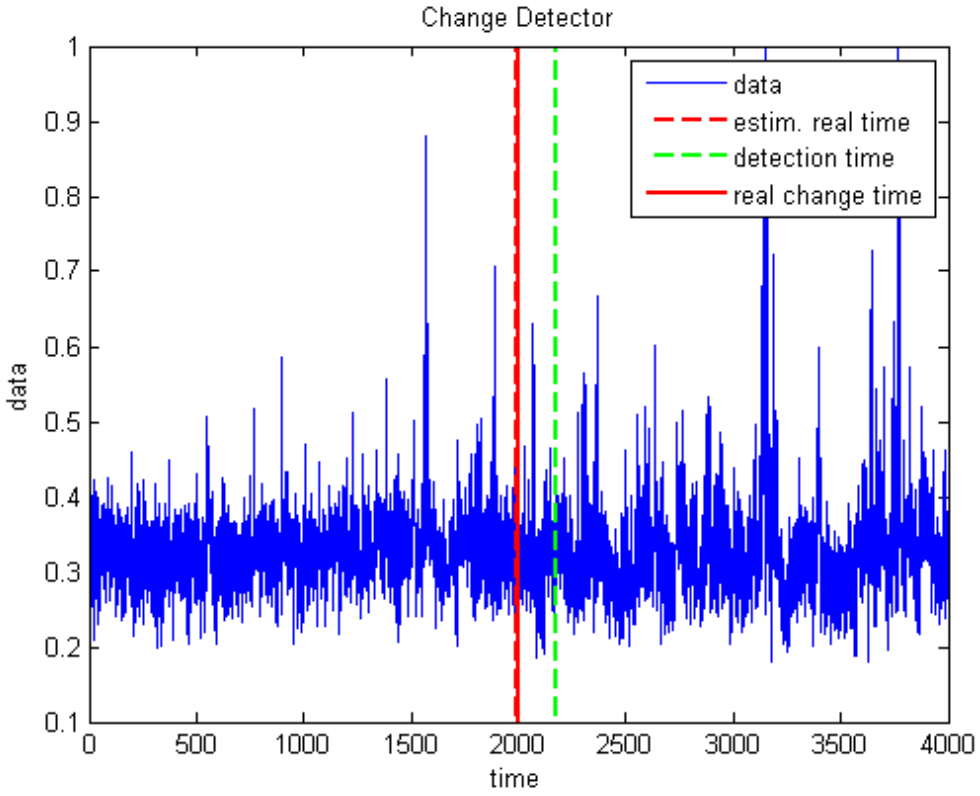

*Σχήμα 5.2.13: Το dataset του 4ου πειράματος.*

Τα σήματα που προέκυψαν ήταν τα παρακάτω:




*Σχήμα 5.2.14: Τα σήματα του 4ου πειράματος.*

Ο ανιχνευτής μεταβολών έδωσε τα εξής αποτελέσματα:



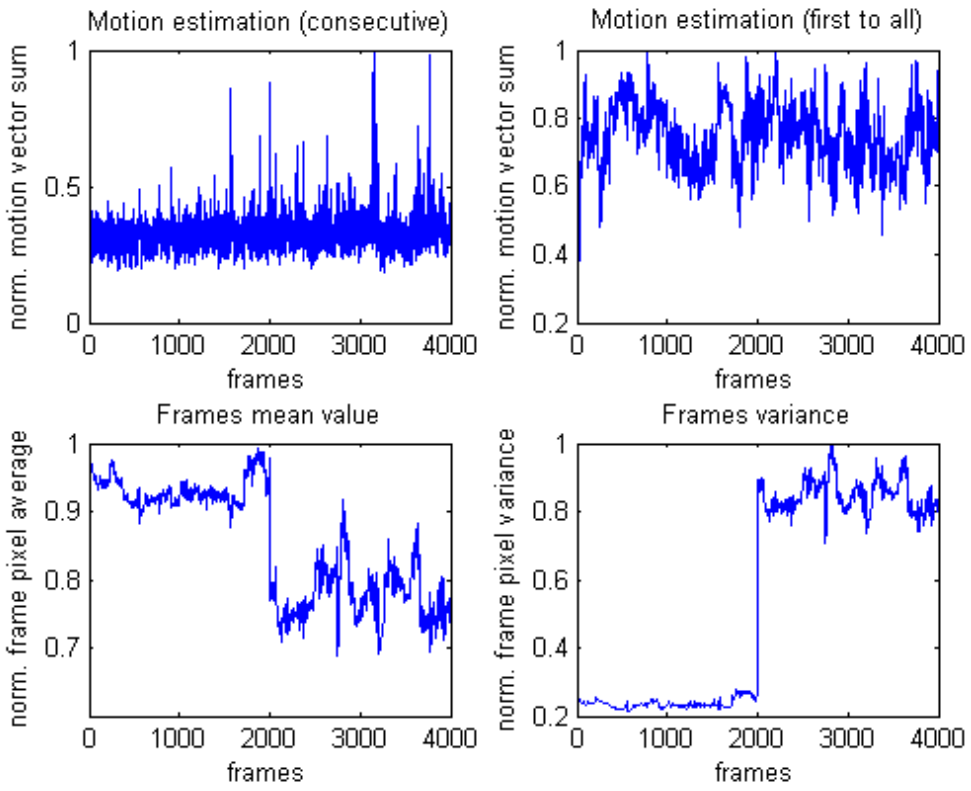

*Σχήμα 5.2.15: Ανίχνευση μεταβολών του 4ου πειράματος.*

Όπως βλέπουμε, η μεταβολή εντοπίστηκε και πάλι, αν και λίγο αργότερα απ' ότι στην περίπτωση του θορύβου.

- **Πείραμα 5**: Εδώ χρησιμοποιήσαμε 2000 διαδοχικά frames της πρώτης βάσης, τα οποία ακολουθούνται από ισάριθμα frames της ίδιας βάσης, στα οποία μεταβάλλαμε την αντίθεση, όπως φαίνεται παρακάτω:

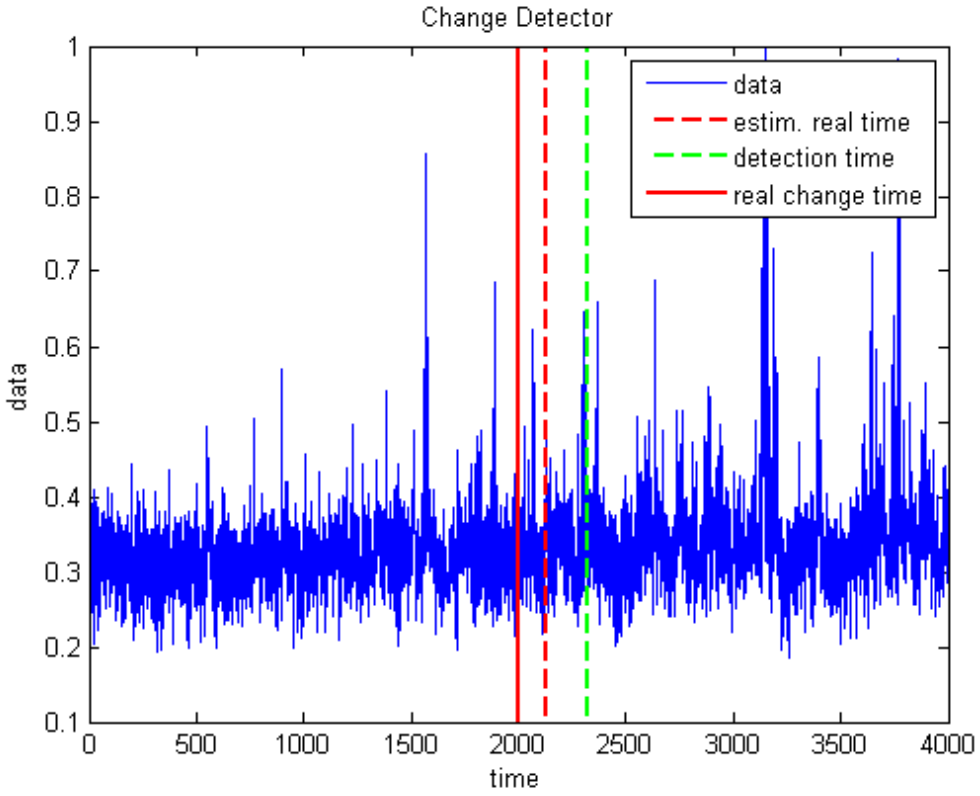

*Σχήμα 5.2.16: Το dataset του 5ου πειράματος.*

Τα σήματα που προέκυψαν ήταν τα παρακάτω:




*Σχήμα 5.2.17: Τα σήματα του 5ου πειράματος.*

Ο ανιχνευτής μεταβολών έδωσε τα εξής αποτελέσματα:



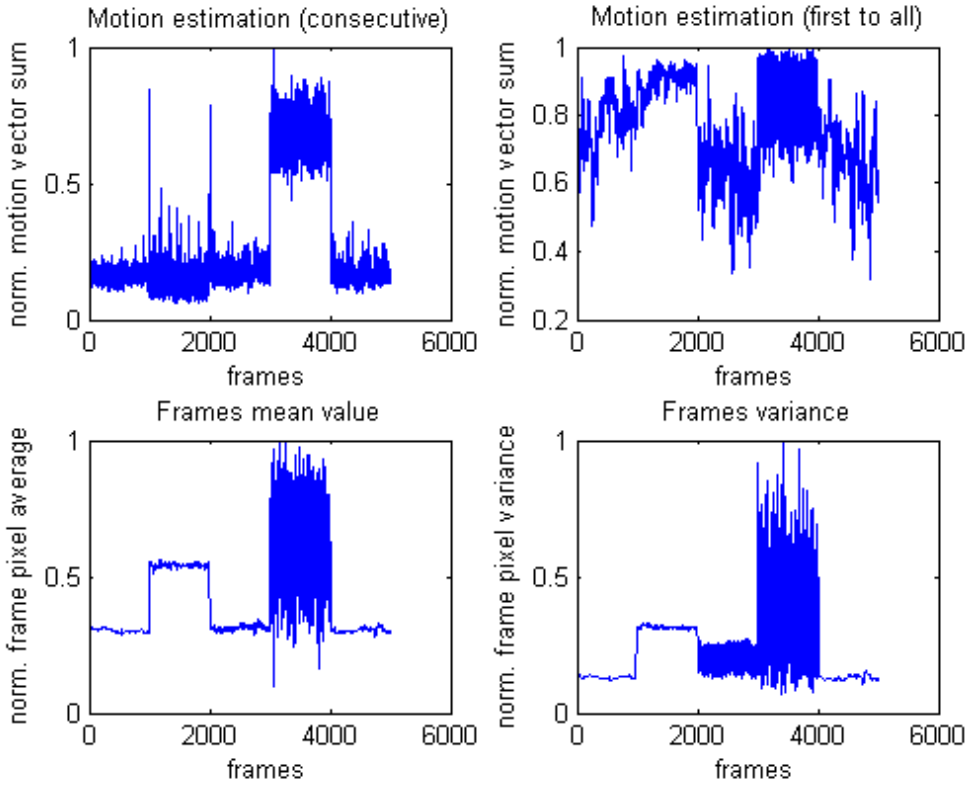

*Σχήμα 5.2.18: Ανίχνευση μεταβολών του 5ου πειράματος.*

Όπως βλέπουμε, η μεταβολή εντοπίστηκε και πάλι, αν και αρκετά πιο αργά απ' ότι στις προηγούμενες περιπτώσεις. Ο λόγος γι' αυτό είναι ότι η αλλαγή του contrast δεν επηρεάζει σε μεγάλο βαθμό τα διανύσματα κίνησης (τα οποία εξαρτώνται περισσότερο απ' τη γεωμετρία της εικόνας), οπότε η ανίχνευση εδώ ήταν πιο αργή.

- **Πείραμα 6**: Τέλος, στο τελευταίο πείραμα, χρησιμοποιήσαμε ένα μείγμα από διάφορα καρέ όλων των βάσεων, όπως φαίνεται παρακάτω:

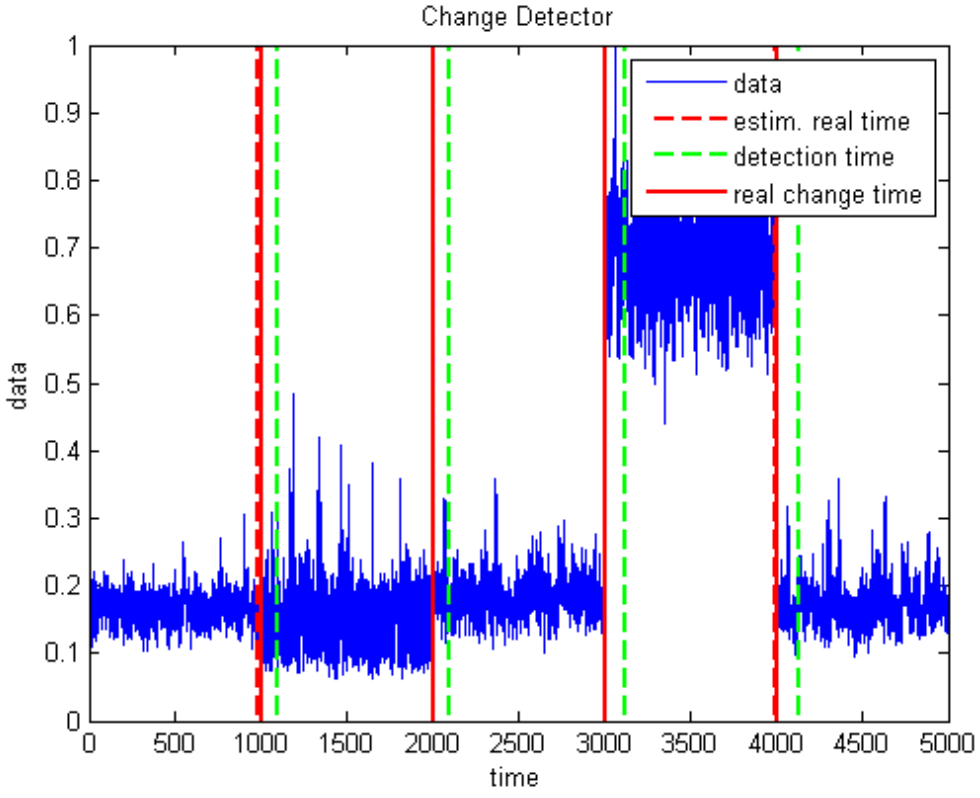

*Σχήμα 5.2.19: Το dataset του 6ου πειράματος.*

Τα σήματα που προέκυψαν ήταν τα παρακάτω:




*Σχήμα 5.2.20: Τα σήματα του 6ου πειράματος.*

Ο ανιχνευτής μεταβολών έδωσε τα εξής αποτελέσματα:



*Σχήμα 5.2.21: Ανίχνευση μεταβολών του 6ου πειράματος.*

Όπως βλέπουμε, όλες οι μεταβολές ανιχνεύτηκαν με ευκολία.

### Συμπεράσματα

Στα παραπάνω σχεδιάσαμε μία απλή διαδικασία για την ανίχνευση concept drift σε ακολουθίες βίντεο, με στόχο την βελτίωση της ανάλυσης συναισθήματος, π.χ. κάποιου συστήματος ανθρώπου μηχανής. Όπως είδαμε, το σύστημα τα πήγε αρκετά καλά στην περίπτωση διαφορετικών χρηστών, καθώς και σε κάποιες άλλες περιπτώσεις, όπως π.χ. θορύβου, φιλτραρίσματος, προσθήκης contrast, κλπ., αλλά άρχισαν να υπάρχουν και κάποιες δυσκολίες σε πιο απαιτητικές περιπτώσεις, π.χ. το contrast, που ανιχνεύτηκε αρκετά αργά. Βλέπουμε λοιπόν ότι, παρόλο που η παραπάνω προσέγγιση είχε, δεδομένης της απλότητάς της, μία αρκετά καλή απόδοση, η εφαρμογή σε πιο απαιτητικές καταστάσεις απαιτεί περαιτέρω βελτίωση.

Η λύση είναι αντί μόνο το άθροισμα των διανυσμάτων κίνησης, να πάρουμε περισσότερες πληροφορίες για την εικόνα, όπως π.χ. οι θέσεις και τα μέτρα των διανυσμάτων κίνησης, οι αναλογίες των χρωμάτων, κλπ. Συνδυάζοντας τα χαρακτηριστικά αυτά θα μπορούσαν να κατασκευαστεί ένα διάνυσμα που να περιγράφει την κάθε εικόνα (π.χ. σαν ένα bag of words), και στη συνέχεια θα μπορούσαμε να ελέγξουμε τα διανύσματα αυτά για ανίχνευση μεταβολών. Το αποτέλεσμα που θα προκύπτει θα ήταν εμφανώς καλύτερο. Στην περίπτωσή μας, κύριο αντικείμενο της διπλωματικής ήταν η μελέτη και η κατασκευή των change detectors, οπότε δεν προχωρήσαμε παραπέρα με το ζήτημα αυτό.

## 5.3 Βελτίωση του ταξινομητή

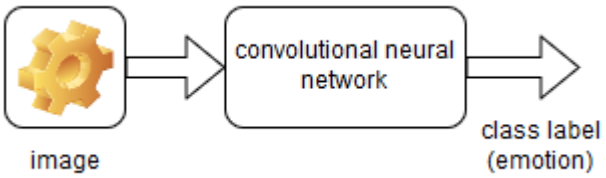
Στην ενότητα αυτή θέλουμε να χρησιμοποιήσουμε τον προηγούμενο ανιχνευτή μεταβολών για να βελτιώσουμε την απόδοση ενός ταξινομητή, ο οποίος αναλύει το συναίσθημα με βάση την εικόνα. Μία τέτοια εφαρμογή θα ήταν χρήσιμη σε διάφορους τομείς της αλληλεπίδρασης ανθρώπου μηχανής.

Η αρχιτεκτονική που χρησιμοποιήθηκε αποτελείται από ένα βαθύ συνελικτικό δίκτυο, το οποίο λαμβάνει ως είσοδο την εικόνα και εξάγει το συναίσθημα του χρήστη.

Το νευρωνικό δίκτυο αναπτύχθηκε με την πλατφόρμα Caffe, και εκπαιδεύτηκε με βάση την 1η βάση δεδομένων, στην οποία έχουν επεκταθεί τα annotated καρέ στα γειτονικά τους, έτσι ώστε να αυξηθεί το σύνολο των διαθέσιμων δεδομένων.

Το δίκτυο που χρησιμοποιήθηκε περιλαμβάνει 4 στάδια. Τα πρώτα στάδια του δικτύου αποτελούνται από δύο συνελικτικά επίπεδα, τα οποία περιέχουν 20 και 50 φίλτρα αντίστοιχα, όλα μεγέθους 5x5 πίξελ. Οι παράμετροι μάθησης ήταν 0.001 για τα φίλτρα, 0.002 για το bias, 0.9 για τον συντελεστή ορμής, και 0.004 για την παράμετρο αποσύνθεσης βαρών ανά εποχή. Στα pooling επίπεδα, το μέγεθος του παραθύρου ήταν 2x2. Στη συνέχεια, τα πλήρως διασυνδεδεμένα επίπεδα είχαν τις ίδιες παραμέτρους, με εξαίρεση την αποσύνθεση βαρών, η οποία τέθηκε ίση με 1. Τέλος, τα δύο τελευταία επίπεδα περιλάμβαναν 500 μονάδες Relu και 5 εξόδους, οι

οποίες αντιστοιχούν στις διαφορετικές κατηγορίες συναισθημάτων στον κύκλο Wiessel.
Η αρχιτεκτονική του συστήματος φαίνεται παρακάτω:




*Σχήμα 5.3.1: Η αρχιτεκτονική του ταξινομητή.*

Στη συνέχεια, εκπαιδεύσαμε το δίκτυο σε ένα σύνολο δεδομένων, και στη συνέχεια το βάλαμε να ταξινομήσει ένα τροποποιημένο σύνολο δεδομένων. Για να βελτιώσουμε την απόδοση, χρησιμοποιήσαμε τον ανιχνευτή μεταβολής της προηγούμενης ενότητας, έτσι ώστε να εντοπίσουμε τη μεταβολή στο σύνολο των δεδομένων. Μόλις η μεταβολή εντοπιστεί, παίρνουμε ένα μικρό υποσύνολο των νέων δεδομένων (π.χ. ζητάμε μερικά δεδομένα από το χρήστη), και εφαρμόζουμε τη διαδικασία του fine - tuning στο νευρωνικό δίκτυο. Κάνοντας τέτοιου τύπου πειράματα με διάφορα σύνολα δεδομένων, βλέπουμε ότι η απόδοση του ταξινομητή αυξάνεται σε κάθε περίπτωση, ανάλογα φυσικά με το βαθμό τροποποίησης των δεδομένων, την αρχιτεκτονική του δικτύου, τις παραμέτρους του fine – tuning, κλπ. Με τον τρόπο αυτό φαίνεται ότι η ανίχνευση μεταβολών και η μάθηση σε μεταβαλλόμενα περιβάλλοντα έχουν νόημα, και μπορούν να βελτιώσουν την απόδοση των συστημάτων μηχανικής μάθησης.

Τα πειράματα που έγιναν περιγράφονται παρακάτω:

**Πείραμα 1**: Στο πείραμα αυτό, εκπαιδεύσαμε αρχικά το δίκτυό μας στις εικόνες της πρώτης βάσης δεδομένων, αφού τις σμικρύναμε σε διαστάσεις 120x170 pixels, έτσι ώστε να βελτιώσουμε την ταχύτητα (οι διαστάσεις αυτές επιλέχθηκαν έναντι μίας τετραγωνικής διάστασης, επειδή διατηρούν τις αναλογίες του προσώπου). Χωρίσαμε το σύνολο δεδομένων κατά τα γνωστά, σε ένα σύνολο εκπαίδευσης (80%), και ένα σύνολο ελέγχου (20%). Η αρχική ακρίβεια του δικτύου ήταν 82.7%, ένα αποτέλεσμα που είναι αρκετά ικανοποιητικό, δεδομένων των λίγων σχετικά δεδομένων που διαθέτουμε.
Στη συνέχεια, κατασκευάσαμε ένα test set με εικόνες της $1^{ης}$ βάσης, στις οποίες όμως προσθέσαμε Γκαουσσιανό θόρυβο, με μέση τιμή $\mu = 0$, και διακύμανση $\sigma^2 = 0.025$. Η ακρίβεια του δικτύου στο νέο σύνολο δεδομένων (θορυβώδεις εικόνες) έπεσε στο 80.5%.
Στη συνέχεια, χρησιμοποιώντας τον CDT της προηγούμενης παραγράφου, το σύστημα εντόπισε τη μεταβολή των δεδομένων, και έκανε fine tuning σε ένα μικρό σύνολο με 1000 θορυβώδεις εικόνες. Με τον τρόπο αυτό, η ακρίβεια του συστήματος ανέβηκε στο 81.7%. Αυτό το πετύχαμε μειώνοντας τον βασικό ρυθμό μάθησης του δικτύου κατά 10, και πενταπλασιάζοντας παράλληλα το ρυθμό μάθησης του τελευταίου επιπέδου επί 5. Φυσικά, μία καλύτερη επιλογή των παραμέτρων ίσως να έδινε ακόμη καλύτερα αποτελέσματα – η βελτίωση φυσικά

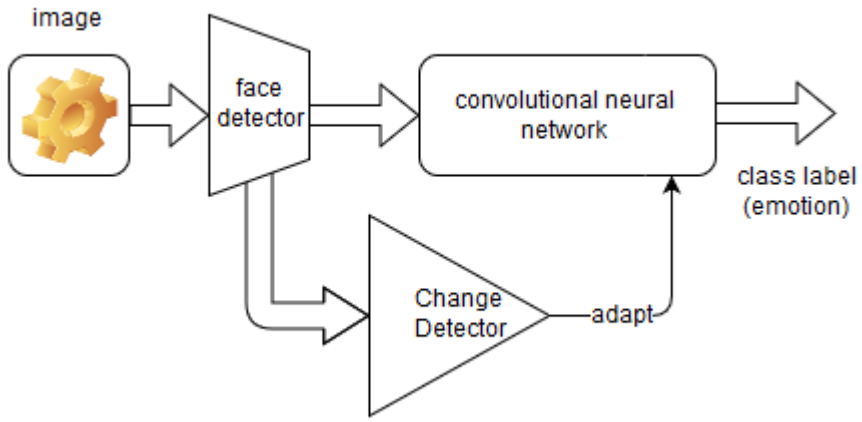
που μπορούμε να πετύχουμε εξαρτάται και απ' το επίπεδο του θορύβου: όσο μεγαλώνει ο θόρυβος, η βελτίωση της ακρίβειας αυξάνεται, αλλά αυξάνοντας το θόρυβο υπερβολικά οι εικόνες θα γίνουν μη αναγνωρίσιμες, οπότε η βελτίωση της ακρίβειας έχει έναν φυσικό άνω όριο.

**Πείραμα 2**: Το πείραμα αυτό είναι παρόμοιο με το πρώτο, με τη διαφορά ότι εδώ κατασκευάζουμε ένα test set όχι εισάγοντας θόρυβο, αλλά φιλτράροντας τις εικόνες με Γκαουσσιανό φίλτρο διάστασης 7x7. Το αρχικό δίκτυο που χρησιμοποιείται είναι αυτό του πρώτου πειράματος, οπότε η αρχική ακρίβεια στο μη τροποποιημένο σύνολο ελέγχου είναι και πάλι 82.7%. Εισάγοντας το φιλτράρισμα, η ακρίβεια του συστήματος πέφτει στο 80.3%. Ανιχνεύοντας τη μεταβολή όπως και πριν, και εφαρμόζοντας fine tuning σε 1000 εικόνες (με τις ίδιες παραμέτρους όπως και προηγουμένως), η ακρίβεια του συστήματος ανέβηκε στο 82.5%. Φυσικά, οι τιμές αυτές εξαρτώνται τόσο απ' τις παραμέτρους του fine tuning, όσο και από τον βαθμό τροποποίησης των δεδομένων.

**Πείραμα 3**: Το πείραμα αυτό είναι εντελώς όμοιο με τα δύο προηγούμενα, με τη διαφορά ότι τώρα εισάγουμε στο δίκτυο εικόνες της δεύτερης βάσης δεδομένων μας, χρησιμοποιώντας και εδώ τη μέθοδο επέκτασης των FAPs σε γειτονικά καρέ, ώστε να αυξήσουμε τον όγκο των δεδομένων. Η αρχική ακρίβεια του δικτύου στο νέο αυτό σύνολο δεδομένων ήταν πολύ χαμηλή – περίπου 14% - πράγμα που δείχνει ότι η ανάλυση του συναισθήματος διαφέρει αρκετά από άτομο σε άτομο (σύμφωνα με τη θεωρία, η αντιστοίχιση προσώπου σε FAPs είναι περισσότερο αμετάβλητη, αφού τα FAPs σχετίζονται με τα γεωμετρικά χαρακτηριστικά του προσώπου, ενώ αντίθετα, η αντιστοίχιση μεταξύ FAPs και συναισθήματος εξαρτάται αρκετά απ' τον χρήστη). Εφαρμόζοντας και πάλι fine – tuning με 1000 εικόνες, η ακρίβεια του συστήματος ανεβαίνει στο 60.4%, το οποίο είναι μία σημαντική αύξηση. Βλέπουμε λοιπόν ότι και στην περίπτωση αυτή, η μεθοδολογία της ανίχνευσης μεταβολών μπορεί να οδηγήσει στην βελτίωση της ταξινόμησης.

**Πείραμα 4**: Στο πείραμα αυτό, εκπαιδεύουμε αρχικά ένα συνελικτικό δίκτυο αναγνώρισης συναισθήματος στη βάση FER. Πριν γίνει αυτό, για λόγους συμβατότητας με τις άλλες δύο βάσεις, μετατρέψαμε τις τιμές των συναισθημάτων της βάσης FER (0 - neutral, 1 – angry, κλπ.), σε τιμές positive και active, με βάση τον κύκλο Wiessel του κεφαλαίου 4. Επίσης, λόγω της μικρής διάστασης των εικόνων της βάσης FER, μειώσαμε τη διάσταση του πυρήνα συνέλιξης των convolution επιπέδων του δικτύου από 5x5 σε 3x3. Εκτός απ' αυτά, οι υπόλοιπες παράμετροι παρέμειναν ίδιες. Στη συνέχεια, κατασκευάσαμε ένα test set, το οποίο περιείχε εικόνες της 1$^{ης}$ βάσης, οι οποίες μετατράπηκαν σε ασπρόμαυρες και άλλαξε η διάστασή τους, για λόγου συμβατότητας με τη βάση FER. Η ακρίβεια του συστήματος ανήρθε στο 51.8%. Στη συνέχεια, εντοπίζοντας τη μεταβολή και εκτελώντας το fine – tuning, η ακρίβεια αυξήθηκε στο 65.2%. Έχουμε λοιπόν βελτίωση και σε αυτήν την περίπτωση.

Τέλος, η συνολική αρχιτεκτονική του συστήματος που χρησιμοποιήθηκε φαίνεται παρακάτω:




*Σχήμα 5.3.2: Η συνολική αρχιτεκτονική του συστήματος ταξινόμησης.*

Επίσης, εδώ πρέπει να τονίσουμε ότι τα αποτελέσματα που πήραμε παραπάνω εξαρτώνται από την αρχιτεκτονική και τις παραμέτρους του δικτύου, οπότε μία προσεκτικότερη επιλογή τους μπορεί να μας έδινε ακόμη καλύτερα αποτελέσματα. Δυστυχώς, αυτό απαιτεί λεπτομερή γνώση των νευρωνικών δικτύων, η οποία ξεφεύγει απ' τα όρια αυτής της διπλωματικής.

# 6 Παρατηρήσεις και Συμπεράσματα

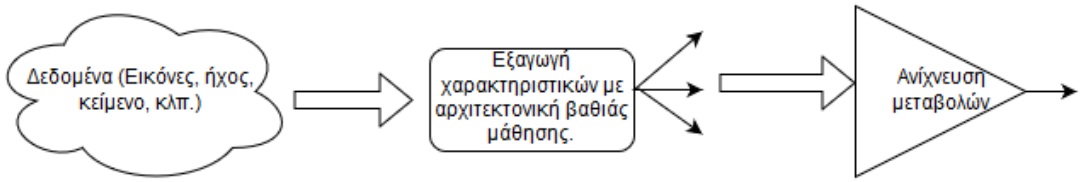
Στην εργασία αυτή μελετήθηκε μία σχετικά νέα περιοχή της μηχανικής μάθησης, η μάθηση σε μεταβαλλόμενα περιβάλλοντα. Συγκεκριμένα, μελετήθηκε μία κατηγορία ανιχνευτών μεταβολής, οι λεγόμενοι ιεραρχικοί ανιχνευτές μεταβολής, και υλοποιήθηκαν μερικές βασικές αρχιτεκτονικές. Στη συνέχεια, έγινε προσπάθεια στο να επεκταθούν οι τεχνικές αυτές σε ένα δυσκολότερο πρόβλημα, το πρόβλημα της ανίχνευσης μεταβολών σε ακολουθίες βίντεο ενός προσώπου, με στόχο τη βελτίωση της αναγνώρισης συναισθήματος σε συστήματα αλληλεπίδρασης ανθρώπου – μηχανής.

Χρησιμοποιώντας μία σχετικά απλή τεχνική, κατασκευάσαμε μερικά βασικά μονοδιάστατα σήματα απ' τις εικόνες, και μέσω αυτών καταφέραμε να πάρουμε μία αρκετά καλή ανίχνευση των μεταβολών σχεδόν σε όλες τις περιπτώσεις που δοκιμάσαμε, τόσο όταν το πρόσωπο του χρήστη αλλάζει, όσο όταν παρεμβάλλονται άλλοι παράγοντες που αλλάζουν την κατανομή, όπως π.χ. θόρυβος, αλλαγές στη φωτεινότητα, κλπ. Στη συνέχεια, με βάση την ανίχνευση αυτή, καταφέραμε να βελτιώσουμε την απόδοση ενός συστήματος αναγνώρισης συναισθήματος από βίντεο.

Παρόλα αυτά, υπάρχουν αρκετά πράγματα ακόμη που πρέπει να γίνουν. Ένα από αυτά για παράδειγμα θα ήταν η κατασκευή ενός ακόμη καλύτερου ανιχνευτή μεταβολής για εικόνες, ο οποίος να κατασκευάζει μία χρήσιμη αναπαράσταση της εικόνας, βασισμένη στα διανύσματα κίνησης, το χρώμα, τη μέση τιμή – διακύμανση, τα χαρακτηριστικά της, κλπ., έτσι ώστε να μπορούμε να εντοπίζουμε τις μεταβολές με μεγαλύτερη ακρίβεια, ακόμα και σε περιπτώσεις αργά μεταβαλλόμενων δεδομένων.

Επίσης, γενικά, η ανίχνευση μεταβολής σε δεδομένα πολλών διαστάσεων, και ιδιαίτερα σε δεδομένα μεγάλης διάστασης είναι ένα πρόβλημα που αφενός δεν έχει μελετηθεί αρκετά, και αφετέρου θεωρείται αρκετά δύσκολο, με την έννοια ότι τα υπάρχοντα στατιστικά τεστ πολλών μεταβλητών είναι δύσχρηστα και χρειάζονται πολλά δεδομένα για να βγάλουν ένα ασφαλές συμπέρασμα. Θα ήταν λοιπόν ιδιαίτερα χρήσιμη μία θεωρία, η οποία εντοπίζει τις συσχετίσεις στα πολυδιάστατα δεδομένα και να τα μετατρέπει σε μία αναπαράσταση χαμηλότερης διάστασης, η οποία να προσφέρεται περισσότερο για ανίχνευση μεταβολών. Εδώ θα μπορούσαν να χρησιμοποιηθούν διάφορες τεχνικές, απ' το κλασσικό PCA, μέχρι η θεωρία των αραιών αναπαραστάσεων. Αυτό θα μπορούσε να γίνει και σε συνδυασμό με τη βαθιά, μάθηση, δηλ. ένα σύστημα βαθιάς μάθησης να παράγει μία χρήσιμη αναπαράσταση των δεδομένων, και στη συνέχεια ο ανιχνευτής μεταβολής να λειτουργεί στο χώρο της αναπαράστασης. Ας υποθέσουμε λ.χ. ότι έχουμε ένα γενικό σύνολο δεδομένων, το οποίο μπορεί να αποτελείται από εικόνες, ήχους,

κείμενο, κλπ. Αντί να σχεδιάσουμε εμείς τα προς ανίχνευση χαρακτηριστικά, πιθανώς θα μπορούσαμε να παράγουμε χρήσιμα χαρακτηριστικά αυτόματα χρησιμοποιώντας μία βαθιά αρχιτεκτονική, π.χ. ένα cnn στην περίπτωση της επιβλεπόμενης μάθησης, ή έναν autoencoder στην περίπτωση της μη επιβλεπόμενης μάθησης. Στη συνέχεια, αφού η αρχιτεκτονική μας εξάγει χρήσιμες αναπαραστάσεις από τα δεδομένα, μπορούμε να εφαρμόσουμε έναν ανιχνευτή μεταβολών στα χαρακτηριστικά αυτά. Η διαδικασία αυτή φαίνεται συνοπτικά στο παρακάτω σχήμα:




*Σχήμα 5.3.3: Αυτόματη εύρεση χαρακτηριστικών για ανίχνευση μεταβολών μέσω βαθιάς μάθησης.*

Η περαιτέρω διερεύνηση των παραπάνω θα ήταν κατά τη γνώμη μας μία σημαντική περιοχή έρευνας.

Γενικά, ένα ουσιώδες χαρακτηριστικό της νοημοσύνης είναι η προσαρμογή σε νέα δεδομένα. Ο ανθρώπινος εγκέφαλος για παράδειγμα διαθέτει το χαρακτηριστικό αυτό σε ιδιαίτερα μεγάλο βαθμό, αφού είναι σε θέση να μαθαίνει νέες γνώσεις μόνο με τη βοήθεια πολύ λίγων δειγμάτων εκπαίδευσης. Για παράδειγμα, απαιτούνται μόνο μερικές δεκάδες παρτίδες μέχρι να μάθει κανείς να παίζει σκάκι, ενώ διάφορα ευφυή συστήματα που έχουν αναπτυχθεί, βασισμένα σε νευρωνικά δίκτυα, απαιτούν δεκάδες εκατομμύρια παραδείγματα. Βλέπουμε λοιπόν ότι ο ανθρώπινος εγκέφαλος έχει την ικανότητα να προσαρμόζεται και να γενικεύει τη γνώση σε μεγάλο βαθμό. Επομένως, η έρευνα γύρω απ’ το μηχανισμό αυτό, και του πεδίου της μάθησης σε μεταβαλλόμενα περιβάλλοντα γενικότερα, πιστεύουμε ότι είναι ιδιαίτερα σημαντική για την περαιτέρω ανάπτυξη της μηχανικής μάθησης.

# 7 Βιβλιογραφία


[1] *Pattern Recognition and Machine Learning*. C. Bishop. Springer Verlag, 2006.

[2] *Machine Learning in Non - Stationary Environments*. M.Sugiyama and M.Kawanabe. Cambridge, MA: MIT Press, 2012.

[3] *Introduction to Machine Learning*. E. Alpaydin. Second Ed., MIT Press, 2010.

[4] *Deep Learning*. I. Goodfellow, Y. Bengio, A. Courville, MIT Press, 2016.

[5] *A survey on Concept Drift Adaptation*. J. Gama, I. Zlibaite, A. Bifet, M. Pechenizkiy, A. Bouchachia. ACM Computing Surveys, Vol. 1, No. 1, Jan. 2013.

[6] *Learning in Nonstationary Environments: A Survey*. G. Ditzler, M. Roveri, C. Alippi, R. Polikar. IEEE Comp. Intel. Magazine, Nov. 2015.

[7] *Learning under Concept Drift: an Overview*. I. Zlibaite. Vilnius University, Technical Report, 2009.

[8] *An overview of concept drift Applications*. I. Zlibaite, M. Pechenizkiy, J. Gama. Big Data Analysis: New Algorithms for a New Society, Japkowicz, N. and Stefanowski, J. (Eds.), Springer Verlag, 2016.

[9] *A Review of Concept Drift.* Y. Kadwe, V. Suryawanshi, IOSR Journal of Comp. Eng., Vol. 17, Feb. 2015, pp. 20 – 26.

[10] *Detection and Management of Concept Drift*. L. Mak, P. Krause. In Proc. 5th Inter. Conf. on M. Learning and Cybern., Aug. 2016.

[11] *Learning in the presence of concept drift and hidden contexts*. G. Widmer and M. Kubat. In Machine Learning, pages 69–101, 1996.

[12] *Tracking Concept Drift in Malware Families*. A. Singh, A. Walenstein, A. Lakhoita. 5th ACM Works. On Sec. And Artif. Intel., 2012.

[13] *Stanley: The Robot that Won the DARPA Grand Challenge*. S. Thrun et. all. Journal of Field Robotics 23(9), 661–692 (2006).

[14] *The dictionary of affect in language*. C. M. Whissel. R. Plutchnik and H. Kellerman (Eds) Emotion: Theory, research and experience: vol 4, The measurement of emotions. Academic Press, New York, 1989.

[15] *Emotion Recognition in Human – Computer Interaction*. R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, J.G. Taylor. IEEE Signal

Processing Magazine, Jan. 2001.

[16] *Face and 2-D Mesh Animation in MPEG-4*. M. Tekalp. Tutorial Issue On The MPEG-4 Standard, Image Communication Journal, Elsevier, 1999.

[17] *Emotion recognition through facial expression analysis based on a neurofuzzy network*. S. Ioannou, A. Raouzaiou, V. Tzouvaras, T. Mailis, K. K Karpouzis, S. Kollias. Special Issue on Emotion: Understanding & Recognition, Neural Networks, Elsevier, Volume 18, Issue 4, May 2005, Pages 423-435.

[18] *Emotion: A psycho-evolutionary synthesis*. R. Plutchik. Harper and Row, New York, 1980.

[19] *Incremental learning of concept drift in non-stationary environments*. R. Elwell and R.Polikar. IEEE Trans. Neural Netw., vol.22, no. 10, pp. 1517–1531, Oct.2011.

[20] *Adaptive Concept Drift Detection*. A. Dreis, U. Ruckert. Stat. Analysis and Data Mining, 2(5-6):311-327, 2009.

[21] *Just-in-time adaptive classifiers-part II: Designing the Classifier*. C. Alippi and M. Roveri. IEEE Trans. Neural Netw., vol. 19, no. 12, pp. 2053–2064, Dec. 2008.

[22] *Learning with drift Detection*. J. Gama, P. Medas, G. Castillo, and P.Rodrigues. In Proc. Advances Artificial Intelligence–SBIA, 2004, pp. 286–295.

[23] *Learning from time-changing data with adaptive windowing*. A. Bifet and R.Gavalda. In Proc. SIAM Int. Conf. Data Mining, 2007.

[24] *Just-in-time classifiers for recurrent concepts*. C. Alippi, G. Boracchi, and M.Roveri. IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 4, pp. 620–634, Apr.2013.

[25] *Intelligence for Embedded Systems*. C. Alippi. Berlin, Germany: Springer-Verlag, 2014.

[26] *Detecting change in data streams*. D. Kifer, S. Ben-David, and J.Gehrke. In Proc. 30th Int. Conf. Very Large Data Bases, 2004, vol. 30, pp. 180–191.

[27] *Hellinger distance based drift detection for Nonstationary environments*. G. Ditzler and R.Polikar. InProc. IEEE Symp. Computational Intelligence Dynamic in Uncertain Environments, 2011, pp. 41–48.

[28] *Optimal window change detection*. J. P.Patist. In Proc. 7th IEEE Int. Conf. Data Mining Workshops, 2007, pp. 557–562.

[29] *Kalman filters and adaptive windows for learning in data streams*. A. Bifet and R.Gavalda. In Proc. Int. Conf. Discovery Science, 2006, pp. 29– 40.

[30] *An adaptive cusum-based test for signal change detection*. C. Alippi and M.Roveri. In Proc. Int. Symp. Circuits Systems, 2006, pp. 1–4.

[31] *Just-in-time adaptive classifiers—Part I: Detecting nonstationary changes*. C. Alippi and M. Roveri. IEEE Trans. Neural Netw., vol.19, no.7, pp. 1145–1153, July 2008.

[32] *Change detection tests using the ICI rule*. C. Alippi, G. Boracchi, and M.Roveri. In Proc.Int. Joint Conf. Neural Networks, 2010, pp. 1–7.

[33] *Just-in-time ensemble of classifiers*. C. Alippi, G. Boracchi, and M.Roveri. In Proc. Int. Joint Conf. Neural Networks, 2012, pp. 1–8.

[34] *Real-Time data mining of non-stationary data streams from sensor Networks*. L. Cohen, G. Avrahami-Bakish, M.Last, A. Kandel, and O. Kipersztok. Inform. Fusion, vol. 9, no.3, pp. 344–353, July 2008.

[35] *Concept drift detection through resampling*. M. Harel, K. Crammer, R. Yaniv, and S. Mannor. In Proc. Int. 31st Conf. Machine Learning, 2014. pp. 1009–1017.

[36] *Learning, detecting, understanding, and predicting concept changes*. K. Nishidaand, K. Yamauchi. In Proc.Int. Joint Conf. Neural Networks, 2009, pp. 2280–2287.

[37] *Early drift detection method*. M.Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales Bueno. In Proc. 4th Int. Workshop Knowledge Discovery from Data Streams, 2006, pp. 1–4.

[38] *Detecting concept drift using statistical testing*. K. Nishida and K. Yamauchi. In Discovery Science. Berlin, Germany: Springer-Verlag, 2007, pp. 264–269.

[39] *The change-point model for statistical process control*. D. M. Hawkins, Q. Peihua, and W. K. Chang. J. Qual. Technol., vol. 35, no. 4, pp. 355–366, Oct. 2003.

[40] *Nonparametric monitoring of data streams for changes in location and scale*. G. J. Ross, D. K. Tasoulis, and N. M.Adams. Technometrics, vol. 53, no. 4, pp. 379–389, 2011.

[41] *Sequential tests of statistical hypotheses*. A. Wald. Ann. Math. Stat., vol. 16, no. 2, pp.117–186, June 1945.

[42] *One pass concept change detection for data streams*. S. Sakthithasan, R. Pears, and Y. S. Koh. In Advances in Knowledge Discovery and Data Mining, Berlin, Germany: Springer-Verlag, 2013, pp. 461–472.

[43] *Detecting concept change in dynamic data streams*. R. Pears, S. Sakthithasan, and Y. S. Koh. Mach. Learn., vol. 97, no. 3, pp. 259–293, Jan.2014.

[44] *Online and non-parametric drift detection methods based on hoeffding's bounds*. I. Frías-Blanco, J. del Campo-Ávila, G. Ramos-Jiménez, R. Morales-Bueno, A. Ortiz-Díaz, and Y.Caballero-Mota, IEEE Trans. Knowledge Data Eng., vol. 27, no. 3, pp. 810–823, Aug. 2014.

[45] *A just-in-time adaptive classification System based on the intersection of confidence intervals rule*. C. Alippi, G. Boracchi, and M.Roveri. Neural Netw., vol. 24, no. 8, pp.791–800, Oct. 2011.

[46] *A hierarchical, nonparametric, Sequential change-detection test*. C. Alippi, G. Boracchi, and M. Roveri. In Proc. Int. Joint Conf. Neural Networks, 2011, pp. 2889–2896.

[47] *A cognitive monitoring system for contaminant detection in intelligent buildings*. G. Boracchi, M. Michaelides, and M. Roveri. In Proc. Int. Joint Conf. Neural Networks, July 2014, pp. 69–76.

[48] *An effective just-in-time adaptive Classifier for gradual concept drifts*. C. Alippi, G. Boracchi, and M. Roveri. InProc. Int. Joint Conf. Neural Networks, 2011, pp. 1675–1682.

[49] *Just in time classifiers: Managing the slow drift case*. C. Alippi, G. Boracchi, and M.Roveri. In Proc. Int. Joint Conf. Neural Networks, 2009, pp. 114–120.

[50] *Gradual forgetting for adaptation to concept drift*. I. Koychev. In Proc. ECAI Workshop Current Issues Spatio-Temporal Reasoning, 2000, pp. 101–106.

[51] *Maintaining time-decaying stream aggregates*. E. Cohen and M. Strauss. In Proc. 22nd ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Systems, 2003, pp. 223–233.

[52] *Learning drifting concepts: Example selection vs. example weighting*. R. Klinkenberg. Intell. Data Anal., vol. 8, no. 3, pp. 281–300, Aug. 2004.

[53] *Random sampling with a reservoir*. S.Vitter. ACM Trans. Math. Softw., vol. 11, no.1, pp. 37–57, Mar. 1985.

[54] *On biased reservoir sampling in the presence of stream evolution*. C. C. Aggarwal. In Proc. 32nd Int. Conf. Very Large Data Bases, 2006, pp. 607–618.

[55] *A test paradigm for detecting changes in Transactional data streams*. W. Ngand M. Dash. In Database Systems for Advanced Applications. Berlin, Germany: Springer-Verlag, 2008, pp. 204–219.

[56] *Mining high-speed data streams*. P. Domingos and G. Hulton. In Proc. 6th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, 2000, pp. 71–80.

[57] *Mining time-changing data streams*. G. Hulten, L. Spencer, and P. Domingos. In Proc. Conf. Knowledge Discovery Data, 2001, pp. 97–106.

[58] *Info-fuzzy algorithms for mining dynamic data streams*. L. Cohen, G. Avrahami, M. Last, and A. Kandel. Appl. Soft Comput., vol. 8, no. 4, pp. 1283–1294, Sept. 2008.

[59] *Classifier ensembles for changing environments*. L. I. Kuncheva. In Proc. 5th Int. Workshop Multiple Classifier Systems, 2004, pp. 1–15.

[60] *Dynamic Integration of classifiers for handling concept drift*. A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen Inform. Fusion, vol. 9, no. 1, pp. 56–68, Jan. 2008.

[61] *Nonlinear neural networks: Principles, mechanisms, and architectures*. S. Grossberg. Neural Netw., vol. 1, no. 1, pp. 17–61, 1988.

[62] *Discounted expert weighting for concept drift*. G. Ditzler, G. Rosen, and R. Polikar. In Proc. IEEE Symp. Computational Intelligence Dynamic Uncertain Environments, 2013, pp. 61–67.

[63] *Domain adaptation bounds for multiple Expert systems under concept drift*. G.Ditzler, G.Rosen, and R. Polikar in Proc. Int. Joint Conf. Neural Networks, 2014, pp. 595–601.

[64] *The impact of diversity on online ensemble learning in the presence of concept drift*. L. L.Minku, A. P. White, and X.Yao. IEEE Trans. Knowledge Data Eng., vol.22, no.5, pp. 731–742, May 2010.

[65] *DDD: A new ensemble approach for dealing with concept drift*. L. L.Minku and X.Yao. IEEE Trans. Knowledge Discovery Data Eng., vol. 24, no. 4, pp. 619–633, Apr. 2012.

[66] *A streaming ensemble algorithm (SEA) for large-scale classification*. W. N. Street and Y. Kim. In Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, 2001, pp. 377–382.

[67] *An ensemble classifier for drifting concepts*. M. Scholzand, R. Klinkenberg. In Proc. 2nd Int. Workshop Knowledge Discovery Data Streams, 2005, pp. 53-64.

[68] *Online nonstationary boosting*. A. Pocock, P. Yiapanis, J. Singer, M. Lujan, and G. Brown. In Proc. Int. Workshop Multiple Classifier Systems, 2010, pp. 205–214.

[69] *Online ensemble learning*. N. Oza. Ph.D. dissertation, Univ. California, Berkeley, CA, 2001.

[70] *New ensemble methods for evolving data streams*. A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R.Gavalda. In Proc. Knowledge Data Discovery, 2009, pp. 139–148.

[71] *Improving adaptive bagging methods for evolving data streams*. A. Bifet, G. Holmes, B. Pfahringer, and R. Gavalda. In Proc. 1st Asian Conf. Machine Learning: Advances Machine Learning, 2009, pp. 27–37.

[72] *Dynamic weighted majority: An ensemble method for drifting concepts*. J. Kolter and M. Maloof. J. Mach. Learn. Res., vol. 8, pp. 2755–2790, Dec. 2007.

[73] *The weighted majority algorithm*. N. Littlestone and M. K. Warmuth. Inform. Comput., vol. 108, no. 2, pp. 212–261, Feb. 1994.

[74] *Reacting to different types of concept drift: The accuracy updated ensemble algorithm*. D. Brzezinski and J. Stephanowski. IEEE Trans. Neural Netw. Learn. Syst., vol. 25, no. 1, pp. 81–94, Jan. 2014.

[75] Classification using streaming random forests. H. Abdulsalam, D. Skillicorn, and P. Martin. IEEE Trans. Knowledge Data Eng., vol. 23, no. 1, pp. 22–36, Jan. 2011.

[76] *A decision-theoretic generalization of online learning and an application to boosting*. Y. Freundand, R. Shapire. J. Comput. Syst. Sci., vol. 55, pp. 119–139, Aug. 1997.

[77] *Incremental learning in nonstationary environments with controlled forgetting*. R. Elwell and R.Polikar. In Proc. Int. Joint Conf. Neural Networks, 2009, pp. 771–778.

[78] *Learning from imbalanced data*. H. He and E. A. Garcia. IEEE Trans. Data Knowledge Discovery, vol. 12, no. 9, pp. 1263–1284, Sept. 2009.

[79] *SMOTE: Synthetic minority over-sampling technique*. N. V. Chawla, K. W.Bowyer, L. O. Hall, and W. P. Kegelmeyer. J. Artif. Intell. Res., vol. 16, pp. 321–357, June 2002.

[80] *Editorial: Special issue on learning from imbalanced data sets*. N. V. Chawla, N. Japkowicz, and A. Kolcz. SIGKDD Expl., vol. 6, no. 1, pp. 1–6, June 2004.

[81] *Classifying data streams with skewed class distributions and concept drifts*. J. Gao, B. Ding, W. Fan, J. Han, and P. S. Yu. IEEE Internet Comput., vol. 12, no. 6, pp. 37–49, Nov.–Dec. 2008.

[82] *A general framework for mining concept-drifting data streams with skewed distributions*. J. Gao, W. Fan, J. Han, and P. S. Yu. In Proc. SIAM Int. Conf. Data Mining, 2007, pp. 203–208.

[83] *SERA: Selectively recursive approach towards nonstationary imbalanced stream data mining*. S. Chen and H. He. In Proc. Int. Joint Conf. Neural Networks, 2009, pp. 552–529.

[84] *Towards incremental learning of nonstationary imbalanced data stream: A multiple selectively recursive approach*. S. Chen and H. He Evolving Syst., vol. 2, no. 1, pp. 35–50, Mar. 2011.

[85] *Incremental learning of concept drift from streaming imbalanced data*. G. Ditzler and R.Polikar. IEEE Trans. Knowledge Data Eng., vol. 25, no. 10, pp. 2283–2301, Oct.2013.

[86] *An incremental learning framework for concept drift and class imbalance*. G. Ditzler and R. Polikar. In Proc. Int. Joint Conf. Neural Networks, 2010, pp. 736–473.

[87] Using class imbalance learning for software defect prediction. S. Wangand, X. Yao. IEEE Trans. Reliab., vol. 62, no. 2, pp. 434–443, June 2013.

[88] *Resampling-based ensemble methods for online class imbalance learning*. S. Wang, L. L. Minku, and X. Yao IEEE Trans. Knowledge Data Eng., vol. 27, no. 5, pp. 1356–1368, May 2015.

[89] *Dealing with concept drift and class imbalance in multi-label stream classification*. E. S. Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas. In Proc. Int. Joint Conf. Artificial Intelligence, 2011, pp. 1583–1588.

[90] *Incremental learning and model selection under virtual concept drifting environments*. K. Yamauchi. In Proc. Int. Joint Conf. Neural Networks, 2010, pp. 1–8.

[91] *Transductive learning algorithms for nonstationary environments*. G. Ditzler, G. Rosen, and R. Polikar. In Proc. Int. Joint Conf. Neural Networks, 2012, pp. 1–8.

[92] Semi-supervised learning in nonstationary environments. G. Ditzler and R. Polikar. In Proc. Int. Joint Conf. Neural Networks, 2011, pp. 2741–2748.

[93] *Density-based clustering over an evolving data stream with noise*. F. Cao, M. Ester, W. Qian, and A. Zhou. In Proc. SIAM Conf. Data Mining, 2006, pp. 328–339.

[94] *Active learning with drifting streaming data*. I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes. IEEE Trans. Neural Networks Learn. Syst., vol. 25, no. 1, pp. 27–39, Jan. 2014.

[95] *Classifier and cluster ensembles for mining concept drifting data streams*. P. Zhang, X. Zhu, J. Tan, and L. Guo. In Proc. Int. Conf. Data Mining, 2010, pp. 1175–1180.

[96] *Classification in presence of drift and latency*. G. Krempl and V. Hofer. In Proc. IEEE Int. Conf. Data Mining Workshops, 2011, pp. 596–603.

[97] *Class and subclass probability re-estimation to adapt a classifier in the presence of concept drift*. R. Alaiz-Rodriguez, A. Guerrero-Curieses, and J.Cid-Sueiro. Neurocomputing, vol. 74, no. 16, pp. 2614–2623, Sept. 2011.

[98] *Drift mining in data: A framework for addressing drift in classification*. V. Hofer and G. Krempl. Comput. Stat. Data Anal., vol. 57, no. 1, pp. 377–391, Jan. 2013.

[99] *The algorithm apt to classify in concurrence of latency and drift*. G. Krempl. Adv. Intell. Data Anal., vol. 7014, pp. 222–233, 2011.

[100] *Semi-supervised learning in initially labeled non-stationary environments with gradual drift*. K. Dyer and R. Polikar. In Proc. Int. Joint Conf. Neural Networks, 2012, pp. 1–9.

[101] *COMPOSE: A semi-supervised learning framework for initially labeled non-stationary streaming data*. K. B.Dyer, R. Capo, and R. Polikar. IEEE Trans. Neural Networks Learn. Syst., vol.25, no. 1, pp. 12–26, Jan. 2013.

[102] *Active learning in nonstationary environments*. R. Capo, K. B. Dyer, and R. Polikar. In Proc. Int. Joint Conf. Neural Networks, 2013, pp. 1–8.

[103] *Randomized algorithms for analysis and control of uncertain systems*. Tempo R., Calafiore G., Dabbene F. Springer, Berlin, 2005.

[104] *A Gaussian approximation to the distribution of the sample variance for non-normal populations*. Mudholkar G. S., Trivedi M. C. J. Am. Stat. Assoc. 76 (374), pp. 479–485, 1981.

[105] *On spatial adaptive estimation of nonparametric regression*. Goldenshluger A., Nemirovski A. Math. Meth. Stat. 6, pp. 135–170, 1997.

[106] *Detecting and reacting to changes in sensing units: the active classifier case*. Alippi C., Bu D. L., Zhao D. IEEE Trans. Syst. Man. Cybern. — Part A: Syst. Hum., 2013.

[107] *Detection of Abrupt Changes: Theory and Application*. Michele Basseville and Igor V. Nikiforov. Prentice-Hall, 1993.