



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ
ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ

Αλγόριθμοι εκπαίδευσης για Hidden Markov Models

Ιωάννης Μηντσόπουλος

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Υπεύθυνοι

Παντελεήμων Μπάγκος

Αναπληρωτής Καθηγητής

Βασίλειος Πλαγιανάκος

Αναπληρωτής Καθηγητής

Λαμία, 1/6 έτος 2016



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ
ΒΙΟΙΑΤΡΙΚΗ**

Αλγόριθμοι εκπαίδευσης για Hidden Markov Models

Ιωάννης Μηντσόπουλος

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιβλέποντες

**Παντελεήμων Μπάγκος
Αναπληρωτής Καθηγητής
Βασίλειος Πλαγιανάκος
Αναπληρωτής Καθηγητής**

Λαμία, 1/6 έτος 2016

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 1/6/2016

Ο – Η Δηλ.

(Υπογραφή)

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.

Αλγόριθμοι εκπαίδευσης για Hidden Markov Models

Ιωάννης Μηντσόπουλος

Τριμελής Επιτροπή:

Μπάγκος Παντελεήμων, Αναπληρωτής Καθηγητής του Τμήματος Πληροφορικής με Εφαρμογές στην Βιοϊατρική του Πανεπιστημίου Θεσσαλίας (επιβλέπων)

Πλαγιανάκος Βασίλειος, Αναπληρωτής Καθηγητής του Τμήματος Πληροφορικής με Εφαρμογές στην Βιοϊατρική του Πανεπιστημίου Θεσσαλίας (επιβλέπων)

Μάρκου Ευριπίδης, Επίκουρος Καθηγητής του Τμήματος Πληροφορικής με Εφαρμογές στην Βιοϊατρική του Πανεπιστημίου Θεσσαλίας

Πίνακας Περιεχομένων

ΠΕΡΙΛΗΨΗ	8
ABSTRACT	9
ΠΡΟΛΟΓΟΣ	10
ΚΕΦΑΛΑΙΟ 1: Hidden Markov Models	11
1.1: Εισαγωγή.....	11
1.2: Ορισμοί	12
1.3: Παραδείγματα	13
1.4: Τα 3 βασικά ερωτήματα σε ένα HMM.....	15
1.5: Υπολογισμός πιθανοφάνειας	15
1.5.1: Αλγόριθμος Forward	16
1.5.2: Αλγόριθμος Backward.....	17
1.6: Αποκωδικοποίηση	17
1.6.1: Αλγόριθμος Viterbi.....	17
1.7: Εκτίμηση Παραμέτρων Των HMM.....	18
1.7.1: Ο Αλγόριθμος Baum-Welch	19
1.7.2: Μέθοδοι Gradient-descent	20
ΚΕΦΑΛΑΙΟ 2: Class Hidden Markov Models.....	23
2.1: Ορισμοί	23
2.2: Πιθανοφάνεια	24
2.3: Εκτίμηση παραμέτρων.....	26
Κεφάλαιο 3: Εκτίμηση Παραμέτρων με την Χρήση του Viterbi Training	29
3.1: Γενική περιγραφή του viterbi training.....	29
3.2: Τροποποιημένος αλγόριθμος Viterbi	30
3.3: Viterbi Training	30
ΚΕΦΑΛΑΙΟ 4: Αποτελέσματα και Συζήτηση	33
4.1: Υλοποίηση	33
4.2: Αποτελέσματα	40
4.3: Συζήτηση	42
Βιβλιογραφία.....	44

ΠΕΡΙΛΗΨΗ

Αντικείμενο της πτυχιακής εργασίας αποτελούν τα κρυπτομαρκοβιανά μοντέλα εξάρτησης Hidden Markov Models, και η επέκταση αυτών, τα μοντέλα για σημασμένες ακολουθίες Class Hidden Markov Models. Στα πλαίσια της εργασίας αναπτύχθηκε ένας νέος αλγόριθμος για την εύρεση των εκτιμητών μέγιστης πιθανοφάνειας, ο αλγόριθμος Viterbi training, ο οποίος εφαρμόζεται κατά την διαδικασία της εκτίμησης των παραμέτρων. Ο αλγόριθμος υλοποιήθηκε με την γλώσσα προγραμματισμού java, και ενσωματώθηκε σε ένα πρόγραμμα, το οποίο πραγματοποιεί εύρεση από κοινού πιθανότητας, αποκωδικοποίηση, εκτίμηση παραμέτρων, και πρόβλεψη.

Με την ανάπτυξη του αλγορίθμου Viterbi training καταφέραμε να μειώσουμε τον χρόνο εκπαίδευσης σημαντικά, με μία μικρή μείωση της αποδοτικότητας του μοντέλου. Συγκεκριμένα, με την χρήση του αλγορίθμου μας, κατά την διαδικασία εκτίμησης των παραμέτρων από τους αλγορίθμους Gradient-Descent και Baum-Welch, η χρήση των αλγορίθμων Forward και Backward για τον υπολογισμό των εκτιμητών μέγιστης πιθανοφάνειας αντικαθίσταται από τον Viterbi training.

Στα πλαίσια της εργασίας αυτής, πεδίο εφαρμογής των μοντέλων αποτελεί ο τομέας της βιοπληροφορικής, και συγκεκριμένα η πρόβλεψη πρωτεϊνών. Η εκπαίδευση του μοντέλου πραγματοποιήθηκε από ένα σύνολο πρωτεϊνών γνωστής δομής (training set).

Τέλος, συγκρίνουμε την αποδοτικότητα του μοντέλου, όπως αυτή προκύπτει με την εκτίμηση των παραμέτρων να γίνεται με την χρήση των Forward και Backward με την αποδοτικότητα του μοντέλου, όταν πραγματοποιείται εκτίμηση παραμέτρων με την χρήση του Viterbi training.

ABSTRACT

The objective of this dissertation pertains to the Hidden Markov Models and their extension, the so-called Class Hidden Markov Models. As part of the dissertation, a new algorithm about finding the maximum likelihood estimates has been developed. It is the Viterbi training algorithm which is applied during the process of parameter estimation. The algorithm has been implemented in Java programming language and incorporated into a program that carries out joint probability finding, decoding, parameter estimation and prediction.

Developing the Viterbi training algorithm, we have managed to diminish considerably the time spent on training, with a small decrease in the model's efficiency. In particular, during the process of parameter estimation with the Gradient-Descent and Baum-Welsh algorithms, the use of the Forward and Backward algorithms for the calculation of the maximum likelihood estimates is replaced by the Viterbi training algorithm.

In this dissertation, scope of the models is the field of Bioinformatics and particularly the protein precipitation. The training of the model has been realized by a protein training set of known structure.

Finally, the model's efficiency in the parameter estimation while using the Forward and Backward algorithms, is compared to the model's efficiency when the parameter estimation is carried out making use of the Viterbi training algorithm.

ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία με θέμα Αλγόριθμοι εκπαίδευσης για Hidden Markov Models πραγματοποιήθηκε εξ ολοκλήρου στο Τμήμα Πληροφορικής με Εφαρμογές στην Βιοϊατρική στη Σχολή Θετικών Επιστημών του Πανεπιστημίου Θεσσαλίας, κατά την διάρκεια του ακαδημαϊκού έτους 2015-2016, υπό την επίβλεψη του Αναπληρωτή Καθηγητή Παντελεήμων Μπάγκου και του Αναπληρωτή Καθηγητή Βασιλείου Πλαγιανάκου.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω ιδιαίτερα τους επιβλέποντες καθηγητές μου, για την καθοδήγηση που μου παρείχαν κατά την διάρκεια εκπόνησης της εργασίας μέσω υποδείξεων, παρατηρήσεων, υλικού, και συμβουλών. Επίσης, θέλω να τους ευχαριστήσω και για την ευκαιρία που μου έδωσαν να ασχοληθώ με ένα τόσο ενδιαφέρον αντικείμενο. Επίσης, ευχαριστώ τον συμφοιτητή μου Στράτο Τσότρα, για την βοήθεια που μου παρείχε κατά την διάρκεια της συγγραφής του κειμένου.

Τέλος, ευχαριστώ πάρα πολύ την οικογένεια μου, για την στήριξη που μου έχει προσφέρει σε όλο τον ακαδημαϊκό μου βίο.

ΚΕΦΑΛΑΙΟ 1: Hidden Markov Models

Σύνοψη

Στο κεφάλαιο αυτό θα γίνει η παρουσίαση των πολύ σημαντικών για την βιοπληροφορική εργαλείων των Hidden Markov Models, περιγράφοντας αναλυτικά το μαθηματικό υπόβαθρο, τις ιδιότητες, τα χαρακτηριστικά αυτών, και τους αλγορίθμους μέσω των οποίων γίνεται ο υπολογισμός της πιθανοφάνειας, της αποκωδικοποίησης και της εκτίμησης των παραμέτρων στα μοντέλα αυτά.

1.1: Εισαγωγή

Στο κεφάλαιο αυτό θα γίνει η παρουσίαση-ανάλυση των κρυπτομαρκοβιανών μοντέλων (Hidden Markov Models). Τα Hidden Markov Models ως μέλη της οικογένειας των Markov Models είναι στοχαστικά-πιθανοθεωρητικά, μαθηματικά μοντέλα. Η δόμηση των Hidden Markov Models από κρυφές καταστάσεις, οι οποίες δεν βρίσκονται σε ένα προς ένα αντιστοιχία με τα σύμβολα της εκάστοτε ακολουθίας, δεν αποτελεί απλά ένα διακριτικό στοιχείο έναντι των άλλων Markov Models, προσδίδει σε αυτά ένα διαφορετικό τρόπο λειτουργίας, ο οποίος τα καθιστά ικανότερα για την περιγραφή ενός μεγάλου πλήθους συστημάτων. Τα HMMs (Hidden Markov Models) έχουν πολλά πεδία εφαρμογής όπως η επεξεργασία εικόνας, ήχου, σήματος, βιοπληροφορική κ.α. και γενικά χρησιμοποιούνται ως pattern recognition methods. Η δομή του εκάστοτε HMM βρίσκεται σε πλήρη εξάρτηση με το προς περιγραφή σύστημα, ενώ η πραγμάτωση (δημιουργία και λειτουργία) αυτού απαιτεί υπολογιστικούς πόρους (μνήμη, υπολογιστική ισχύ κ.α.). Από τα παραπάνω συμπεραίνουμε πως για την πραγμάτωση αυτή είναι αναγκαία η χρήση (H/Y). Για τα μοντέλα Markov σε κάθε σύστημα υπάρχει ένα αλφάβητο και τα σύμβολα αυτού, για παράδειγμα, σε μία πρωτεΐνη το αλφάβητο αποτελείται από τα 20 αμινοξέα και ως σύμβολο ορίζεται ένα αμινοξύ, όπως και στην περίπτωση μίας φυσικής γλώσσας τα γράμματα του εκάστοτε αλφαβήτου αποτελούν τα σύμβολα.

Η εξάρτηση των συμβόλων σε ένα κωδικοποιημένο σύστημα αποτέλεσε την βασική ιδέα ανάπτυξης των μοντέλων Markov, ένα παράδειγμα εξάρτησης συναντάμε συναντάμε στην αγγλική γλώσσα όπου το γράμμα Q ακολουθείται σχεδόν πάντα από το γράμμα U, οπότε η πιθανότητα να εμφανιστεί το U σε κάποια θέση δεν είναι ίδια πάντα, αλλά εξαρτάται από το αν προηγουμένως εμφανίστηκε το γράμμα Q.

Για την ιστορία, ο Ρώσος μαθηματικός Andrey Markov (1856-1922) εμπνεύστηκε τα ομώνυμα μοντέλα, μελετώντας τις εναλλαγές των συμφώνων και φωνηέντων σε ένα ποίημα του Pushkin (Markov, 1913).

1.2: Ορισμοί

Για την δημιουργία ενός Hidden Markov Model απαιτούνται, ένα σύνολο κρυφών καταστάσεων $\Omega = \{\omega_1, \omega_2, \omega_3, \dots, \omega_m\}$, ένα σύνολο παρατηρούμενων συμβόλων και δύο σύνολα πιθανοτήτων, οι πιθανότητες μετάβασης των καταστάσεων του μοντέλου (transitions) και οι πιθανότητες γεννήσεως ή εκπομπής των συμβόλων (emissions). Θεωρώντας μία πρωτεύινη ακολουθία \mathbf{x} μήκους L καταλοίπων:

$$\mathbf{X} = x_1, x_2, \dots, x_{L-1}, x_L \quad (1.1)$$

Με x_i συμβολίζουμε ένα παρατηρούμενο σύμβολο της ακολουθίας, δηλαδή ένα εκ των 20 αμινοξέων. Σε ένα HMM δεν υπάρχει ένα προς ένα αντιστοιχία των καταλοίπων της εκάστοτε ακολουθίας με τις καταστάσεις του μοντέλου. Η αλληλουχία των καταστάσεων στην οποία έχει περιέλθει το μοντέλο έως την θέση i της αλληλουχίας, συνηθίζεται να ονομάζεται ως μονοπάτι (path) και συμβολίζεται με π_i . Για κάθε κατάσταση του μοντέλου υπάρχουν οι πιθανότητες μετάβασης προς τις άλλες καταστάσεις του μοντέλου, έτσι, δύο καταστάσεις k, l συνδέονται μεταξύ τους μέσω των πιθανοτήτων μετάβασης a_{kl} , δημιουργώντας μια αλυσίδα 1^{ης} τάξης. Ο τυπικός ορισμός των πιθανοτήτων μεταβάσεως είναι ο ακόλουθος:

$$a_{kl} = P(\pi_i = l / \pi_{i-1} = k) \quad (1.2)$$

και συμβολίζει την πιθανότητα η κατάσταση l να καταλάβει την θέση i στην αλληλουχία των καταστάσεων, εφόσον η κατάσταση k κατέχει την θέση $i-1$, πιο απλά, την πιθανότητα να μεταβούμε από την κατάσταση k στην l . Όπως και σε ένα απλό Markov Model, έχουμε την δυνατότητα να θέσουμε καταστάσεις ενάρξεως και τερματισμού της ακολουθίας των καταστάσεων, οι οποίες εν συντομία ονομάζονται B (begin) και E (end) αντίστοιχα.

$$a_{Bk} = P(\pi_i = k / B) \quad (1.3)$$

$$a_{kE} = P(E / \pi_i = k) \quad (1.4)$$

Η σχέση (1.3) συμβολίζει την πιθανότητα μετάβασης από την κατάσταση B (begin) στην κατάσταση k , και η (1.4) την πιθανότητα μετάβασης από την κατάσταση k στην κατάσταση E (end). Το δεύτερο σύνολο πιθανοτήτων των καταστάσεων είναι οι πιθανότητες εκπομπής ή γεννήσεως:

$$e_k(b) = P(x_i = b / \pi_i = k) \quad (1.5)$$

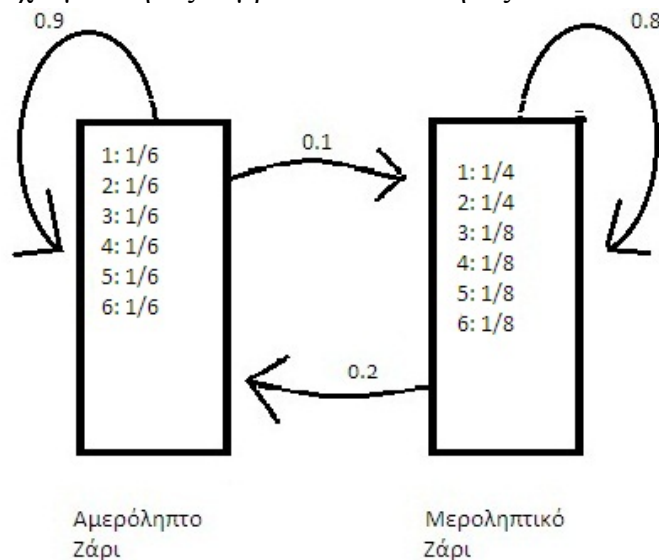
σχετίζουν την ακολουθία των συμβόλων (η οποία προέρχεται από το προς περιγραφή σύστημα), με τις καταστάσεις του μοντέλου, έτσι, η σχέση (1.5) εκφράζει την πιθανότητα εμφάνισης στην θέση i της ακολουθίας, ενός συγκεκριμένου συμβόλου b (αμινοξύ, καθώς η ακολουθία των συμβόλων είναι πρωτεύινη), εφόσον το σύστημα βρίσκεται την κατάσταση k . Η από κοινού πιθανότητα μιας ακολουθίας \mathbf{x} και του μονοπατιού π υπολογίζεται ως εξής:

$$P(\mathbf{x}, \pi) = P(x_L, x_{L-1}, \dots, x_1, \pi) = a_{B\pi_1} \prod_{i=1}^L e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}} \quad (1.6)$$

1.3: Παραδείγματα

Παράδειγμα 1

Ένα απλό παράδειγμα για την εύκολη επεξήγηση ενός HMM είναι το παράδειγμα του “ανέντιμου καζίνο”, για το οποίο υπάρχει αναφορά στην βιβλιογραφία (Durbin, et al., 1998). Στο παράδειγμα αυτό, το καζίνο χρησιμοποιεί κατά το δοκούν δύο ζάρια, ένα αμερόληπτο, στο οποίο οι πιθανότητες εμφάνισης των συμβόλων (1,2,..6) καθορίζονται από τον παράγοντα τύχη, και ένα δεύτερο μεροληπτικό ζάρι, στο οποίο υπάρχει μεροληπτική-προκαθορισμένη κατανομή των πιθανοτήτων εμφάνισης. Στην εικόνα 1.1 τα δύο ζάρια απεικονίζονται ως δύο καταστάσεις, οι πιθανότητες εμφάνισης των συμβόλων αναγράφονται στο εσωτερικό των καταστάσεων, και οι πιθανότητες μετάβασης στα βέλη του σχήματος. Βάσει του παραδείγματος αυτού, όταν ο παίχτης παίζει με το αμερόληπτο ζάρι, η πιθανότητα να πετύχει τον αριθμό π.χ. 5 είναι $1/6$, ενώ αν παίζει με το μεροληπτικό $1/8$. Η ευχέρεια του καζίνο να αλλάζει το ζάρι εξαρτάται από τις πιθανότητες μετάβασης έτσι, π.χ. η εναλλαγή από το αμερόληπτο σε μεροληπτικό γίνεται με πιθανότητα 0.1. Από την πλευρά του ο παίκτης, όταν βλέπει ένα αποτέλεσμα π.χ. τον αριθμό 2, δεν γνωρίζει από ποια κατάσταση-ζάρι προήλθε, γι αυτόν τον λόγο το μοντέλο ονομάζεται κρυμμένο (Hidden). Όπως έχει αναφερθεί σε ένα HMM δεν υπάρχει ένα προς ένα αντιστοίχιση των συμβόλων με τις καταστάσεις, σε αντίθεση με τα απλά Markov models. Ως εκ τούτου, από την εκάστοτε κατάσταση μπορεί να εμφανιστεί δυνητικά το οποιοδήποτε σύμβολο του αλφαβήτου και ο αριθμός των καταστάσεων μπορεί να είναι διαφορετικός από το πλήθος των συμβόλων της ακολουθίας. Στο παράδειγμά μας έχουμε πλήθος συμβόλων 6 και πλήθος καταστάσεων 2.



Εικόνα 1.1 Η απεικόνιση του ανέντιμου καζίνο. Ως καταστάσεις ορίζονται τα παραλληλόγραμμα, έχοντας μία κατάσταση για το αμερόληπτο ζάρι και μία για το μεροληπτικό. Μέσα σε κάθε κατάσταση απεικονίζονται οι πιθανότητες εμφάνισης των συμβόλων, ενώ τα εξωτερικά βέλη συμβολίζουν τις πιθανότητες μετάβασης.

Παράδειγμα 2

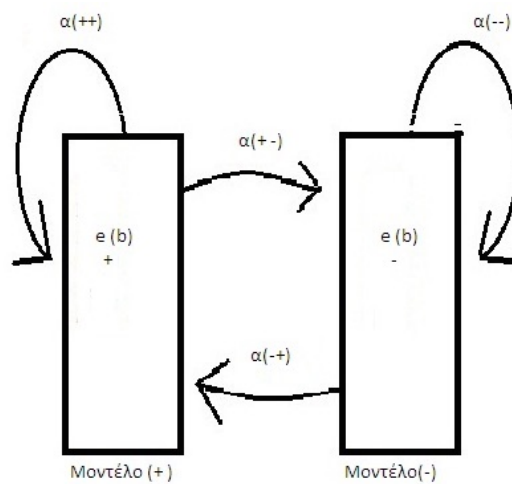
Στο παράδειγμα αυτό παρουσιάζεται η χρήση μοντέλου HMM στην βιοπληροφορική και συγκεκριμένα σε διαμεμβρανικές πρωτεΐνες.

Έχουμε μία ακολουθία αμινοξέων:

.....A D E R P G G S D A D K **L I L V C I G F V L I F** V T Y T.....

.....- - - - - + + + + + + + + + + + - - - -.....

Στην πρώτη γραμμή έχουμε μία ακολουθία αμινοξέων διαμεμβρανικής πρωτεΐνης, όπου το εικονιζόμενο διαμεμβρανικό τμήμα της έχει τονιστεί με κόκκινο χρωματισμό. Στην δεύτερη γραμμή παρουσιάζεται μία αλληλουχία καταστάσεων του μοντέλου μας (Εικόνα 1.2).



Εικόνα 1.2

Η εξήγηση του HMM στην περίπτωση των διαμεμβρανικών τμημάτων είναι πολύ απλή, και θα μας βοηθήσει στην κατανόηση της εφαρμογής των μοντέλων HMMs στα βιολογικά συστήματα. Είναι γνωστό ότι τα διαμεμβρανικά τμήματα των πρωτεϊνών αποτελούνται κυρίως από υδρόφοβα αμινοξέα, επομένως, η πιθανότητα να εμφανιστεί π.χ. λευκίνη (υδρόφοβο αμινοξύ) σε ένα διαμεμβρανικό τμήμα είναι μεγαλύτερη από την πιθανότητα εμφάνισης σε ένα μη διαμεμβρανικό. Επίσης, η διαδοχή στο διαμεμβρανικό κομμάτι της πρωτεΐνης είναι πιθανότερη να πραγματοποιηθεί από υδρόφοβο αμινοξύ. Ως εκ τούτου, το μοντέλο (+) της εικόνας 1.2 έχει αυξημένες πιθανότητες εμφάνισης υδρόφοβων αμινοξέων.

1.4: Τα 3 βασικά ερωτήματα σε ένα HMM

Στο σημείο αυτό παρατίθενται τα τρία βασικά ερωτήματα, τα οποία μπορούμε να θέσουμε σε ένα HMM όπως διατυπώθηκαν από τον Rabiner (Rabiner, 1989).

- Πώς μπορούμε να υπολογίσουμε την συνολική πιθανότητα μία ακολουθία \mathbf{x} να έχει εμφανιστεί από ένα καθορισμένο μοντέλο θ ; Δηλαδή, ποιά είναι η τιμή $P(\mathbf{x}|\theta)$;
- Δεδομένου ενός μοντέλου θ και μίας ακολουθίας \mathbf{x} , πως μπορούμε να υπολογίσουμε το πιθανότερο μονοπάτι π , δηλαδή την πιθανότερη αλληλουχία καταστάσεων από την οποία διήλθε το μοντέλο; Δηλαδή, πως μπορούμε να υπολογίσουμε το μονοπάτι π έτσι ώστε:

$$\pi^{max} = \arg \max_{\pi} P(\mathbf{x}, \pi)$$

- Δεδομένου ενός μοντέλου θ , πως μπορούμε να μεταβάλουμε τις παραμέτρους αυτού, με την χρήση νέων δεδομένων, ώστε το μοντέλο να γίνει καλύτερο; Το πρόβλημα αυτό είναι γνωστό ως εκτίμηση παραμέτρων, επομένως, αναζητούμε τον υπολογισμό των παραμέτρων, έτσι ώστε:

$$\theta^{ML} = \arg \max_{\theta} P(\mathbf{x} | \theta)$$

Στα τρία αυτά ερωτήματα στηρίζονται, τόσο η εκπαίδευση, όσο και η λειτουργία των Hidden Markov Models.

1.5: Υπολογισμός πιθανοφάνειας

Όπως έχει ήδη προαναφερθεί στους ορισμούς, η σχέση (1.6) είναι η από κοινού πιθανότητα μίας ακολουθίας \mathbf{x} και του μονοπατιού π . Η σχέση αυτή δεν είναι ικανή να απαντήσει στο ερώτημα της ευρέσεως της πιθανοφάνειας, διότι είναι αδύνατον να γνωρίζουμε το μονοπάτι από το οποίο διήλθε το μοντέλο για την παραγωγή της ακολουθίας συμβόλων. Για την εύρεση της συνολικής πιθανότητας μίας ακολουθίας συμβόλων και ενός μοντέλου, θα πρέπει να αθροίσουμε την συνολική πιθανότητα όλων των πιθανών μονοπατιών στα οποία δυνητικά μπορεί να διέλθει το μοντέλο, μέσω της σχέσης:

$$P(x | \theta) = \sum_{\pi} P(\mathbf{x}, \pi | \theta) = \sum_{\pi} a_{B\pi_1} \prod_{i=1}^L e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}} \quad (1.7)$$

Η σχέση αυτή δεν μπορεί να εφαρμοστεί στην πραγματικότητα, διότι το πλήθος των πιθανών μονοπατιών αυξάνεται εκθετικά, όσο το μήκος της ακολουθίας αυξάνεται. Παίρνοντας ως παράδειγμα ένα μοντέλο 60 καταστάσεων και μία ακολουθία 400 συμβόλων, ο αστρονομικός αριθμός των πιθανών μονοπατιών διέλευσης του μοντέλου είναι 60^{400} . Για τον λόγο αυτόν, η σχέση (1.7) κρίνεται ως μη λειτουργική. Την απάντηση στο ερώτημα της πιθανοφάνειας μπορεί να δώσει ο δυναμικός προγραμματισμός μέσω αλγορίθμων. Ο πιο γνωστός αλγόριθμος είναι ο αλγόριθμος Forward (Εικόνα 1.3), ο οποίος αναλύεται διεξοδικά παρακάτω (Durbin, et al., 1998; Rabiner, 1989).

1.5.1: Αλγόριθμος Forward

$$\forall k \neq B, i = 0: f_B(0) = 1, f_k(0) = 0$$

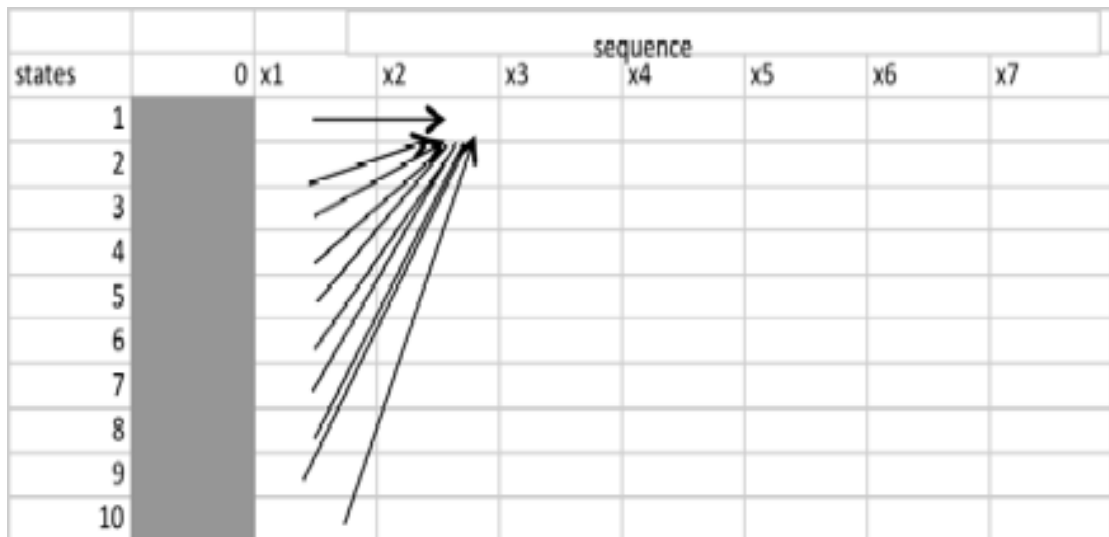
$$\forall 1 \leq i \leq L: f_l(i) = e_i(x_i) \sum_k f_k(i-1) a_{kl} \quad (1.8)$$

$$P(x | \theta) = \sum_k f_k(L) a_{kE} \quad (1.9)$$

Η τιμή της $f_l(i)$ ισούται με την πιθανότητα εμφάνισης του καταλοίπου x_i από την κατάσταση l , επί το άθροισμα για όλες τις καταστάσεις, της τιμής $f_k(i-1)$ επί την πιθανότητα μετάβασης της εκάστοτε κατάστασης στην l . Στο τέλος αθροίζεται για όλες τις καταστάσεις, η τιμή της $f_k(l)$ επί την πιθανότητα μετάβασης της εκάστοτε κατάστασης προς την κατάσταση λήξεως (End), αν αυτή υπάρχει.

Ο αλγόριθμος αυτός, δημιουργεί έναν διδιάστατο πίνακα $N(L+1)$, όπου ο αριθμός N αντιστοιχεί στο πλήθος των καταστάσεων του μοντέλου και ο L στο μήκος της ακολουθίας, και χρησιμοποιεί μία μεταβλητή $f_k(i)$ για κάθε θέση i και κατάσταση k της ακολουθίας. Η τιμή της μεταβλητής ισούται με την από κοινού πιθανότητα της ακολουθίας μέχρι το κατάλοιπό i (π.χ. για το κατάλοιπο x_4 $i=4$) και του μονοπατιού που αντιστοιχεί στην κατάσταση k . Ο τυπικός ορισμός είναι:

$$f_k(i) = P(x_1, x_2, \dots, x_i, \pi_i = k) \quad (1.10)$$



Εικόνα 1.3 Η αναπαράσταση του πίνακα Forward, για ένα μοντέλο 10 καταστάσεων (states) και μία ακολουθία 7 καταλοίπων. Τα βέλη δείχνουν την συνεισφορά των καταστάσεων στον υπολογισμό της τιμής $f_1(2)$

Ο αλγόριθμος υλοποιείται σε τρία βήματα(steps) στο πρώτο βήμα γίνεται η αρχικοποίηση του πίνακα $f_k(i)$, στο δεύτερο συμπληρώνονται οι τιμές του με κατεύθυνση από το κατάλοιπο x_1 προς το κατάλοιπο x_L (σύμφωνα με την σχέση (1.8)), και στο τρίτο βήμα αθροίζονται για να προκύψει η τελική πιθανοφάνεια σύμφωνα με την σχέση (1.9). Αν δεν υπάρχει κατάσταση λήξεως (End), τότε οι πιθανότητες απλώς απαλείφονται. Η πολυπλοκότητα του αλγορίθμου είναι $O(NL)$, διότι απαιτούνται NL υπολογισμοί.

1.5.2: Αλγόριθμος Backward

$$\forall k, i = L: b_k(L) = a_{kE}$$

$$\forall 1 \leq i \leq L: b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1) \quad (1.11)$$

$$P(\mathbf{x} | \theta) = \sum_l a_{Bl} e_l(x_1) b_l(L) \quad (1.12)$$

Ο αλγόριθμος Backward (Durbin, et al.,1998; Rabiner,1989) είναι παρόμοιος με τον Forward, η μοναδική διαφορά τους είναι η κατεύθυνση προς την οποία διατρέχει την ακολουθία, δηλαδή, ο Backward ξεκινά να γεμίζει τον πίνακα από το τελευταίο κατάλοιπο της ακολουθίας συμβόλων με κατεύθυνση το πρώτο ($x_L \rightarrow x_1$). Ο Backward χρησιμοποιεί ως ενδιάμεση μεταβλητή την $b_k(i)$, η τιμή της οποίας είναι η από κοινού πιθανότητα από την θέση $i+1$, για κάθε κατάλοιπο που βρίσκεται στην θέση i , εφόσον το μοντέλο βρίσκεται στην κατάσταση k . Επομένως:

$$b_k(i) = P(x_{i+1}, \dots, x_L | \pi_i = k) \quad (1.13)$$

Εάν το μοντέλο δεν έχει κατάσταση λήξης (End), τότε στο βήμα της αρχικοποίησης οι πιθανότητες (μετάβασης προς αυτήν την κατάσταση) παίρνουν την τιμή 1. Το αποτέλεσμα που δίνει ο αλγόριθμος είναι ίδιο με αυτό του Forward.

1.6: Αποκωδικοποίηση

Η εύρεση του πιθανότερου μονοπατιού των καταστάσεων από το οποίο διήλθε το μοντέλο, για την εμφάνιση-παραγωγή της ακολουθίας των παρατηρήσεων, απαντά στο δεύτερο ερώτημα, το οποίο τίθενται σε ένα HMM, και είναι γνωστή ως αποκωδικοποίηση (decoding). Ένας πολύ γνωστός αλγόριθμος αποκωδικοποίησης, ο οποίος είναι παρόμοιος με τους προαναφερθέντες αλγορίθμους πιθανοφάνειας, είναι ο αλγόριθμος Viterbi (Durbin, et al.,1998; Rabiner,1989).

1.6.1: Αλγόριθμος Viterbi

$$\forall k \neq B, i = 0: u_B(0) = 1, u_k(0) = 0$$

$$\forall 1 \leq i \leq L: u_l(i) = e_l(x_i) \max_k \{u_k(i-1) a_{kl}\} \quad (1.14)$$

$$P(\mathbf{x}, \pi^{max} | \theta) = \max_k \{u_k(L) a_{kE}\} \quad (1.15)$$

Ο αλγόριθμος Viterbi είναι παρόμοιος με τον αλγόριθμο Forward, χρησιμοποιώντας τα ίδια δεδομένα με αυτόν για την παραγωγή αποτελέσματος, δηλαδή, τις πιθανότητες μετάβασης και εμφάνισης. Η μοναδική διαφορά αυτών, έγκειται στο γεγονός, ότι στον αλγόριθμο Viterbi τα αθροίσματα αντικαθίστανται από μεγιστοποιήσεις. Η διαφορά αυτή είναι εμφανής ανάμεσα στις σχέσεις (1.8)-(1.14) και (1.9)-(1.1). Με π^{\max} συμβολίζεται το μονοπάτι με την μεγαλύτερη πιθανότητα και η πιθανότητα αυτή συμβολίζεται ως $P(\mathbf{x}, \pi^{\max} / \theta)$. Συμπερασματικά από τα παραπάνω, ισχύει πάντα $P(\mathbf{x}, \pi^{\max} / \theta) \leq P(\mathbf{x} | \theta)$. Ένα επιπλέον χαρακτηριστικό του Viterbi είναι η ανάγκη αποθήκευσης του μονοπατιού, η οποία μπορεί να καλυφθεί με την χρήση διαφόρων δομών δεδομένων (πίνακες, λίστες, ουρές κ.α.).

1.7: Εκτίμηση Παραμέτρων Των HMM

Η αποδοτικότητα ενός HMM εξαρτάται από τις παραμέτρους (πιθανότητες μετάβασης-γεννήσεως) αυτού, για τον λόγο αυτό, οι παράμετροι ενός μοντέλου θα πρέπει να εκτιμώνται “προσαρμόζονται”, ώστε να περιγράφεται όσο το δυνατόν καλύτερα το προς περιγραφή σύστημα. Η εκτίμηση των παραμέτρων ενός πιθανοθεωρητικού-στατιστικού μοντέλου, όπως στην περίπτωση των HMM, είθισται να πραγματοποιείται με την μέθοδο της μέγιστης πιθανοφάνειας (Maximum Likelihood). Οι τιμές των παραμέτρων του μοντέλου θ^{ML} μέσω των οποίων μεγιστοποιείται η συνάρτηση πιθανοφάνειας, θέτονται ως Εκτιμητές Μέγιστης Πιθανοφάνειας. Η συνάρτηση πιθανοφάνειας λειτουργεί ως η από κοινού συνάρτηση των παρατηρήσεων του μοντέλου, δοθέντος των παραμέτρων του μοντέλου, θεωρώντας τις τελευταίες ως τυχαίες μεταβλητές. Επομένως:

$$\theta^{ML} = \arg \max_{\theta} P(\mathbf{x} | \theta) \quad (1.16)$$

Συνήθως, για λόγους υπολογιστικής ευκολίας χρησιμοποιούμε τον λογάριθμο της $l(\mathbf{x} | \theta)$, ο οποίος μεγιστοποιείται στην ίδια περιοχή με αυτή.

$$l(\mathbf{x} | \theta) = \log P(\mathbf{x} | \theta)$$

Εάν χρησιμοποιούμε τον λογάριθμο της πιθανοφάνειας το ζητούμενο είναι να μεγιστοποιήσουμε την τιμή του, ενώ αν χρησιμοποιήσουμε το αντίθετο (αρνητικό λογάριθμο) ο σκοπός είναι η ελαχιστοποίηση αυτού.

Αν οι ακολουθίες που χρησιμοποιούνται είναι πρωτεϊνικές, οι παρατηρήσεις (σύμβολα) είναι τα αμινοξικά κατάλοιπα. Στην περίπτωση όπου οι αλληλουχίες-πρωτεΐνες, οι οποίες μετέχουν στην εκπαίδευση είναι περισσότερες από μία, τότε η συνολική πιθανοφάνεια είναι το γινόμενο των πιθανοφάνειών τους. Επομένως, η συνολική λογαριθμική πιθανοφάνεια είναι το άθροισμα των πιθανοφάνειών όλων των αλληλουχιών. Από το σημείο αυτό και σε όλα τα παρακάτω, η αλληλουχία \mathbf{x} τίθεται ως το σύνολο των δεδομένων εκπαίδευσης.

Στην ιδεώδη (αλλά και μη εφικτή) περίπτωση, κατά την οποία έχουμε την δυνατότητα να γνωρίζουμε τα ακριβή μονοπάτια στα οποία διήλθε το μοντέλο για τις αλληλουχίες εκπαίδευσης, ο υπολογισμός των ΕΜΠ είναι πολύ απλός. Θα πρέπει απλούστατα να μετρήσουμε πόσες φορές πραγματοποιήθηκε μία συγκεκριμένη μετάβαση

από κάθε κατάσταση, και πόσες φορές κάθε κατάλοιπο-αμινοξύ γεννήθηκε από κάθε κατάσταση. Επομένως, οι ΕΜΠ για τις πιθανότητες μετάβασης θα είναι:

$$\hat{a}_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \quad (1.17)$$

$$\hat{e}_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b)} \quad (1.18)$$

Όπου τα αθροίσματα στους παρανομαστές περιλαμβάνουν όλη την έκταση των παραμέτρων. Ένα πρόβλημα που μπορεί να δημιουργηθεί με αυτή την προσέγγιση, είναι να έχει κάποια παράμετρος μηδενική τιμή, γεγονός που θα οδηγήσει αυτόματα στην εμφάνιση μηδενικών πιθανοτήτων. Το πρόβλημα αυτό λύνεται με την πρόσθεση ψευδοτιμών.

1.7.1: Ο Αλγόριθμος Baum-Welch

Ο αλγόριθμος Baum-Welch εφαρμόζεται στην διαδικασία της εκτίμησης των παραμέτρων. Όπως έχει προαναφερθεί παραπάνω, είναι σύνηθες να μην γνωρίζουμε τις αλληλουχίες των καταστάσεων από τις οποίες διήλθε το μοντέλο, κατά την εκπαίδευση. Βάσει αυτής της παραδοχής το πρόβλημα γίνεται διττό, διότι τώρα εκτός από την εκτίμηση των παραμέτρων, θα πρέπει ταυτόχρονα να εκτιμηθούν και τα μονοπάτια. Έτσι, ο Baum και η ερευνητική του ομάδα πρότειναν τον αλγόριθμο Baum-Welch (Baum, 1972). Ο αλγόριθμος αποτελεί ειδική περίπτωση του Expectation-Maximisation (EM) (Dempster, Laird, & Rubin, 1977), ο οποίος έχει προταθεί ως μία γενική μέθοδος εκτίμησης παραμέτρων με ελλείπουσες τιμές. Από το σημείο αυτό θα γίνει η παρουσίαση του Baum-Welch "υπό όρους" EM. Το κύριο χαρακτηριστικό του EM είναι η εκτίμηση παραμέτρων με ελλείπουσες τιμές, στο πρόβλημα μας οι ελλείπουσες τιμές είναι οι άγνωστες καταστάσεις π (μονοπάτια). Στην παράγραφο αυτή με $\theta, \theta^t, \theta^{t+1}, \dots$, συμβολίζονται οι παράμετροι για κάθε επανάληψη, κατά την διαδικασία της εκπαίδευσης, και με \mathbf{x} το σύνολο των αλληλουχιών.

Στον Baum-Welch για τις μεταβάσεις έχουμε:

$$A_{kl} = \sum_{\pi} P(\pi | \mathbf{x}, \theta') A_{kl}(\pi) = \frac{1}{P(\mathbf{x})} \sum_i f_k(i) a_{kl} e_l(x_{i+1}) b_l(i+1) \quad (1.19)$$

και για τις πιθανότητες γεννήσεως:

$$E_k(b) = \sum_{\pi} P(\pi | \mathbf{x}, \theta') E_k(b, \pi) = \frac{1}{P(\mathbf{x})} \sum_{\{i: x_i=b\}} f_k^j(i) b_k^j(i) \quad (1.20)$$

$$Q(\theta | \theta') = \sum_{k=1}^K \sum_b E_k(b) \log e_k(b) + \sum_{k=0}^K \sum_{l=1}^K A_{kl} \log a_{kl} \quad (1.21)$$

Περληηπτικά, ο Baum-Welch αποτελείται από 2 στάδια, στο πρώτο οι $f_k(i)$, $b_k(i)$ υπολογίζονται από τους αλγορίθμους Forward και BackWard αντίστοιχα, και έπειτα υπολογίζονται οι τιμές των $A_{kl}, E_k(b)$ από τους τύπους (1.19) και (1.20)

αντίστοιχα. Στο δεύτερο στάδιο οι τιμές των $A_{kl}, E_k(b)$ τοποθετούνται στις σχέσεις (1.17), (1.18), στη συνέχεια υπολογίζονται ξανά οι εκτιμητές μέγιστης πιθανοφάνειας, και η πιθανοφάνεια του μοντέλου. Η διαδικασία της εκπαίδευσης τερματίζεται, όταν η διαφορά της πιθανοφάνειας (log Likelihood) της επανάληψης n και $n-1$, έχει τιμή μικρότερη από μία προκαθορισμένη τιμή, την οποία θέτουμε εμείς. Επίσης, μπορούμε να θέσουμε ως συνθήκες τερματισμού ένα μέγιστο πλήθος επαναλήψεων ή όταν $\log \text{Likelihood}=0$ (δηλαδή πιθανότητα=1).

1.7.2: Μέθοδοι Gradient-descent

Ο αλγόριθμος Baum-Welch έχει όπως προαναφέρθηκε αρκετά θετικά χαρακτηριστικά. Τα σημαντικότερα εξ αυτών είναι δύο, πρώτον είναι μαθηματικά αποδεδειγμένη η σύγκλιση του, και δεύτερον είναι γρήγορος. Το κύριο μειονέκτημα του είναι ότι η ανανέωση των παραμέτρων πραγματοποιείται, εφόσον έχει εμφανιστεί όλο σύνολο εκπαίδευσης (batch mode of learning). Επίσης, δεν έχει την δυνατότητα αλλαγής μίας τιμής παραμέτρου, ε αν αυτή έχει μηδενιστεί. Δηλαδή αν μια παράμετρος πάρει την τιμή 0 δεν θα αλλάξει ποτέ από αυτόν τον αλγόριθμο. Οι Baldi και Chauvin (Baldi & Chauvin, 1994) έδωσαν την δική τους λύση για την αντιμετώπιση των παραπάνω μειονεκτημάτων.

Η μέθοδος Gradient-descent προτάθηκε ως μία γενική ευριστική μέθοδος ελαχιστοποίησης ενέργειας. Ορίζοντας μία συνάρτηση f με n μεταβλητές:

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$$

η οποία παραγωγίζεται:

$$(\mathbf{x}_0) = (x_{10}, x_{20}, \dots, x_{n0})$$

έχουμε την δυνατότητα να βρούμε ένα τοπικό της ελάχιστο προσεγγιστικά μέσω της σχέσης:

$$f(\mathbf{x}_{i+1}) = f(\mathbf{x}_i) - \eta \Delta f(\mathbf{x})$$

Το διάνυσμα των μερικών παραγώγων της συνάρτησης ορίζεται ως Δ , και ως η θέτουμε ο ρυθμός μάθησης (learning rate). Φυσικά τις παραμέτρους αποτελούν τα σύνολα των πιθανοτήτων μεταβάσεως και γεννήσεως. Για κάθε παράμετρο του μοντέλου η ανανέωση πραγματοποιείται σύμφωνα με την σχέση:

$$\omega^{t+1} = \omega^t - \eta \frac{\partial \ell(\mathbf{x} | \theta)}{\partial \omega} \quad (1.22)$$

Επομένως, το ζητούμενο είναι ο υπολογισμός των μερικών παραγώγων του λογαρίθμου της πιθανοφάνειας, ως προς τις παραμέτρους του μοντέλου. Η μερική παράγωγος ως προς τις πιθανότητες μεταβάσεως υπολογίζονται σύμφωνα με την σχέση:

$$\frac{\partial \log P(\mathbf{x} | \theta)}{\partial a_{kl}} = \sum_{\pi} P(\pi | \mathbf{x}, \theta) \frac{A_{kl}(\pi)}{a_{kl}} = \frac{A_{kl}}{a_{kl}} \quad (1.23)$$

και για τις πιθανότητες γεννήσεως :

$$\frac{\partial \log P(\mathbf{x} | \theta)}{\partial e_k(b)} = \sum_{\pi} P(\pi | \mathbf{x}, \theta) \frac{E_k(\pi, b)}{e_k(b)} = \frac{E_k(b)}{e_k(b)} \quad (1.24)$$

Τελικά, η μερική παράγωγος του λογαρίθμου της πιθανοφάνειας ως προς τις παραμέτρους, ισούται με την παράγωγο της συνάρτησης Q από την σχέση (1.21). Στο τέλος καταλήγουν στο ίδιο αποτέλεσμα, εφόσον και οι δύο μηδενίζονται στο τοπικό ελάχιστο (Baldi & Chauvin, 1994). Εάν χρησιμοποιήσουμε την σχέση (1.22), τρέχοντας Gradient-descent με πραγματικές τιμές παραμέτρων, ενυπάρχει ο κίνδυνος να λάβουμε ακόμη και αρνητικούς εκτιμητές για τις πιθανότητες αυτές. Ως εκ τούτου, είναι αναγκαίο να θέσουμε ορισμένες βοηθητικές μεταβλητές, οι οποίες παίρνουν τιμές μέσα στο εύρος των τιμών πιθανοτήτων (από 0 έως 1). Έπειτα, προχωρούμε στην ελαχιστοποίηση και στην συνέχεια ανακτούμε τις τιμές των πιθανοτήτων. Στην συγκεκριμένη περίπτωση χρησιμοποιούμε την μέθοδο Krogh και Riss (Krogh, & Riss, 1999). Λαμβάνουμε τις ανανεωμένες παραμέτρους για τις πιθανότητες από την:

$$a_{kl}^{(t+1)} = \frac{z_{kl}^{(t)} \exp\left(-\eta \frac{\partial \ell^{(t)}}{\partial z_{kl}}\right)}{\sum_{l'} z_{kl'}^{(t)} \exp\left(-\eta \frac{\partial \ell^{(t)}}{\partial z_{kl'}}\right)} \quad (1.25)$$

Για τις πιθανότητες μετάβασης από την σχέση:

$$a_{kl}^{(t+1)} = \frac{a_{kl}^{(t)} \exp\left(-\eta \left[A_{kl} - a_{kl} \sum_{l'} A_{kl'} \right]\right)}{\sum_{l'} a_{kl'}^{(t)} \exp\left(-\eta \left[A_{kl} - a_{kl} \sum_{l'} A_{kl'} \right]\right)} \quad (1.26)$$

Για τις πιθανότητες γεννήσεως από την σχέση:

$$e_k^{(t+1)} = \frac{e_k^{(t)}(b) \exp\left(-\eta \left[E_k(b) - e_k(b) \sum_k E_k(b) \right]\right)}{\sum_{b'} e_k^{(t)}(b) \exp\left(-\eta \left[E_k(b) - e_k(b) \sum_{b'} E_k(b) \right]\right)} \quad (1.27)$$

Με την μέθοδο αυτή μπορούμε να πραγματοποιήσουμε την διαδικασία της εκπαίδευσης εξαλείφοντας τον κίνδυνο του μηδενισμού κάποιων παραμέτρων, ταυτόχρονα μας δίνεται η δυνατότητα να πραγματοποιήσουμε την διαδικασία “online training”, μέσω της οποίας οι παράμετροι δύναται να ενημερώνονται κατάλοιπο-κατάλοιπο, χωρίς την ανάγκη της παρουσίασης όλου του συνόλου της εκπαίδευσης.

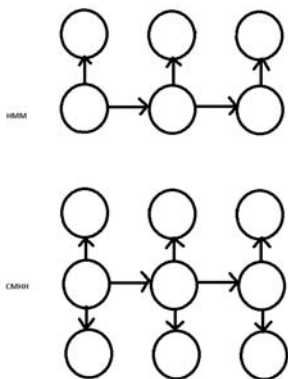
ΚΕΦΑΛΑΙΟ 2: Class Hidden Markov Models

Σύνοψη

Στο κεφάλαιο αυτό γίνεται η αναλυτική παρουσίαση των Class Hidden Markov Models (CHMM) μέσω μαθηματικών σχέσεων, αλγορίθμων και εικόνων. Η ειδοποιός διαφορά ενός CHMM έναντι του HMM, έγκειται στο γεγονός ότι στο πρώτο, εκτός από την ακολουθία των συμβόλων έχουμε και την ύπαρξη ακολουθίας σημάτων. Με την χρήση των σημάτων μπορούμε να ομαδοποιούμε τις καταστάσεις ενός μοντέλου, με αποτέλεσμα την καλύτερη περιγραφή των συστημάτων.

2.1: Ορισμοί

Οι προσεγγίσεις στην εκπαίδευση των CHMM είναι γνωστές ως “μέθοδοι με επίβλεψη” (supervised methods) ή μετά διδασκάλου. Στον αντίποδα, όταν δεν χρησιμοποιούμε μεθόδους επίβλεψης, από την πλευρά του ο χρήστης δίνει απλά τις ακολουθίες στο μοντέλο, και αυτό με την σειρά του πραγματοποιεί την βέλτιστη εκτίμηση των παραμέτρων, για την όσο καλύτερη περιγραφή του συστήματος. Η αξία των CHMM διακρίνεται εύκολα στον τομέα της βιολογίας, όταν πρέπει να εκπαιδεύσουμε ένα μεγάλο μοντέλο το οποίο θα περιγράψει με την δυνατότερη ακρίβεια μία πρωτεΐνη απαρτιζόμενη από αμινοξέα, τα οποία ανήκουν σε διαφορετικές περιοχές μέσα σε αυτή.



Εικόνα 2.1 Γραφική αναπαράσταση δύο μοντέλων: Στο HMM όπως έχει παρουσιαστεί έως τώρα, η αναπαραγωγή της ακολουθίας συμβόλων εξαρτάται από τα σύνολα των πιθανοτήτων μετάβασης και γεννήσεως, ενώ το μοντέλο σχηματίζει $1^{η}$ τάξης αλυσίδα. Το CHMM “γεννά” 2 ακολουθίες, η πρώτη είναι η ακολουθία συμβόλων, όμοια με το HMM, ενώ η δεύτερη μία ακολουθία σημάτων. Το CHMM είναι αλυσίδα $1^{η}$ τάξης.

Ένα κατατοπιστικό παράδειγμα για την επεξήγηση “των ανωτέρων”, είναι αυτό της πρόγνωσης διαμεμβρανικών πρωτεϊνών. Πρώτα από όλα πρέπει να κατανοήσουμε

το πρόβλημα, έτσι για την περίπτωση αυτή, έχουμε μία πρωτεΐνη της οποίας τα αμινοξέα “χωρίζονται” σε 3 κατηγορίες, τα διαμεμβρανικά, τα αμινοξέα τα οποία ανήκουν στην εξωτερική περιοχή της μεμβράνης και αυτά που ανήκουν στην εσωτερική. Εάν επιχειρούσαμε να λύσουμε το πρόβλημα με μεθόδους “άνευ διδασκάλου”, θα έπρεπε να δημιουργήσουμε 3 μοντέλα, ένα για κάθε κατηγορία, και στην συνέχεια να προχωρήσουμε στην ένωση των μοντέλων μέσω αυθαίρετων (μη τεκμηριωμένων) πιθανοτήτων μεταβάσεων. Το πρόβλημα αυτό παύει να υφίσταται χρησιμοποιώντας την μέθοδο “μετά διδασκάλου” ή με επίβλεψη. Η μέθοδος αυτή στηρίζεται στην ύπαρξη σημασμένων ακολουθιών ή αλληλουχίες με ετικέτες, δημιουργήθηκε από τον Krogh και είναι γνωστή ως Class Hidden Markov Model (Krogh, 1994).

Σύμφωνα με την μέθοδο αυτή κάθε αλληλουχία συμβόλων:

$$\mathbf{x} = x_1, x_2, \dots, x_{L-1}, x_L$$

συνδέεται με μία ακολουθία σημάνσεων:

$$\mathbf{y} = y_1, y_2, \dots, y_{L-1}, y_L$$

Στο δικό μας παράδειγμα, αυτό της πρόγνωσης των διαμεμβρανικών πρωτεϊνών, το “αλφάβητο” των σημάνσεων αποτελείται από 3 γράμματα, αριθμός βεβαίως ίσος με το πλήθος των κατηγοριών στις οποίες υπάγονται τα αμινοξέα της πρωτεΐνης. Έτσι, έχουμε μία σήμανση για την διαμεμβρανική περιοχή (M), μία για την εσωτερική πλευρά της μεμβράνης (I), και μία για την εξωτερική πλευρά της μεμβράνης (O). Επιπροσθέτως, θα πρέπει να θέσουμε μία κατανομή για την πιθανότητα ταύτισης μίας κατάστασης με μία συγκεκριμένη σήμανση από το σύνολο (“αλφάβητο”) των σημάνσεων. Εφόσον έχουν προηγηθεί τα παραπάνω, προχωρούμε στην ομαδοποίηση των καταστάσεων, η οποία συντελείται βάσει βιολογικών κριτηρίων, δηλαδή δημιουργούμε μία ομάδα καταστάσεων για την διαμεμβρανική περιοχή, μία για την εξωτερική περιοχή της μεμβράνης, και μία για την εσωτερική. Έτσι με αυτόν τον τρόπο επιτυγχάνουμε να μοντελοποιήσουμε την πρωτεΐνη σε ένα μοντέλο, αντί για 3. Για την ταύτιση των καταστάσεων με τις σημάνσεις θέτουμε μία μεταβλητή $\delta_k(c)$, η οποία εκφράζει την πιθανότητα η κατάσταση k να έχει σήμανση c . Πρακτικά χρησιμοποιούμε μία δίτιμη συνάρτηση με τιμές 0 και 1, έτσι η συνάρτηση αυτή δίνει τιμή 1, εάν η κατάσταση συμφωνεί με την σήμανση, και 0 στην αντίθετη περίπτωση, με αποτέλεσμα την εξασφάλιση της ταύτισης μίας κατάστασης με μία μόνο σήμανση.

2.2: Πιθανοφάνεια

Ως λογική συνέχεια των παραπάνω για τον υπολογισμό της πιθανοφάνειας, όταν χρησιμοποιούμε μέθοδο “μετά διδασκάλου”, θα πρέπει να λαμβάνεται υπόψη η ακολουθία των σημάνσεων. Επομένως, ορίζεται ως αντικειμενική συνάρτηση η από κοινού πιθανότητα $P(\mathbf{x}, \mathbf{y} | \theta)$ της ακολουθίας \mathbf{x} με την ακολουθία \mathbf{y} , δοθέντος του μοντέλου θ , ως εξής:

$$P(\mathbf{x}, \mathbf{y} | \theta) = \sum_{\pi} P(\mathbf{x}, \mathbf{y}, \pi | \theta) = \sum_{\pi \in \Pi_{\mathbf{y}}} P(\mathbf{x}, \pi | \theta) = \sum_{\pi \in \Pi_{\mathbf{y}}} a_{B\pi_1} \prod_{i=1}^L e_{\pi_i}(x_i) a_{\pi_i, \pi_{i+1}} \quad (2.1)$$

Συγκρίνοντας την σχέση αυτή με την (1.7), γίνεται εύκολα αντιληπτό ότι η διαφορά είναι ο τρόπος άθροισης, έτσι, στην (2.1) η άθροιση πραγματοποιείται μόνο για τα μονοπάτια Π_y , τα οποία σχηματίζονται από καταστάσεις που ταυτίζονται με την σήμανση. Για τον υπολογισμό της από κοινού πιθανότητας των ακολουθιών και των σημάνσεων δοθέντος του μοντέλου, χρησιμοποιούμε τους παραλλαγμένους αλγόριθμους Forward και Backward, εφόσον έχουν γίνει σ' αυτούς κάποιες κοινότυπες τροποποιήσεις. Η τροποποίηση αυτή λαμβάνει χώρα μέσω του πολλαπλασιασμού των ενδιάμεσων μεταβλητών με μία δίτιμη συνάρτηση (0,1, την οποία αναλύσαμε παραπάνω), η οποία κατ' ουσίαν πραγματοποιεί τον έλεγχο της συμφωνίας καταστάσεων και σημάνσεων. Επομένως, ο τροποποιημένος αλγόριθμος Forward είναι ο εξής:

Τροποποιημένος αλγόριθμος Forward

$$\forall k \neq B, i = 0: f_B(0) = 1, f_k(0) = 0$$

$$\forall 1 \leq i \leq L: f_i(i) = e_i(x_i) \delta_i(y_i) \sum_k f_k(i-1) a_{ki} \quad (2.2)$$

$$P(x, y, | \theta) = \sum_k f_k(L) a_{kE} \quad (2.3)$$

Βάσει της ίδιας λογικής σχηματίζεται ο τροποποιημένος αλγόριθμος Backward:

		Sequence						
		I	I	M	M	O	O	
states	labels	0	x_1	x_2	x_3	x_4	x_5	x_6
1	I							
2	I							
3	I							
4	M							
5	M							
6	M							
7	O							
8	O							
9	O							

Σημειώσεις:

- Το κελί (2,7) περιέχει το $f=0$.
- Το κελί (6,3) περιέχει το $f=0$.
- Το κελί (7,5) περιέχει το $f=0$.
- Το κελί (5,5) περιέχει το κείμενο: "f υπολογίζεται ως συνήθως".

Εικόνα 2.2 Η απεικόνιση του αλγορίθμου Forward με την χρήση σημασμένων ακολουθιών. Απεικονίζεται ένα υποθετικό μοντέλο 9 καταστάσεων και μία ακολουθία με 6 κατάλοιπα, των οποίων οι σημάνσεις αναγράφονται στην ίδια στήλη. Όπως είναι ευδιάκριτο, όταν οι καταστάσεις και τα κατάλοιπα ταυτίζονται ως προς τις σημάνσεις, έχουμε μη μηδενικές τιμές για την f , ενώ στην αντίθετη περίπτωση έχουμε τιμή ίση με το 0.

Τροποποιημένος αλγόριθμος Backward

$$\forall k, i = L: b_k(L) = a_{kE}$$

$$\forall k, i = L: b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) \delta_l(y_{i+1}) b_l(i+1) \quad (2.4)$$

$$P(x, y | \theta) = \sum_l a_{Bl} e_l(x_1) b_l(1) \quad (2.5)$$

Όπως είναι φυσικό οι αλγόριθμοι αυτοί δίνουν μηδενικές τιμές στα κελιά των πινάκων τους, όταν δεν πληρείτε η συμφωνία των καταστάσεων και των σημάνσεων

(Εικόνα 2.2). Εξαιτίας του ότι το πλήθος των επιτρεπτών μονοπατιών Π_y είναι μικρότερο του Π , οδηγούμαστε στο λογικό συμπέρασμα $P(\mathbf{x}, \mathbf{y} | \theta) \leq P(\mathbf{x} | \theta)$.

2.3: Εκτίμηση παραμέτρων

Μέγιστη πιθανοφάνεια

Η εκτίμηση μέγιστης πιθανοφάνειας με την χρήση σημασμένων ακολουθιών ορίζεται από την σχέση:

$$\theta^{ML} = \arg \max_{\theta} P(\mathbf{x}, \mathbf{y} | \theta)$$

Όπως περιγράφηκε παραπάνω, οι αλγόριθμοι που εφαρμόζονται στις μεθόδους “χωρίς επίβλεψη”, μπορούν να εφαρμοστούν και στις “μετά διδασκάλου” μεθόδους, μετά την τροποποίηση τους. Βάσει αυτής της παραδοχής, αλλά και του γεγονότος ότι υπάρχει ανεξαρτησία μεταξύ των αλληλουχιών και των καταστάσεων, οι σχέσεις (1.19) και (1.20) μετατρέπονται ως εξής:

$$A_{kl} = \frac{1}{P(\mathbf{x}, \mathbf{y} | \theta)} \sum_i f_k(i) a_{kl} e_l(x_{i+1}) \delta_l(y_{i+1}) b_l(i+1) \quad (2.6)$$

$$E_k(b) = \frac{1}{P(\mathbf{x}, \mathbf{y} | \theta)} \sum_{\{i|x_i^l=b\}} f_k(i) b_k(i) \quad (2.7)$$

Μέσω των τροποποιημένων αλγορίθμων μπορούμε να υπολογίσουμε τις τιμές των $P(\mathbf{x}, \mathbf{y} | \theta)$, $f(i)$ και $b_k(i)$, επομένως, έχουμε την δυνατότητα να πραγματοποιήσουμε εκπαίδευση μέγιστης πιθανοφάνειας χρησιμοποιώντας τους αλγορίθμους Baum-Welch και Gradient Descent. Έτσι, ο υπολογισμός των εκτιμητών μέγιστης πιθανοφάνειας από τον αλγόριθμο Baum-Welch με σημάνσεις (labels) προκύπτει από τις σχέσεις:

$$\hat{a}_{kl} = \frac{A_{kl}^c}{\sum_{l'} A_{kl'}^c} \quad (2.8)$$

$$\hat{e}_k(b) = \frac{E_k^c(b)}{\sum_{b'} E_k^c(b)} \quad (2.9)$$

Επίσης, από τον αλγόριθμο Gradient Descent με την χρήση σημάνσεων (labels), οι εκτιμητές μέγιστης πιθανοφάνειας υπολογίζονται από τις σχέσεις:

$$a_{kl}^{(t+1)} = \frac{a_{kl}^{(t)} \exp\left(-\eta \left[A_{kl}^c - a_{kl} \sum_{l'} A_{kl'}^c \right]\right)}{\sum_{l'} a_{kl'}^{(t)} \exp\left(-\eta \left[A_{kl}^c - a_{kl} \sum_{l'} A_{kl'}^c \right]\right)} \quad (2.10)$$

$$e_k^{(t+1)} = \frac{e_k^{(t)}(b) \exp\left(-\eta \left[E_k^c(b) - e_k(b) \sum_k E_k^c(b) \right]\right)}{\sum_{b'} e_k^{(t)}(b) \exp\left(-\eta \left[E_k^c(b) - e_k(b) \sum_{b'} E_k^c(b) \right]\right)} \quad (2.11)$$

Δεσμευμένη Μέγιστη πιθανοφάνεια

Για την χρήση των σημασμένων ακολουθιών ο Krogh (Krogh,1994) δημιούργησε μία νέα μέθοδο εκπαίδευσης, την επονομαζόμενη μέθοδο της Δεσμευμένης Μέγιστης πιθανοφάνειας (Conditional Maximum Likelihood). Βάσει αυτής προσπαθούμε να μεγιστοποιήσουμε την πιθανότητα των σημάνσεων, δοθέντος του μοντέλου και των ακολουθιών:

$$\theta^{\text{CLM}} = \underset{\theta}{\operatorname{argmax}} P(\mathbf{y} | \mathbf{x}, \theta) = \underset{\theta}{\operatorname{argmax}} \frac{P(\mathbf{x}, \mathbf{y} | \theta)}{P(\mathbf{x} | \theta)}$$

Ο Krogh απέδειξε επίσης ότι αυτή η μέθοδος-προσέγγιση είναι μία γενική περίπτωση της γνωστής διαδικασίας εκπαίδευσης με το κριτήριο της Μέγιστης αμοιβαίας πληροφορίας (Maximum Mutual Information), η οποία αναφέρεται από τον (Rabiner, 1989). Ο αρνητικός λογάριθμος αυτής της δεσμευμένης πιθανοφάνειας μπορεί να τεθεί ως η διαφορά:

$$\ell = -\log P(\mathbf{y} | \mathbf{x}, \theta) = \ell_c - \ell_f$$

Ορίζοντας:

$$\ell_c = -\log P(\mathbf{x}, \mathbf{y} | \theta)$$

$$\ell_f = -\log P(\mathbf{x} | \theta)$$

Με τους δείκτες f και c , ορίζουμε αντίστοιχα την πιθανοφάνεια στην περίπτωση όπου οι σημάνσεις δεν λαμβάνονται υπόψη (free-running phase), και στην περίπτωση όπου λαμβάνονται υπόψη (clamped phase). Από τις σχέσεις (1.23) και (1.24) υπολογίζονται οι αναμενόμενες τιμές και τις μερικές παραγώγους των παραμέτρων του μοντέλου.

$$\frac{\partial \ell}{\partial a_{kl}} = \frac{\partial \ell_c}{\partial a_{kl}} - \frac{\partial \ell_f}{\partial a_{kl}} = -\frac{A_{kl}^c - A_{kl}^f}{a_{kl}} \quad (2.12)$$

$$\frac{\partial \ell}{\partial e_k(b)} = \frac{\partial \ell_c}{\partial e_k(b)} - \frac{\partial \ell_f}{\partial e_k(b)} = -\frac{E_k^c(b) - E_k^f(b)}{e_k(b)} \quad (2.13)$$

Όπως είναι φυσικό ο αλγόριθμος Baum-Welch δεν είναι δυνατόν να χρησιμοποιηθεί, εξαιτίας του ότι η διαφορά των αναμενόμενων τιμών θα παράξει αρνητικές εκτιμήσεις για τις παραμέτρους. Στον αντίποδα, με την χρήση των μερικών παραγώγων μπορούμε να εκτελέσουμε εκπαίδευση με τον αλγόριθμο Gradient Descent. Η σχέση μέσω της οποίας υπολογίζονται οι ανανεωμένες τιμές των παραμέτρων στην επανάληψη t , ορίζεται ως:

$$a_{kl}^{(t+1)} = \frac{a_{kl}^{(t)} \exp\left(-\eta \left[A_{kl}^c - A_{kl}^f - a_{kl} \sum_{l'} (A_{kl'}^c - A_{kl'}^f) \right]\right)}{\sum_{l'} a_{kl'}^{(t)} \exp\left(-\eta \left[A_{kl}^c - A_{kl}^f - a_{kl} \sum_{l'} (A_{kl'}^c - A_{kl'}^f) \right]\right)} \quad (2.14)$$

$$e_k^{(t+1)} = \frac{e_k^{(t)} \exp\left(-\eta \left[E_k^c(b) - E_k^f(b) - e_k(b) \sum_k (E_k^c(b) - E_k^f(b)) \right]\right)}{\sum_{b'} e_k^{(t)} \exp\left(-\eta \left[E_k^c(b) - E_k^f(b) - e_k(b) \sum_{b'} (E_k^c(b) - E_k^f(b)) \right]\right)} \quad (2.15)$$

Τα δύο βασικά μειονεκτήματα της μεθόδου αυτής, είναι πρώτον ότι «καταναλώνει» διπλάσιους υπολογιστικούς πόρους, τόσο στην μνήμη, όσο και στην υπολογιστική ισχύ, καθώς απαιτούνται δύο τρεξίματα των αλγορίθμων για τον υπολογισμό των δύο πιθανοφαιών, και δεύτερον η εύρεση της βέλτιστης τιμής για τον ρυθμό μάθησης. Το πρόβλημα του ρυθμού μάθησης αντιμετωπίστηκε ως ένα μεγάλο βαθμό μέσω ενός αλγορίθμου, ο οποίος προτάθηκε από τους (Bagos, Liakopoulos, & Hamodrakas, 2004), και στηρίζεται στην χρήση διαφορετικών ρυθμών μάθησης και στην αναπροσαρμογή αυτών, κατά την διάρκεια της εκπαίδευσης. Το πλεονέκτημα της μεθόδου αυτής είναι η καλύτερη ικανότητα περιγραφής-μοντελοποίησης των συστημάτων.

Κεφάλαιο 3: Εκτίμηση Παραμέτρων με την Χρήση του Viterbi Training

3.1: Γενική περιγραφή του viterbi training

Στην ενότητα αυτή παρουσιάζεται ένας πολύ απλός αλγόριθμος εκτίμησης παραμέτρων, ο οποίος εντάχθηκε στην βιβλιογραφία σχετικά αργά. Ο αλγόριθμος αυτός ονομάζεται Viterbi training ή αλλιώς segmentalk-means algorithm και προτάθηκε από τους Juang και Rabiner το 1990 (Juang & Rabiner,1990). Όπως έχει προαναφερθεί παραπάνω, είναι σύνηθες να μην γνωρίζουμε τις αλληλουχίες των καταστάσεων από τις οποίες διήλθε το μοντέλο, κατά την εκπαίδευση. Βάσει αυτής της παραδοχής, το πρόβλημα στην διαδικασία της εκτίμησης των παραμέτρων γίνεται διττό, διότι τώρα εκτός από την εκτίμηση των παραμέτρων, θα πρέπει ταυτόχρονα να εκτιμηθούν και τα μονοπάτια. Η λύση του προβλήματος αυτού μέσω του Viterbi training περιλαμβάνει 2 βήματα, τα οποία αποτελούν και την βάση της δημιουργίας του.

- εύρεση του πιθανότερου μονοπατιού των καταστάσεων με την χρήση του αλγορίθμου viterbi,
- θεώρηση-παραδοχή του μονοπατιού αυτού ως πραγματικό, και εν συνεχεία χρήση των τύπων (1.17) και (1.18) για την εκτίμηση των παραμέτρων.

Ο αλγόριθμος παρουσιάστηκε για πρώτη φορά το 1968 με το όνομα K-means algorithm (MacQueen, 1967) στην βιβλιογραφία της στατιστικής ομαδοποίησης (clustering). Με την εύρεση του πιθανότερου μονοπατιού (μέσω του Viterbi), και την θεώρηση αυτού ως πραγματικό, η από κοινού πιθανοφάνεια μίας αλληλουχίας και του μονοπατιού αυτού υπολογίζεται από ένα απλό γινόμενο (1.6), επομένως, η εφαρμογή των τύπων για τον υπολογισμό των εκτιμητών μέγιστης πιθανοφάνειας είναι το επόμενο λογικό βήμα. Οι Juang και Rabiner απέδειξαν ότι η επαναλαμβανόμενη και διαδοχική εκτέλεση των παραπάνω δύο βημάτων, έχει ως αποτέλεσμα την μονοτονική αύξηση της πιθανοφάνειας, η οποία ονομάστηκε πιθανοφάνεια βελτιστοποιημένη για τις καταστάσεις (state-optimized likelihood). Ένα πολύ σημαντικό πλεονέκτημα του αλγορίθμου, είναι το γεγονός ότι συγκλίνει πάντα σε ένα τοπικό μέγιστο της πιθανοφάνειας σε ένα πεπερασμένο πλήθος επαναλήψεων, λόγω των πεπερασμένων πιθανών μονοπατιών.

Η τιμή που θα έχει το τοπικό μέγιστο, βάσει της οποίας κρίνεται η προγνωστική αξία των μοντέλων, βρίσκεται σε εξάρτηση με τον καθορισμό των αρχικών τιμών. Όπως και στην περίπτωση των άλλων αλγορίθμων που έχουν αναφερθεί, δεν μπορεί να δοθεί διαβεβαίωση ότι το μέγιστο της πιθανοφάνειας που θα βρει ο αλγόριθμος είναι ταυτόχρονα και ολικό. Ο Viterbi training μοιάζει πολύ με τον Baum-welch, η βασική διαφορά τους έγκειται στο γεγονός ότι ο πρώτος λαμβάνει υπόψη την συνεισφορά μόνο του πιθανότερου μονοπατιού για τον υπολογισμό της τιμής, ενώ ο δεύτερος αθροίζει την συνεισφορά όλων των πιθανών μονοπατιών.

Βάσει αυτής της ειδοποιού διαφοράς ο Viterbi training θεωρητικά θα παράγει αποτελέσματα λίγο χειρότερα, όμως κατέχει το πλεονέκτημα ότι για τον υπολογισμό του απαιτεί για κάθε επανάληψη μόνο ένα “τρέξιμο” του Viterbi, ενώ ο αλγόριθμος Baum-Welch ένα “τρέξιμο” του Forward και ένα του Backward, γεγονός που τον καθιστά σχεδόν δύο φορές ταχύτερο.

3.2: Τροποποιημένος αλγόριθμος Viterbi

$$\forall k \neq B, i = 0: u_B^y(0) = 1, u_k^y(0) = 0$$

$$\forall 1 \leq i \leq L: u_i^y(i) = \delta_l(y_i) e_l(x_i) \max_k \{u_k^y(i-1) a_{ki}\} \quad (3.1)$$

$$P(\mathbf{x}, \mathbf{y}, \pi^y | \theta) = \max_k \{u_k^y(L) a_{kE}\} \quad (3.2)$$

Ο τροποποιημένος αλγόριθμος Viterbi (Bagos, Liakopoulos, & Hamodrakas, 2006) χρησιμοποιήθηκε για την διαδικασία αποκωδικοποίησης (decoding), με την χρήση κάποιων ακολουθιών με σημάνσεις (labels).

Ο τροποποιημένος αλγόριθμος Viterbi είναι παρόμοιος με τον Viterbi, όπως αυτός αναλύθηκε στο κεφάλαιο 1. Η διαφορά τους έγκειται στο γεγονός ότι ο τροποποιημένος Viterbi λαμβάνει υπόψη τις σημάνσεις των ακολουθιών.

Συγκεκριμένα, η συνάρτηση δ_l είναι μία δίτιμη συνάρτηση, η οποία ελέγχει αν οι καταστάσεις ταυτίζονται με τις σημάνσεις της ακολουθίας, έτσι, όταν υπάρχει ταύτιση η τιμή της είναι ένα, ενώ στην αντίθετη περίπτωση 0. Η συνάρτηση δ_l πολλαπλασιάζεται με την ενδιάμεση μεταβλητή, για τον υπολογισμό της u_i^y . Έτσι προκύπτει η από κοινού πιθανότητα της ακολουθίας και των σημάνσεων, δοθέντος του μοντέλου $P(\mathbf{x}, \mathbf{y}, \pi^y | \theta)$.

3.3: Viterbi Training

Στα πλαίσια αυτής της εργασίας δημιουργήσαμε τον Viterbi training για ακολουθίες με σημάνσεις (labels). Ο αλγόριθμος μας είναι μία μέθοδος εκτίμησης παραμέτρων “μετά διδασκάλου”, επομένως, απαιτεί την ύπαρξη σημασμένων ακολουθιών. Όπως έχει προαναφερθεί, κατά την διαδικασία της εκτίμησης των παραμέτρων αναζητούμε ταυτόχρονα το πιθανότερο μονοπάτι των καταστάσεων και τους εκτιμητές των παραμέτρων. Η δική μας προσέγγιση είναι η εξής: Αρχικά βρίσκουμε το πιθανότερο μονοπάτι μέσω του τροποποιημένου Viterbi, έπειτα θεωρούμε το μονοπάτι αυτό ως πραγματικό, και στο τελευταίο βήμα προχωρούμε στην εκτίμηση των παραμέτρων. Οι εκτιμητές μέγιστης πιθανοφάνειας (EMΠ) υπολογίζονται για τις μεταβάσεις μέσω της σχέσης (1.17) και για τις γεννήσεις μέσω της (1.18). Η διαφορά του αλγορίθμου μας με όλους τους άλλους που έχουν προαναφερθεί, έγκειται στον τρόπο υπολογισμού των εκτιμητών μέγιστης πιθανοφάνειας.

Συγκεκριμένα, για τον υπολογισμό των A_{kl} και $E_k(b)$, οι οποίοι αποτελούν τους αριθμητές των σχέσεων (1.17) και (1.18) αντίστοιχα, οι μέθοδοι Baum-Welch και Gradient Descent απαιτούσαν για τον υπολογισμό τους, τις τιμές των αλγορίθμων

Forward και Backward, σχέσεις (1.19) και (1.20). Στον αντίποδα, με τον αλγόριθμο Viterbi training ο υπολογισμός των A_{kl} , $E_k(b)$ πραγματοποιείται ως εξής:

- Θεωρώντας το μονοπάτι του τροποποιημένου Viterbi ως πραγματικό μετράμε τον αριθμό των μεταβάσεων της κάθε κατάστασης προς τις άλλες, και οι αριθμοί (score) που προκύπτουν από αυτήν την καταμέτρηση αποτελούν τις τιμές του A_{kl} .
- Βάσει της ίδιας λογικής υπολογίζονται και οι τιμές του $E_k(b)$, έτσι, θεωρώντας ότι οι καταστάσεις του μονοπατιού γέννησαν την ακολουθία των συμβόλων, μετρούμε πόσες φορές γέννησε η εκάστοτε κατάσταση το εκάστοτε αμινοξύ, και οι αριθμοί που προκύπτουν, αποτελούν τις τιμές του $E_k(b)$.

Για διαχωριστικούς λόγους οι A_{kl} και $E_k(b)$ αναφέρονται ως A^{c*}_{kl} και $E^{c*}_k(b)$. Έτσι οι σχέσεις (1.17) και (1.18) μετασχηματίζονται αντίστοιχα σε:

$$\hat{a}_{kl} = \frac{A^{c*}_{kl}}{\sum_{l'} A^{c*}_{kl'}} \quad (3.3)$$

$$\hat{e}_k(b) = \frac{E^{c*}_k(b)}{\sum_{b'} E^{c*}_k(b)} \quad (3.4)$$

Ο Viterbi training μπορεί να εφαρμοστεί και με τον αλγόριθμο Gradient-Descent. Στην περίπτωση αυτή οι εκτιμητές μέγιστης πιθανοφάνειας υπολογίζονται από τις σχέσεις:

$$a_{kl}^{(t+1)} = \frac{a_{kl}^{(t)} \exp\left(-\eta \left[A_{kl}^{c*} - a_{kl} \sum_{l'} A_{kl'}^{c*} \right]\right)}{\sum_{l'} a_{kl'}^{(t)} \exp\left(-\eta \left[A_{kl}^{c*} - a_{kl} \sum_{l'} A_{kl'}^{c*} \right]\right)} \quad (3.5)$$

$$e_k^{(t+1)} = \frac{e_k^{(t)}(b) \exp\left(-\eta \left[E_k^{c*}(b) - e_k(b) \sum_k E_k^{c*}(b) \right]\right)}{\sum_{b'} e_k^{(t)}(b) \exp\left(-\eta \left[E_k^{c*}(b) - e_k(b) \sum_{b'} E_k^{c*}(b) \right]\right)} \quad (3.6)$$

Επίσης, ο αλγόριθμος Viterbi training μπορεί να χρησιμοποιηθεί με μέθοδο της Δεσμευμένης Μέγιστης πιθανοφάνειας (Conditional Maximum Likelihood). Χρησιμοποιώντας την μέθοδο αυτή, οι ΕΜΠ προκύπτουν από τις σχέσεις:

$$a_{kl}^{(t+1)} = \frac{a_{kl}^{(t)} \exp\left(-\eta \left[A_{kl}^{c*} - A_{kl}^{f*} - a_{kl} \sum_{l'} (A_{kl'}^{c*} - A_{kl'}^{f*}) \right]\right)}{\sum_{l'} a_{kl'}^{(t)} \exp\left(-\eta \left[A_{kl}^{c*} - A_{kl}^{f*} - a_{kl} \sum_{l'} (A_{kl'}^{c*} - A_{kl'}^{f*}) \right]\right)} \quad (3.7)$$

$$e_k^{(t+1)} = \frac{e_k^{(t)} \exp\left(-\eta \left[E_k^{c^*}(b) - E_k^{f^*}(b) - e_k(b) \sum_k (E_k^{c^*}(b) - E_k^{f^*}(b)) \right] \right)}{\sum_{b'} e_k^{(t)} \exp\left(-\eta \left[E_k^{c^*}(b) - E_k^{f^*}(b) - e_k(b) \sum_{b'} (E_k^{c^*}(b) - E_k^{f^*}(b)) \right] \right)} \quad (3.8)$$

Συμπερασματικά, μέσω του Viterbi training αντικαθίσταται η χρήση των δύο αλγόριθμων Forward και Backward σε κάθε επανάληψη της διαδικασίας υπολογισμού των ΕΜΠ, με την χρήση του τροποποιημένου Viterbi. Αυτό έχει ως αποτέλεσμα την μείωση του χρόνου εκπαίδευσης σε σημαντικό βαθμό, ενώ τα αποτελέσματα που προκύπτουν είναι λίγο χειρότερα από αυτά της μέχρι τώρα μεθοδολογίας.

ΚΕΦΑΛΑΙΟ 4: Αποτελέσματα και Συζήτηση

4.1: Υλοποίηση

Η νέα κλάση Viterbi training ενσωματώθηκε σε μία προϋπάρχουσα υλοποίηση (Πρόγραμμα), η οποία χρησιμοποιήθηκε σε παλαιότερες εργασίες. Συγκεκριμένα, ο Viterbi training ενσωματώθηκε στο τμήμα της προϋπάρχουσας υλοποίησης, στο οποίο πραγματοποιείται η εκτίμηση των παραμέτρων. Με την εισαγωγή του Viterbi training στο τμήμα αυτό, η εύρεση των εκτιμητών μέγιστης πιθανοφάνειας πραγματοποιείται πλέον με δύο τρόπους, ο πρώτος είναι με τους αλγορίθμους Forward και Backward, και ο δεύτερος με τον Viterbi training, σύμφωνα με την ανάλυση του έγινε στο κεφάλαιο 3.

Ο κώδικας του αλγορίθμου Viterbi training

```
package javaomp2;
import java.io.*;
import java.util.*;
public class Viterbi Training extends HMMAlgo
{

public double [][] A;      //A=AC+AF
public double [][] AC;    //  $A_{kl}^{c*}$ 
public double [][] AF;    //  $A_{kl}^{f*}$ 
public double [][] EC;    //  $E_k^{c*}(b)$ 
public double [][] EF;    //  $E_k^{f*}(b)$ 
public String [] PathF;
public static double VPROBC; //the log viterbi likelihood with
//labels
public static double VPROBF; //the log viterbi likelihood without
//labels

public Viterbi Training(HMM hmm, Seq x, String flow)
{
    //hmm the model, seg x the protein

super(hmm, x);
String viterbiPath,viterbiPathC,viterbiPathF;
int length=x.getLen(); //protein length
String [] Path;
String [] PathC;

if(flow.equals("CML")) //conditional maximum likelihood case
{
viterbiPathC=GetPath(hmm,x,false); //viterbi path with labels
viterbiPathF=GetPath(hmm,x,true); //viterbi path without labels
```

```

PathC=new String[length]; //viterbi path with labels
PathC=CutPathAndSet(viterbiPathC,length);

AC=new double[Params.nstate][Params.nstate]; //nstate=model's
number of states
EC=new double[Params.nstate][Params.nesym]; //nesym=model's
number of symbols
Score(PathC,AC,x,EC);

PathF=new String[length]; //viterbi path without labels
PathF=CutPathAndSet(viterbiPathF,length);

AF=new double[Params.nstate][Params.nstate];
EF=new double[Params.nstate][Params.nesym];
Score(PathF,AF,x,EF);
A=new double[Params.nstate][Params.nstate];

for(int i=0;i<Params.nstate;i++)
{
for(int j=0;j<Params.nstate;j++)
{
A[i][j]=AC[i][j]-AF[i][j];
}
}

}
else //if we're computing the maximum likelihood
{
viterbiPath=GetPath(hmm,x,false); //viterbi path with labels

PathC=new String[length];
PathC=CutPathAndSet(viterbiPath,length);
AC=new double[Params.nstate][Params.nstate];
EC=new double[Params.nstate][Params.nesym];
Score(PathC,AC,x,EC);

}

}
private String GetPath(HMM hmm, Seq x, boolean free )
{ //if free is true then viterbi is executed without labels
//if free is false then viterbi is executed with labels

String Path;
Double prob;
Viterbi vit5 = new Viterbi(hmm, x, free ); //callout of viterbi
function
Path=vit5.getPath(); //viterbi path
prob=vit5.getProb(); //log viterbi likelihood

```

```

SetViterbiProb(prob,free); //initialization of the propability
return Path;
}
private String[] CutPathAndSet(String Path,int sequencelength)
{ //gets the states of the path as a single string and
seperates every state and it stores them in a matrix in which
every cell is a different state and it returns the matrix

String [] statepath;
String Pathseq=Path;
int length=sequencelength;
int lengthOfCharacter;
statepath=new String[length];
for(int m=0;m<length;m++)
statepath[m]=" ";
char k;
char [] table;
table=new char[1];
int i=0;
String p="";
int position=0;
lengthOfCharacter=charactercounter();
while(i<length*lengthOfCharacter) //for every type of state
{

position=i/lengthOfCharacter;
table[0]=Pathseq.charAt(i);
p=newString(table);
statepath[position]=p;

for(int j=i+1;j<i+lengthOfCharacter;j++)
{
table[0]=Pathseq.charAt( j );
p=new String(table);

statepath[position]=statepath[position]+p;
}
i=i+lengthOfCharacter;

} //end of while

return statepath;
}
Private int character counter() //counts the number of the
characters of the states and then it returns it
{
String state value;
int amount;
statevalue=Params.state[0];
amount=statevalue.length();
return amount;
}

```

```

private void Score(String sequencePath[],double
A[][],Seqx,double E[][])
{
    //at first, the matrices A and E have zero values
    //uses the path and calculates the score of the transitions
of the states eg. If from the 3rd state we're transitioning to
the 5th state then  $A[3-1][5-1]=A[3-1][5-1]+1$ . For the emissions
matrix E we match every amino acid with every state eg. If we
have from the path the 4th state and from the amino acid sequence
the 7th amino acid then  $E[4-1][7-1]=E[4-1][7-1]+1$ . P.S. we
subtract 1 from the matrices indexes because we're using Java.

    Int row,column;
    int previous=0;
    int next=1;
    int lengthTable;
    String StatePrevious;
    String StateNext;
    StatePrevious=sequencePath[0];
    StateNext=sequencePath[1];
    int L = x.getLen();
    row=column=0;
    if(Params.ALLOW_BEGIN) //if we have a starting state(Begin)
    {
    for(int k=0;k<Params.nstate;k++)
    {
    if(StatePrevious.equals(Params.state[k])) //we're calculating to
which state the model will transition
    A[0][k]=A[0][k]+1;
    }

    }
    lengthTable=sequencePath.length;

    int pos;

    while(next<lengthTable)
    {
    int k=0;
    StatePrevious=sequencePath[previous];
    StateNext=sequencePath[next];

    for(int i=0;i<Params.nstate;i++)
    {

    if(StatePrevious.equals(Params.state[i])) //find the number of
the state
    {
    row=i;
    k++;
    }

    if(StateNext.equals(Params.state[i])) //find the number of the
state

```

```

{
column=i;
k++;
}
if(k==2)
i= Params.nstate;
} //end of for params
A[row][column]=A[row][column]+1;
pos=x.getNESym(previous); //the position of the amino acid
E[row][pos]=E[row][pos]+1;
previous++;
next++;
} //end of while
if(next==lengthTable) //we we're checking the last amino acid
{
pos=x.getNESym(lengthTable-1);

E[column][pos]=E[column][pos]+1;
}

if(Params.ALLOW_END) //if we have an End state
{

StatePrevious=sequencePath[lengthTable-1];
for(int k=0;k<Params.nstate;k++)
{
if(StatePrevious.equals(Params.state[k])) //calculating the last
state before the End state
A[k][(Params.nstate)-1]=A[k][(Params.nstate)-1]+1;
}
}

}

private void SetViterbiProb(double prob,boolean free)
{ //initialization of viterbi propability

if(!free) //check for the use of labels
VPROBC=prob;
else
VPROBF=prob;
}
public double SendTheA(int i,int j)
{
double value;
value=A[i][j];
return value;
}
public double SendTheAF(int i,int j)
{
double value;
value=AF[i][j];
return value;
}
}

```

```

public double SendTheEC(int i,int j)
{
double value;
value=EC[i][j];
return value;
}
public double SendTheEF(int i,int j)
{
double value;
value=EF[i][j];
return value;
}
public String SendThePathF(int i)
{
String State;
State=PathF[i];
return State;
}
}

```

Το τμήμα του προγράμματος όπου γίνεται ο υπολογισμός των πινάκων A και E , από τους αλγορίθμους Forward και Backward, ή από τον Viterbi training.

```

.....
for (int s=0; s<nseqs; s++) // Foreach sequence
{
System.out.print( "." );

if(!TrainingWithViterbi) //if we're not executing viterbi
training
{
Forward fwd = fwds[s]; //forward execution for the protein s
Backward bwd = bwds[s]; //backward execution for the protein s

int L = seqs.seq[s].getLen(); //sequence length
double P = logP[s];

for (int i=1; i<=L; i++)
for (int k=0; k<Params.nstate; k++)
{

E[k][esyminv[seqs.seq[s].getSym(i-1)]] += exp(fwd.f[i][k] +
bwd.GetVal( i, k ) - P); //computing  $E_k^c(b)$  with forward and
backward

}
AddExpC_A( A, seqs.seq[s], P, fwd, bwd ); //computing  $A_{kl}^c$  with
forward and backward
}
else //if we're executing viterbi training
{

```

```

ViterbiTraining SecondWay=new
ViterbiTraining(hmm,seqs.seq[s],"ML");

Viterbilikelihood=Viterbilikelihood+SecondWay.VPROBC; //we're
adding the log viterbi likelihood for every protein

for(int i=0;i<Params.nstate;i++)
{
for(intj=0;j<Params.nstate;j++)
{
A[i][j]=A[i][j]+SecondWay.SendTheAC(i,j); //computing  $A_{kl}^{c*}$  with
viterbi training
}

}
for(int n=0;n<Params.nstate;n++)
{
for(intm=0;m<Params.nesym;m++)
{
E[n][m]=E[n][m]+SecondWay.SendTheEC(n,m); //computing  $E_k^{c*}(b)$  with
viterbi training
}
}
}

} //end foreach sequence
.....

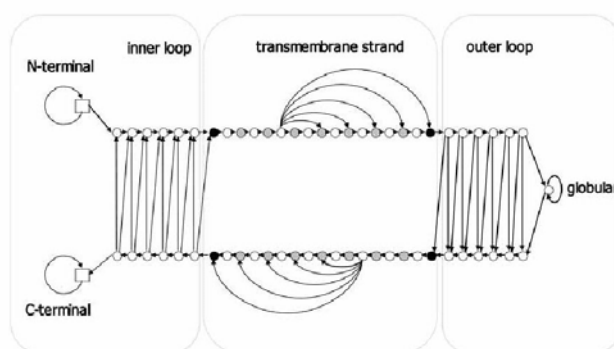
```

4.2: Αποτελέσματα

Εισαγωγή

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα του μοντέλου μας, με την μορφή εικόνων και πίνακα. Στις εικόνες απεικονίζεται η πιθανοφάνεια, ενώ στον πίνακα δείκτες αποδοτικότητας του μοντέλου. Το μοντέλο που χρησιμοποιήθηκε είναι 62 καταστάσεων, ενώ ως training set χρησιμοποιείται ένα set 20 πρωτεϊνών β-βαρελιών.

Η απεικόνιση του μοντέλου:

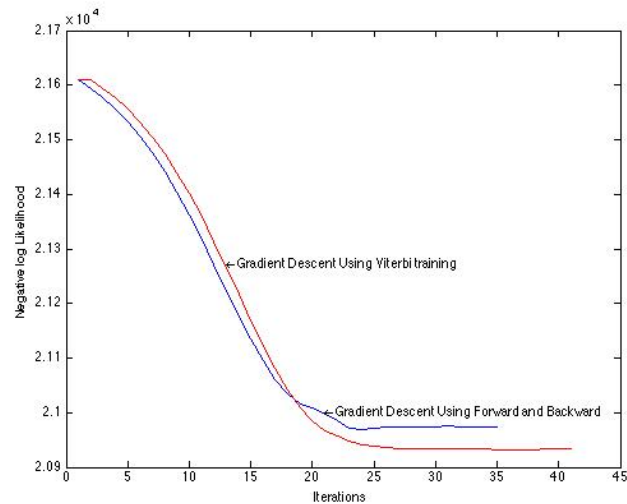


Μοντέλο Εκπαίδευσης

Η εικόνα προέρχεται από το (Bagos, Liakopoulos, & Hamdrakas, 2004)

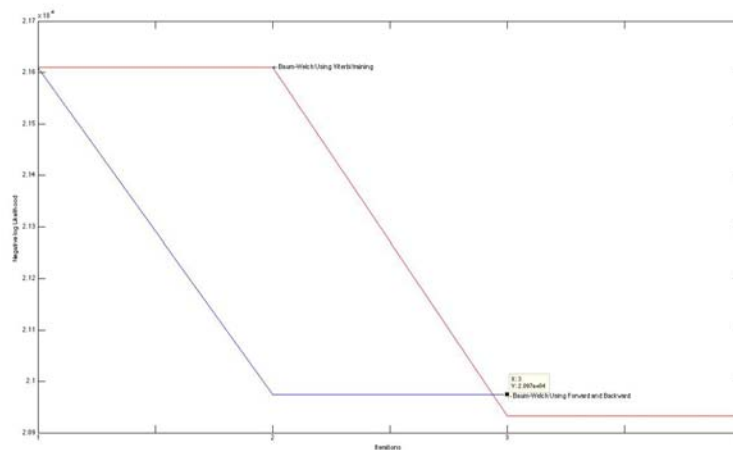
Στην παρακάτω εικόνα εμφανίζονται, η γραφική παράσταση του αρνητικού λογαρίθμου της πιθανοφάνειας ενός σέτ 20 πρωτεϊνών β-βαρελιών (training set), η οποία προκύπτει μετά την εκτίμηση παραμέτρων από τον τροποποιημένο αλγόριθμο Gradient Descent (Bagos, Liakopoulos, & Hamdrakas, 2004), χρησιμοποιώντας για την εύρεση των εκτιμητών μέγιστης πιθανοφάνειας τους αλγορίθμους Forward και Backward, και η γραφική παράσταση του αρνητικού λογαρίθμου της πιθανοφάνειας για το ίδιο training set, η οποία προκύπτει μετά την εκτίμηση παραμέτρων από τον τροποποιημένο αλγόριθμο Gradient Descent (Bagos, Liakopoulos, & Hamdrakas, 2004), χρησιμοποιώντας για την εύρεση των εκτιμητών μέγιστης πιθανοφάνειας τον αλγόριθμο Viterbi training.

Εμφάνιση των δύο παραπάνω γραφικών παραστάσεων σε ένα διάγραμμα



Στην παρακάτω εικόνα εμφανίζονται, η γραφική παράσταση του αρνητικού λογαρίθμου της πιθανοφάνειας ενός σέτ 20 πρωτεϊνών β-βαρελιών (training set), η οποία προκύπτει μετά την εκτίμηση παραμέτρων από τον αλγόριθμο Baum-Welch, χρησιμοποιώντας για την εύρεση των εκτιμητών μέγιστης πιθανοφάνειας τους αλγορίθμους Forward και Backward, και η γραφική παράσταση του αρνητικού λογαρίθμου της πιθανοφάνειας ενός σέτ 20 πρωτεϊνών β-βαρελιών (training set), η οποία προκύπτει μετά την εκτίμηση παραμέτρων από τον αλγόριθμο Viterbi training. Αξίζει να σημειωθεί ότι το μέγιστο πλήθος των επαναλήψεων σε αυτή την περίπτωση είναι 4.

Εμφάνιση των δύο παραπάνω γραφικών παραστάσεων σε ένα διάγραμμα



Πίνακας αποδοτικότητας

Ο παρακάτω πίνακας παρουσιάζει την αποδοτικότητα του μοντέλου, όπως αυτή προέκυψε με την αποκωδικοποίηση από τον αλγόριθμο viterbi. Οι δείκτες αποδοτικότητας είναι: Q= το ποσοστό των σωστά προβλεπόμενων καταλοίπων, C= ο συντελεστής συσχέτισης Matthews, sov= το μέτρο των επικαλυπτόμενων τμημάτων, ο αριθμός των επαναλήψεων, ο χρόνος εκπαίδευσης, και η τιμή της πιθανοφάνειας. Ο αλγόριθμος Gradient Descent είναι ο τροποποιημένος Gradient Descent (Bagos, Liakopoulos, & Hamodrakas, 2004).

Δείκτες	Gradient Descent using Forward and Backward	Gradient Descent Using Viterbi training	Baum-Welch Using Forward And Backward	Viterbi training
Q	0.884	0.880	0.880	0.880
C	0.754	0.746	0.746	0.746
Sov	0.870	0.850	0.850	0.850
Negative Log Likelihood	20973	20932	20974	20932
Iterations	35	41	3	4
Training Time (sec)	32	25	3	3

4.3: Συζήτηση

Σύμφωνα με τα αποτελέσματα που παρουσιάστηκαν στην ενότητα 4.2, η αποδοτικότητα του μοντέλου μειώνεται ελάχιστα “οριακά”, στην περίπτωση όπου, κατά την εκτίμηση των παραμέτρων από τους αλγόριθμους, τροποποιημένος Gradient Descent (Bagos, Liakopoulos, & Hamodrakas, 2004) και Viterbi training, ο υπολογισμός των εκτιμητών μέγιστης πιθανοφάνειας γίνεται από τον Viterbi training, σε σύγκριση με την περίπτωση όπου, κατά την εκτίμηση των παραμέτρων από τους αλγόριθμους, τροποποιημένος Gradient Descent (Bagos, Liakopoulos, & Hamodrakas, 2004) και Baum-welch, ο υπολογισμός των εκτιμητών μέγιστης πιθανοφάνειας, γίνεται με την χρήση των Forward και Backward.

Το κύριο πλεονέκτημα του αλγορίθμου Viterbi training, όπως παρουσιάζεται και στον πίνακα αποδοτικότητας, είναι η μείωση του χρόνου εκπαίδευσης.

Επομένως, με τον αλγόριθμο Viterbi training μπορούμε να εκπαιδεύουμε τα μοντέλα σε λιγότερο χρόνο, με μία μικρή μεταβολή της απόδοσης των μοντέλων. Το γεγονός αυτό τον καθιστά ιδανικό σε περιπτώσεις όπου τα προς περιγραφή

συστήματα χρειάζονται μεγάλη υπολογιστική ισχύ (μεγάλα μοντέλα, πολύ μεγάλος όγκος πληροφορίας)

Βιβλιογραφία

- Bagos, P. G., Liakopoulos, T. D., & Hamodrakas, S. J. (2004). Faster Gradient Descent Conditional Maximum Likelihood Training of Hidden Markov Models, Using Individual Learning Rate Adaptation. In G. Paliouras & Y. Sakakibara (Eds.), *Grammatical Inference: Algorithms and Applications* (Vol. 3264, pp. 40-52): Springer Berlin/Heidelberg.
- Bagos, P. G., Liakopoulos, T. D., & Hamodrakas, S. J. (2006). Algorithms for incorporating prior topological information in HMMs: application to transmembrane proteins. *BMC Bioinformatics*, 7, 189.
- Bagos, P. G., Liakopoulos, T. D., Spyropoulos, I. C., & Hamodrakas, S. J. (2004). A Hidden Markov Model method, capable of predicting and discriminating β -barrel outer membrane proteins. *BMC bioinformatics*, 5(1), 1.
- Baldi, P., & Chauvin, Y. (1994). Smooth On-Line Learning Algorithms for Hidden Markov Models. *Neural Comput*, 6(2), 305-316.
- Baum, L. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3, 1-8.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J Royal Stat Soc B*, 39, 1-38.
- Durbin, R., Eddy, S. R., Krogh, A., & Mithison, G. (1998). *Biological sequence analysis, probabilistic models of proteins and nucleic acids.*: Cambridge University Press.
- Juang, B. H., & Rabiner, L. R. (1990). The Segmental K-Means Algorithm for Estimating Parameters of Hidden Markov Models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(9), 1639-1641.
- Krogh, A. (1994). Hidden Markov models for labelled sequences. *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 140-144.
- Krogh, A. (1997). Two methods for improving performance of an HMM and their application for gene finding. *Proc Int Conf Intell Syst Mol Biol*, 5, 179-186.
- Krogh, A., Brown, M., Mian, I. S., Sjolander, K., & Haussler, D. (1994). Hidden Markov models in computational biology. Applications to protein modeling. *J Mol Biol*, 235(5), 1501-1531.
- Krogh, A., & Riis, S. K. (1999). Hidden neural networks. *Neural Comput*, 11(2), 541-563.
- MacQueen, B. (1967). *Some Methods for classification and Analysis of Multivariate Observations*. Paper presented at the Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability.
- Markov, A. A. (1913). An example of statistical study on text of Eugeny Onegin illustrating the linking of events to a chain. *Izvestija Imp. Akad. nauk*, 6(3), 153-162.
- Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2), 257-286.

