2η Εργασια κατανεμημενα λειτουργικα συστηματα

Νικος Γεωργιαδης αεμ 2043

Θοδωρης Φασουλας αεμ 2096

Δημητρης Αγτζιδης αεμ 2040

ΠΡΟΒΛΗΜΑ ΤΑΞΙΝΟΜΗΣΗΣ

Στην εργασια αυτη καλουμαστε να υλοποιησουμε με τη χρηση του Hadoop ενα δεντρο αποφασεων και πιο συγκεκριμενα τον αλγοριθμο Id3 για να προσεγγισουμε ενα προβλημα ταξινομησης.Το συνολο δεδομενων που θα χρησιμοποιησουμε περιεχει αξιολογησεις σε αυτοκινητα με βαση ορισμενα χαρακτηριστικα.

ΥΛΟΠΟΙΗΣΗ

Το προγραμμα ξεκιναει μπαινοντας σε ενα βρογχο με 10 επαναληψεις οπου ειναι και η υλοποιηση του ten-fold cross validation.Ο αλγοριθμος θα τρεξει 10 φορες καθε φορα με διαφορετικο αρχειο εκπαιδευσης και διαφορετικο αρχειο αξιολογησης.Στη συνεχεια μπαινει σε ενα αλλο loop (while) οπου ελεγχεται με ενα flag αν ολοι οι mappers σε μια map/reduce φαση δεν εχουν επικοινωνισει με τον reducer ή αν ειναι η πρωτη φορα που ξεκιναει η map/reduce φαση για μια απο τις 10 επαναληψεις του αλγοριθμου.

Αφου φτιαξω καταλληλα τις εισοδους και εξοδους για καθε map/reduce φαση ,τα στελνω σαν παραμετρους στην Start_Job() απο οπου και ξεκιναει η διαδικασια του hadoop.Αυτο που θα τρεχει παραλληλα στο hadoop θα ειναι οι κομβοι-χαρακτηριστικα που βρισκονται καθε φορα στο ιδιο υψος του δεντρου αποφασεων.Δηλαδη θα τρεχουν διαφορετικα μονοπατια που στην προηγουμενη map/reduce φαση ειχαν το ιδιο μονοπατι. Το key και value που θα εξαγονται απο την mapper και την reducer καθε φορα θα ειναι τυπου text αφου το key στην mapper ειναι το μονοπατι μαζι με το αρχειο εκπαιδευσης και το value ειναι μονο το αρχειο εκπαιδευσης ενω στην reducer το value θα ειναι null για λογους ευκολιας .

Map(key=δεν μας χρειαζεται,value=μονοπατι_αρχειο εκπαιδευσης)

Εαν το count==0 δηλαδη εαν ειναι η πρωτη map/reduce φαση μιας επαναληψης του αλγοριθμου τοτε το μονοπατι θα ειναι κενο αρα το value περιεχει μονο το αρχειο εκπαιδευσης οπου και το παιρναμε στο samples[][].Εαν ειναι !=0 τοτε σπαμε το value και στο path αποθηκευουμε το μονοπατι ενω στο samples[][] το αρχειο εκπαιδευσης.Στην συνεχεια εαν το path ειναι κενο σημαινει οτι μπηκαμε για πρωτη φορα στην map και γιαυτο θα παρουμε ενα κομματι απο το samples[][] οπου θα το χρησιμοποιησουμε σαν αρχειο αξιολογησης (samples _ [][]). Παρακατω εαν το μονοπατι ειναι κενο , βρισκουμε το Gain ολων των χαρακτηριστικων με την Gain_Calculation() αλλιως βρισκουμε το Gain απο τα χαρακτηριστικα τα οποια δεν υπαρχουν ηδη μεσα στο μονοπατι .Βρισκουμε το max Gain

μετα και αν το max Gain ειναι ==0 αυτο σημαινει ειτε οτι εχουμε χρησιμοποιησει ολα τα χαρακτηριστικα μεσα στο μονοπατι ειτε οτι το αρχειο εκπαιδευσης ειναι μιας κλασης.Αυτο σημαινει οτι ειμαστε σε φυλλο αρα θα γραψουμε το μονοπατι σε ενα αρχειο paths.txt. και δεν θα χρησιμοποιησουμε καθολου τον Reducer.Εαν ειναι !=0 τοτε βρισκουμε ολες τις τιμες του χαρακτηριστικου που επιλεξαμε και δημιουργουμε τοσα μονοπατια τα οποια θα τα εκχωρησουμε ως key μαζι με το αντιστοιχο αρχειο εκπαιδευσης του και ως value θα εκχωρησουμε το αρχειο εκπαιδευσης. **(context.write(key=path+αρχειο εκπαιδευσης,value=αρχειο εκπαιδευσης).**

Reducer(key,value)

Εδω ελεγχουμε εαν το value δηλαδη το αρχειο εκπαιδευσης ειναι κενο δλδ αν δεν εχω στειλει κανενα αρχειο αρα ειμαι σε φυλλο.Αν δεν ειναι κενο τοτε γραφω το key που εχω ως τωρα και value θα το εχω κενο αλλιως δεν κανω τιποτα.

(context.write(key=path+αρχειο εκπαιδευσης, value=null)

Gain_Calculation()

Εδω μεσα υπολογιζω το Κερδος Πληροφοριας του χαρακτηριστικου που εξεταζω καθε φορα.

Total_Entropy= p1*(log(p1)/log2)+ p2*(log(p2)/log2)+ p3*(log(p3)/log2)+ p4*(log(p4)/log2)

S(u=value)= p1(value)*(log(p1(value))/log2)+ p2(value)*(log(p2(value))/log2)+ p3(value)*(log(p3(value))/log2)+ p4(value)*(log(p4(value))/log2)

S(u=value) …

S(u=value)..

Gain=Total_Entropy-(new_c_all/c_all)*S(u=value) + …..)

Το new_c_all ειναι το ποσα δειγματα εχουν την συγκεκριμενη τιμη απο το χαρακτηριστικο που εξεταζουμε και το c_all ειναι το ποσα δειγματα υπαρχουν συνολικα ανεξαρτητα απο την τιμη του χαρακτηριστικου.

ΠΙΣΩ ΣΤΗΝ MAIN() Evaluation Implementation

Αφου εχει τελειωσει η διαδικασια της εκπαιδευσης του ταξινομητη με μια επαναληψη του αλγοριθμου με την βοηθεια hadoop, το επομενο κομματι ειναι η αξιολογηση του ταξινομητη. Διαβαζουμε το αρχειο με τα μονοπατια που δημιουργθηκε και το αποθηκευω στο results[][].Στην συνεχεια για να το εχω σε καλυτερη μορφη και να γινει ευκολοτερη η επεξεργασια το αποθηκευω στο new_results .Για καθε κλαση καλω την Evaluation()

Evaluation()

Στη συναρτηση αυτη με το αρχειο αξιολογησης που εχω (samples _ [][]) εξεταζω για καθε δειγμα του σε ποιο μονοπατι αντιστοιχει.Βρισκω αυτο το μονοπατι και περνω την κλαση του.Εαν η κλαση ειναι ιδια με αυτη του δειγματος τοτε True positive++ αλλιως False positive++ . Εαν εξεταζουμε αλλη κλαση και προβλεψουμε την παραπανω κλαση τοτε False positive ++

Οταν τελειωσω αυτη την διαδικασια για ολες τις κλασεις τοτε για καθε κλαση θα βρω το precision , recall , fmeasure και στην συνεχεια τον μεσο ορο γιαυτες τις μετρικες απο ολες της κλασεις.(average_precision,average_recall,average_fmeasure)

Οταν εχω πραγματοποιησει το ten fold cross validation δηλαδη εχω τρεξει τον αλγοριθμο 10 φορες, θα υπολογισω τον μεσο ορο απο ολους τους μεσους ορους που εχω βρει για καθε μετρικη.(final_precision,final_recall,final_fmeasure) .Τελος θα τα γραψω σε ενα αρχειο με ονομα “results.txt”.

Τα αποτελεσματα απο το αρχειο ειναι : Final_precision 81% Final_recall 68% Final_fmeasure 65%

Τελος προγραμματος

ΣΥΝΔΕΣΗ

ΜΕ ΤΗΝ ΒΟΗΘΕΙΑ 2 VIRTUAL BOX ΤΟ ΕΝΑ ΛΕΙΤΟΥΡΓΩΝΤΑΣ ΩΣ MASTER ΚΑΙ SLAVE ΚΑΙ ΤΟ ΑΛΛΟ ΩΣ SLAVE ΚΑΝΟΝΤΑΣ ΤΟ ΚΑΤΑΛΛΗΛΟ SETUP ΚΑΙ ΜΕ ΤΗΝ ΒΟΗΘΕΙΑ ΤΟΥ HADOOP ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΑΝ ΟΙ MAP/REDUCE ΦΑΣΕΙΣ.