Γεια σας. Ονομάζομαι Αλέξανδρος Γαϊτάνης κ θα σας παρουσιάσω τις εργασίες μου για το μάθημα της Υπολογιστικής Νοημοσύνης.

Στην 1η και στη 2η Εργασία έγινε κατηγοριοποίηση σε 2 dataset. Υλοποιήθηκαν και αξιολογήθηκαν αλγόριθμοι SVM, Kernel PCA + LDA κι έγινε σύγκριση με τους αντίστοιχους αλγορίθμους της βιβλιοθήκης sklearn.

Το 1ο dataset που χρησιμοποιήθηκε είναι η MNIST η οποία αποτελείται από ασπρόμαυρες εικόνες χειρόγραφων ψηφίων 0-9. Γίνεται classification σε μονούς και ζυγούς αριθμούς. Το training set έχει μέγεθος 60000 και το test set 10000. Σε κάποιες περιπτώσεις χρησιμοποιήθηκε ένα μικρό training set 10000 εικόνων γιατί ορισμένοι αλγόριθμοι ήταν πολύ απαιτητικοί για τον υπολογιστή μου, ιδιαίτερα σε μνήμη.

Η διαδικασία έχει ως εξής. Γίνεται πρώτα κανονικοποίηση και σε αρκετές περιπτώσεις PCA όπου μειώνεται η διάσταση από 784 σε 87. Έπειτα γίνεται επιλογή των παραμέτρων των αλγορίθμων αξιολογώντας τους σε ένα holdout validation set. Η τελική αξιολόγηση γίνεται στο test set και ως μετρική χρησιμοποιήθηκε το accuracy.

Υλοποιήθηκε αλγόριθμος SVM στον οποίο γίνεται εύρεση των πολλαπλασιαστών Lagrange του δυικού προβλήματος των SVM με τον quadratic programming solver CVXOPT. Ξόδεψα αρκετό χρόνο προσπαθώντας να ­βρω ποια είναι τα κατάλληλα tolerances για να θεωρήσω μηδενικούς τους πολλαπλασιαστές γιατί ο αριθμός των support vectors επηρέαζε πολύ τα αποτελέσματα. Τελικά επέλεξα κάποια tolerances έτσι ώστε να βγάλω παρόμοια αποτελέσματα με την sklearn.­­

Υλοποιήθηκε αλγόριθμος Kernel PCA. Υπολογίζεται πρώτα ο centralized Gramian matrix. Γίνεται ιδιοανάλυση και ταξινομούνται οι μη μηδενικές ιδιοτιμές από την μικρότερη στην μεγαλύτερη. Στο τέλος προβάλλονται τα δείγματα στον χώρο που έχει ως βάση τα αντίστοιχα ιδιοδιανύσματα.

Επίσης υλοποιήθηκε αλγόριθμος LDA με την απλή τεχνική των δύο κλάσεων με την οποία υπολογίζεται το διάνυσμα w στο οποίο προβάλλονται τα δείγματα με τον τύπο που βλέπουμε εδώ. Ως μελλοντική βελτίωση προτείνω τη χρήση του ψευδοαντίστροφου για την αποφυγή περιπτώσεων όπου ο Sw είναι μη αντιστρέψιμος.

Εδώ βλέπουμε τα αποτελέσματα της sklearn για ολόκληρο το training set. Βλέπουμε ότι παρατηρείται υποπροσαρμογή για το linear και το sigmoid ενώ το polynomial και το RBF δίνουν πολύ καλά αποτελέσματα.

Εδώ βλέπουμε και τους χρόνους εκπαίδευσης. Για το linear μετά από κάποια τιμή του C, τα πράγματα γίνονται απαγορευτικά και δεν συνέχισα τα πειράματα. Βλέπουμε για το polynomial ότι καθώς αυξάνεται ο βαθμός του πολυωνύμου αυξάνεται και ο χρόνος, το οποίο είναι αναμενόμενο.

Προχωράμε στις συγκρίσεις. Για linear τα πράγματα είναι ίδια, εκτός από τον χρόνο, όπου σε μένα είναι πολύ πιο μεγάλος λόγω των μεγάλων πινάκων που χρησιμοποιώ, ενώ η sklearn εφαρμόζει πιο efficient μεθόδους.

Παρόμοια αποτελέσματα και για polynomial. Πάλι ισχύει η ίδια παρατήρηση για τον χρόνο.

Παρόμοια και για RBF

Προχωράμε στους γείτονες. Οι nearest neighbors δίνουν πολύ καλά αποτελέσματα ενώ το nearest class centroid όχι.

Πάμε τώρα στο Kernel PCA + LDA, όπου βλέπουμε παρόμοια αποτελέσματα για το linear. Να σημειώσω εδώ ότι μάλλον δεν είχε και πολύ νόημα που έβαλα linear PCA πριν από το LDA αλλά παρόλα αυτά σας παρουσιάζω τα αποτελέσματα.

Για το polynomial με τη μέθοδο μου, βλέπουμε απόλυτο διαχωρισμό των κλάσεων και πολύ καλά score, ενώ η sklearn δεν τα πηγαίνει καθόλου καλά με τον svd solver που χρησιμοποιεί. Επίσης οι χρόνοι είναι καλύτεροι σε μένα.

Παρόμοια αποτελέσματα και για το RBF όπου πάλι η sklearn δεν τα πηγαίνει καλά.

Εδώ βλέπουμε αποτελέσματα σε ολόκληρο το training set. Τα SVM με polynomial και RBF kernels υπερτερούν και πετυχαίνουν test accuracy 0.99.

Και αποτελέσματα στο μικρό training set. Πολύ καλά τα πηγαίνουν οι nearest neighbors και γενικά οι αλγόριθμοι με polynomial και RBF kernels.

Προχωράμε στο 2ο dataset. Οι κλάσεις είναι 6 τύποι γυαλιού οι οποίοι αναγνωρίζονται με βάση κάποιες μετρήσεις χημικών στοιχείων. Είναι ένα μικρό dataset, το οποίο χωρίζεται σε training set και test set μεγέθους 128 και 86 αντίστοιχα.

Πρώτα γίνεται πάλι κανονικοποίηση. Επειδή είναι μικρό το dataset χρησιμοποιείται 5-fold cross-validation για το parameter tuning. Ως μετρική αξιολόγησης χρησιμοποιείται το weighted F1. Ως μελλοντική βελτίωση, προτείνω τη χρήση ενός Power Transformer ο οποίος μετατρέπει τις κατανομές σε gaussian και είδα ότι μπορεί να δώσει καλύτερα αποτελέσματα για όλα τα μοντέλα.

Χρησιμοποιήθηκε υλοποίηση LDA όπου επιλύεται το γενικευμένο πρόβλημα ιδιοανάλυσης για τους πίνακες Sb και Sw.

Βλέπουμε τα αποτελέσματα της sklearn. Παρατηρείται υποπροσαρμογή για το linear και το sigmoid και υπερπροσαρμογή για το polynomial και το RBF.

Βλέπουμε αποτελέσματα και για τους γείτονες. Πάλι παρατηρείται υποπροσαρμογή για τον nearest class centroid ενώ καλά αποτελέσματα για το nearest neighbors.

Προχωράμε στο Linear Kernel PCA + LDA όπου βλέπουμε παρόμοια αποτελέσματα μεταξύ της δικής μου υλοποίησης και της sklearn. Κλασικά παρατηρείται υποπροσαρμογή για το linear kernel.

Εδώ για την sklearn προκύπτουν διαφορετικές βέλτιστες παράμετροι. Κατά τα άλλα βλέπουμε παρόμοια αποτελέσματα μετά την LDA.

Εδώ έχω κάπως καλύτερα αποτελέσματα για το RBF σε σχέση με την sklearn. Και στις δυο περιπτώσεις παρατηρείται υπερπροσαρμογή.

Συνοπτικά αποτελέσματα. Καλύτερο F1 έχει ο Nearest Neighbors και ακολουθούν τα SVM και τα Kernel PCA + LDA.

Και πάμε στην 3η εργασία η οποία έχει να κάνει με ομαδοποίηση.

Υλοποιήθηκε spectral clustering χρησιμοποιώντας unnormalized και normalized graph laplacian και έγινε σύγκριση με τον αντίστοιχο αλγόριθμο της sklearn ο οποίος χρησιμοποιεί τον normalized.

Ας δούμε λίγο την διαδικασία. Ξεκινάμε πάλι με κανονικοποίηση. Έπειτα γίνεται μείωση διάστασης σε 2 με κάποιο embedding το οποίο επιλέγουμε. Το embedding γίνεται μέσω sklearn. Ακολουθεί spectral clustering με δικιά μου υλοποίηση και με sklearn το οποίο αξιολογείται με τις μετρικές ομοιογένεια, completeness και v-measure, αφού γνωρίζουμε τα labels του training set. Τέλος θεωρώντας ότι όλα τα δείγματα της κάθε ομάδας έχουν το ίδιο label, συγκεκριμένα επιλέγεται το πιο συχνό, γίνεται classification με Nearest Class Centroid και αξιολόγηση στο test set. Χρησιμοποιούνται οι κλασικές μετρικές για το classification.

Βλέπουμε εδώ τον αλγόριθμο του Spectral Clustering. Αρχικά χτίζεται ο γράφος ομοιότητας με βάση τους k πλησιέστερους γείτονες. Από αυτόν προκύπτει ο πίνακας γειτνίασης G. Αυτός ο πίνακας δεν είναι συμμετρικός γιατί αν ένα δείγμα Α έχει γείτονα το Β δε σημαίνει ότι ισχύει και το αντίστροφο. Ο S όμως είναι συμμετρικός και χρησιμοποιείται ως πίνακας ομοιότητας. Έπειτα πραγματοποιείται ιδιοανάλυση στον graph laplacian. Τα ιδιοδιανύσματα αποτελούν τα στήλες του νέου πίνακα δειγμάτων τα οποία έχουν μετασχηματιστεί σε ένα νέο χώρο. Χρησιμοποιούνται ειδικές δομές και συναρτήσεις για αραιούς πίνακες για καλύτερη πολυπλοκότητα χώρου και χρόνου. Στο τέλος πραγματοποιείται clustering με k-means.

1ο dataset είναι η MNIST πάλι.

Εδώ βλέπουμε διάφορα embeddings που δοκιμάστηκαν. Επέλεξα να συνεχίσω με το t-SNE γιατί θεώρησα ότι με τα άλλα δεν θα έβγαζα καθόλου καλά αποτελέσματα.

Και ξεκινάμε και βλέπουμε τα αποτελέσματα της ομαδοποίησης για διαφορετικούς αριθμούς ομάδων…………………..

Και πάμε τώρα στα scores. Τα clustering και classification scores δείχνουν ότι ο βέλτιστος αριθμός ομάδων είναι 11 ή 12. Μετά από αυτή την τιμή δε φαίνεται κάποια βελτίωση στα classification scores και το v-measure χειροτερεύει.

Βλέπουμε παρόμοια αποτελέσματα για n\_clusters = 10 για όλους τους αλγορίθμους. Παρατηρείται σύγχυση μεταξύ των ψηφίων 1, 8 και 4, 9 σε όλες τις περιπτώσεις.

Και πάμε στο 2ο dataset. Οι κλάσεις είναι 5 τύποι καρκίνου και τα χαρακτηριστικά είναι επίπεδα έκφρασης διαφόρων γονιδίων. To training set έχει μέγεθος 480 και το test set 321.

Επιλέγεται spectral embedding με RBF affinity και gamma=10-4. Επιλέχθηκαν αυτές οι τιμές για να μας δώσουν κάτι ενδιαφέρον για να συνεχίσουμε με το clustering. Κάποιες τιμές έδιναν διαχωρισμένες ομάδες που θα μπορούσαν να αναγνωριστούν εύκολα μόνο με k-means.

Και ξεκινάμε και βλέπουμε τα αποτελέσματα της ομαδοποίησης για διαφορετικούς αριθμούς ομάδων…………………………

Και πάμε τώρα στα scores τα οποία δείχνουν ότι ο βέλτιστος αριθμός ομάδων είναι 5. 5 ήταν και οι κλάσεις του dataset.

Σύμφωνα με το eigengap heuristic όμως, όπως βλέπουμε εδώ, ο βέλτιστος αριθμός ομάδων είναι 4.

Και βλέπουμε εδώ τα αποτελέσματα για n\_clusters = 5, τα οποία είναι ίδια και για τους 3 αλγόριθμους.

Ευχαριστώ.