

Οι αλγόριθμοι που θα χρησιμοποιηθούν είναι οι:

(Isomap , LLE , MDS ,tSNE) οι οποίοι ανοίκουν όλοι στην οικογένεια των Spectral Embedding αλγορίθμων και επιλέχθηκαν αυτοί προκειμένου να παρουσιάσουμε περισσότερο τις διαφορές που υπάρχουν.

Ας αρχίσουμε λοιπόν με τον απλοικότερο αλγόριθμο ο οποίος είναι ο Locally Linear Embedding η αλλιώς (LLE)

LLE

Περιγραφή Αλγορίθμου:

Ο LLE αλγόριθμος επιχειρεί να δημιουργήσει μια ενσωμάτωση σε έναν χώρο χαμηλότερης διάστασης τέτοια ώστε τα σημεία που είναι «κοντά» στην υψηλή διάσταση να παραμένουν κοντά και στην χαμηλή διάσταση.

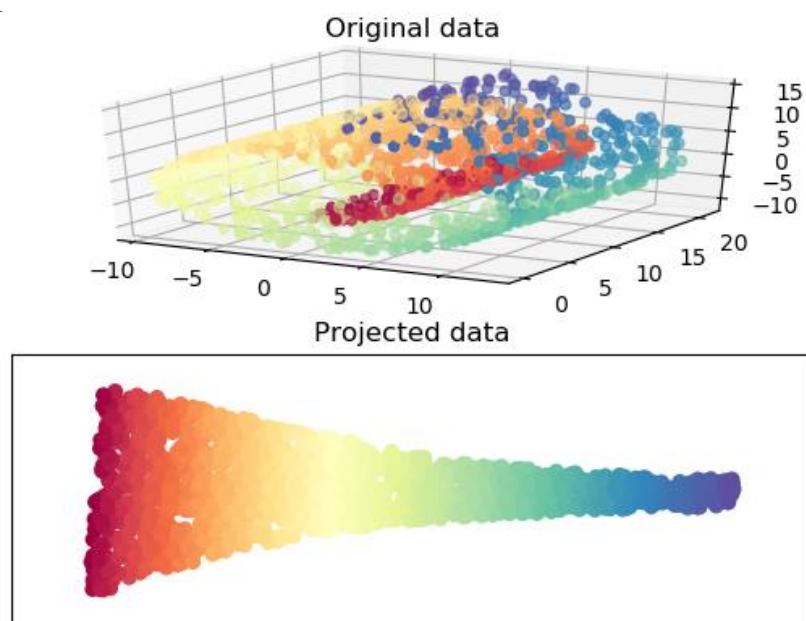
Το σφάλμα ελαχιστοποιείται όταν γειτονίες σημείων στην υψηλή διάσταση παραμένουν αναλλοίωτες στην χαμηλή διάσταση.

Ο LLE δημιουργεί την ενσωμάτωση βασιζόμενος μόνο τις τοπικές αποστάσεις, παραλείποντας τις σφαιρικές (global) αποστάσεις. Υποθέτει επιπλέον ότι τα δεδομένα απλώνονται πάνω σε μια λεία επιφάνεια το οποίο στα μαθηματικά σημαίνει ότι δεν έχει τρύπες και ότι για κάθε σημείο των δεδομένων υπάρχει μια γειτονιά από υπόλοιπα σημεία.

Ο LLE παίρνει το όνομα του ακριβώς επειδή η κατασκευή του βασίζεται μόνο στις γειτονιές των τοπικών σημείων (local).

Δεν θα επιχειρήσουμε να δώσουμε την μαθηματική περιγραφή του αλγορίθμου παρα μόνο να δείξουμε ένα παράδειγμα εφαρμογής του σε ένα κλασικό Swiss Roll

Σχήμα 1.1



Πλεονεκτήματα : Ο LLE έχει μερικές πολύ ωραίες ιδιότητες. Πρώτον διαθέτει μια κλειστή λύση του προβλήματος διάστασης δεδομένων, δηλαδή η βελτιστοποίηση του είναι κυρτή σε αντίθεση με άλλους αλγορίθμους. Δεύτερον ο πίνακας των βαρών “Weight Matrix” W είναι αραιός εννοώντας ότι περιέχει αρκετά μηδενικά επειδή κάθε σημείο βασίζεται μόνο σε μια γειτονία από άλλα σημεία το οποίο κάνει γρήγορους τους υπολογισμούς μας. Τρίτον οι μόνες παράμετροι που έχουμε να προσδιορίσουμε είναι η διάσταση της ενσωμάτωσης και το πλήθος των γειτόνων.

Μειονεκτήματα : The Collapse Problem .Το κύριο πρόβλημα του αλγορίθμου είναι ότι προσπαθεί να βελτιστοποιήσει το λάθος πρόβλημα εννοώντας ότι χρησιμοποιεί μόνο έναν πίνακα βαρών για να δημιουργήσει σημεία στον καινούριο χώρο που θα έχουν τις ίδιες αποστάσεις με τον υψηλότερης διάστασης χώρο. Εξαιτίας αυτού δεν έχει επίσης κάποιο κίνητρο να κρατήσει απομακρυσμένα τα δεδομένα στον χώρο μικρότερης διάστασης.

Isometrical Mapping (Isomap)

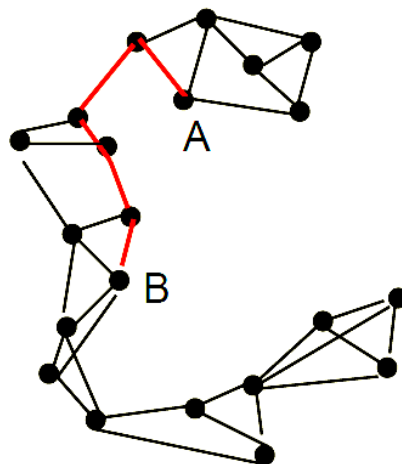
Περιγραφή αλγορίθμου:

Ο Isomap είναι ένας αλγόριθμος μη γραμμικής μείωσης διάστασης ο οποίος προσπαθεί να διατηρήσει αναλλοίωτες τις γεοδεσιακές αποστάσεις στον χώρο μικρότερης διάστασης. Έπειτα μέσω ιδιοανάλυσης του πίνακα των γεοδεσιακών αποστάσεων βρίσκει την ενσωμάτωση στην χαμηλότερη διάσταση (μπορούμε να το φανταστούμε σαν το KernelPCA). Σε μη γραμμικές πολλαπλότητες η ευκλείδεια μετρική είναι κατάλληλη αν και μόνο αν η κατασκευή των γειτονικών περιοχών μπορεί να θεωρηθεί ως γραμμική. Αν όμως η πολλαπλότητα (manifold) περιέχει τρύπες τότε η ευκλείδεια μετρική μπορεί να είναι παραπλανητική.

Πώς βρίσκει τις γεοδεσιακές αποστάσεις?

Ο τρόπος με τον οποίο ο αλγόριθμος προσεγγίζει τις γεοδεσιακές αποστάσεις μεταξύ κάθε σημείου των δεδομένων είναι δημιουργώντας έναν γράφο και στην συνέχεια βρίσκοντας το κοντινότερο μονοπάτι κάθε σημείου από οποιοδήποτε άλλο. Ακολουθεί μια εικόνα για να το καταλάβουμε καλύτερα.

Σχήμα 1.2



Στην συνέχεια αφού έχει υπολογίσει τον πίνακα ομοιοτήτων με τις γεωδειακές αποστάσεις χρησιμοποιεί ιδιοανάλυση στον πίνακα προκειμένου να βρεί την ενσωμάτωση στον νέο χώρο μικρότερης διάστασης.

Πλεονεκτήματα και Μειονεκτήματα: Ο Isomap είναι ένας ισχυρός αλγόριθμος μείωσης διάστασης δεδομένων ο οποίος σε αντίθεση με τους υπόλοιπους δεν βασίζεται μονάχα στις τοπικές αποστάσεις των σημείων μιας γειτονιάς αλλά επιπροσθέτως χρησιμοποιεί τις αυτή την πληροφορία για να φτιάξει έναν καθολικό πίνακα αποστάσεων μεταξύ των σημείων, δηλαδή χρησιμοποιεί ταυτόχρονα και την τοπική αλλά και την καθολική γεωμετρία του συνόλου δεδομένων μας. Παρ'όλα αυτά τα αποτελέσματα μας μπορεί να είναι αρκετά φτωχά όταν δεν υπάρχει σωστή δειγματοληψία ανάμεσα στα δεδομένα μας και περιέχουν «τρύπες».

MultiDimensionalScaling (MDS)

Περιγραφή Αλγορίθμου:

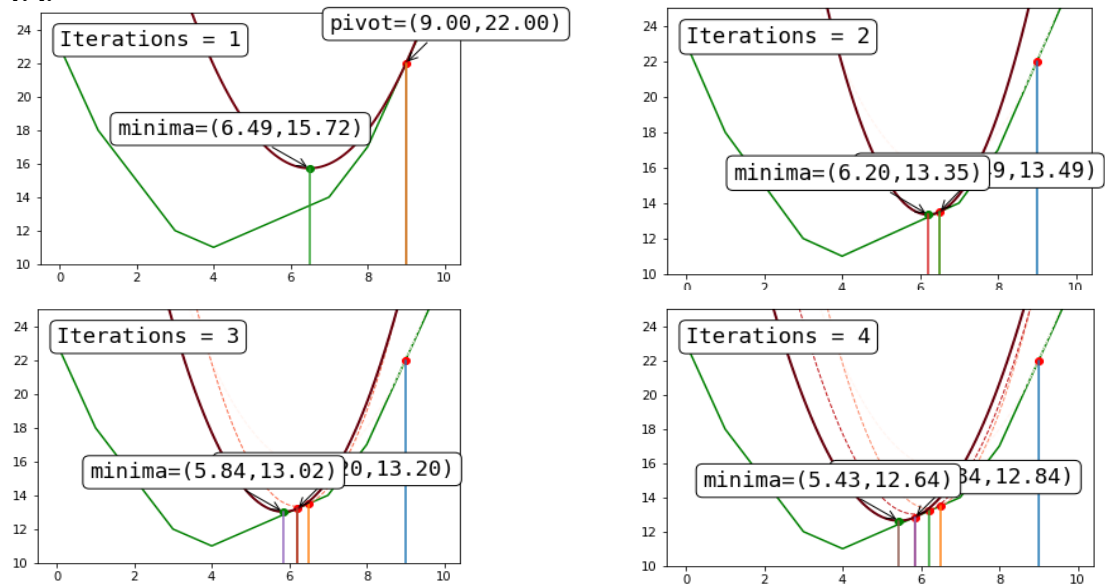
Ο MDS δεν είναι μια συγκεκριμένη τεχνική αλλά μια οικογένεια τεχνικών. Αυτό που κάνει είναι να δέχεται έναν πίνακα ομοιοτήτων D όπου D_{ij} αποτελεί την ομοιοότητα του i με το j και στην συνέχεια παράγει μια ενσωμάτωση των δεδομένων μας σε μικρότερη διάσταση βασιζόμενος σε αυτόν τον πίνακα ομοιοτήτων.

Ο MDS διακρίνεται σε δύο κατηγορίες (Metric Mds, Non Metric) θα περιγράψουμε όμως μόνο τον metric ο οποίος μας ενδιαφέρει σε αυτήν την περίπτωση

Metric MDS : Ο metric mds προσπαθεί να διατηρήσει την αρχική ομοιότητα των μετρικών (πχ Ευκλίδεια). Δοθέντος ενός πίνακα D , μια **μονότονη** συνάρτηση f και την διάσταση της ενσωμάτωσης p ο metric mds προσπαθεί να βρεί έναν χώρο $X \subset \mathbb{R}^p$ s.t. $f(D_{ij}) \approx d_{ij} = (x_i - x_j)^2$. Επίσης μια άλλη μορφή του MDS είναι ο κλασσικός MDS ο οποίος αντι να προσπαθεί να βρεί τον κατάλληλο μετασχηματισμό που θα ρίχνει την διάσταση, χρησιμοποιεί ιδιοανάλυση για να βρεί την λύση και το οποίο παρέχει μια κυρτή λύση του προβλήματος (δεν αλλάζει μετα από πολλαπλές επαναλήψεις).

Ο τρόπος με τον οποίο ο Mds προσπαθεί να βρεί τον κατάλληλο μετασχηματισμό είναι χρησιμοποιώντας έναν MM αλγόριθμο (Majorize-Minimization) όπως για παράδειγμα ο αλγόριθμος SMACOF προκειμένου να βρεί ένα τοπικό βέλτιστο της συνάρτησης μετασχηματισμού. Ακολουθεί μια οπτικοποίηση για να το καταλάβουμε καλύτερα.

Σχήμα 1.3



Πλεονεκτήματα και Μειονεκτήματα: Ο Metric MDS είναι καλύτερος από τους υπόλοιπους κλασσικούς MDS αλγορίθμους σε ‘μη γραμμικές’ πολλαπλότητες αλλά βέβαια απαιτεί αρκετούς υπολογισμούς ($O(N^2)$) καθώς υπολογίζει όλες τις αποστάσεις μεταξύ των σημείων.

Επίσης είναι αρκετά δύσκολο να ενσωματώσεις καινούρια δεδομένα με τον MDS καθώς δεν παρέχεται ένας πίνακας μετασχηματισμού όπως στην περίπτωση του (LLE, Isomap) αλλά η ενσωμάτωση προήλθε από ένα μη κυρτό πρόβλημα βελτιστοποίησης μιας συνάρτησης f . Ακόμη δεν είμαστε σίγουροι ότι η ενσωμάτωση που παρέχει είναι η καλύτερη καθώς ο αλγόριθμος βρίσκει τοπικό και όχι ολικό βέλτιστο και τέλος πολλαπλές επαναλήψεις οδηγούν σε διαφορετικά αποτελέσματα.

t-Stochastic Neighbor Embedding (tSNE)

Περιγραφή αλγορίθμου:

Ο tSNE είναι μια επέκταση του αλγορίθμου SNE οπότε για να το καταλάβουμε καλύτερα πρέπει πρώτα να πούμε μερικά λόγια για τον SNE αλγόριθμο.

SNE (Stochastic Neighbor Embedding) : Ο SNE χρησιμοποιεί πιθανοκρατική προσέγγιση προκειμένου να βρεί την ενσωμάτωση σε χώρο μικρότερης διάστασης διατηρώντας την δομή των γειτονιών στο dataset. Για κάθε σημείο του συνόλου δεδομένων μας ορίζεται μια Gaussian συνάρτηση κατανομής σε κάθε πιθανό γείτονα του ορίζοντας έτσι την γειτονιά αυτού του σημείου. Ο SNE αυτό που προσπαθεί να κάνει είναι να ελαχιστοποιήσει την διαφορά των κατανομών ανάμεσα στην αρχική διάσταση και την μειωμένη. Για κάθε σημείο i και γείτονα του j ορίζεται η P_{ij} η οποία αντιπροσωπεύει την πιθανότητα το j να είναι γείτονας του i .

$P_{i|j} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ik}^2)}$ όπου d_{ij}^2 είναι η ‘ανομοιότητα’ μεταξύ του i και του j η οποία μπορεί να υπολογιστεί και ως $d_{ij}^2 = \|x_i - x_j\|^2 / 2\sigma_t^2$

Στην συνέχεια οι ίδιοι υπολογισμοί γίνονται και για τα Y_i, Y_j στην χαμηλότερη διάσταση βάζοντας βέβαια σε αυτή την περίπτωση $\sigma = 0.5$.

$$q_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_k \exp(-\|y_k - y_i\|^2)}.$$

Τέλος το SNE προσπαθεί να ελαχιστοποιήσει την διαφορά μεταξύ των δύο κατανομών χρησιμοποιώντας ως μέτρο την Kullback-Liebler απόκλιση η οποία ορίζεται ως εξής

$D_{KL}(P \parallel Q) = \sum_i P_i \log\left(\frac{P_i}{Q_i}\right)$ η συνάρτηση κόστος C δίνεται από τον τύπο

$$C = \sum_i \sum_j P_{ij} \log\left(\frac{P_{ij}}{Q_{ij}}\right) \text{ απ' όπου προκύπτει ότι}$$

$\frac{\delta C}{\delta y} = 2 \sum_i (P_{ij} - q_{ij} + P_{ji} - q_{ji})(y_i - y_j)$ από το οποίο όμως εμπίπτουν δύο ζητήματα:

- Κοντινά σημεία χαρακτηρίζονται ως μακρίνα στην νέα διάσταση όταν $(P_{ij}, P_{ji} > q_{ij}, q_{ji})$
- Μακρινά σημεία χαρακτηρίζονται ως γείτονες στην νέα διάσταση όταν $(P_{ij}, P_{ji} < q_{ij}, q_{ji})$

Ακόμη ο πίνακας των πιθανοτήτων P που προκύπτει είναι ασύμμετρος κάτι που έχει ως αποτέλεσμα να δημιουργείται το Crowding problem το οποίο πρακτικά σημαίνει ότι για παράδειγμα σημεία τα οποία φαινομενικά έχουν ίση απόσταση από ένα σημείο στον χώρο μεγάλης διάστασης δεν μπορούν να αντιπροσωπευτούν από σημεία που θα έχουν ίσες αποστάσεις στον δισδιάστατο χώρο. Αυτό το πρόβλημα δεν συναντάται βέβαια μόνο στον SNE αλγόριθμο αλλά και σε άλλες τεχνικές ενσωμάτωσης (πχ Shannon mapping)

Ας περάσουμε όμως και στον tSNE ο οποίος η διαφορά που έχει από τον SNE είναι ότι χρησιμοποιεί την t-students κατανομή του Cauchy για να μετρήσει την ομοιότητα των Y_i και Y_j στην χαμηλή διάσταση και ακόμη για να μπορέσει να αντιμετωπίσει το πρόβλημα με τον ασύμμετρο πίνακα ορίζει ως πιθανότητα $P_{ij} = \frac{(P_{ij} + P_{ji})}{2^n}$.

Τέλος χρησιμοποιώντας την εντροπία $H(x) = -\sum_i (p(x_i) \log_2(p(x_i)))$ υπολογίζει την perplexity $Perp(x) = 2^{H(x)}$ η οποία χοντρικά σημαίνει τον αριθμό των σημαντικών γειτόνων για το κάθε σημείο. Αν το σύνολο είναι πυκνό τότε συνιστάται να χρησιμοποιείται μεγάλη perplexity, αντίθετα αν το σύνολο είναι αραιό τότε συνιστάται η perplexity να κυμαίνεται από 5 έως 50.

Πλεονεκτήματα και Μειονεκτήματα:

Ο tSNE είναι ένας πολύ δυνατός αλγόριθμος για ‘μη γραμμικά’ datasets και δουλεύει καλύτερα από οποιονδήποτε άλλον αλγόριθμο σε τέτοια προβλήματα, παρ’ όλα αυτά τα προβλήματα εγείρονται όταν οι διαστάσεις είναι μεγαλύτερες οι διαστάσεις. Ακόμη

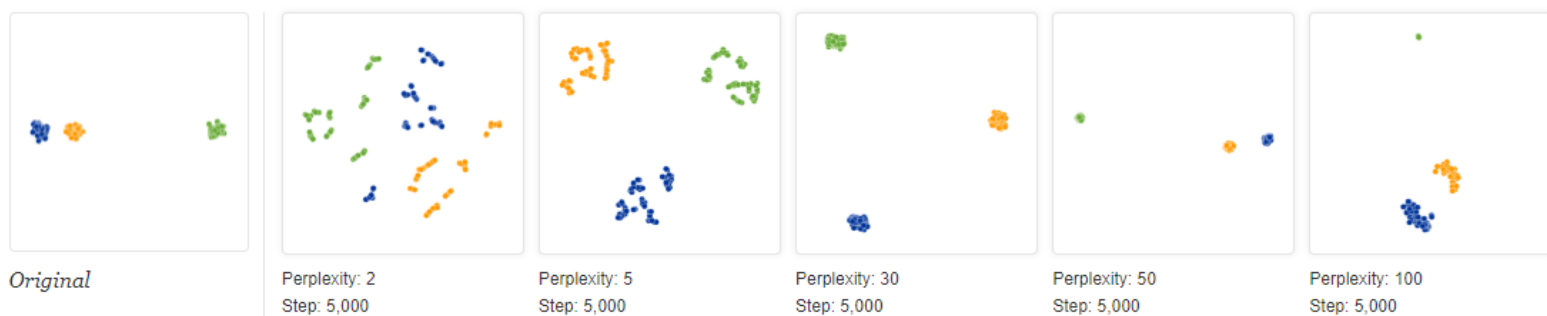
όπως και άλλοι αλγόριθμοι επειδή είναι ευριστικός η λύση που θα προκύψει ενδέχεται να μην είναι η βέλτιστη γιατί το πρόβλημα ελαχιστοποιήσεις της συνάρτησης κόστους είναι μη κυρτό. Τέλος όπως και ο MDS δεν επιδέχεται ενσωμάτωση σε νέα σημεία παρ'όλα αυτά υπάρχουν μερικές τεχνικές που θα εφαρμόσουμε για να μπορέσουμε να αντιμετωπίσουμε αυτό το πρόβλημα.

Η μία η οποία βέβαια δεν συνιστάται είναι να εκπαιδεύσουμε τον αλγόριθμο μας σε όλο το σύνολο δεδομένων (εκπαίδευσης και ελέγχου) και η άλλη την οποία προτείνει και ο δημιουργός του αλγορίθμου Laurens van der Maaten είναι να εκπαιδεύσουμε έναν Multilinear Regressor για να μπορέσουμε να ενσωματώσουμε τα νέα σημεία.

Επίσης να αναφέρουμε ότι όσο αυξάνεται η παράμετρος perplexity τα clusters που σχηματίζονται τείνουν να μαζεύονται πιο κοντά. Η εξήγηση αυτού είναι ότι όσο αυξάνουμε το πλήθος των σημαντικών γειτόνων που θέλουμε να κρατάει κάθε σημείο ο αλγόριθμος χάνει την ιδιότητα να κρατάει απομακρυσμένα τα clusters.

Ακολουθεί ένα παράδειγμα που συλλέξαμε από την σελίδα:

<https://distill.pub/2016/misread-tsne/>



Το σύνολο των δεδομένων μας είναι 50 για κάθε κλάση.

Η διαφορά φαίνεται ξεκάθαρα από το 30 στο 50. Όταν το perplexity είναι 30 κάθε σημείο έχει 30 αποτελεσματικούς γείτονες ενώ όταν το perplexity ορίζεται 50 τότε κάθε σημείο θα πάρει όλα τα σημεία της κλάσης του 'ή' θα επιλέξει στοιχεία από άλλη κλάση ως σημαντικούς γείτονες που σημαίνει ότι θα κρατήσει την ίδια απόσταση στην ενσωμάτωση με αυτήν που είχε στην αρχική διάσταση.

References

- [1] CSC 2535: 2013 Lecture 11 Non-linear dimensionality reduction Geoffrey Hinton
- [2] Laplacian Eigenmaps for Dimensionality Reduction and Data Representation
M.Belkin , P.Niyogi