Υπολογιστική Γεωμετρία
Δεύτερη Εργασία

Σιώρος Βασίλειος - 1115201500144
Ανδρινοπούλου Χριστίνα - 1115201500006

Απρίλιος 2020

1.Implement from scratch the k-NN algorithm in python programming language. Present your method and how you worked, conclude by discussing the disadvantages of the k- NN algorithm, if any.

Thought Process

Algorithm

Implementation

Running the code

Example Usage

2. Using the code provided at the class (or implementing your own) explain what is the curse of dimensionality and how is related to the k-NN algorithm.

Είναι συχνό φαινόμενο, σε επίπεδο εκμάθησης αλγορίθμων μηχανικής μάθησης ή εξόρυξης δεδομένων, να περιορίζεται η μελέτη σε προβλήματα που ορίζονται πάνω σε λίγες διαστάσεις, Συνήθως μελετώνται προβλήματα στις 2 ή στις 3 διαστάσεις και αυτό είναι εύλογο, διότι μπορούμε πολύ εύκολα να αναπαραστήσουμε οπτικά τα αποτελέσματα των αλγορίθμων. Ωστόσο, στην πραγματικότητα τα προβλήματα ορίζονται σε περισσότερες διαστάσεις από 2 ή 3.

Οι διαστάσεις σε ένα πρόβλημα ανάλυσης δεδομένων αντικατοπτρίζουν τα χαρακτηριστικά των αντικειμένων που χρησιμοποιούνται ως βάση για την εκπαίδευση του εκάστοτε αλγορίθμου και ως στόχοι του. Για να γίνει πιο εύκολα κατανοητή η έννοια των χαρακτηριστικών θεωρούμε το εξής παράδειγμα: έστω ότι τα αντικείμενα που επεξεργάζεται ο αλγόριθμος αναπαριστούν τα οχήματα: 'αυτοκίνητο' και 'ποδήλατο' και επιθυμούμε να αποφανθούμε για ένα νέο αντικείμενο αν ανήκει στην κλάση των αυτοκινήτων ή των ποδηλάτων. Τα χαρακτηριστικά που μπορούν να χρησιμοποιηθούν για την περιγραφή των αντικείμενων μπορούν ενδεικτικά να είναι, το πλήθος των τροχών, το μέγεθος, το βάρος κ.α.

Φαινεταί λογικό πως όσα περισσότερα χαρακτηριστικά κατέχει το κάθε αντικείμενο, τόσο πιο εύκολο θα είναι για τον αλγόριθμο να προβλέψει σε ποιά κλάση αντικειμένων ανήκει το προς εξέταση αντικείμενο. Ωστόσο, δεν πρέπει να αποκρυφθεί ότι ένα πολύ σημαντικό ζήτημα στην περίπτωση των πολλών χαρακτηριστικών (ή αλλιώς πολλών διαστάσεων) είναι οι μεγάλες απαιτήσεις σε μνήμη, υπολογιστική ισχύ και χρόνο. Πέρα από την ανάκυψη αυτού του πρακτικού ζητήματος, οι πολλές διαστάσεις οδηγούν και σε ένα πολύ σημαντικό πρόβλημα που καλείται "η κατάρα των πολλών διαστάσεων" ή αλλιώς "η κατάρα της διαστατικότητας" (The Curse of Dimensionality). Ο όρος προέκυψε το 1961 από τον Richard E. Bellman.

Η κατάρα των πολλών διαστάσεων δεν είναι ένα πρόβλημα που μπορεί να οριστεί αυστηρά σε κάθε αλγόριθμο ανάλυσης δεδομένων. Αυτό συμβαίνει διότι εξαρτάται άμεσα από το σύνολο των δεδομένων και τον εκάστοτε αλγόριθμο. Πιο συγκεκριμένα, το ίδιο μοντέλο μπορεί να παρουσιάζει αρκετά καλή συμπεριφορά, δηλαδή ο αλγόριθμος να δίνει καλό Accuracy, για ένα σύνολο δεδομένων όπου η διάστασή του είναι n και για μεγαλύτερη διάσταση από αυτήν το Accuracy να μειώνεται σημαντικά. Να σημειώσουμε εδω ότι το Accuracy χρησιμοποιείται στους αλγορίθμους μηχανικής μάθησης ως μετρική για την αξιολόγηση του εκάστοτε αλγορίθμου. και δίνεται από τον τύπο

$$\text{Accuracy} = \frac{\text{\# correct predictions}}{\text{\# predictions}}.$$

Ουσιαστικά το n είναι ένα threshold για τον συγκεκριμένο αλγόριθμο και τα συγκεκριμένα δεδομένα. Το threshold αυτό δεν είναι ένα σταθερό μέγεθος και μεταβάλεται αναλόγως τον αλγόριθμο και τα δεδομένα.

Η κατάρα των πολλών διαστάσεων σίγουρα αντιβαίνει στη διαίσθηση που έχει ο άνθρωπος. Θα πίστευε κανείς πως όσα περισσότερα χαρακτηριστικά έχει στη διάθεσή του, τόσο πιο εύκολη θα ήταν η κατηγοριοποίηση των αντικειμένων. Κάτι τέτοιο όμως δε συμβαίνει στην πραγματικότητα. Για να εξηγήσουμε το φαινόμενο αυτό αρχικά θα το περιγράψουμε διαισθητικά και απο τη γεωμετρική

σκοπιά των πραγμάτων και στη συνέχεια θα το ορίσουμε με αυστηρά μαθηματικά.

Για τη γεωμετρική προσέγγιση του ζητήματος επιστρατεύουμε ξανά το παράδειγμα με τα οχήματα που προαναφέραμε. Για να αποφανθούμε για την κατηγορία οχήματος για κάποιο αντικείμενο που δε γνωρίζουμε την κατηγορία του μία απλή τεχνική είναι να διαμερίσουμε τον χώρο στον οποίο ζουν τα δεδομένα και το καινούριο αντικείμενο να λάβει label ανάλογα με την περιοχή του χώρου στην οποία εντοπίζεται. Αν δηλαδη βρίσκεται σε περιοχή που υπερισχύει η κλάση των ποδηλάτων, δηλαδή σε αυτήν την περιοχή εντοπίζονται περισσότερα ποδήλατα, τότε θα λάβει το label "ποδήλατο" και αντίστοιχα για οποιαδήποτε άλλη κατηγορία οχήματος.

Αν αποφασίσουμε τα αντικείμενα να περιγράφονται από ένα χαρακτηριστικό, τότε τα σημεία που αναπαριστούν τα οχήματα τοποθετούνται πάνω σε μία ευθεία. Αν διαιρέσουμε την ευθεία σε ίσα τμήματα, παράγονται m το πλήθος τέτοια τμήματα. Αν ακολουθήσουμε την ίδια διαδικασία στις 2 διαστάσεις, δηλαδή αν τα αντικείμενά μας τώρα περιγράφονται από 2 χαρακτηριστικά, παρατηρούμε ότι τα κελιά που δημιουργούνται, τα οποία έχουν τη μορφή τετραγώνων είναι πολλά περισσότερα σε αριθμό, ενώ στις 3 διαστάσεις η ίδια διαδικασία θα παράξει ίσους κύβους ως κελιά και το πλήθος αυτών θα είναι πολύ μεγαλύτερο σε σχέση με το πλήθος των κελιών στις 2 διαστάσεις και στη 1 διάσταση.

πλήθος κελιών στη 1 διάσταση $\ll$ πλήθος κελιών στις 2 διαστάσεις $\ll$ πλήθος κελιών στις 3 διαστάσεις

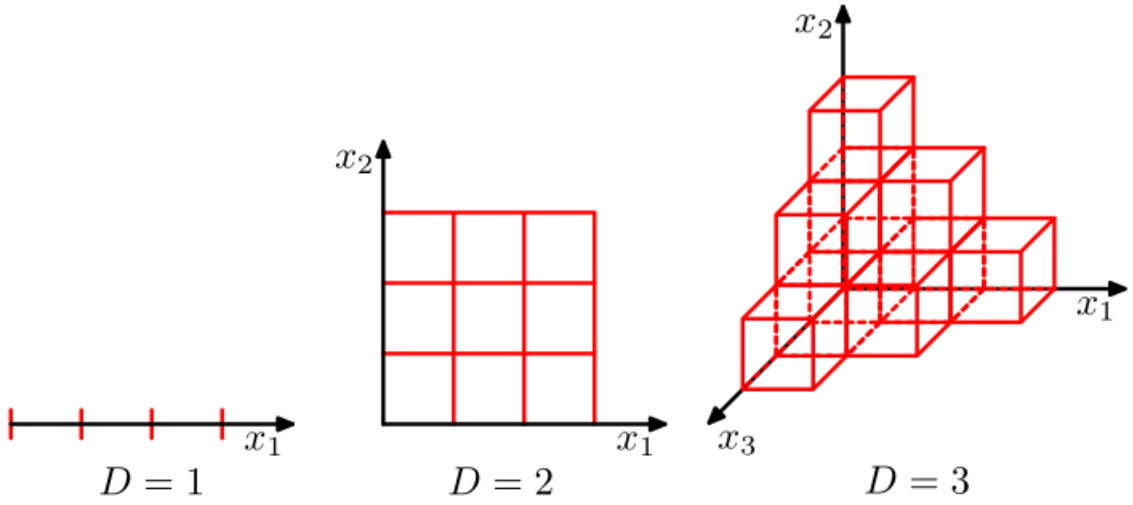

Figure 1: Διάσπαση του χώρου αναπαράστασης των δεδομένων σε ίσα κελιά, για χώρο διάστασης D = 1, D = 2 και D = 3. Το πλήθος των κελιών αυξάνεται σημαντικά όσο αυξάνεται η διάσταση του χώρου. (πηγή: Pattern Recognition and Machine Learning, Christopher M. Bishop)

Να επισημάνουμε στο σημείο αυτό, ότι ο όρος "κελί" εδώ χρησιμοποιείται καταχρηστικά και αναφέρεται είτε σε ευθύγραμμο τμήμα, είτε σε τετράγωνο, είτε σε κύβο αναλόγως με το D.

Μπορεί κανείς εύκολα να συμπεράνει πώς όσο αυξάνεται το D, δηλαδή το πλήθος των χαρακτηριστικών, τόσα περισσότερα κελιά θα μένουν χωρίς σημεία. Καθώς το πλήθος των δεδομένων είναι

σταθερό και καθορισμένο και για τις τρεις διαφορετικές περιπτώσεις D, το ίδιο πλήθος σημείων αντιστοιχεί σε περισσότερα κελιά καθώς το D αυξάνεται. Συνεπώς, όσο αυξάνεται η διάσταση του προβλήματος τόσο τα σημεία διασπείρονται στο χώρο και οι μεταξύ τους αποστάσεις μεγαλώνουν σημαντικά.

Η παρατήρηση αυτή μπορεί να επιβεβαιωθεί και πειραματικά. Επιλέξαμε να τρέξουμε τον κώδικα που αφορά την κατάρα της διαστατικότητας που παρουσιάστηκε στο μάθημα και να τον παραμετροποιήσουμε με τέτοιον τρόπο, ώστε να μπορέσουμε να παράξουμε ασφαλή συμπεράσματα σχετικά με το φαινόμενο. Παραθέτουμε παρακάτω τον κώδικα για λόγους διευκόλυνσης του αναγνώστη.

```
def random_point(dim):
    return [random.random() for _ in range(dim)]

def random_distances(dim, num_pairs):
    return [distance(random_point(dim), random_point(dim))
            for _ in range(num_pairs)]

dimensions = range(1, 10000, 1000)

avg_distances = []
min_distances = []

random.seed(0)

for dim in dimensions:
    distances = random_distances(dim, 10000)  # 10,000 random pairs
    avg_distances.append(mean(distances))     # track the average
    min_distances.append(min(distances))      # track the minimum
    print(f"DIMENSION: {dim} MIN DISTANCE {min(distances)} MEAN DISTANCE:
    {mean(distances)}")

plt.title('The Curse of Dimensionality')

plt.plot(dimensions, avg_distances, label = "avg_distances")
plt.plot(dimensions, min_distances, label = "min_distances")

# naming the x axis
plt.xlabel('dimensions')
# naming the y axis
plt.ylabel('distances')

plt.legend()
plt.show()
```

Τρέξαμε τον κώδικα για διαφορετικό εύρος διαστάσεων και πλήθος σημείων. Ενδεικτικά αποτελέσματα παρουσιάζονται στα διαγράμματα παρακάτω.

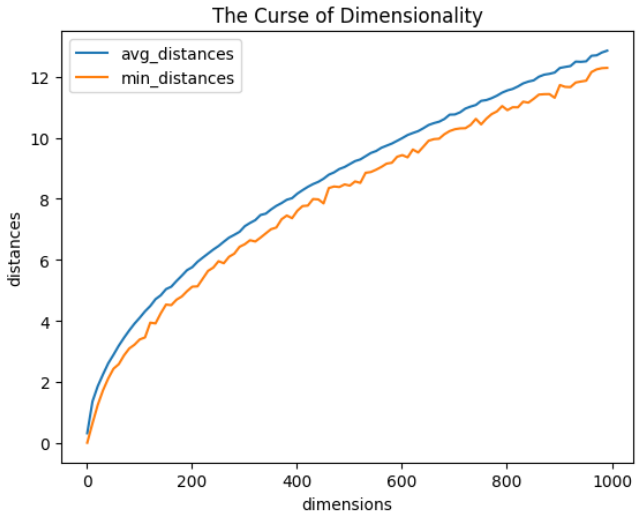

Figure 2: Η κατάρα της διαστατικότητας: αναπαράσταση της μέσης και της ελάχιστης απόστασης δύο σημειών σε χώρους διάστασης από D = 1 εως και D = 1000

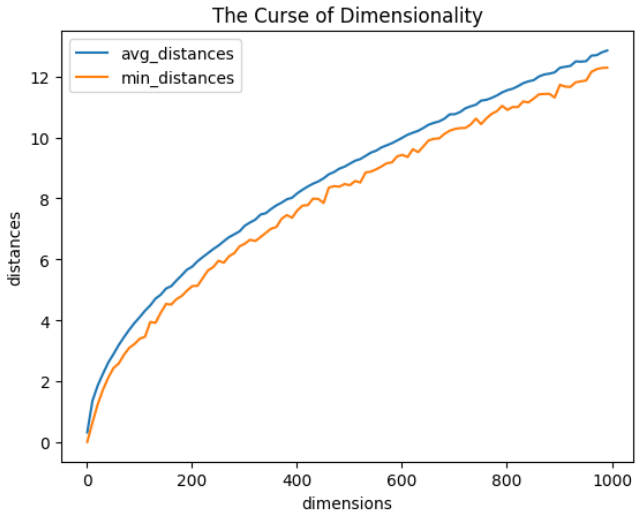

Figure 3: Η κατάρα της διαστατικότητας: αναπαράσταση της μέσης και της ελάχιστης απόστασης δύο σημειών σε χώρους διάστασης από D = 1 εως και D = 1000

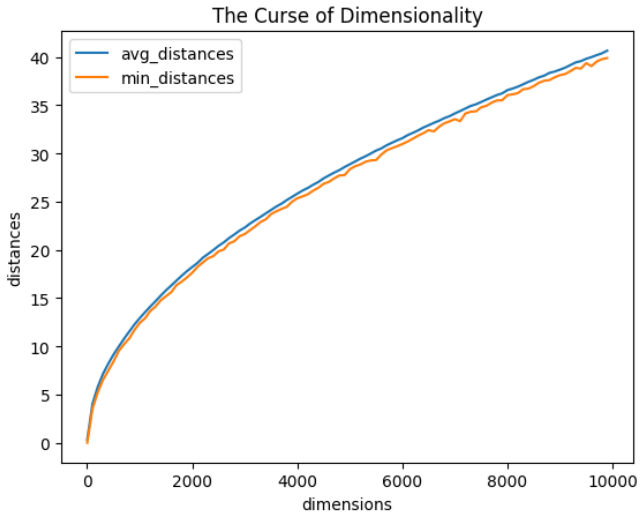

Figure 4: Η κατάρα της διαστατικότητας: αναπαράσταση της μέσης και της ελάχιστης απόστασης δύο σημειών σε χώρους διάστασης από D = 1 εως και D = 10000

Παρατηρούμε ότι όσο αυξάνεται το πλήθος των χαρακτηριστηκών, τόσο αυξάνεται και η απόσταση των σημείων μεταξύ τους και μάλιστα με πολύ γρήγορο ρυθμό, γεγονός που πιστοποιεί όσα ήδη έχουμε αναφέρει για την κατάρα των πολλών διαστάσεων. Η αύξηση της μέσης απόστασης μεταξύ των σημείων τείνει να εκφυλίσσει την εννοια της απόστασης, πλέον το πόσο απέχουν δύο αντικείμενα δεν επαρκεί για να παραχθούν ασφαλή συμπεράσματα. Όσο μεγαλώνει η διάσταση, τόσο αραιώνει ο χώρος απο σημεία που αναπαριστούν δεδομένα και αυτό μπορεί να οδηγήσει στη σκέψη ότι τα σημεία αυτά μπορεί να μην είναι πλέον αντιπροσωπευτικά του προβλήματος και να μην πρόκειται για σημεία που αναπαριστούν τον "κανόνα", αλλά την εξαίρεση, γεγονός που δυσκολεύει την παραγωγή ορθών συμπερασμάτων.

Αφού κατανοήσαμε διαισθητικά και πειραματικά την κατάρας της διαστατικότητας μπορούμε πλέον να περάσουμε στον πιο αυστηρό και με μαθηματικό τρόπο ορισμό του φαινομένου.

Αρχικά, πρέπει να αναφέρουμε ορισμένες σημαντικές μαθηματικές έννοιες, για να μπορέσουμε να ορίσουμε το φαινόμενο με αυστηρό τρόπο. Βασιζόμαστε στην ανάλυση και στον συμβολισμό που παρουσιάζεται στο [1]

Στην περίπτωση του 1-NN, θεωρούμε ότι τα σημεία που αποτελούν το σύνολο εκπαίδευσης ανήκουν στον χώρο X, για τον οποίον ισχύει χωρίς βλάβη της γενικότητας ότι

$$X = [0,1]^d$$

, όπου d είναι η διάσταση του χώρου ($d = 1, 2, ...$). Ο χώρος X είναι εξοπλισμένος με μία μετρική ρ, για την οποία ισχύει

$$\rho : X \times X \rightarrow \mathbb{R}$$

και η οποία ακολουθεί μερικούς κανόνες:

- $\rho(x, x') \geq 0$
- $\rho(x, x') = 0 \Rightarrow x = x'$
- $\rho(x, x') = \rho(x', x)$
- $\rho(x, x') \leq \rho(x, x'') + \rho(x'', x')$

Μία τέτοια μετρική είναι η Ευκλείδια απόσταση:

$$\rho(x, x') = \|x - x'\| = \sqrt{\sum_{i=1}^{d} (x_i - x_i')^2}$$

Υιοθετούμε την ευκλείδια απόσταση ως μετρική σε αυτήν την περίπτωση. Τα labels, δηλαδή οι κατηγορίες που αποδίδονται σε κάθε αντικείμενο ανήκουν στον χώρο Y, για τον οποίο ισχύει χωρίς βλάβη της γενικότητας ότι

$$Y = [0, 1]$$

Θεωρούμε σύνολο S τέτοιο ώστε

$$S = \{(x_1, y_1), ..., (x_m, y_m)\}$$

Με m συμβολίζεται το πλήθος των αντικειμένων του συνόλου εκπαίδευσης S του αλγορίθμου. Έστω D μία κατανομή στο $X \times Y$ και η μία συνάρτηση για την οποία ισχύει ότι

$$\eta : \mathbb{R}^d \rightarrow \mathbb{R}$$

και εκφράζει την πιθανότητα

$$\eta(x) = P[y = 1|x]$$

Η η είναι c-Lipschitz συνάρτηση, δηλαδή ισχύει ότι υπάρχει κάποια σταθερά c, τέτοια ώστε

$$|\eta(x) - \eta x'| \leq c \|x - x'\|$$

Αν με h_s συμβολίζουμε τον κανόνα που έχουμε επιλέξει για την αξιολόγηση των σημείων, με h_s^* τον βέλτιστο κανόνα αξιολόγησης σημείων και με L_D το λάθος ενός κανόνα, είτε αυτός είναι ο h, είτε ο h^*, τότε το αναμενόμενο λάθος στον 1-NN αλγόριθμο φράσσεται ως εξής:

$$E[L_D(h_s)] \leq 2L_D(h^*) + 4c\sqrt{d}m^{-\frac{1}{d+1}}$$

Στην περίπτωση όπου $k \geq 2$, το αντίστοιχο άνω φράγμα γίνεται

$$E[L_D(h_s)] \leq (1 + \sqrt{\frac{8}{k}})L_D(h^*) + (6c\sqrt{d} + k)m^{-\frac{1}{d+1}}$$

Συνεπώς, το σφάλμα που παράγεται από τον αλγόριθμο (ανεξάρτητα από την επιλογή του k) εξαρτάται από τις ιδιότητες της κατανομης και το m, δηλαδή το πλήθος των αντικειμένων εκπαίδευσης του αλγορίθμου. Ιδανικά, το σφάλμα θέλουμε να είναι μικρότερο από μία μικρή ποσότητα ε. Για να συμβεί αυτό πρέπει

$$E[L_D(h_s)] \leq \varepsilon$$

Στην περίπτωση του 1-NN

$$\begin{aligned}
4c\sqrt{d}m^{-\frac{1}{d+1}} &\leq \varepsilon \Leftrightarrow \\
m^{-\frac{1}{d+1}} &\leq \frac{\varepsilon}{4c\sqrt{d}} \Leftrightarrow \\
\frac{1}{m^{\frac{1}{d+1}}} &\leq \frac{\varepsilon}{4c\sqrt{d}} \Leftrightarrow \\
m^{\frac{1}{d+1}} &\geq \frac{4c\sqrt{d}}{\varepsilon} \Leftrightarrow \\
\sqrt[d+1]{m} &\geq \frac{4c\sqrt{d}}{\varepsilon} \Leftrightarrow \\
m &\geq \frac{4c\sqrt{d}}{\varepsilon}^{d+1}
\end{aligned}$$

Στην περίπτωση του k-NN, με $k \geq 2$

$$\begin{aligned}
(6c\sqrt{d}+k)m^{-\frac{1}{d+1}} &\leq \varepsilon \Leftrightarrow \\
m^{-\frac{1}{d+1}} &\leq \frac{\varepsilon}{6c\sqrt{d}+k} \Leftrightarrow \\
\frac{1}{m^{\frac{1}{d+1}}} &\leq \frac{\varepsilon}{6c\sqrt{d}+k} \Leftrightarrow \\
m^{\frac{1}{d+1}} &\geq \frac{6c\sqrt{d}+k}{\varepsilon} \Leftrightarrow \\
\sqrt[d+1]{m} &\geq \frac{6c\sqrt{d}+k}{\varepsilon} \Leftrightarrow \\
m &\geq \frac{6c\sqrt{d}+k}{\varepsilon}^{d+1}
\end{aligned}$$

Συνεπώς, το πλήθος των αντικειμένων που απαρτίζουν το σύνολο εκπαίδευσης του k-NN πρέπει να αυξάνεται εκθετικά με βάση τη διάσταση του χώρου στον οποίον ανήκει το σύνολο των αντικειμένων που εξετάζονται, ώστε ο αλγόριθμος να παράγει ασφαλή αποτελέσματα. Αυτή η εκθετική εξάρτηση του αλγορίθμου από τη διάσταση είναι η κατάρα της διαστατικότητας.

Παραθέτουμε ενδεικτικά στους παρακάτω πίνακες το μέγεθος ενός συνόλου εκπαίδευσης που χρειάζεται ο k-NN, για να παράξει σωστά αποτελέσματα. Το c εδώ είναι 2 και το ε ισούται με 0.8

1-NN	
Διάσταση Χώρου Συνόλου Εκπαίδευσης (d)	Πλήθος Σημείων Συνόλου Εκπαίδευσης (m)
1	100
3	83521
5	113379904
7	208827064576
9	590490000000000

3-NN	
Διάσταση Χώρου Συνόλου Εκπαίδευσης (d)	Πλήθος Σημείων Συνόλου Εκπαίδευσης (m)
1	225
3	331776
5	729000000
7	2251875390625
9	8140406085191601

10-NN	
Διάσταση Χώρου Συνόλου Εκπαίδευσης (d)	Πλήθος Σημείων Συνόλου Εκπαίδευσης (m)
1	576
3	1185921
5	3518743761
7	14048223625216
9	64925062108545024

Ο υπολογισμός των τιμών που παρουσιάζονται στους παρακάτω πίνακες έγινε με βάση το πρόγραμμα που βρίσκεται στο αρχείο number_of_points.py.

Θα μπορούσε κανείς να υποστηρίξει ότι η κατάρα της διαστατικότητας μπορεί να αντιμετωπιστεί με την προσθήκη κάθε φορά όλων των απαραίτητων σημείων, ώστε ο αλγόριθμος να παράγει ασφαλή αποτελέσματα. Ωστόσο, αυτή δεν είναι μία αποδοτική λύση, καθώς το πλήθος των σημείων που θα χρειάζεται κάθε φορά ο αλφόριθμος θα αυξάνεται εκθετικά σύμφωνα με τη διάσταση του χώρου των αντικειμένων. Συνεπώς, μία πιο ασφαλής προσέγγιση του ζητήματος είναι να περιορίσουμε τις διαστάσεις, δηλαδή τα χαρακτηριστικά των αντικειμένων. Δεν είναι πάντοτε όλα τα χαρακτηριστικά σημαντικά για την εξαγωγή συμπερασμάτων στη μηχανική μάθηση. Οι διαστάσεις μπορούν να μειωθούν με διάφορους τρόπους (ενδεικτικά παραδείγματα αποτελούν τα PCA και SVD, που είναι κλασσικές τεχνικές στην μηχανική μάθηση), έτσι ώστε ο αλγόριθμος να μην αντιμετωπίζει πρόβλημα ως προς τη διάσταση και τα δεοδμένα.

3.

1. Suppose there is a set of points on a two-dimensional plane from two different classes. Points in class Red are (0, 1), (2, 3), (4, 4) and points in class Blue are (2, 0), (5, 2), (6, 3). Draw the k-nearest-neighbor decision boundary for k = 1 as we discussed in the lecture. Experiment yourself with two or more differrent distance metrics. Present your results.

2. If the y-coordinate of each point was multiplied by 5, what would happen to the k = 1 boundary? Draw a new picture. Explain whether this effect might cause problems in practice.

3. Can you draw the decision boundary for k=3?

4. Suppose now we have a test point at (1, 2). How would it be classied under 3-NN? Given that you can modify the 3-NN decision boundary by adding points to the training set in the diagram, what is the minimum number of points that you need to add to change the classication at (1, 2)? Provide also the coordinates for these new points and justify your answer.

4. How long does it take for k-NN to classify one point? Or in other words what is the testing complexity for one instance? Assume your data has dimensionality d, you have n training examples and use Euclidean distance. Assume also that you use a quick select implementation which gives you the k smallest elements of a list of length m in O(m).

5. To see an application of the k-NN algorithm in a real world classification prob- lem consider the data found at https://www.kaggle.com/uciml/iris. Download from there the Iris.csv file. Ignore the id column and consider the columns: SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm as point coordinates in a four dimensional space. Consider the column Species as the class/label column. The file contains 150 rows. Run the algorithm on the first 100 rows and make predictions for the rest 50. How your predictions are compared to the actual? Explain your methodology.

References

[1] Understanding Machine Learning: From Theory to Algorithms, Shai Shalev-Shwartz and Shai Ben-David, Cambridge University Press, 2014, pages. 258-267, ISBN: 978-1-107-05713-5

[2] Πολύτροπη Μείωση Διάστασης Δεδομένων με Χρήση Πυρήνα, Διπλωματική εργασία, Πέτρος Χ. Δρακούλης, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Τμήμα ΠΛηροφορικής, 2016

[3] Pattern recognition and machine learning, Christopher Bishop, Springer, 2006, pages. 33-38, ISBN: 978-0-387-31073-2