למידה חישובית וזיהוי תבניות

תרגיל כיתה

מבוא

פונקציית צפיפות ההסתברות הגאוסיאנית Gaussian pdf

בזיהוי תבניות משתמשים בפונקציית צפיפות ההסתברות הגאוסיאנית באופן נרחב. הסיבות הן נוחות מתימטית, וכן משפט הגבול המרכזי הקובע כי פונקציית צפיפות ההסתברות של סכום של מספר משתנים אקראיים שואפת להתפלגות גאוסית כאשר מספר המשתנים שואף לאינסוף. באופן מעשי זה נכון בקירוב גם עבור מספר מספיק גדול של משתנים.

ההתפלגות הגאוסית החד-מימדית מוגדרת על-ידי:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right)$$

כאשר m הוא התוחלת של ההתפלגות ו- σ^2 היא השונות.

ההתפלגות הגאוסית הרב-מימדית מוגדרת על-ידי:

$$p(x) = \frac{1}{(2\pi)^{l/2}|S|^{1/2}} \exp\left(-\frac{1}{2}(x-m)^T S^{-1}(x-m)\right)$$

כאשר m=E(x) הוא וקטור התוחלות, והמטריצה S היא מטריצת הקווריאנס המוגדרת על-ידי:

$$S = E[(x-m)(x-m)^T]$$

וכאשר $|S|$ היא הדטרמיננטה של מטריצת הקווריאנס.

1. חשבו את ערך ה- pdf הגאוסייני N(m,S) עבור :

$$x_1 = \begin{pmatrix} 0.2 \\ 1.3 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 2.2 \\ -1.3 \end{pmatrix}$$

כאשר : $m = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

הדרכה : כיתבו פונקציית Matlab : my_comp_Gauss_dens_value לחישוב הערכים הדרושים.

2. בבעיית סווג עם שתי מחלקות יש לסווג תבנית עבורה וקטור המאפיינים הוא דו-מימדי. ה- data בשתי המחלקות מתפלג גאוסית עם ההתפלגויות

$N(m_1,S_1), \quad N(m_2,S_2)$.

ידוע כי $m_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad m_2 = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \quad S_1 = S_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

כמו-כן ידוע כי $p(\omega_1) = p(\omega_2)$.

א. סווגו את $x_1 = \begin{pmatrix} 1.8 \\ 1.8 \end{pmatrix}$ ל- ω_1 או ל- ω_2.

ב. חזרו על פעולת הסווג כאשר $p(\omega_1) = \frac{1}{6}, \quad p(\omega_2) = \frac{5}{6}$

וכאשר: $p(\omega_1) = \frac{5}{6}, \quad p(\omega_2) = \frac{1}{6}$

3. יצרו data עבור תבניות במרחב דו-מימדי, כאשר כל תבנית מתפלגת בהתאם להתפלגות גאוסית $N(m,S)$ עם $N = 500$, ועם

$$m = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad S = \begin{pmatrix} {\sigma_1}^2 & \sigma_{12} \\ \sigma_{12} & {\sigma_2}^2 \end{pmatrix}$$

עבור המקרים הבאים:

$\sigma_1^2 = \sigma_2^2 = 1, \sigma_{12} = 0$

$\sigma_1^2 = \sigma_2^2 = 0.2, \sigma_{12} = 0$

$\sigma_1^2 = \sigma_2^2 = 2, \sigma_{12} = 0$

$\sigma_1^2 = 0.2, \sigma_2^2 = 2, \sigma_{12} = 0$

$\sigma_1^2 = 2, \sigma_2^2 = 0.2, \sigma_{12} = 0$

$\sigma_1^2 = \sigma_2^2 = 1, \sigma_{12} = 0.5$

$\sigma_1^2 = 0.3, \sigma_2^2 = 2, \sigma_{12} = 0.5$

$\sigma_1^2 = 0.3, \sigma_2^2 = 2, \sigma_{12} = -0.5$

ציירו כל אחד מהמקרים והעירו על צורת הצבירים הנוצרים על-ידי נקודות ה- data.

הדרכה: כדי ליצור את נקודות ה- data עבור הסעיף הראשון השתמשו בפקודות ה- Matlab הבאות:

```
randn('seed',0) %Initialization of the randn function
m=[0 0]';
S=[1 0;0 1];
N=500;
X = mvnrnd(m,S,N)';
```

סווג תבניות באמצעות אלגוריתם כלל KNN.

Nearest neighbor (NN) is one of the most popular classification rules, although it is an old technique. We are given c classes, ω_i, $i = 1,2,\ldots,c$, and a point $x \in \mathcal{R}^l$, and N *training* points, x_i, $i = 1,2,\ldots,N$, in the l-dimensional space, with the corresponding class labels. Given a point, x, whose class label is unknown, the task is to classify x in one of the c classes. The rule consists of the following steps:

1. Among the N training points, search for the k neighbors closest to x using a distance measure (e.g., Euclidean, Mahalanobis). The parameter k is user-defined. Note that it should not be a multiple of c. That is, for two classes k should be an odd number.
2. Out of the k-closest neighbors, identify the number k_i of the points that belong to class ω_i. Obviously, $\sum_{i=1}^{c} k_i = k$.
3. Assign x to class ω_i, for which $k_i > k_j$, $j \neq i$. In other words, x is assigned to the class in which the majority of the k-closest neighbors belong.

For large N (in theory $N \to \infty$), the larger k is the closer the performance of the k-NN classifier to the optimal Bayesian classifier is expected to be [Theo 09, Section 2.6]. However, for small values of N (in theory, for its finite values), a larger k may not result in better performance [Theo 09, Problem 2.34].

A major problem with the k-NN classifier, as well as with its close relative the k-NN density estimator, is the computational complexity associated with searching for the k-nearest neighbors, especially in high-dimensional spaces. This search is repeated every time a new point x is classified, for which a number of suboptimal techniques have been suggested [Theo 09, Section 2.6].

1. Consider a 2-dimensional classification problem where the data vectors stem from two equiprobable classes, ω_1 and ω_2. The classes are modeled by Gaussian distributions with means $m_1 = [0,0]^T$, $m_2 = [1,2]^T$, and respective covariance matrices

$$S_1 = S_2 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$$

Generate two data sets X_1 and X_2 consisting of 1000 and 5000 points, respectively.

2. Taking X_1 as the training set, classify the points in X_2 using the k-NN classifier, with $k = 3$ and adopting the squared Euclidean distance. Compute the classification error.

מסווגים במשפחה זו מוגדרים ישירות על המידע (כלומר סדרת הלימוד), ללא שלב של כיוונון פרמטרים. מסווג נפוץ במשפחה זו הוא "מסווג K השכנים הקרובים" (K Nearest Neighbors, K-NN) אותו נתאר כאן בקצרה.

א. <u>מסווג השכן הקרוב</u>: תהי $\left\{x^{(k)},\omega^{(k)}\right\}_{k=1}^{n}$ סדרת הלימוד, אותה אנו שומרים בזיכרון.

בהינתן קלט חדש x, נמצא את תבנית הקלט $x^{(k)}$ <u>הקרובה ביותר</u> ל- x, ונסווג את x בהתאם לתווית של $x^{(k)}$:

$$f_{NN}(x)=\omega_{k(x)}$$

כאשר

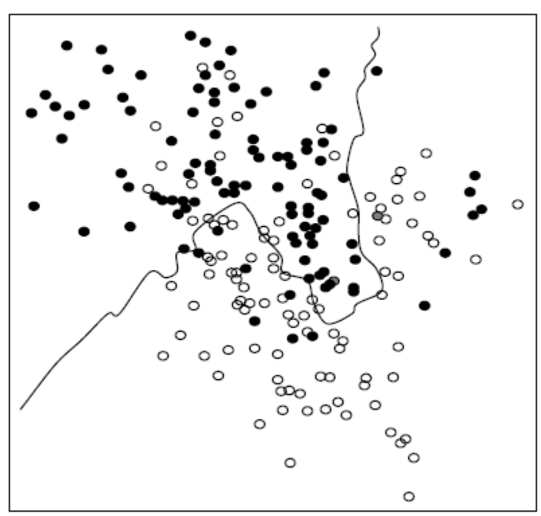
$$k(x)=\underset{k=1,\ldots,n}{\arg\min}\, d(x,x^{(k)})$$

ואילו $d(x,x^{(k)})$ הוא המרחק בין x ל- x_k.

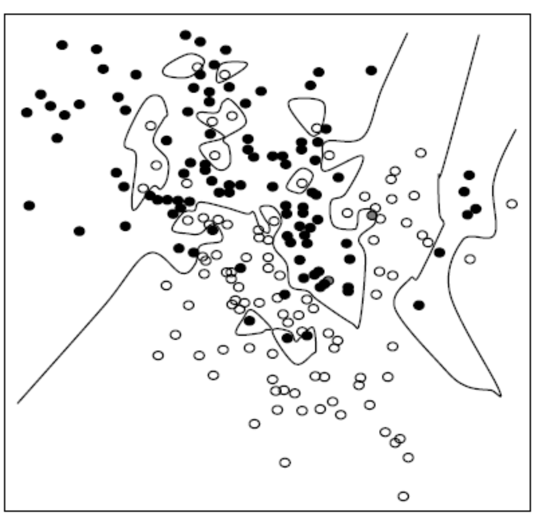
הערה: אלגוריתם זה דורש הגדרת פונקצית-מרחק מתאימה.

פעולת מסווג "השכן הקרוב" מודגמת בציור הבא. מובן כי מאחורי הגדרת מסווג זה עומדת הנחת רציפות כלשהי של הסיווג הנכון על פני מרחב הקלט X.

(מתוך מערכות לומדות תשס"ז, פרופ' רון מאיר, הטכניון)



סיווג לשתי קטגוריות באמצעות מסווג K-NN עם K=15
לפי Hastie et. al. (2001), ציור 2.2



סיווג לשתי קטגוריות באמצעות מסווג K-NN עם K=1
לפי Hastie et. al. (2001), ציור 2.3

הערות:

1. נציין כי פעולת מסווגים אלה מחייבת לשמור בזיכרון את סדרת הלימוד $\left\{x^{(k)}, \omega^{(k)}\right\}_{k=1}^{n}$, ובכל פעם שמתבצע סיווג של קלט חדש יש למצוא את האיבר הקרוב ביותר (או k האיברים הקרובים) מתוך סדרה זו. כאשר מספר הדוגמאות n גדול נדרש זיכרון גדול בהתאם ועומס חישובי ניכר. קיימים מספר אלגוריתמים שמטרתם "לדלל" את סדרת הלימוד המקורית על ידי מחיקת דוגמאות שהשפעתם על המסווג קטנה.

2. מסווגים אלה הינם פשוטים יחסית לתכנון ומימוש, אולם זמן החישוב של המסווג עשוי היות גדול כאשר מספר הדגימות גדול, והביצועים תת-אופטימליים כאשר מספר הדגימות אינו גדול מספיק.

3. הגדרת מסווג זה משתמשת במידת מרחק d על מרחב הקלט X. בדוגמאות לעיל השתמשנו במרחק האוקלידי (ה"טבעי"), אולם בשימושים מסוימים הגדרת מרחק נכונה עשויה להיות מסובכת בהרבה. לדוגמא: מה המרחק בין שתי סדרות באורך N, כאשר ידוע כי 3 איברים מכל סדרה חסרים (במיקום לא ידוע)? כמו כן שימוש במרחק ,לא נכון, יכול להיות מסוכן, למשל, מרחק אוקלידי לזיהוי ספרות.

```
function [z]=knn_classifier(Xtrain,ytrain,X,k)
% knn_classifier implements the k-nearest neighbor classifier for c
classes.
% The classification is based on a reference data set, Xtrain, for
which the class
% labels of its vectors are known.
% Input Arguments:
%   Xtrain:  dxN1  matrix, whose i-th column corresponds to the i-th
reference vector.
%   ytrain:  N1 dimensional vector whose i-th  component contains the
label of the class
%              where the i-th reference vector belongs.
%   X:  dxN matrix whose columns are the data vectors to be classfied.
%   k:  the number of nearest neighbors of the reference set that are
%       taken into account for the classification of a given vector.
% Output argurments:
%   z:  N dimensional vector whose i-th component contains the label
%       of the class where the i-th vector of X is assigned.
%   Usage: [z]=knn_classifier(Xtrain,ytrain,X,k)
```