Л.Н. Гумилев атындағы Евразия Ұлттық Университеті

«Ақпараттық технологиялар» факультеті

«Ақпараттық жүйелер» кафедрасы

**ӨЗДІК ЖҰМЫС**

Пән атауы: Интеллектуалды жүйелердің математикалық негіздері.

Тақырыбы: SVM, TSVM және SVMT әдістері бойынша салыстыру.

Орындаган: Жұмаш Мөлдір

Бақытқали Дина

Адильбекова Сабина

Топ: Ақпараттық жүйелер – 37

Тексерген: Жукабаева Тамара Кокеновна

Астана, 2023

|  |  |  |  |
| --- | --- | --- | --- |
|  | SVM | Tsvm | SvmT |
| Артықылығы | +Бірінші рет қолдану жасау қиын болуы мүмкін. Бірақ SVM мәліметтерінің жұмыс істеуі үшін үйрену көбі жеткілікті және алдағы дерекқорлар мен категориялар арасында қандай бір салыстыру мадағында орналастыру керек.  +Ол, жаңа мәліметтерге көбейтілген жағдайда сапасын жағдайынан арттыруға болады. Осында, оқу алгоритмын шектеулі мәліметтермен қолдануы бар қандай дерекқорларға тез жұмыс жасай алады артықшылықты жоғарылатуға болады.  +сөздіктері, белгішелерді, жаттығушыларды жылдам алып тұруда әділдікке қолдау иесін береді. | +Кез келген метрикті пайдалану: TSVM алгоритмының негізгі артықшылығы, кез келген қалыпты метриктерді пайдалануы мүмкін. Бұл оларды қолданып, бірінші және екінші дереу өрістер арасындағы артықшылықтарды түсіндіруге мүмкіндік береді.  +Бірінші және екінші дерек өрістерді аралау: TSVM, бірінші дерек өрістері мен екінші дерек өрістері арасындағы артықшылықты түсіндіруге мүмкіндік береді. Бұл артықшылықтар айырмашылықтарды түсіну үшін алдымен жаттығуды ашықтай түсіндіретін екі тармақталған аралықтарға айырмашылықтар көрсетеді.  +Мөлшерлік күшті таңдау: TSVM алгоритмының мөлшерлік күшті таңдау мүмкіндігі бар. Осы арқылы, деректердін жаңа деректер жасау  Мүмкін | +Кез келген метрикті пайдалану: TSVM алгоритмының негізгі артықшылығы, кез келген қалыпты метриктерді пайдалануы мүмкін. Бұл оларды қолданып, бірінші және екінші дереу өрістер арасындағы артықшылықтарды түсіндіруге мүмкіндік береді.  +Бірінші және екінші дереу өрістерді аралау: TSVM, бірінші дереу өрістері мен екінші дереу өрістері арасындағы артықшылықты түсіндіруге мүмкіндік береді. Бұл артықшылықтар айырмашылықтарды түсіну үшін алдымен жаттығуды ашықтай түсіндіретін екі тармақталған аралықтарға айырмашылықтар көрсетеді.  +Мөлшерлік күшті таңдау: TSVM алгоритмының мөлшерлік күшті таңдау мүмкіндігі бар. Осы арқылы, деректердін жаңа деректер жасау мүмкін. |
| Кемшілігі | -SVM нашар сапалы жағдайларда жұмыс істеуі мүмкін. Ол, әділ жағдайдан алдын келеді, оларды қолданушының мәліметтеріне сәйкес келетін мәліметтерді табу үшін қолжетімсіздік мүмкін.  SVM, жоғары дерекқорлар санында көбейтілген жағдайды жақсы қолдайтын болуы мүмкін. Бұл, орташа дерекқор санын көпірлілікке тиетін жағдайдарда әділдікті жұмыс істеу жаттығушыны қателестіруі мүмкін.  Сапалы мәліметтер жойылған жағдайды анықтау және оқу алгоритмын жаттығушыларын тексеру арқылы басқа жағдайды қамтамасыз ете алмауы мүмкін. | -Тарау аудармауы керек: TSVM алгоритмын қолдану аллдында, кемшіліктерді жаңадан тарау аудару керек.  -Мәліметтер жеткіліксіз болса, кемшіліктерді дұрыст жүзеге асыру мүмкін емес: Деректердің аяқталмағаны, салыстырулар тарату мәселесінің кең желілігін жоюы мүмкін.  -Тарату алгоритмы таңдауы керек: Әрбір деректердің тіл тарату алгоритмын таңдау кезінде кейінгі уақытта сізге көмек көрсететін техникалық жағдайларды түсіндіруге мүмкін болады.  Сізге деректердін жаттығу жолында таңдау және кемшіліктері туралы мәліметтерді көрсету үшін TSVM алгоритмын қолдану кезінде өз жұмысыңызда жасау кезінде егжей-тегжей сөз айту үшін дайын боламын. | TSVM алгоритмын қолдану алғысы кезінде, кемшіліктерді жаңадан тарау аудару керек.  Мәліметтер жеткіліксіз болса, кемшіліктерді дұрыст жүзеге асыру мүмкін емес: Деректердің аяқталмағаны, салыстырулар тарату мәселесінің кең желілігін жоюы мүмкін.  Тарату алгоритмы таңдауы керек: Әрбір деректердің тіл тарату алгоритмын таңдау кезінде кейінгі уақытта сізге көмек көрсететін техникалық жағдайларды түсіндіруге мүмкін болады.  Сізге деректердін жаттығу жолында таңдау және кемшіліктері туралы мәліметтерді көрсету үшін TSVM алгоритмын қолдану кезінде өз жұмысыңызда жасау кезінде егжей-тегжей сөз айту үшін дайын боламын. |

## **Модельдерді енгізу**

Тапсырманы орындау үшін Python бағдарламалау тілін қолданамыз. Және де төменгі модульдерді қолданамыз. Олар:

1. pandas (pd):

* Кодта пайдалану: CSV файлынан деректерді жүктеу және оны деректер жақтауы ретінде көрсету үшін қолданылады.

1. sklearn.model\_selection:

* Кодта пайдалану: train\_test\_split функцияны қолдана отырып, деректерді оқу және тест үлгілеріне бөлу үшін қолданылады`.

1. sklearn.svm (SVC):

* Кодта пайдалану: SVC(kernel='linear') көмегімен SVM классификатор қосымшасын құру үшін қолданылады.

1. skmultilearn.problem\_transform (BinaryRelevance):

* Кодта қолдану: "BinaryRelevance" классификаторын құру үшін қолданылады. BinaryRelevance(classifier=SVC(kernel='linear'))

1. sklearn.metrics (accuracy\_score):

* Кодта пайдалану: accuracy\_score(y\_true, y\_pred) функциясы арқылы дәл нәтиже алу үшін қолданылады

1. scipy.sparse (csr\_matrix):

* Кодта қолдану: кескіндерді екілік байланыста пайдалану үшін сирек матрицаларға түрлендіру үшін қолданылады.

# 

# **ҚОРЫТЫНДЫ:**

Деректерді талдау және қант диабетін жіктеу тапсырмасы үшін әртүрлі Машиналық оқыту әдістерін (SVM, TSVM, SVMT) қолдану барысында біз келесі нәтижелерге қол жеткіздік:

1. SVM моделі 0.1875 дәлдігін көрсетті. Алайда, бұл модельдің оқу жылдамдығы өте жылдам болды-0.0260.

2. TSVM моделі 0.1407 дәлдігін көрсетті. Tsvm оқу жылдамдығы 0.1425 болды, бұл да жылдам нәтиже.

3. SVMT моделі 0.1823 дәлдігін көрсетті, оқу жылдамдығы басқаларына қарағанда айтарлықтай төмен болды, 23.077.

Айта кету керек, барлық модельдердің дәлдігі салыстырмалы түрде төмен болды. Бұл деректердің әртүрлілігінің жеткіліксіздігіне және жіктеу тапсырмасының күрделілігіне байланысты болуы мүмкін, әсіресе белгілер ұқсас мәндерге ие болса.

Нәтижелерді одан әрі жақсарту үшін келесілер ұсынылады:

1. Деректерді өңдеу: өткізіп алған мәндермен жұмыс істеуді және шығарындыларды жоюды қоса алғанда, деректерді алдын ала өңдеуді Мұқият жүргізіңіз.

2. Машиналық оқытудың басқа әдістерін қолданып көріңіз: Берілген тапсырмаға жақсырақ сәйкес келетін балама алгоритмдер мен модельдерді іздеңіз.

3. Белгілерді зерттеу: белгілерді талдау және деректерде жіктеуді жақсарту үшін қолдануға болатын үлгілердің бар-жоғын анықтау.

4. Параметрлерді таңдау: жақсы өнімділікке қол жеткізу үшін модель параметрлерін реттеңіз.

Тұтастай алғанда, деректерді талдау және модельдерді Оқыту нәтижелері Берілген тапсырмадағы жіктеу дәлдігін жақсарту үшін қосымша күш салу қажеттілігін көрсетеді. Бұл қосымша деректерді дайындауды, тереңірек зерттеуді және машиналық оқытудың қолайлы әдістерін таңдауды қажет етуі мүмкін.

**ҚОСЫМША**

SVM

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

import time

data = pd.read\_csv("C:/Users/Lenova/Desktop/archive/diabetes.csv")

X = data.drop("Pregnancies", axis=1)

y = data["Pregnancies"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

clf = SVC(kernel='linear')

start\_time = time.time()

clf.fit(X\_train\_scaled, y\_train)

end\_time = time.time()

y\_pred = clf.predict(X\_test\_scaled)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Точность SVM:", accuracy)

print("Время обучения модели:", end\_time - start\_time, "секунд"

TSVM

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from tslearn.svm import TimeSeriesSVC

from sklearn.metrics import accuracy\_score

import time

data = pd.read\_csv("C:/Users/Lenova/Desktop/archive/diabetes.csv")

X = data.drop(columns=["Pregnancies"])

y = data["Pregnancies"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

clf = TimeSeriesSVC(kernel="rbf")

start\_time = time.time()

clf.fit(X\_train, y\_train)

end\_time = time.time()

y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Точность TSVM:", accuracy)

print("Время обучения модели:", end\_time - start\_time, "секунд")

SVMT

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from skmultilearn.problem\_transform import BinaryRelevance

from sklearn.metrics import accuracy\_score

from scipy.sparse import csr\_matrix

import time

data = pd.read\_csv("C:/Users/Lenova/Desktop/archive/diabetes.csv")

X = data.drop(columns=["Pregnancies"])

y = data["Pregnancies"]

y\_sparse = csr\_matrix(y.values.reshape(-1, 1))

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_sparse, test\_size=0.25, random\_state=42)

base\_classifier = SVC(kernel='linear')

classifier = BinaryRelevance(classifier=base\_classifier)

classifier.fit(X\_train, y\_train)

y\_pred\_sparse = classifier.predict(X\_test)

y\_pred = y\_pred\_sparse.toarray().reshape(-1)

accuracy = accuracy\_score(y\_test.toarray().reshape(-1), y\_pred)

start\_time = time.time()

classifier.fit(X\_train, y\_train)

end\_time = time.time()

training\_time = end\_time - start\_time

print("Время обучения модели:", training\_time, "секунд")

print("Точность SVMT:", accuracy)