Дәріс 12: Кластерлеу

Оқытушы: Сарсембаева Т.С.

Дәрістің мақсаты

Қадағаланбайтын оқыту (Unsupervised Learning) тұжырымдамасымен және оның қадағаланатын оқытудан айырмашылығымен таныстыру. Осыдан кейін, кластерлеудің негізгі идеясын – деректерді ішкі құрылымына қарай топтарға бөлуді түсіндіру қарастырылады.

Дәріс центроидқа негізделген ең танымал әдіс – K-Means алгоритмін, оның жұмыс істеу қадамдарын, артықшылықтары мен кемшіліктерін талдаумен жалғасады. Сонымен қатар, тығыздыққа негізделген DBSCAN әдісін зерттеу, оның K санын қажет етпейтінін, шуды анықтай алатынын және күрделі пішіндерді табатынын көрсету көзделген.

Оған қоса, K-Means үшін K кластерлерінің оңтайлы санын таңдау әдістерін меңгерту мақсат етіледі. Соңында, Иерархиялық кластерлеу және оның нәтижесін дендрограмма арқылы визуализациялау қарастырылады.

Негізгі сұрақтар

1. Қадағаланбайтын оқыту дегеніміз не және оның қадағаланатын оқытудан айырмашылығы неде?
2. Кластерлеудің негізгі принципі қандай?
3. K-Means алгоритмінің негізгі қадамдары қандай?
4. DBSCAN K-Means-тен несімен ерекшеленеді және оған қандай параметрлер қажет?
5. K-Means үшін K-ның оңтайлы санын таңдауға арналған "Шынтақ" әдісі (Elbow Method) қалай жұмыс істейді?
6. Силуэт коэффициенті нені өлшейді және оның мәні -1-ге жақын болса, бұл нені білдіреді?
7. Иерархиялық кластерлеудің "тегіс" кластерлеуден (K-Means) айырмашылығы неде?

Қысқаша мазмұны (тезистер)

Біз көбінесе "белгіленген" деректері (X және Y) бар қадағаланатын оқытуды (Supervised Learning) қарастырдық. Көп жағдайда бізде "дұрыс жауаптар" (y) болмайды, тек деректер жиынтығы (X) болады. Бұл **қадағаланбайтын оқыту (Unsupervised Learning)** деп аталады.

Кластерлеу – бұл қадағаланбайтын оқытудың ең танымал міндеттерінің бірі. Негізгі идеясы – деректер жиынтығын бір-біріне ұқсас нысандардан тұратын топтарға (кластерлерге) бөлу.

K-Means (К-Орташалар) әдісі - бұл ең танымал, центроидқа (кластердің геометриялық орталығы) негізделген алгоритм. Мақсаты – деректерді алдын-ала берілген K саны бойынша K кластерге бөлу.

DBSCAN (Тығыздыққа негізделген кластерлеу) - бұл тығыздық ұғымына негізделген. Идеясы, кластер – бұл бір-біріне жақын (тығыз) нүктелер, ал сирек нүктелер – шу. K санын талап етпейді, кез-келген пішіндегі кластерлерді таба алады және шуды анықтайды. Екі параметрді қажет етеді: eps (көршілес радиусы) және min_pts (негізгі нүкте болу үшін қажетті минималды нүктелер саны).

Оңтайлы K санын таңдау

"Шынтақ" әдісі (Elbow Method) - WCSS (Within-Cluster Sum of Squares) метрикасына негізделген. WCSS – бұл кластер ішіндегі нүктелерден центроидқа дейінгі қашықтықтардың қосындысы.

Силуэт коэффициенті

Әр нүктенің өз кластерінде қаншалықты "жақсы" орналасқанын өлшейді. Ол a(i) (нүктенің өз кластерімен орташа қашықтығы) және b(i) (нүктенің ең жақын басқа кластермен орташа қашықтығы) мәндерін салыстырады.

Иерархиялық кластерлеу

K-Means немесе DBSCAN сияқты "тегіс" емес, кластерлердің "ағаш тәрізді" құрылымын (иерархиясын) құрады. Агломеративті ("төменнен жоғары") әдіс жиі қолданылады, әрбір нүкте жеке кластер ретінде басталып, ең жақын екеуі бір үлкен кластерге біріктіріледі.

Нәтижесі **Дендрограмма** арқылы визуализацияланады. K-ны алдын ала таңдаудың орнына, дендрограмманы белгілі бір биіктікте "кесуге" болады.

Кластерлеу идеясы

Осыға дейінгі дәрістерде біз көбінесе **қадағаланатын оқыту (supervised learning)** әдістерін қарастырдық. Бұл әдістерде бізде "белгіленген" (labeled) деректер болды: біз модельге кіріс деректерді (X) ғана емес, сонымен қатар оларға сәйкес "дұрыс жауаптарды" (y) да бердік (мысалы, "бұл сурет – мысық", "бұл email – спам"). Модельдің мақсаты X пен y арасындағы байланысты үйрену болды.

Бірақ көп жағдайда бізде "дұрыс жауаптар" болмайды. Бізде тек деректер жиынтығы (X) болады. Бұл **қадағаланбайтын оқыту (unsupervised learning)** деп аталады. Мұндай жағдайда біздің мақсатымыз – модельдің осы деректердің ішкі құрылымын, жасырын заңдылықтарын немесе топтарын өздігінен табуы.

Кластерлеу (Clustering) – бұл қадағаланбайтын оқытудың ең танымал және іргелі міндеттерінің бірі.

Кластерлеудің негізгі идеясы – деректер жиынтығын бір-біріне ұқсас нысандардан (нүктелерден) тұратын топтарға, яғни кластерлерге бөлу.

> Негізгі принцип
>
> - Бір кластердің ішіндегі нысандар бір-біріне максималды ұқсас болуы керек.
> - Әртүрлі кластерлердегі нысандар бір-біріне максималды ұқсас емес (алыс) болуы керек.

Ұқсастық көбінесе қашықтық метрикасы арқылы өлшенеді. Ең жиі қолданылатын метрика – **Евклидтік қашықтық**:

Екі $p = (p_1 , p_2)$ және $q = (q_1 , q_2)$ нүктелерінің арақашықтығы: $d (p , q) = \sqrt{((q_1 - p_1)^2 + (q_2 - p_2)^2)}$

Бүгін біз кластерлеудің екі негізгі және бір-бірінен түбегейлі өзгеше әдісін қарастырамыз: **K-Means және DBSCAN**.

Кластерлеу модельдері

K-Means алгоритмі

K-Means – кластерлеу алгоритмдерінің ішіндегі ең танымал және қарапайымдарының бірі. Бұл центроидқа (centroid) негізделген алгоритм. "Центроид" – бұл кластердің геометриялық орталығын білдіретін нүкте.

Алгоритмнің негізгі идеясы – алдын-ала берілген K саны бойынша деректерді K кластерге бөлу. Ол әрбір нүктені сол нүктеге ең жақын орналасқан центроидқа жатқызу арқылы жұмыс істейді.

K-Means алгоритмінің қадамдары:

1 **K санын таңдау**

Алгоритмді іске қоспас бұрын, біз деректерді қанша кластерге бөлгіміз келетінін (K санын) өзіміз шешуіміз керек.

2 **Центроидтарды инициализациялау**

Деректер кеңістігінен кездейсоқ K нүктені таңдап, оларды бастапқы центроидтар ретінде белгілеу.

3 **Нүктелерді тағайындау (Assignment Step)**

Деректер жиынындағы әрбір нүктенің барлық K центроидқа дейінгі қашықтығын есептеп, нүктені өзіне ең жақын центроидқа тиесілі кластерге тағайындау.

4 **Центроидтарды жаңарту (Update Step)**

Барлық нүктелер тағайындалып болғаннан кейін, әрбір K кластер үшін жаңа центроидты есептеу. Жаңа центроид – сол кластерге тиесілі барлық нүктелердің орташа арифметикалық мәні.

5 **Қайталау**

3-ші және 4-ші қадамдарды центроидтардың орны тұрақтанғанға дейін немесе нүктелердің кластерге тиесілілігі өзгермейтін болғанға дейін қайталау.

Артықшылықтары:

- Өте жылдам және есептеу тұрғысынан тиімді.
- Түсінуге және іске асыруға оңай.
- Үлкен деректер жиынында жақсы жұмыс істейді.

Кемшіліктері:

- K санын алдын-ала білу керек – ең үлкен мәселе.
- Бастапқы инициализацияға сезімтал.
- "Сфералық" кластерлерді болжайды.
- "Шуға" (Outliers) сезімтал.

DBSCAN алгоритмі

K-Means кластерлерді орталық нүкте (центроид) арқылы анықтаса, **DBSCAN** мүлдем басқаша жұмыс істейді. Оның негізінде **тығыздық (density)** ұғымы жатыр. DBSCAN-ның негізгі идеясы: "Кластер – бұл бір-біріне жақын орналасқан (тығыз) нүктелердің жиынтығы". Тығыз аймақтардың арасындағы сирек нүктелер шу (noise) немесе шығарындылар (outliers) ретінде қарастырылады.

Бұл алгоритм K-Means-тен екі нәрсесімен ерекшеленеді:

- Ол кластерлердің санын (K) алдын-ала талап етпейді. Ол кез-келген пішіндегі (мысалы, "жарты ай" немесе "шеңбер") кластерлерді таба алады және шуды өте жақсы анықтайды.

DBSCAN параметрлері:

eps (Epsilon, ϵ)

Бір нүктенің айналасындағы "көршілес" аймақтың радиусы.

min_pts

Бір нүктені "негізгі" нүкте деп санау үшін оның ϵ-радиусында болуы керек нүктелердің минималды саны (нүктенің өзін қоса алғанда).

Осы параметрлерге сүйене отырып, DBSCAN барлық нүктелерді үш түрге бөледі:

- **Негізгі нүкте (Core Point)**: Егер нүктенің ϵ-радиусында кем дегенде min_pts санындай нүкте болса. Бұл – кластердің "ішкі" нүктесі.
- **Шекаралық нүкте (Border Point)**: Егер нүктенің ϵ-радиусында min_pts санынан аз нүкте болса, бірақ ол кем дегенде бір негізгі нүктенің ϵ-радиусында жатса. Бұл – кластердің "шеткі" нүктесі.
- **Шу нүктесі (Noise Point/Outlier)**: Негізгі де, шекаралық та емес нүкте. Ол ешбір кластерге жатпайды.

DBSCAN алгоритмінің қадамдары:

01

Кездейсоқ, әлі қаралмаған нүктені таңдау.

02

Оның ϵ-көршілерін табу: Егер ол негізгі нүкте болса, жаңа кластер басталады және барлық тығыздық арқылы қол жетімді нүктелер қосылады.

03

Егер ол шекаралық немесе шу нүктесі болса, уақытша "Шу" деп белгіленеді.

04

Барлық нүктелер қаралғанға дейін қайталау.

Практикалық мысал: IT-конференция нетворкинг сессиялары

Елестетіңіз, сіз үлкен IT-конференциясын ұйымдастырып жатырсыз және 1000 қатысушыға арналған нетворкинг (танысу) сессияларын жасауыңыз керек. Мақсат – ұқсас қызығушылықтары немесе дағдылары бар адамдарды бір топқа жинау.

1. Деректерді жинау:

Әрбір қатысушыдан сауалнама алынды:

- q1: Бағдарламалау тәжірибесі (0-20 жыл)
- q2: Негізгі технологиясы (Python, Java, C++, JS, ...)
- q3: Қызығушылық саласы (AI, Web, Mobile, DevOps, ...)

2. Деректерді дайындау:

"Python", "AI" сияқты категориялық деректерді сандық түрге (мысалы, One-Hot Encoding арқылы) айналдыру керек. Нәтижесінде әрбір қатысушы көп өлшемді векторға айналады (мыс: [5, 1, 0, 0, 1, 0, 0] - 5 жыл тәжірибе, Python, AI).

K-Means қолдансақ:

- Біз алдымен нетворкинг үшін қанша үстел (топ) қажет екенін шешуіміз керек (мысалы, K = 10).
- K-Means 1000 қатысушыны 10 топқа бөледі.
- Нәтиже: Бір топ "5-7 жыл тәжірибесі бар Java/Web әзірлеушілері", екінші топ "0-2 жыл тәжірибесі бар Python/AI студенттері" болуы мүмкін.
- Мәселе: K-Means барлық 1000 адамды міндетті түрде 10 топтың біріне тағайындайды.

DBSCAN қолдансақ:

- Біз K санын бермейміз. Оның орнына eps және min_pts параметрлерін береміз.
- Нәтиже: DBSCAN бір-біріне өте ұқсас 30 адамнан тұратын "Senior DevOps мамандары" тобын, 15 адамнан тұратын "Mobile (Flutter) әзірлеушілері" тобын табады.
- Ең маңыздысы: DBSCAN сонымен қатар 50 адамды "Шу" деп таңбалауы мүмкін. Бұл адамдардың профильдері ешбір тығыз топқа сәйкес келмеген.
- Бұл "ерекше" мамандарға бөлек сессия ұйымдастыруға болады.

Бұл мысал **DBSCAN-ның икемділігін және K-Means-тің қарапайымдылығын** көрсетеді.

Оңтайлы кластер санын таңдау (K-Means үшін)

Жоғарыда айтқанымыздай, K-Means-тің басты кемшілігі – K санын алдын-ала білу қажеттілігі. Егер біз K = 3 деп таңдасақ, бірақ деректерде 5 табиғи топ болса, нәтиже нашар болады. K санын таңдауға көмектесетін екі танымал әдіс бар.

"Шынтақ" әдісі (Elbow Method)

Бұл әдіс кластерлеудің сапасын өлшейтін **WCSS (Within-Cluster Sum of Squares)** метрикасына негізделген.

WCSS – бұл әрбір кластердегі нүктелерден сол кластердің центроидына дейінгі қашықтықтардың квадраттарының қосындысы.

$$WCSS = \sum_{i=1}^{K} \sum_{x \in C_i} \| x - centroid(C_i) \|^2$$

Егер кластерлер "тығыз" (жақсы) болса, WCSS мәні төмен болады. Егер кластерлер "шашыраңқы" болса, WCSS жоғары болады.

Әдістің қадамдары:

1 **K-Means** алгоритмін қайталау

K-Means алгоритмін K-ның әртүрлі мәндері үшін қайталап іске қосу.

2 **WCSS** мәнін есептеу

Әрбір K үшін WCSS мәнін есептеу.

3 Графикті салу

K (X осі) және WCSS (Y осі) арасындағы графикті салу. Графикте K өскен сайын WCSS әрқашан төмендейтінін көреміз.

4 "Шынтақ" нүктесін табу

Біздің іздейтініміз – WCSS-тің төмендеу жылдамдығы кенеттен баяулайтын нүкте. Бұл нүкте графикте "шынтаққа" (elbow) ұқсайды. Осы "шынтақ" нүктесі K-ның оңтайлы мәні болып саналады.

Силуэт коэффициенті **(Silhouette Score)**

Бұл – кластерлеу сапасын бағалаудың неғұрлым сенімді әдісі. Ол әрбір нүктенің өз кластерінде қаншалықты "жақсы" орналасқанын өлшейді.

Әрбір i нүктесі үшін:

- **a(i)**: Осы нүктеден өзінің кластеріндегі басқа барлық нүктелерге дейінгі орташа қашықтық. Бұл i нүктесінің өз кластерінде қаншалықты тығыз орналасқанын көрсетеді (төмен болса жақсы).
- **b(i)**: Осы нүктеден оған ең жақын басқа кластердегі барлық нүктелерге дейінгі орташа қашықтық. Бұл ең жақын "бәсекелес" кластердің қаншалықты алыс екенін көрсетеді (жоғары болса жақсы).

i нүктесінің силуэт коэффициенті s(i):

$$s(i) = (b(i) - a(i)) / \max(a(i), b(i))$$

s(i) ≈ 1-ге жақын

b(i) ≫ a(i). Нүкте өз кластерінде өте тығыз және басқа кластерлерден өте алыс орналасқан. Бұл – өте жақсы кластерлеу.

s(i) ≈ 0-ге жақын

b(i) ≈ a(i). Нүкте екі кластердің шекарасында орналасқан.

s(i) ≈ -1-ге жақын

b(i) < a(i). Нүкте басқа кластерге өзінің кластеріне қарағанда жақынырақ орналасқан. Бұл нүкте дұрыс тағайындалмаған.

Әдістің қадамдары:

01

K-Means алгоритмін K-ның әртүрлі мәндері үшін іске қосу.

02

Әрбір K үшін барлық нүктелердің орташа силуэт коэффициентін есептеу.

03

Ең жоғары орташа силуэт коэффициентін берген K мәні ең оңтайлы болып саналады.

Иерархиялық кластерлеу

Біз қарастырған K-Means және DBSCAN **"тегіс" (flat)** кластерлеуді орындайды, яғни олар деректерді бір деңгейлі топтарға бөледі.

Иерархиялық кластерлеу басқаша тәсілді ұсынады. Ол кластерлердің "ағаш тәрізді" құрылымын (иерархиясын) құрады. Бұл әсіресе таксономия (биологиядағы түрлерді жіктеу) сияқты салаларда пайдалы.

Иерархиялық кластерлеудің екі негізгі түрі

Аггломеративті **(Agglomerative)**

"Төменнен жоғары" (bottom-up) әдісі. Әрбір нүкте жеке кластер ретінде басталады (N нүкте = N кластер). Содан соң әр қадамда бір-біріне ең жақын екі кластер табылып, бір үлкен кластерге біріктіріледі. Соңында барлық нүктелер бір ғана үлкен кластерге біріккенше жалғасады.

Дивизивті **(Divisive)**

"Жоғарыдан төмен" (top-down) әдісі. Барлық нүктелер бір ғана үлкен кластерде болады. Процесс барысында әр қадамда ең үлкен (немесе ең әртекті) кластер екі кішірек кластерге бөлінеді. Соңында әрбір нүкте жеке кластер болғанша жалғасады.

Аггломеративті әдіс жиі қолданылады. Оның жұмыс істеуі үшін "кластерлер арасындағы қашықтықты" қалай өлшеу керектігін анықтайтын **біріктіру критерийін (linkage criterion)** таңдау керек:

Single-Linkage

Екі кластер арасындағы қашықтық – олардың ең жақын екі нүктесінің арақашықтығы.

Complete-Linkage

Екі кластер арасындағы қашықтық – олардың ең алыс екі нүктесінің арақашықтығы.

Average-Linkage

Екі кластер арасындағы қашықтық – барлық мүмкін жұптар арасындағы орташа қашықтық.

Дендрограмма **(Dendrogram)**

Иерархиялық кластерлеудің нәтижесін визуализациялаудың ең жақсы жолы – **дендрограмма** салу.

- Дендрограмма – бұл кластерлердің қалай біріккенін (немесе бөлінгенін) көрсететін ағаш тәрізді диаграмма.
- Тік ось (Y) – кластерлердің бірігу қашықтығын көрсетеді. Ұзын тік сызықтар – бір-бірінен алыс орналасқан кластерлердің біріккенін білдіреді.
- **Кластер санын таңдау**: K-Means-тен айырмашылығы, мұнда K-ны алдын ала таңдаудың қажеті жоқ. Біз дендрограмманы "кесе" аламыз. Диаграмма арқылы көлденең сызық жүргізу арқылы біз сол сызық қиып өтетін кластер санын аламыз. Бұл бізге K=2, K=3, K=4 және т.б. үшін нәтижелерді бірден көруге мүмкіндік береді.

Дәрісті меңгеруге арналған бақылау сұрақтары:

1. Қадағаланатын (Supervised) және қадағаланбайтын (Unsupervised) оқытудың негізгі айырмашылығы неде? Мысал келтіріңіз.

2. K-Means алгоритмінің 4 негізгі қадамын атаңыз (Инициализация, Тағайындау, Жаңарту, Қайталау).

3. DBSCAN алгоритмінің K-Means-тен екі негізгі артықшылығы қандай?

4. DBSCAN-да "Негізгі нүкте" (Core Point) мен "Шекаралық нүкте" (Border Point) арасында қандай айырмашылық бар?

5. K-Means үшін K санын таңдауға арналған "Шынтақ" әдісі (Elbow Method) қалай жұмыс істейді? Графикте біз нені іздейміз?

6. Силуэт коэффициенті 0-ге жақын болса, бұл нені білдіреді?

7. Иерархиялық кластерлеуде "Дендрограмма" не үшін қолданылады?

Әдебиеттер тізімі:

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer. https://hastie.su.domains/ElemStatLearn/

Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media. https://github.com/ageron/handson-ml3

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96). https://www.dbs.ifi.lmu.de/Publikationen/Papers/KDD-96.final.frame.pdf

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Scikit-learn Development Team. (2025). User Guide: 2.3. Clustering. Scikit-learn. https://scikit-learn.org/stable/modules/clustering.html