

Лекция 4. Расс-Лассо регрессиясы: олардың дұрыс қолдануы

1. Лекция мақсаты мен міндеттері: OLS сызықтық регрессия туралы түсінік беру. Расс-Лассо регрессиясы: оларды қалай дұрыс ұстау керектігін айқындау. Бақыланатын машиналық оқыту: Регрессия Vs классификациясына анықтама беру.

2. Лекция мазмұны:

1. Расс-Лассо регрессиясы.
2. Жоталардың шығыс функциясы.
3. Lasso шығындар функциясы

3. Оқытудың техникалық құралдары: Zoom онлайн оқыту платформасы

4. Лекция оқудың тәртібі, оқыту әдістері мен түрлері: баяндау, сұрақ – жауап, түсіндіру, кіріспе лекция

Расс-Лассо регрессиясы: оларды қалай дұрыс ұстау керек?

Жақында мен қазіргі кездегі деректанушы ғалымның жоталар мен лассо регрессиясының арасындағы айырмашылықты еске түсіру үшін күресетін мақаласын оқыдым. Бұл лауазымның мақсаты - деректер ғылымымен жаңадан келген және соған ұқсас күресіп жүрген адамдарға көмектесу, жақсы идеялар тізбегі мен лассо-регрессияны алуға көмектеседі. Бұл лауазымға қарапайым квадраттарды (OLS) және сызықты алгебраны білу ұсынылады

Өзіңіз білетін нәрседен бастаңыз

Жоталар мен лассо регрессиясы қарапайым ең кіші квадраттардың сызықтық регрессиясының «модификациясы» болып табылады. OLS бірнеше сызықтық регрессияда біздің шығындар функциясы:

$$\frac{1}{n} \| \mathbf{y} - \mathbf{X}\beta \|^2 .$$

Егер жоғарыда айтылғандар сізге түсініксіз болса, онда сізде сызықтық алгебра мен векторлық белгілерде білім болмағандықтан болуы мүмкін. Бұл OLS құны функциясымен таныс болған кезде жақсы болуы мүмкін, егер сіз, ең болмағанда, төмендегі түйсікті ұстануыңыз мүмкін. Біздің $\mathbf{y}$ мәндеріміз $\mathbf{y}$ векторында, $\mathbf{x}$ -мәндері $\mathbf{X}$ матрицасында, ал сызықтық регрессия коэффициенттеріміз - векторда β .

Егер сіз осы лауазымға өту барысында әлі де шатасып жатсаңыз, бірнеше сызықтық регрессияны мұқият түсініп, осы лауазымға оралмас бұрын негізгі сызықтық алгебраны құруды ұсынамын.

Білгенге сұрақ

OLS сызықтық регрессия - бұл пайдалы әдіс, бірақ бұл әрқашан идеалды әдіс емес. Мысалы, егер бізде сатушылар бар болса ше? Немесе біздің көп модельділік модельге қалай әсер етуі мүмкін? Осындай сұрақтар қоюда OLS әрдайым регрессия үшін қолайлы әдіс емес екені белгілі болады.

Сызықтық регрессияны көп полиноларлықты көрсететін қасиеттерде қолданғымыз келеді делік. Мультиколлинарлық біздің дисперсиямыздың жоғарылауына әкелуі мүмкін, бұл біздің модельге шамадан тыс сәйкес келуіне әкеледі. Біздің модельдеріміз жаңа деректерге жалпыланғанын қалаймыз, сондықтан мұндай жарамсыздықтың алдын-алу әдісі пайдалы.

Регуляризация - бұл біздің моделімізде жоғары қарама-қайшылықты болдырмауға арналған әдіс. Ол үшін жоғарыда көрсетілген шығын функциясына айыппұл мерзімін енгіземіз. Осы жаза мерзімін қосу арқылы біз сызықтық функцияның коэффициенттерінің тым үлкен болуына жол бермейміз. Бұл жағымсыздықты енгізу дисперсияны азайтып, сәйкестікке жол бермеу керек.

Жоталардың шығыс функциясы

Жоталардың шығыны функциясы L2 нормасын ℓ_2 -ге айыппұл мерзімі ретінде қосады, бұл біз «барлық коэффициенттерді квадрат қосамыз» деп айтудың қиял тәсілі. Бұл өте дұрыс емес, өйткені L2 norm нормасы біздің модельге қаншалықты әсер ететінін бақылауды қалаймыз, сондықтан біз баптау параметріне көбейтеміз,.

$$\frac{1}{n} \| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \|^2 + \lambda \| \boldsymbol{\beta} \|^2$$

Егер λ едәуір үлкен болса, квадраттық қателіктер жиынтығы біздің шығындар функциясында аз болады. Жоғарыда аталған функцияны азайтуға тырысқанымыз үшін, бұл күш 0-ден 0-ге дейін жетеді. Сонымен, λ көбейген сайын values мәні азайғанын көреміз. Бұл біз қалаған мінез-құлық, өйткені ол дисперсияны азайтады. Неліктен бұлай екенін білу үшін дисперсияны кіші коэффициенттермен есептеу туралы ойлана аласыз.

Lasso шығындар функциясы

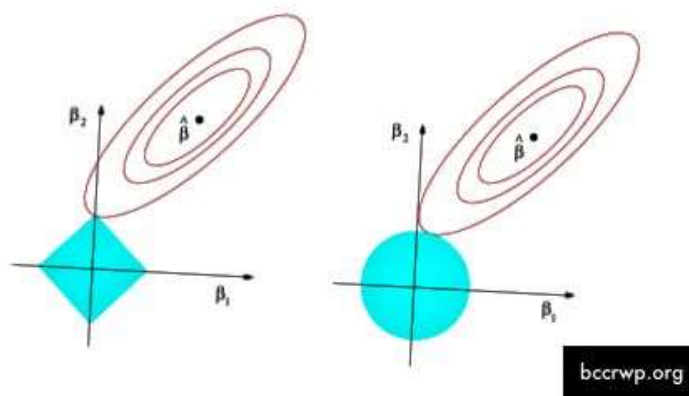
Лассо құны функциясы, екінші жағынан, L1 norm нормасын өзінің жаза мерзімі ретінде қолданады. L1 нормасы β - бұл біздің коэффициенттеріміздің абсолютті шамаларының қосындысы.

$$\frac{1}{n} \| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \|^2 + \lambda \| \boldsymbol{\beta} \|$$

Сол сияқты, significantly едәуір үлкен болған сайын 0 0-ге мәжбүр болады. Бірақ лассо құны функциясы туралы айта кететін маңызды нәрсе, біз коэффициенттеріміздің квадратының орнына абсолютті мәнді қолданатындықтан, маңызды емес коэффициенттер аз болуы мүмкін. бұл 0 мәнін тағайындады. Бұл әсіресе мүмкіндік таңдау үшін пайдалы.

Айыппұл мерзімі коэффициенттердің нөлге тең болуының себебі мен шектелетін аймаққа қатысты болуы мүмкін, бірақ мен мұнда кірмеймін, бірақ мен оқуды ұсынамын, өйткені бұл L1 айыппұлына мүмкіндік беретін идеяны бекіту үшін пайдалы болады. кейбір коэффициенттер нөлге тең болады.

Lasso шектеу аймағы көк (сол жақта), жотасы (оң жақта)



Бірге қою

Жоталар мен лассо регрессиясының не істейтінін түсінгеннен кейін, қайсысы регуляризация әдісін қолданатынын еске түсіру оңай болады.

Жеке өзім сандық жолда сандарды нөлге дейін түсіретін елестеткенді ұнатамын. Бұл маған лассо регрессиясы модельдің кейбір коэффициенттерін нөлге әкелуі мүмкін екенін есіме салады. L1 нормасы мұны мүмкін ететін шектеуші аймақты құрайтындықтан, Лассо L1 нормасын өзінің жаза мерзімі ретінде қолданатынын білеміз. Егер сіз Лассо мен жотаның L1 және L2 нормаларын айыппұл шарты ретінде қолданатындығын есіңізде сақтасаңыз, онда жотаның L2 нормасы қолданылады.

Бұл біреуіне осы екі регрессиялық әдісті түзу ұстау қиынға соғатын адамға көмектесе алады деп үміттенемін.

Бақыланатын машиналық оқыту: Регрессия Vs классификациясы

Бақыланатын машиналық оқыту: Регрессия Vs классификациясы

Бұл мақалада мен басқарылатын машинаны оқыту алгоритмдерінің жіктеу мен жіктеу арасындағы негізгі айырмашылықтарды түсіндіремін. Тиісті машинаны оқыту алгоритмін таңдағанға дейін айырмашылықтарды түсіну керек.

Хабарламадан бас тартуды оқып шығыңыз.



Мен қысқаша 7 негізгі бағытты сипаттаймын:

1. Регрессия мен жіктеу арасындағы айырмашылық
2. Жалпы регрессия және жіктеу алгоритмдерінің атаулары
3. Алогритмаңыздың жақсылығын тексеру
4. Үстеме бағаны түсіндіру
5. Асып кетудің алдын алу әдістері
6. Реттеудің қысқаша мазмұны
7. Градиенттің түсуі туралы айтып беріңіз

1. Регрессия мен жіктеу арасындағы негізгі айырмашылықтар қандай?

Екеуі де:

- Жетекшілік ететін оқыту алгоритмдері
- Тарихи деректерді болжау және шешім қабылдау үшін пайдаланыңыз
- Фитингтің ең жақсы сызығына назар аударыңыз

Жетекшілік етілетін оқу алгоритмдері мәліметтерді белгілеуді талап етеді. Бақыланатын машинамен оқыту туралы қосымша ақпарат алу үшін менің мақалама қараңыз: 8 минут ішінде машиналық оқыту

Регрессия:

Регрессия сіздің деректер нүктелеріңіздің үздіксіз мәндерге ие болуын талап етеді. Алдымен факторлар (тәуелсіз айнымалылар) табылған. Содан кейін нақты және болжанатын шамалар арасындағы айырмашылықты азайтуға мүмкіндік беретін тәуелсіз айнымалыларға коэффициенттер (көбейткіштер) есептеледі. Соңында формула есептеледі. Формула тәуелсіз айнымалылардан (сіз өлшегіңіз келетін) тәуелді айнымалы мәнді болжау үшін қолданылады (сіздің мақсатыңыз қандай болатынына байланысты). Болжалды шамалар үздіксіз. Регрессия сізге үнемі нәтиже береді.

Жіктеу:

Жіктеу сіздің деректер нүктелеріңізде дискретті мәндердің болуын талап етеді, мысалы категориялары. Алғашқы тарихи деректер санаттарға (класстарға) бөлінген. Содан кейін жаңа енгізілген деректер тарихи деректер негізінде жіктеледі және соңында шешімдер қабылданады. Болжалды мәндер дискретті болып табылады. Жіктеу дискретті мәндер мен қатаң санаттарға арналған мәліметтер жиынтығын шығарады.

Қолданады

- Егер нәтижелер сіздің деректер жинағыңыздағы деректер нүктелерінің сыныбын белгілі бір нақты санаттарға көрсеткіңіз келсе, жіктеуді қолданыңыз, мысалы, егер сіз есімнің ер немесе әйел екенін білгіңіз келсе.
- Егер сіз жеке тармақтарды, мысалы, ылғалдылық пен ластанудың арақатынасын жақсы ажыратқыңыз келсе, регрессияны қолданыңыз.

2. Жалпы регрессия және жіктеу алгоритмдері

3 белгілі алгоритм.

Регрессия: сызықтық регрессия, регрессия орманы, регрессиялық нейрондық желілер.

Жіктеу: К жақын көрші, логистикалық регрессия, тірек векторлық машиналар

Алгоритмдердің егжей-тегжейлі салыстыруы мына жерде келтірілген: Машиналарды оқыту алгоритмдерін салыстыру

3. Менің регрессия немесе жіктеу үлгісі қаншалықты жақсы?

Үлгіңіздің дұрыстығын тексеру үшін әр түрлі шаралар бар:

Реттелген R-квадраты (Регрессия): Теңдеудегі еркіндік дәрежесі үшін жазалағаннан кейін нақты және болжамды шамалар арасындағы айырмашылықты есептейді. Мен бұл мақалада қалай есептелетінін түсіндірдім: Менің болжамды моделім қаншалықты жақсы - регрессиялық талдау

F1 (Жіктеу): F1 көрсеткіші - модельдің жұмысының өлшемі. Бұл модельдің дәлдігі мен еске түсірілуінің орташа мәні. Нәтижелер 1-ден 0-ге дейін. 1-ге ұмтылған нәтижелер ең жақсы деп саналады, ал 0-ге бағытталғандар ең нашар деп саналады. F1 классификациялық сынақтарда қолданылады, онда шынайы теріс мәндер онша маңызды емес.

Шатастыру матрицасы (жіктеу): қарапайым түрде, шатастыру матрицасы нақты нақты мәндер белгілі болған кезде жіктеу алгоритмінің нәтижелерін жинақтайтын нәтижелер кестесі. Бірнеше термин қолданылады:

- Шын позитивті: нақты нәтиже шын болғанда және болжамды мән де дұрыс болса
- Шын теріс: нақты нәтиже жалған болса және болжамды мән де жалған болса
- Жалған позитив: нақты нәтиже жалған болса, бірақ болжамды мәні шын болса
- Жалған теріс: нақты нәтиже шын болса, бірақ болжамды мәні жалған болса

4. Асып кету дегеніміз не?

Асып кету - модельдің экспрессивтілігі тым жоғары болған кезде. Асып кету дегеніміз - модель жаттығу мәліметтеріне сәйкес келетін жағдай, бірақ модельді тестілік деректерге қарсы тексергенде, ол жаман болады. Егер сіз өзіңіздің модельіңізді жаттығу деректеріне үйретіп жатсаңыз және ол оның ережелері мен заңдылықтарын жаттығу деректерінің айналасында қалыптастыратын болса, ол көрінбейтін мәліметтер бойынша жалпылай алмайды. Бұл деректердің шуылына (кездейсоқтыққа) байланысты. Соның салдарынан модель бұрын болмаған сценарийлерді болжай алмайды. Бұл модель бастапқы деректерді оқытуда стохастикалық мінез-құлықты ескере отырып аяқталады және жақсы жинақтала алмайды. Бұл артық шығындар деп аталады.

Асып кету - бұл жалпылау кезінде модель нашар болған кезде. Асып кету - бұл машиналарды оқыту алгоритмдерінің жиі кездесетін мәселесі. Бұл жаттығулар туралы мәліметтер шу болғандықтан және модель өзінің алгоритміне шу енгізгендіктен болады.

Әрі қарай түсіндіру, болжау моделін дайындау үшін жаттығу және тест мәліметтерін жинау керек. Егер сіздің жаттығу деректеріңізде кездейсоқтық болса, онда сіз шығаратын модель нақты мәндер деп болжайды, ол болжанған мәндерді нақты мәндерге мүмкіндігінше жақын болатын

теңдеулерді құрастырады. Алайда, сынақ деректері көбірек енгізілгеннен кейін, модельдің болжамдылығы сәтсіз болады. Ол дәл емес жалпылауды береді, өйткені ол шуды өзімен бірге алып жүреді.

Екінші жағынан, жарамсыздық асып кетудің керісінше. Егер модель жарамсыз болса, онда ол мәліметтерді жете түсінбейді және мәндерді болжай алмайды.

5. Артық салмауды болдырмаңыз

Шамадан тыс кияудің бірнеше әдісі бар:

1. Сіздің жаттығуларыңыздың көлемін және тестілеу деректерін көбейтіңіз.
2. Өз модельіңіздің параметрлерін, еркіндік дәрежелерін және параметрлерін азайтыңыз. Бұл сіздің моделіңіздің қарапайым болуын қамтамасыз етеді және жаттығу деректеріндегі шуды (стохастикалық мінез-құлықты) азайтуға мүмкіндік береді.
3. Айқас тексеру әдісін қолданыңыз. Ол модельдің жалпылау қатесінің орташа мәнін алдыңғы орташа деңгеймен салыстырады. Крестті тексеру әдісі k-қатпарларды қамтиды.
4. Модель параметрлерін шамадан тыс сәйкестендіруге әкелетін болса, айыппұл салыңыз. Бұл процесс жүйелеу деп аталады.

6. Реттеу дегеніміз не?

Асып кетуді азайтудың бір тәсілі - бұл жүйелеу. Үлгіге айыппұл салу үшін модельге қосымша шарттар енгізілуі мүмкін. LASSO (L1) және Ridge (L2) - белгілі жүйелеу әдісі. L1 және L2 - коэффициенттердің мөлшеріне / квадратына сәйкес келетін екі жоғалту функциясы.

- L1 есептелген және нақты мәндер арасындағы абсолютті айырмашылықтардың қосындысын азайтады.
- L2 есептелген және нақты мәндер арасындағы квадраттық айырмашылықты азайтады.

L1 берік, бірақ L2 тұрақты деп саналады.

7. Градиент түсу дегеніміз не?

Градиентті түсіру - оңтайландыру алгоритмі. Ол функцияның қателіктерін азайтуға мүмкіндік береді. Градиентті түсіру машинаны оқыту алгоритмдерінің барлығында қолданылады. Машина үйрену алгоритмі мәліметтерді болжаған кезде, алгоритмнің қаншалықты жақсы екенін бағалау үшін оның шығындар функциясын таба аламыз. Шығындар функциясы машинаны оқыту алгоритміндегі болжамды қателерді бақылайды. Машиналарды оқыту алгоритмінің болжамды қуатын оның параметрлерін өзгерту арқылы жақсартуға болады. Шығын функциясы максималды дәлдікте максималды болатындығын білдіретін шығын функциясы ең төменгі нүктеге жеткенге дейін біз параметрлерді жедел көбейтуге болады. Бұл процесс градиентті түсіру деп аталады.

Алгоритмнің бірнеше өзгерісі бар, оның ішінде стохастикалық градиент түседі. Стохастикалық градиентті түсіру (SGD) нейрондық желілерді оқытуда қолданылады.

5. Деңгейлік тапсырмалар:

1-деңгей.

- Жоталардың шығыс функциясы дегеніміз не?.
- Lasso шығыс функциясы қалай беріледі?

2-деңгей.

- Жалпы регрессия және жіктеу алгоритмдердің берілуі?
- Асып кету дегеніміз не?

3-деңгей.

- Реттеу дегеніміз не?
- Градиентті түсу дегеніміз не?

6. БОӨЖ тапсырмалары:

Ауыстыру және дисперсия (bias and variance). Орта гипотеза ұғымы.

7. БӨЖ тапсырмалары:

ADALINE, персептрон Розенблатта, Хебб ережесі.

8. Пайдаланылған әдебиеттер:

1. Ғ. Айғараева, Қ. Асанова, М. Нысанов. Сандық қондырғылар және микропроцессорлық жүйелер. Оқулық. 2010
2. Д. С. Козен. Есептеу теориясы. Сандық. Оқулық Алматы. 2013.
3. Асамбаев А.Ж. Жасанды интеллект негіздері.- Алматы: ЖШС РПБК "Дәуір", 2011-135б. Оқу құралы.
4. Рассел С., Норвиг П. Жасанды интеллект: Жаңашыл әдіс -Алматы, 2014.. Оқу құралы.
5. К. Пол. Цифровое искусство. Учебник. Москва , 2017, 220
6. Андреас Мюллер, Сара Гвидо. Введение в машинное обучение с помощью Python. –Москва. 2016.
7. В.Н.Негода. Машинно-ориентированное программирование. Учебное пособие Ульяновск 2015.