Лекция 3. Подготовка на данни

Съществуващите реални бази от данни съдържат зашумени, непълни и противоречиви данни, обикновено поради своят огромен размер, често стигащ до няколко гигабайта или още повече. *Непълнотата* в данни може да възникне по няколко причини. Някои от интересуващи нас атрибути могат да липсват защото просто не са били записани по време на създаване на база данни, някои стойности могат да липсват поради повреди в оборудването, използвано за тяхното събиране и т.н.

Причините за наличието на зашумени данни (некоректни стойности на атрибути) могат да бъдат повреди в събиращото данни оборудване, човешки грешки при въвеждането на данни или грешки при тяхното получаване чрез комуникационни канали (например, поради ограничен размер на буфера). Всички тези "замърсени" данни могат да объркат впоследствие ИЗД алгоритъм, водейки до ненадеждни и неправдоподобни резултати. По тази причина *"изчистване" на данни* е една важна стъпка от процеса на предварителната подготовка на данни преди от тях да бъдат направен опит за извличане на скритите закономерности.

Често данните, използвани за анализа, постъпват от различни източници (бази данни). По тази причина едни и същи по смисъл атрибути могат да имат различни имена. Същото важи и за имена на отделни атрибутни стойности. Това води до възникване на противоречия и излишества в анализираните данни. Освен това, някои от използваните атрибути могат да се окажат нерелевантни за текущата цел или излишни, тъй като могат да бъдат изведени от други атрибути. Наличието на излишни данни може значително да забави процеса на ИЗД или да доведе до намирането на погрешни зависимости. По тази причина процесът на подготовката на данни освен "изчистването" включва и процедурите по *тяхно интегриране*.

Различни ИЗД алгоритми, използвани за моделиране, имат свои специфични изисквания към типа, качеството и обема на използвани данни. Така, например, алгоритмите от типа на най-близкия съсед или невронни мрежи работят по-добре с нормализирани данни, наивен Бейсов класификатор работи с номинални атрибути и т.н. Това изисква предварително използване на различни алгоритми за *трансформация на данни*, както и за *намаляване на техния обем*.

И така, методите за предварителната подготовка на данни подобряват качество на реални данни, помагайки по този начин за увеличаване на точността и ефективност на последващия процес на построяване на модели на данни. Етапът на предварителната подготовка на данни е много важна стъпка в общия процес на ИЗД, тъй като качеството на извлечените закономерности се базира на качеството на използваните при анализа данни. Намирането и отстраняването на аномалии, както и намаляване на обема на данни, използвани при анализа, може да доведе до значителното подобрение на получаваните при моделиране резултати.

3.1. "Изчистване" на данни

Повечето от реални данни са непълни, зашумени и противоречиви. Процедурите за изчистване на данни опитват да попълват липсващите стойности, да изглаждат шума чрез идентифициране на екстремните стойности и да премахват противоречията в данни. В този раздел ще се запознаете с базовите методи за изчистване на данни.

3.1.1. Липсващите стойности

Следващите методи за обработка на липсващите атрибутни стойности могат да се приемат като базови:

1. *Игнориране на записа.* Този метод обикновено се прилага, когато в записа липсва стойност на целевия атрибут (като се предполага, че задачата на ИЗД е класификацията) и не е много ефективен, освен в случаи, когато записът съдържа няколко атрибута с липсващите стойности. Методът води до лоши резултати в случаи, когато процентът на липсващите атрибутни стойности е значителен.
2. *Ръчно попълване на липсващите стойности.* В общия случай този подход изисква много време и практически не е приложим за големи бази данни с голям процент на липсващите атрибутни стойности.
3. *Използване на една глобална константа за заместване на липсващата стойност.* Всички липсващите стойности се заменят с една и съща константа, например "?" (т.е. "неизвестно"). Някои версии на ИЗД алгоритми (например, методи за класификация, използващи MVDM метрика са сходство) успешно построяват модели върху така "изчистени" бази данни.
4. *Запълване на липсващата стойност със стойността на средно аритметично за даден атрибут.* При този метод неизвестните стойности на *непрекъснат атрибут* се заменят със средно аритметично на всички известни му стойности, а неизвестните стойности на *номинален атрибут* – с най-често срещана известна му стойност. Това е един доста разпространен метод за "изчистване" на данни, използван, например, при задачи на класификация, решавани чрез невронни мрежи.
5. *Запълване на липсващата стойност със стойността на средно аритметично за даден атрибут с отчитане на класа.* Този подход се използва при наличие на целевия атрибут. Неизвестната стойност на някой атрибут за всички записи от един и същ клас се попълва със средно аритметично на атрибута за същия клас, ако атрибутът е непрекъснат, или с най-често срещаната му стойност в класа, ако атрибутът е номинален.
6. *Запълване на липсващата стойност с най-вероятното значение на съответния атрибут.* При този подход атрибутът с липсващите стойности се разглежда като целевия и задачата за попълване на тези стойности се третира като задача за класификация, ако атрибутът е номинален, или като регресия, ако той е непрекъснат. За тяхното решение могат да бъдат използвани такива методи като

Бейсов класификатор, алгоритмите за най-близък съсед, класификационни или регресионни дървета и др.

Всички методи от 3 до 6 променят разпределение на данни и не отчитат възможни връзки между различни атрибути. От тази гледна точка последният метод използва най-много налична информация за предсказване на липсващите стойности, като запазва връзки между атрибутите. Обаче той е доста по-сложен за изпълнение, тъй като изисква прилагане (и наличие) на различни класификационни или регресионни техники за моделиране.

3.1.2. "Зашумени" и противоречиви данни

Шумът е случайната грешка или разсейване в стойността на измерваната величина. За непрекъснатите атрибути като измерител на разсейване (и, по този начин, показател за наличие на шума в данни) може да се използва коефициент на вариация ($V_{\%}$), който изразява стандартното отклонение като процент от средна аритметична:

$$V_{\%}(X) = \frac{\sigma}{\tilde{x}} 100 \quad (3.1)$$

където $\tilde{x}$ и σ са съответно средно аритметично и стандартно отклонение на атрибута X.

Наличието на сравнително голямо разсейване (над 2%) може да бъде причинено, например, от груби грешките при измерването или от наличието на сгрешени екстремални стойности. В първия случай "шумът" може да бъде намален чрез прилагане на методи за "изглаждане" (smoothing) на данни, докато във втория – чрез идентификация и премахване на съществуващите крайности (outliers).

Един от методите за "изглаждане" на данни се състои в тяхното "интервализиране" (binning). При този метод подредените в нарастващ ред стойности на анализирания непрекъснат атрибут се разбиват на определени интервали (bins) и намиращи се във всеки интервал данни локално си изглаждат чрез консултиране със съседните в интервала данни. А именно, при *изглаждане чрез средните стойности на интервала* всяка стойност в интервала се заменя със средно аритметично на този интервал. Аналогично, при *изглаждане чрез медианни стойности на интервала* всяка стойност в интервала се заменя със стойността на медиана на този интервал. В метода за *изглаждане чрез интервални граници* всяка стойност в интервала се заменя с най-близка до нея долна или горна граница на интервала.

Самото разбиването на интервали може да бъде *с еднаква дълбочина* (т.е. всеки интервал да съдържа еднакъв брой на атрибутни стойности) или *с еднаква ширина* (т.е. еднаква разлика между горната и долната граница на интервала).

Пример 3.1. Нека сортираните стойности са 4, 8, 15, 21, 21, 24, 25, 28, 34.
Разбиване на интервали с еднаква дълбочина:

Интервал 1: 4, 8, 15
Интервал 2: 21, 21, 24
Интервал 3: 25, 28, 34

Изглаждане чрез средните стойности:

Интервал 1: 9, 9, 9
Интервал 2: 21, 21, 21
Интервал 3: 29, 29, 29

Изглаждане чрез медиани:

Интервал 1: 8, 8, 8
Интервал 2: 21, 21, 21
Интервал 3: 28, 28, 28

Изглаждане чрез граници на интервала:

Интервал 1: 4, 4, 15
Интервал 2: 21, 21, 24
Интервал 3: 25, 25, 34

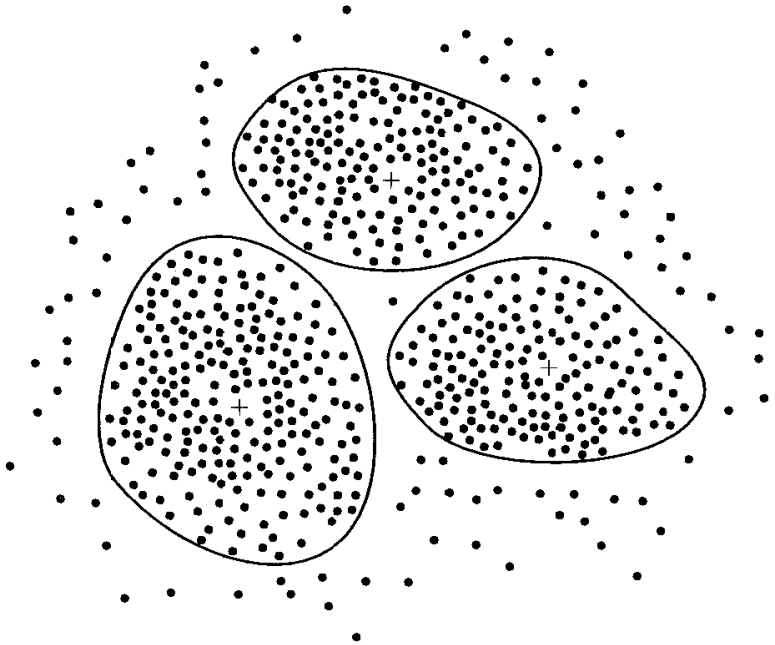
Важно е да се помни, че голямото разсейване не винаги е признак на шум в данни – обикновено това се отнася само за еднородни данни, т.е. данни отнасящи към един и същ клас (например, когато става дума за стойността на температура на класа "*Здрави хора*"). Обратно, често това е признак на наличието в данни на отделни групи – клъстери или класове (например, голямото разсейване може да се получи при измерване на температура на здрави и болни хора). По-подробно тази тема е засегната в лекция, посветена на клъстерния анализ.

Изчистване на зашумени данни може да стане чрез откриване и премахване на случайни грешки, които се проявяват като отклонение от общото поведение – екстремални стойности или крайности. Крайностите могат да бъдат определени чрез вече известния на нас статистическия анализ, например за такива могат да се считат стойностите, лежащи на разстояние 1.5 пъти по-голямо от междуквартилното разстояние от първия и третия квартил (виж предишната лекция). Обаче, макар че стойността на всеки атрибут от определена запис може да не бъде екстремна, самата запис като цяло може да се отличава значително от подобните записи от бази данни. Такива записи могат да бъдат идентифицирани чрез клъстерния анализ (виж фиг. 3-1) или други методи, които по-подробно ще разгледаме в лекция, посветена на анализа на крайностите.

Противоречиви се наричат данни с грешни стойности на целевия атрибут. За откриване на подобни противоречия се използват налични основни знания за проблемната област, например списък на допустими стойности на целевия атрибут - за откриване на грешки при въвеждане или неприемливи съкращения, известни функционални ограничения на стойности на някои атрибути от съответните класове (например атрибут *Цвят* за клас *Ябълка* не може да приема стойност *Син*, или стойността на атрибута *Тегло* за клас *Бебе* да бъде по-голяма от 10 кг) и т.н.

Най-лесно се откриват синтактичните противоречия – това са случаи, когато в база данни съществуват идентични записи, различаващи се само по стойността на целевия атрибут. Начина за разрешаване на подобни противоречия – пълно премахване на всички противоречащи записи или само някои от тях - зависи както

от честотата на тяхно срещане, така и размера на самата база. Ако базата съдържа достатъчно записи за "спорни" класове, то подобни противоречиви записи могат да бъдат изтрити напълно. В противен случай окончателното решение може да се взима на база на определяне на степента на сходство на спорните записи със всеки от противоречащи класове или чрез прилагане на някой класификационен алгоритъм, предсказващ стойността на класа на спорния запис.



Фиг. 3-1. Идентификация на крайностите чрез клъстерния анализ

3.2. Интеграция на данни

В повечето реални случаи процесът на ИЗД изисква интегриране на данни от различни източници, като бази данни или файлове, в един еднороден източник на данни, например *n x p* таблицата на данни. При реализацията на този процес могат да възникнат множество проблеми. Един от тях е *идентификацията на обекти*. Например, как изследователят може да бъде сигурен, че атрибутът *ИД_потребител* в една база данни и атрибутът *Потребител_№* имат един и същ смисъл (уникален идентификатор на потребителя) и сочат към едни и същи обекти? За решаването на подобни проблеми обикновено се използват метаданни – данни за данни, които е част от бази данни.

Другият проблем при интеграция е излишество на атрибути. Един атрибут е излишен, ако той може да бъде изведен от другата таблица. Наличието на излишни атрибути може да бъде предизвикано и от противоречия в системата за наименования на атрибути. Някои излишества могат да бъдат открити с помощта

на корелационен анализ, разгледан в предишната лекция. Така например, чрез изчисляване на коефициента на корелация могат да бъде решен проблем с атрибути *ИД_потребител* и *Потребител_№*.

Като резултат от интеграция може да възникне проблемът с наличие на дублиращи се записи.

Много важен проблем при интеграцията е. *намирането и разрешаване на конфликти между атрибутните стойности.* Например, атрибутните стойности на една и съща наблюдаема величина в различни бази данни могат да бъдат различни. Причината за това може да бъде различното им представяне, мащабиране или кодиране. Например, в една база атрибутът *Тегло* може да бъде съхраняван в метричната система, а в другата – в британската. Атрибутът *Цена* на стаята за различни хотели може не само да бъде представена в различни валути, но и да включва различни услуги (например закуска) и т.н. Наличието на подобни семантично нееднородни данни изисква от изследователя внимателна предварителна работа по тяхна правилна интеграция. Създаването на еднородна база от данни, не съдържаща противоречия и излишества, значително подобрява точността и бързодействие на прилагани към нея в последствие моделиращи ИЗД алгоритми.

3.3. Трансформация на данни

Целта на трансформацията на данни е да представят данни във вида по-удобен за по-нататъшния анализ. Обикновено конкретен вид на трансформации, извършвани над данни, се определя от типа на основната ИЗД задача и изисквания към формата на входни данни, налагани от използвания ИЗД алгоритъм за моделиране.

3.3.1. Нормализация на данни

Една от често срещаните форми на трансформация на числови (непрекъснати) данни е тяхната *нормализация* – т.е. привеждане на стойности на трансфирмирания атрибут в зададения диапазон. Нормализацията е особено полезна за създаване на класификационни модели с помощта на невронни мрежи или за методи, базирани на измерване на разстояния, такива като най-близкия съсед или клъстеризация. Съществуват различни методи за нормализация, като най-популярните от тях са: мини-максна нормализация, z-нормализация и нормализация чрез десетично мащабиране.

Мини-максната нормализация изпълнява линейната трансформация на оригинални данни. Нека min_A и max_A са минималната и максималната стойности на атрибута A. Минимаксната нормализация отобразява всяка стойност v на A в стойността v' от диапазона $[new\_min_A, new\_max_A]$ чрез трансформацията:

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A \quad (3.2)$$

Мини-максната нормализация запазва съотношения между оригиналните стойности на данни. Тя може да се ползва за откриване на грешка в данни от типа "стойност извън диапазона", ако се окаже, че конкретната стойност на атрибута в някои от записи лежи извън оригиналния диапазон на атрибутните стойности .

z-нормализация (или нормализация *с нулево средно аритметично*) извършва нормализиране на данни на базата на средно аритметичното и стандартното отклонение на атрибута *A*. Стойността *v* на *A* се нормализира във *v'* чрез трансформацията:

$$v' = \frac{v - \widetilde{A}}{\sigma_A} \quad (3.3)$$

където $\widetilde{A}$ и σ_A са, съответно, средно аритметично и стандартно отклонение на *A*. Този метод се използва, когато актуалните стойности на минимума и максимума на атрибута са неизвестни, или когато има екстремни стойности (ourliers), доминиращи в мини-максната нормализация.

Нормализация чрез десетично мащабиране изпълнява нормализацията чрез преместване на десетичната точка на стойностите на *A*. Броят на позиции при преместване на точката зависи от максимума на абсолютната стойност на *A*. Стойността *v* на *A* се нормализира във *v'* чрез трансформацията:

$$v' = \frac{v}{10^j} \quad (3.4)$$

където *j* е най-малкото цяло число, такова че $max\{|v'|\} < 1$.

Пример 3.2. Нека записаните стойности на *A* се променят от *–987* до *92*. Максималната абсолютна стойност на *A* е *987*, следователно $j = 3$. Като резултат след нормализацията чрез десетичното мащабиране диапазонът на стойностите на *A* ще бъде *[-0.987, 0.092]*.

Обърнете внимание, чрез след прилагането на нормализацията данните се променят. За да бъдат нормализирани правилно и бъдещите данни, необходимо да се пази параметрите на използваната нормализация (като, например, средното и стандартното отклонение при z-нормализацията).

3.3.2. Дискретизация на данни

Трансформацията на числови данни чрез "изглаждане" вече е разгледана като един от методи за намаляване на шума в данни. Дискретизацията на непрекъснати данни много напомня "изглаждането", тъй като също се базира на разбиването на данни на интервали. Обаче в отличието от "изглаждането", където числовите стойности, попаднали в един и същ интервал, се заменяха с една или няколко *пак числови*

стойности, при дискретизацията всички числови стойности, попаднали в един и същ дискретизационен интервал, се заменят с една и съща *номинална* стойност, която е уникална за всеки интервал. Например, непрекъснатите стойности на измерената човешка температура могат да се разбият на 3 интервала [35.0, 37.0], [37.1, 38.0], [38,1,42.0] след което атрибутът "*Температура*" да бъде разглеждан като номинален със стойностите *[Нормална, Повишена, Висока]*. Дискретизацията на непрекъснатите атрибути се прилага като метод за предварителната обработка на данни в случаите, когато основния ИЗД алгоритъм, използван за моделиране, може да работи само с номинални атрибути (например наивен Бейсов класификатор, някои видове асоциативни правила и т.н.), или това води до значително ускоряване на неговата работа (например при обучение с класификационни дървета) или точността (виж, например, SNMC). Дискретизацията може да се използва и като метод за решаване на такава самостоятелна ИЗД задача, като създаване на йерархия от понятия от едномерни данни, към която ще се върнем по-подробно след запознаване с клъстерния анализ.

Съществуват различни начини за групиране на методи за дискретизация. Разбиването на *локални и глобални* отразява дали процедурата на дискретизация се прилага към отделни локални области от пространството на данни или към цялото пространство. Разбиването на *статични или динамични* отразява дали търсене на параметри на дискретизация се изпълнява независимо за всеки атрибут по-отделно или за всички атрибути заедно, т.е. отчитайки зависимостите между атрибутите.

Най-често използвано разделение на дискретизационните методи е на *управляеми и неуправляеми* (supervised - unsupervised). Неуправляемите методи не използват информация за целевия атрибут – клас, докато при управляеми – тази информация лежи в основата на избора на дискретизационни параметри (например, брой и ширината на интервалите). Очевидно е, че управляемите методи са приложими само за класифицирани данни (т.е. данни с известна стойност на целевия атрибут).

Едни от най-популярни (и лесни за реализация) методи за неуправляемата дискретизация са дискретизация на интервали с еднаква ширина и дискретизация на интервали с еднаква дълбочина.

Дискретизация на интервали с еднаква ширина

При *дискретизацията на интервали с еднаква ширина* цял диапазон на стойности на избрания за дискретизация непрекъснат атрибут се разбива на k интервали с еднакъв диапазон (ширина), където k – е параметър, задаван от потребителя. Т.е. ако наблюдаваните стойности на атрибута X са ограничени от x_{min} и x_{max}, то ширината на интервала се изчислява по формулата:

$$\delta = \frac{x_{\max} - x_{\min}}{k} \quad (3.5)$$

а границите на интервали се създават в точки $x_{min} + i\delta$, $i = 1,\ldots, k-1$.

Най-често използваната стойност на k е 10. За да бъде елиминирано влияние на екстремните стойности, често преди прилагане на тази дискретизация се прави z-нормализация. Така при $k = 10$ получените данни се разбиват на интервали, с дължина равна на половина от стандартно отклонение – по 5 интервала вляво и вдясно от z-нормализирана нулева стойност.

Дискретизация на интервали с еднаква дълбочина

При *дискретизацията на интервали с еднаква дълбочина (честота)* атрибутът се разбива на k интервала с *приблизително еднакъв брой* на атрибутните стойности във всеки интервал, където k – е параметър, задаван от потребителя. Т.е., ако база данни съдържа m (възможно повтарящи се) стойности на наблюдавания атрибут, то *дълбочината* на дискретизационния интервал Δ (брой на стойности в интервала) се определя като най-близкото цяло до m/k.

Дискретизация чрез максимални разлики

Добрите резултати се получават и чрез така наречена *дискретизация чрез максимални разлики*. И в този случай потребителят задава броя на желаните интервали за дискретизация – k. Дискретизацията се изпълнява на две стъпки – на първата се изчисляват разликите между всички съседни стойности на атрибута и се избират $k-1$ най-големи разлики. На втората стъпка $k-1$ граници на дискретизационни интервали се установяват по средата между съответните атрибутни стойности, образуващи изчислените разлики.

Клъстерна дискретизация

Задачата за дискретизация може да се разглежда и като задача за *клъстеризацията* – разбиването на цялото множество от непрекъснати стойности на атрибута на (предварително зададен или автоматично изчислен) брой клъстери (дискретизационни интервали), така че всички обекти (атрибутни стойности) на един и същ интервал са "по-близки" помежду си (съгласно някой избран критерий – например, разстояние до центъра на интервала), отколко с обекти от съседни интервали. По-долу ще разгледаме един такъв подход, базиращ се на идеята за агломеративна клъстеризация (по-подробно за клъстеризацията – в една от следващите лекции).

Основната идея на алгоритъма (J. Li, H. Shen and R. Topor 2000), който ще наречем *агломеративна клъстерна дискретизация (АКД)*, се състои в следното – клъстери (в нашия случай - интервали) се създават итеративно – чрез сливане на някои от съседни интервали, направени на предишната итерация. От всички двойки интервали – кандидати за сливане, се избира тази, която минимизира предварително избран критерий за сливане. Процесът на дискретизацията чрез сливането започва от "елементарни" интервали – т.е. интервали, съдържащи само по една (различна) атрибутна стойност, и продължава докато получените интервали не удовлетворят определен критерий за спиране. Обърнете внимание, че

при този подход потребителят *не задава* предварително брой на дискретизационни интервали – той се определя автоматично от използвания критерий за спиране.

Авторите на алгоритъма предлагат една ефективна процедура за изчисляване на интервали, която се базира на представяне на всеки интервал I_k чрез своя център c_k - средно аритметично от съдържащи се в него стойности, и броя N_k на атрибутните стойности в интервала. Разликата между два съседни интервала I_k и I_{k+1} се дефинира като претегленото (по броя на атрибутните стойности) линейно разстояние между техните центрове:

$$Diff(c_i, c_{i+1}) = \frac{N_i N_{i+1}}{N_i + N_{i+1}} (c_{i+1} - c_i) \quad (3.6)$$

Имайки на някоя итерационна стъпка на алгоритъма разбиването на интервали I_1, ..., I_m със съответните центрове c_1, ..., c_m, ние можем да избираме измежду $m-1$ двойки интервали за сливане. След тестване на всяка двойка интервали се слива тази с най-малко $Diff$. Ако това е двойката (I_t, I_{t+1}), то новият интервал ще бъде I' = $I_t \cup I_{t+1}$ с характеристики:

$$N' = N_t + N_{t+1} \text{ и } c' = \frac{c_t N_t + c_{t+1} N_{t+1}}{N_t + N_{t+1}} \quad (3.7)$$

Критерий за спиране на процедурата за сливане е

$$(c_{t+1} - c_t) > 3 * d \quad (3.8)$$

където d е средното разстояние между последователни стойности на числовия атрибут, подлежащ на дискретизация:

$$d = \frac{1}{m-1} [(v_2 - v_1) + (v_3 - v_2) + ... + (v_m - v_{m-1})] = \frac{1}{m-1} (v_m - v_1) \quad (3.9)$$

където $v_1 \le v_2 \le ... \le v_k$ - са стойности на атрибута.

Границите между получените интервали се установяват между техните центрове с отчитане на техните тегла (брой на съдържащите в интервала стойности).

Авторите използват описаната дискретизация за задача на извличане на асоциативни правила от бази данни с непрекъснати атрибути.

Дискретизация, базирана на ентропия

Един от най-ефективните методи за *управляема* дискретизация е *рекурсивната дискретизация с минимизацията на ентропия (Recursive minimal entropy partitioning)*, предложена от Fayyad & Irani – 1993. Нерекурсивният вариант на дискретизация с използване на ентропия (разбиване само на два интервала) вече ни е познат от курса по машинно самообучение - той се използва при построяване на класификационни дървета за непрекъснати атрибути. Настоящият вариант позволява дискретизацията на няколко интервала с различна ширина.

Задачата за дискретизация се свежда до намиране на границите, разбиващия целия диапазон на непрекъснатия атрибут на няколко непресичащи се интервали. Ако разполагаме с S записи от база данни, процедурата за дискретизация на някой непрекъснат атрибут A има следния вид:

1. Всяка стойност v от A може да се разглежда като една потенциална граница T на двоичен дискретизационен интервал, разбивайки всички записи S на два подмножества S_1 и S_2, удовлетворяващи, съответно, условия $A < v$ и $A \geq v$.
2. Ентропията $E(A, T; S)$ на такава двоична дискретизация се задава от формула:

$$E(A,T;S) = \frac{|S_1|}{|S|} Entropy(S_1) + \frac{|S_2|}{|S|} Entropy(S_2) \quad (3.10)$$

където $|S_1|$ и $|S_2|$ са, съответно, брой на записи в база данни, удовлетворяващи условия $A < v$ и $A \geq v$. Ентропията за даденото множество от записи се изчислява на базата на разпределение на записи по класове. Например, за m класа ентропията на S_1 е:

$$Entropy(S_1) = -\sum_{i-1}^{m} p_i \log_2(p_i) \quad (3.11)$$

където p_i е вероятността на класа i в S_1 , изчислена чрез деление на брой на записи от този клас в S_1 върху общия брой записи в S_1.

От всички възможни двоични разбивания се избира такава граница T, която *максимизира информационната печалба*:

$$Gain(A, T; S) = Entropy(S) - E(A, T; S) \quad (3.12)$$

3. Процесът се прилага рекурсивно към двата интервала, получени при двоична дискретизация, докато не бъде изпълнен критерий за спиране:

$$Gain(A, T; S) < \delta \quad (3.13)$$

В оригиналната версия като критерий за спиране авторите използват принципа за *минималната дължината на описание* (виж отново лекции по МС), който в нашия случай е:

$$Gain(A,T;S) < \frac{\log_2(N-1)}{N} + \frac{\Delta(A,T;S)}{N} \quad (3.14)$$

където N е брой на записи в S, а

$$\Delta(A,T;S) = \log_2(3^k - 2) - [k \cdot Entropy(S) - k_1 \cdot Entropy(S_1) - k_2 \cdot Entropy(S_2)] \quad (3.15)$$

където k_i е брой на класове, представени в S_i.

Тъй като всеки клон на такава рекурсивна дискретизация се оценява независимо чрез този критерий, някои области от диапазона на непрекъснатия атрибут ще бъдат разбити на малки - фини интервали, а други – на груби – големи.