ENGEENIRING AND TECHNOLOGY

Kanev O.K.

FUZZY CLUSTERING WITH USING DIFFERENT METRICS FOR CALCULATING DISTANCES FROM AN OBJECT TO A PROTOTYPE IN DIAGNOSTIC SYSTEMS

Kanev O.K., Russia, Nizhny Novgorod State Technical University n.a. R.E. Alekseev, Master of the Department of Computer Systems and Technologies

Abstract

The article contains description of two standard fuzzy clustering algorithms (Fuzzy C-Means and Possibilistic C-Means) with using different metrics for calculating distances from an object to a cluster's center. Euclidean distance, Hemming distance, Chebyshev distance, Mahalanobis distance and peak distance have been considered. The results of research quality level, time of algorithm's work and count of iterations are given and commented.

Keywords: data mining, fuzzy logic, degree of membership, state of an object, a cluster’s prototype, fuzzy partition.

Введение

На сегодняшний день автоматизация трудоемких аналитических процессов является одной из наиболее востребованных областей прикладных исследований. Бесспорно, это обусловлено тем, что при решении задач интеллектуального анализа, характерных для данной области, на практике нередко приходится обрабатывать большие массивы многомерных данных. Это является основной причиной, по которой активно ведутся исследования в области

автоматизации процессов, так как ручная обработка больших наборов информации требует огромных временных затрат, а также вносит долю субъективной погрешности, которая заметно снижает качество результата.

При автоматизации процессов особое внимание уделяется диагностированию состояний объектов различной природы, как например, оценка состояния здоровья пациента на базе некоего набора проб или оценка состояния оборудования перед началом эксплуатации. В связи с этим одним из наиболее перспективных и актуальных направлений прикладных исследований является проектирование и разработка диагностических систем, которые в настоящее время применяются во всех отраслях народного хозяйства.

Важным вопросом при проектировании диагностических систем является способ задания прототипов (или эталонов) классов для диагностирования состояний объектов. С одной стороны можно задавать их вручную перед началом эксплуатации. Однако со временем заданные эталонные образцы могут устаревать. Как следствие, возникает необходимость корректировки эталонных значений, что при ручном труде также приводит к временным затратам. В виду этого возникает необходимость реализации в рамках диагностической системы обучающего модуля, который будет самостоятельно вырабатывать прототипы для дальнейшей диагностики объектов и корректировать их по запросу на основе некоторого обучающего множества.

Для реализации блока обучения диагностической системы актуально использование алгоритмов кластерного анализа, так как они в большинстве своем направлены на разбиение выборки объектов с попутной выработкой центров кластеров, которые в дальнейшем можно использовать как эталоны для диагностики.

Отметим, что при кластеризации данных особое внимание стоит уделять способу отнесения объекта в тот или иной кластер. Это обусловлено тем, что большая часть алгоритмов кластерного анализа являются метрическими, то есть используют некоторую меру близости для оценки расстояния от объекта до центроида.

Кроме того, также важен и сам выбор алгоритма кластеризации. Основное применение в информационных технологиях нашли неиерархические итерационные алгоритмы кластерного анализа, дающие за конечное число итераций оптимальное разбиение. Среди них выделяются алгоритмы нечеткой кластеризации, так как в отличие от алгоритмов

четкой (жесткой) кластеризации они предполагают отнесение объекта сразу во все кластеры, но с некоторой степенью принадлежности, что улучшает результат, а также делает решение задачи о разбиении более гибким, давая возможность проанализировать данные и выявить те объекты, которые негативно сказываются на результате разбиения.

В виду описанного выше целью исследования является оценка качественных (эффективность и количество ошибок разбиения) и количественных (затрачиваемое время и число итераций, за которые достигается оптимальное разбиение) характеристик алгоритмов нечеткой кластеризации при использовании разных мер близости для вычисления расстояния от объекта до центров кластеров.

Материалы и методы исследования

Пусть нам дано некоторое множество, состоящее из D объектов, каждый из которых характеризуется вектором в N-мерном Евклидовом пространстве. Необходимо разбить предложенное множество на заданное число кластеров. Отметим, что в рамках реализации блока обучения диагностической системы задача кластерного анализа данных сводится не столько к нахождению оптимального разбиения на заданное число кластеров, сколько к поиску наиболее оптимальных положений их центров, которые в дальнейшем будут использоваться в качестве прототипов, необходимых для диагностирования состояний объектов.

Таким образом, блок обучения диагностической системы будет представлен в виде некоторого программного модуля, на вход которого поступает обучающая выборка, представленная в виде матрицы «объект-признак», а на выходе мы наблюдаем заданное число векторов-прототипов для дальнейшей диагностики иных объектов, имеющих ту же природу и структуру, что и элементы обучающего множества. В качестве «начинки» блока обучения будет выступать алгоритм нечеткой кластеризации, который и будет осуществлять процесс разбиения входного множества на кластеры.

Для решения задачи нечеткой кластеризации были выбраны два алгоритма, основанные на итерационном пересчете центроидов кластеров и матриц нечеткого разбиения: Fuzzy C-Means (FCM) [1, 2] и Possibilistic C-Means (PCM) [3, 4]. Выбор был сделан в пользу данных алгоритмов в виду того, что другие известные на сегодняшний день алгоритмы нечеткой кластеризации, такие как Relational Fuzzy C-Means [5], Non-Euclidean Relational Fuzzy C-Means [6], Kernelized Non-Euclidean Relational Fuzzy C-Means [7], Robust Relational Fuzzy C-Means

[8], Robust Non-Euclidean Relational Fuzzy C-Means [9], Similarity-based Possibilistic C-Means [10] и алгоритм Гюстафсона-Кесселя [11], являются всего лишь их модификациями.

Опишем базовую структуру обоих алгоритмов.

В начале мы задаем для работы обоих алгоритмов необходимые условия, а именно, число кластеров C, на которые требуется разбить исходное множество, величину экспоненциального веса w, которую, как правило, принимают равной двум, так как нет четко обоснованного правила его выбора [2], и меру точности ε, которая используется для оценки оптимальности полученного разбиения на текущей итерации.

Далее, согласно базовым трактовкам алгоритмов, генерируется начальная (стартовая) матрица нечеткого разбиения M, которая имеет следующий вид:

$$M = \begin{vmatrix} \mu_{11} & \cdots & \mu_{1i} & \cdots & \mu_{1C} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \mu_{k1} & \cdots & \mu_{ki} & \cdots & \mu_{kC} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ \mu_{D1} & \cdots & \mu_{Di} & \cdots & \mu_{DC} \end{vmatrix}, \mu_{ki} \in [0,1], \quad (1)$$

где μ_{ki} – степень принадлежности k-го объекта i-му кластеру.

При начальной генерации матрицы M ее ячейки заполняются произвольными значениями μ_{ki} из интервала от 0 до 1. При этом для FCM-алгоритма должно строго выполняться следующее требование:

$$\left(\sum_{i=1}^{C} \mu_{ki} = 1 \right) \bigcap \left(0 < \sum_{k=1}^{D} \mu_{ki} < D \right) \quad (2)$$

Для PCM-алгоритма требование (2) может не выполняться.

Далее вычисляются положения центров заданных кластеров:

$$c^{(i)} = \frac{\sum_{k=1}^{D} \left(\mu_{ki}^{w} * x^{(k)} \right)}{\sum_{k=1}^{D} \mu_{ki}^{w}}, \quad (3)$$

где $c^{(i)}$ – центр i-го кластера, $x^{(k)}$ – входной вектор атрибутов k-ого объекта.

После вычисления координат центров, осуществляется расчет квадратов расстояний от каждого объекта до каждого кластера.

Для выбранных алгоритмов базовой мерой близости является Евклидово расстояние, вычисляемое по формуле:

$$\rho E(x,c) = \sqrt{\sum_{l=1}^{N}(x_l - c_l)^2} \quad (4)$$

Следует отметить, что Евклидово расстояние, возведенное в квадрат, как и предполагают алгоритмы FCM и PCM, позволяет придать большие веса разрозненным (более отдаленным друг от друга) объектам. По этой причине оно и было выбрано в рамках стандартных трактовок выбранных алгоритмов.

Но согласно поставленной цели исследования было решено дать оценку качественных и количественных характеристик работы алгоритмов при использовании разных мер близости. Поэтому кроме Евклидова расстояния мы будем использовать и иные достаточно известные метрики, часто применяемые на практике: Хемминогово расстояние, расстояние Чебышева, расстояние Махаланобиса и пиковое расстояние.

Хеммингово расстояние вычисляется по формуле:

$$\rho H(x,c) = \sum_{l=1}^{N}|x_l - c_l| \quad (5)$$

Как сообщается в [11], в большинстве случаев Хеммингова мера близости приводит к таким же результатам, что и при расчете расстояния по Евклидовой метрике. Однако, в отличие от нее, при использовании Хеммингова расстояния ослаблено влияние выбросов (отдельных больших разностей), так как отсутствует возведение в квадрат.

Расстояние Чебышева рассчитывается по следующей формуле:

$$\rho\infty(x,c) = \max_{1\le l\le N}|x_l - c_l| \quad (6)$$

Как видно из (6), использование данной меры близости полезно лишь в том случае, если мы различаем объекты по какой-нибудь одной координате.

Расстояние Махаланобиса рассчитывается по формуле:

$$\rho M(x,y) = (x-c) * S^{-1} * (x-c)^T, \quad (7)$$

где S – ковариационная матрица размерности NxN, вычисляемая по формуле:

$$S = \frac{1}{D-1}\sum_{k=1}^{D}\left[\left(x^{(k)} - \bar{x}\right) * \left(x^{(k)} - \bar{x}\right)^{T}\right] \quad (8)$$

Как сообщается в [11], данная мера плохо работает, когда расчет ковариационной матрицы осуществляется по всему исходному множеству. В виду этого данную меру близости целесообразно применять, сосредотачиваясь на отдельных группах данных.

Пиковое расстояние вычисляется по формуле:

$$\rho L(x,c) = \frac{1}{N}\sum_{l=1}^{N}\frac{|x_l - c_l|}{x_l + c_l} \quad (9)$$

После вычисления квадратов расстояний в FCM-алгоритме выполняется пересчет степеней принадлежности по следующей формуле:

$$\mu_{ki}^{*} = \begin{cases} 1, \ при \ \rho_{ki}^{2} = 0 \\ \dfrac{1}{\left(\sum\limits_{j=1}^{C}\dfrac{\rho_{ki}^{2}}{\rho_{kj}^{2}}\right)^{1/w-1}}, \ при \ \rho_{ki}^{2} \neq 0 \end{cases} \quad (10)$$

где μ^{*}_{ki} – пересчитанное значение степени принадлежности k-ого объекта i-му кластеру.

В PCM-алгоритме перед пересчетом степеней принадлежности для каждого кластера рассчитывается ширина зоны по формуле:

$$\eta_i = \frac{\sum\limits_{k=1}^{D}\left(\mu_{ki}^{w} * \rho_{ki}^{2}\right)}{\sum\limits_{k=1}^{D}\mu_{ki}^{w}} \quad (11)$$

Степени принадлежности пересчитываются по формуле:

$$\mu_{ki}^{*} = \frac{1}{1 + \left(\dfrac{\rho_{ki}^{2}}{\eta_i}\right)^{1/w-1}} \quad (12)$$

После пересчета степеней принадлежности по (10) и (12), для FCM- и PCM-алгоритмов соответственно, выполняется проверка оптимальности полученного результата. Для этого вычисляется матричная норма разности старой и новой матриц

нечеткого разбиения, равная максимальному по модулю элементу, и далее сравнивается с заданным параметром точности ε. Если значение матричной нормы не превышает ε, считается, что достигнуто оптимальное разбиение и алгоритмы завершают свою работу. В противном случае осуществляется переход к расчету центров кластеров.

Для оценки качественных и количественных характеристик выбранных алгоритмов было разработано приложение на языке C#, реализующее считывание входных данных из файла *.xls, работу FCM- и PCM-алгоритмов с выбором метрики для оценки расстояния от объекта до центроида и выдачу статистики по итогам разбиения.

В качестве входных данных, подлежащих кластерному анализу, используются сведения о пациентах, наблюдаемых на предмет анализа состояния микробиоты желудочно-кишечного тракта с целью определения у них степени дисбактериоза. Каждый из них характеризуется набором из 21 атрибута, значения которых предварительно нормируются. Для контроля качества разбиения во входных данных об объекте предусмотрено специальное поле, содержащее экспертную оценку состояния объекта. По результатам разбиения номер кластера, которому принадлежит объект, сравнивается с назначенной ему экспертной оценкой.

Кроме того, в приложении предусмотрен счетчик итераций и таймер, который запускается перед началом работы алгоритма и останавливается после удовлетворения условия оптимальности.

Качество разбиения оценивается по следующей формуле:

$$Q = \left(1 - \frac{E}{D}\right) * 100\%, \quad (13)$$

где E – количество ошибок при разбиении.

Оценки качественных и количественных характеристик обоих алгоритмов проводятся для каждой рассмотренной метрики.

Разбиение предложенного множества проводится на 4 кластера.

Результаты и обсуждение

В таблицы 1 и 2 сведены результаты исследования зависимости времени выполнения разбиения от размера входного множества при использовании разных мер близости для расчета расстояний от объектов до центров кластеров по FCM-алгоритму и PCM-алгоритму соответственно.

Таблица 1

Исследование зависимости времени выполнения алгоритма FCM от размера множества при разных мерах близости

Размер множества	Время работы алгоритма, мс				
	dE	dH	d∞	dM	dL
50	3,74	2,26	34,56	56,77	4,98
100	12,84	10,47	79,03	123,61	9,47
150	23,89	19,79	102,49	189,9	13,29
200	36,76	25,07	135,7	276,8	16,52
250	49,95	39,54	186,65	401,67	20,62
300	58,98	50,48	239,39	678,01	26,61

Таблица 2

Исследование зависимости времени выполнения алгоритма PCM от размера множества при разных мерах близости

Размер множества	Время работы алгоритма, мс				
	dE	dH	d∞	dM	dL
50	5,06	3,99	36,98	61,46	5,78
100	13,15	12,21	82,32	130,54	11,23
150	26,78	20,09	107,74	201,71	15,87
200	39,42	27,91	140,1	310,76	19,12
250	52,78	41,74	192,23	446,6	25,91
300	60,6	51,65	255,89	732,12	31,59

На рисунках 1 и 2 приведены графики зависимости времени выполнения алгоритмов FCM и PCM соответственно от размера исходного множества при разных мерах близости.

Как видно из рисунков 1 и 2, время достижения оптимального разбиения для обоих алгоритмов линейно возрастает с ростом размера исходной выборки объектов независимо от выбранной метрики. Однако следует отметить, что в виду собственной вычислительной сложности разные меры близости приводят к разным временным затратам.

Согласно результатам эксперимента, оба алгоритма работают быстрее при использовании Хеммингова расстояния в качестве меры близости. Наибольшие временные затраты также для обоих алгоритмов наблюдаются при использовании в качестве метрики расстояния Махаланобиса в виду достаточно высокой вычислительной сложности из-за работы с матрицами.

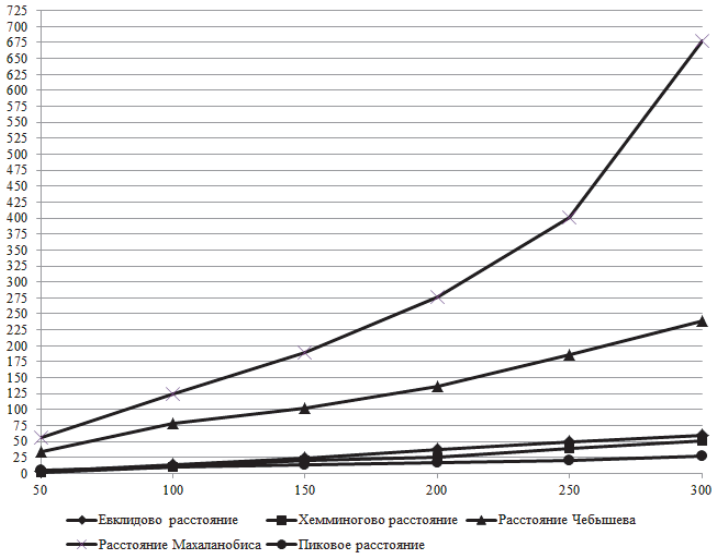

Рисунок 1. Зависимость времени выполнения алгоритма FCM от размера множества при разных мерах близости

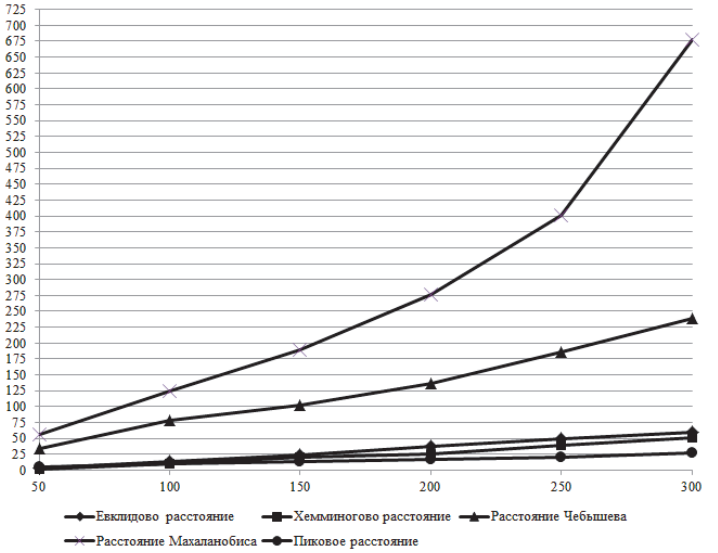

Рисунок 2. Зависимость времени выполнения алгоритма PCM от размера множества при разных мерах близости

В таблицы 3 и 4 сведены результаты исследования зависимости числа итераций, за которое достигается оптимальное разбиение FCM-алгоритмом и PCM-алгоритмом соответственно при использовании разных мер близости для расчета расстояний от объектов до центроидов.

Таблица 3

Исследование зависимости числа итераций от размера множества при разных мерах близости для FCM-алгоритма

Размер множества	Число итераций				
	dE	dH	d∞	dM	dL
50	10	11	9	20	14
100	17	16	13	24	16
150	18	20	10	21	15
200	16	19	8	19	14
250	13	16	12	25	14
300	15	26	11	23	15

Таблица 4

Исследование зависимости числа итераций от размера множества при разных мерах близости для PCM-алгоритма

Размер множества	Число итераций				
	dE	dH	d∞	dM	dL
50	9	12	10	19	17
100	14	14	11	23	18
150	15	12	10	18	20
200	17	13	11	24	22
250	10	11	13	24	21
300	11	15	9	19	16

Как видно из таблиц 3 и 4, количество итераций, затрачиваемых алгоритмом, не зависит от размера выборки.

Однако необходимо отметить, что расстояние Махаланобиса и пиковое расстояние приводят к затрате большего количества итераций в отличие от остальных трех метрик.

Кроме того, следует отметить, что независимость количества итераций от размера выборки обусловлена начальным заданием матрицы нечеткого разбиения.

В таблицах 5, 6, 7, 8 и 9 приведены результаты исследования качества разбиения, получаемого с помощью

FCM- и PCM-алгоритмов соответственно при использовании разных метрик и при разных размерах входного множества.

Таблица 5

Оценка качества разбиения FCM- и PCM-алгоритмами, при использовании Евклидова расстояния

Размер множества	50	100	150	200	250	300
E (FCM)	7	13	17	23	31	42
Q (FCM)	86%	87%	88.67%	88.5%	87.6%	86%
E (PCM)	6	15	16	22	35	50
Q (PCM)	88%	85%	89.33%	89%	86%	83.33%

Таблица 6

Оценка качества разбиения FCM- и PCM-алгоритмами, при использовании Хеммингова расстояния

Размер множества	50	100	150	200	250	300
E (FCM)	8	15	20	27	34	45
Q (FCM)	84%	85%	86.67%	86.5%	86.4%	85%
E (PCM)	9	13	17	23	33	46
Q (PCM)	82%	87%	88.67%	88.5%	86.8%	84.67%

Таблица 7

Оценка качества разбиения FCM- и PCM-алгоритмами, при использовании расстояния Чебышева

Размер множества	50	100	150	200	250	300
E (FCM)	13	25	40	64	79	103
Q (FCM)	74%	75%	73.33%	68%	68.4%	65.67%
E (PCM)	15	30	38	60	79	100
Q (PCM)	70%	70%	74.67%	70%	68.4%	66.67%

Таблица 8

Оценка качества разбиения FCM- и PCM-алгоритмами, при использовании расстояния Махаланобиса

Размер множества	50	100	150	200	250	300
E (FCM)	10	21	33	52	70	99
Q (FCM)	80%	79%	78%	74%	72%	67%
E (PCM)	13	30	45	62	77	102
Q (PCM)	74%	70%	70%	69%	69.2%	66%

Таблица 9

Оценка качества разбиения FCM- и PCM-алгоритмами, при использовании пикового расстояния

Размер множества	50	100	150	200	250	300
E (FCM)	14	23	42	60	75	99
Q (FCM)	72%	77%	72%	70%	70%	67%
E (PCM)	24	45	68	85	113	139
Q (PCM)	52%	55%	54.67%	57.5%	54.8%	53.67%

Таблица 10

Средний уровень качества конечного разбиения FCM- и PCM-алгоритмами при использовании разных метрик

Алгоритм	Средний уровень качества разбиения				
	dE	dH	d∞	dM	dL
FCM	87.3%	85.6%	70.73%	75%	71.33%
PCM	86.78%	86.27%	69.96%	69.7%	54.61%

В таблице 10 приведены сведения о средних уровнях качества разбиения для обоих алгоритмов нечеткой кластеризации при использовании разных мер близости для расчета расстояния.

Выводы

Были подробно рассмотрены два базовых алгоритма нечеткой кластеризации: Fuzzy C-Means и Possibilistic C-Means, которые являются основой для остальных упомянутых выше алгоритмов. В ходе проведения экспериментов было установлено, что с ростом размера исходного множества, подлежащего разбиению, линейно возрастает время обработки, за которое достигается оптимальное разбиение. Как показали полученные данные, при использовании в качестве метрики при расчете расстояния от объекта до центра кластера Хеммингова расстояния мы имеем наименьшие временные затраты на обработку данных.

Также было установлено, что число итераций, затрачиваемых на обработку входных данных, не зависит от размера выборки. На данный показатель влияет только начальное задание матрицы нечеткого разбиения.

Кроме того, было выявлено, что с увеличением размера выборки растет количество ошибок разбиения.

На основе полученных данных можно сделать вывод, что оба алгоритма работают наиболее эффективно при своих

базовых трактовках, то есть при использовании Евклидова расстояния в качестве меры близости.

References:

[1] J.C. Bezdek. Pattern recognition with fuzzy objective function algorithms. New York: Plenum, 1981
[2] Shtovba S.D. Vvedenie v teoriyu nechetkix mnozhestv i nechetkuyu logiku. Vinnica: Kontinent-Prim. - 2003. - 198 s.
[3] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. IEEE Trans. Fuzzy Syst., vol. 1. no. 2, pp. 98-110, Apr. 1993.
[4] Demidova L.A., Titov S.B. Podxod k probleme nechetkoj klasterizacii v usloviyax neopredelennosti vybora celevoj funkcii // Vestnik Pyazanskogo Gosudarstvennogo Radiotexnicheskogo Universiteta. 2009. № 29. S. 54-60.
[5] R.J. Hathaway, J.W. Davenport, and J.C. Bezdek. Relational duals of the c-Means clustering algorithms. Pattern Recognition, vol. 22, no. 2, pp. 205-212, 1989.
[6] R.J. Hathaway and J.C. Bezdek. NerF c-means: Non-Euclidean relational fuzzy clustering. Pattern Recognition, vol. 27, no. 3, pp. 429-437, 1994.
[7] R.J. Hathaway, J.M. Huband, and J.C. Bezdek. A Kernelized Non-Euclidean Relational Fuzzy c-Means Algorithm. Proceedings of FUZZ-IEEE 2005 IEEE Press, Piscataway, N.J., pp. 414-419, 2005.
[8] R.N. Dave and S. Sen. Robust fuzzy clustering of relational data. IEEE Trans on Fuzzy Systems, vol. 10, no. 6, pp. 713-727, 2002.
[9] J.W. Davenport and R.J. Hathaway. Possibilistic c-means clustering for relational data. Proccedings of the 1st International Conference on Neural, Parallel and Scientific Computations, vol. 1, pp. 139-142, 1995.
[10] V.S. Tseng and C. Kao. A novel Similarity-based fuzzy clustering algorithm by integrating PCM and mountain method. IEEE TRANSACTIONS ON FUZZY SYSTEMS, 15:1188-1196. DECEMBER, 2007.
[11] Barsegyan, A.A. Analiz dannyx i processov: ucheb. posobie / A.A. Barsegyan, M.C. Kupriyanov, I.I. Xolod, M.D. Tess, S.I. Elizarov. - 3-e izd., pererab. i dop. - SPb.: BXV-Peterburg, 2009. - 512 s.