УДК 005

ИССЛЕДОВАНИЕ АЛГОРИТМОВ МНОГОМЕРНОЙ КЛАССИФИКАЦИИ НАУЧНЫХ ДАННЫХ

Гусева А.И., Киреев В.С., Кузнецов И.А., Бочкарёв П.В.

Национальный исследовательский ядерный университет «МИФИ» (Московский инженерно-физический институт), Москва,

e-mail: aiguseva@mephi.ru, vskireev@mephi.ru, IAKuznetsov@mephi.ru, PVBochkarev@mephi.ru

Настоящая статья посвящена решению задачи классификации научных направлений. Источником данных является большой объем библиографической информации, содержащийся в международных научных системах и базах данных. Рассматривается задача многомерной классификации в пространстве большого количества признаков. Проводится сравнительный анализ таких алгоритмов классификации, как построение наивного байесовского классификатора, дерево решений и лес деревьев решений. Рассматривается подход усиления алгоритмов, связанный с введением комитетов голосующих алгоритмов. Предлагается алгоритм, представляющий собой модификацию голосования по большинству. Серия вычислительных экспериментов показала, что применение предлагаемого алгоритма повышает точность полученных решений до 90%. Для дальнейшего усовершенствования алгоритма, по нашему мнению, могут использоваться два направления: обучение комитета алгоритмов и введение весов атрибутов в пространстве признаков, описывающие классифицируемые объекты.

Ключевые слова: научное направление, жизненный цикл научного направления, научный результат, многомерная классификация, комитет голосующих алгоритмов

AN INVESTIGATION OF ALGORITHMS FOR MULTI-DIMENSIONAL CLASSIFICATION OF SCIENTIFIC DATA

Guseva A.I., Kireev V.S., Kuznetsov I.A., Bochkarev P.V.

National Research Nuclear University (Moscow Engineering Physics Institute), Moscow,

e-mail: aiguseva@mephi.ru, vskireev@mephi.ru, IAKuznetsov@mephi.ru, PVBochkarev@mephi.ru

The present article is devoted to solving the problem of classification of scientific trends. The data source is a large volume of bibliographic information contained in the international research systems and databases. Thr problem of multivariate classification in the space of a large number of characteristics is considered. A comparative analysis of classification algorithms was conducted, as the construction of a naive Bayesian classifier, a decision tree and a forest of decision trees. The authorsenhanced the algorithms associated, with the introduction of the committees of voting algorithms. The algorithm is proposed, which is a modification of the voting majority. Series of computational experiments showed that the application of the proposed algorithm improves the accuracy of the obtained solutions up to 90%. For further improvement of the algorithm, in our opinion, can be used two approaches: training of voting committee algorithms and the introduction of the weights of the attributes in the feature space, describing the classified objects.

Keywords: scientific trend, life cycle of scientific trend, scientific result, multi-dimensional classification, voting committee of algorithms

В настоящее время, когда финансирование научных исследований и разработок происходит на конкурсной основе, встает задача не только идентификации перспективных научных направлений, но и определение значимости научных результатов тех творческих коллективов, которые могут претендовать на такую поддержку.

Научное (научно-техническое) направление характеризуется совокупностью научных работ, объединенных общностью объекта и методов исследования, общностью тем и их взаимосвязанностью. К научным работам или, более точно, научным результатам (НР) мы относим тезисы докладов, статьи, монографии и учебники, препринты, защищенные диссертации и результаты интеллектуальной деятельности (патенты на изобретения, промышленные образцы, полезные модели и селекционные достижения, свидетельства государственной регистрации на программы для ЭВМ и базы данных).

Стандартная модель жизненного цикла научного направления предполагает четыре этапа своего развития: зарождение, рост, зрелость и насыщение с последующим распадом [3]. Этапы зарождения и роста составляют интенсивную фазу, а этапы зрелости и насыщения – экстенсивную. Во время первой, интенсивной фазы, число ежегодно получаемых принципиально новых научных результатов быстро увеличивается, на экстенсивной – падает. По длительности интенсивная стадия познания заметно короче, чем экстенсивная. На разных этапах жизненного цикла преобладают разные виды научных результатов. Так, на первом

этапе развития научного направления преобладают фундаментальные исследования, на второй – прикладные исследования и разработки. В соответствии с этим изменяется основной вид результатов научной деятельности в сторону увеличения патентов и свидетельств.

Объем накопленной в мире научной информации (тезисы, статьи, монографии, патенты и т.д.) оценивается в петабайтах (большие данные, Big Data) и требует специальных методов обработки. Современные мировые базы данных имеют свои собственные классификаторы научных направлений. Использование имеющихся классификаторов также сопряжено с рядом трудностей. Во-первых, эти классификаторы отличаются друг от друга, во-вторых, динамика изменения направления прикладных исследований чрезвычайно высока, и классификаторы не успевают изменяться так быстро, как нужно. В-третьих, появление нового научного направления не всегда бывает замечено и отражено в классификаторах. Например, в [1] приведены результаты исследования списка предметных кодов JEL в электронной библиографии экономической литературы EconLit. По данным 2006–2013 годов выявлены 62 новых направления экономических исследований на пересечении микрообластей классификатора JEL.

Таким образом, решение задачи многомерной классификации на больших объемах научных данных является весьма актуальным.

Состояние вопроса

Постановка задачи классификации выглядит следующим образом.

Пусть есть множество объектов X, заданных в многомерном пространстве признаков, и конечный набор классов $\{C_i \ ... \ C_n\}$. Известно, что каждый объект $x \in X$ относится к некоторому классу $C_j \in \{C_1, C_2, ... C_n\}$. Таким образом, необходимо построить правило (алгоритм) отображения каждого объекта из X к своему классу $T{:}X \to \{C_1, C_2, ... C_n\}$ с наибольшей точностью.

Учитывая, что реальное пространство признаков очень велико и разрежено, а самих объектов чрезвычайно много, то подгонка параметров алгоритма (его обучение) проводится по некоторому конечному подмножеству $X_L \subseteq X$, называемому обучающей выборкой.

Для решения задачи построения классификатора наилучшей точности, сохраняющего при этом хорошую обобщающую способность на больших данных, могут быть использованы различные алгоритмы. Рассмотрим несколько алгоритмов, хорошо зарекомендовавших себя при работе с большими данными.

Наивный байесовский классификатор (*NAÏVE BAYES, NB***)**. Наивный байесовский классификатор объединяет модель с правилом решения. Одно общее правило должно выбрать наиболее вероятную гипотезу; оно известно как апостериорное правило принятия решения (MAP). Соответствующий классификатор – это функция {classify}, определённая следующим образом [2]:

$$classify(f_1,\ldots,f_n)=$$

$$=\arg\max p(C=c)\prod_{i=1}^{n} p(F_i=f_i \mid C=c),$$

где $f_1, f_2, \ldots, f_n$ – переменные (характеристики объекта); C – класс.

То есть вероятность принадлежности объекта к конкретному классу рассчитывается как произведение вероятностей независимых событий о равенстве переменных конкретным значениям для класса.

К достоинствам алгоритма относятся такие свойства, как устойчивость к изолированным точкам шума; обработка как количественных, так и дискретных данных, быстрое вычисление и эффективное использование памяти, нечувствительность к нерелевантным переменным. При работе алгоритма возникают проблемы, если условная вероятность равна нулю. Ограничением использования алгоритма является предположение, что переменные независимы.

Дерево решений (*DT).* Дерево принятия является одним из самых простых, но в то же время прозрачных и эффективных алгоритмов для выполнения классификации. Дерево решений представляет правила в иерархическом и последовательном виде, где каждому атрибуту соответствует свой узел, на основе которого дается решение [2].

К достоинствам данного алгоритма относится то, что он не требует предобработки данных, обладает высокой скоростью работы, высокой точностью и прозрачностью. Недостатком является то, что слишком сложные конструкции могут не отображать реальной картины, ввиду чего точность может оказаться ниже, чем у более простой конструкции

Лес деревьев решений (*Random Forest, RF*). Само по себе дерево решений не обеспечивает достаточной точности для этой задачи, но отличается быстротой построения. Алгоритм *RF* обучает k решающих деревьев на параметрах, случайно выбранных для каждого дерева, после чего на каждом из тестов проводится голосование среди обученного ансамбля. В основе построения этого алгоритма лежит идея о том, что если

суммировать данные от большого количества различных слабых алгоритмов, сведя их в единый ответ, то результат, скорее всего, будет лучше, чем у одного мощного алгоритма [2].

Подобные алгоритмы обладают способностью эффективно обрабатывать данные с большим числом признаков и классов, мало чувствительны к масштабированию (и вообще к любым монотонным преобразованиям) значений признаков, одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки, обладают внутренней оценкой способности модели к обобщению (тест out-of-bag), возможностью параллельной реализации и масштабируемостью. Существуют методы построения деревьев по данным с пропущенными значениями признаков и оценивания значимости отдельных признаков в модели. В качестве недостатков следует отметить, что алгоритм склонен к переобучению на некоторых задачах, особенно на зашумленных задачах, и требует большого объема памяти для представления полученных результатов. В этом случае требуется $O(Nk)$ памяти для хранения модели, где k – число деревьев.

Совершенствование алгоритма классификации (*Adaptive Boosting, AdaBoost*).

Увеличение точности простых классификаторов может быть проведено путём комбинирования примитивных слабых классификаторов в один сильный алгоритм AdaBoost (от английских слов адаптивность и усиление) [7]. Под силой классификатора в данном случае подразумевается эффективность (качество) решения задачи классификации. Улучшение (Boosting) – это набор методов для порождения последовательности классификаторов, в которой каждый последующий классификатор пытается исправить ошибки предыдущих. Улучшение применяется к некоторому алгоритму классификации, называемому базовым классификатором (BaseLearner). Этот классификатор часто называют слабым классификатором. На каждом шаге алгоритм изменяет веса записей обучающей выборки, увеличивая веса неверно расклассифицированных записей.

Усиление происходит путем объединения их в комитет. Суть алгоритма заключается в комбинировании слабых классификаторов в один финальный, более мощный [7, 9]. Известно, что в качестве слабых классификаторов могут выступать практически любые, а использование комитетов почти всегда приводит к усилению результата [5]. В процессе обучения финального классификатора акцент делится на эталоны, которые распознаются хуже, т.е. выбирается классификатор, который лучше идентифицирует объекты, неверно распознанные предыдущим классификатором, в этом и заключается адаптивность алгоритма, в процессе обучения он подстраивается под наиболее сложные объекты.

Часто используются следующие способы построения комитетов алгоритмов [2, 4, 8].

1. Голосование по большинству (простое голосование), при котором комитет классификаторов относит объект к тому классу, к которому его отнесли большинство входящих в него алгоритмов.

2. Голосование по старшинству (машина покрывающих множеств). Этот метод предполагает последовательную одноклассовую классификацию. То есть первый алгоритм комитета отвечает за отнесение объекта к классу 1. Если он отказывается от классификации, то объект передается второму алгоритму, который может его отнести к классу 2. Если этого не произошло, объект передается к третьему классификатору и т.д., пока один из алгоритмов не примет решения.

3. Взвешенное голосование из смеси экспертов. В этом случае голос каждого из классификаторов T_i, входящих в комитет T, имеет свой вес α_i, зависящий от ошибки данного алгоритма на обучающем множестве:

$$T(x) = \sum_{i=1}^{m} \alpha_i T^i(x).$$

В работе [6] приводятся данные, что усложнение правил построения комитетов, от взвешенного голосования до нейронных сетей, не дает значимого эффекта. Наилучший результат, повышение точности на 7–10 %, дает обучение комитета.

К достоинствам алгоритма AdaBoost следует отнести простоту реализации, возможность идентифицировать объекты, являющиеся шумовыми выбросами, хорошую обобщающую способность. В реальных задачах (не всегда, но часто) удаётся строить композиции, превосходящие по качеству базовые алгоритмы. Обобщающая способность может улучшаться (в некоторых задачах) по мере увеличения числа базовых алгоритмов. Время построения композиции практически полностью определяется временем обучения базовых алгоритмов.

К недостаткам алгоритма относятся следующие. Во-первых, AdaBoost склонен к переобучению при наличии значительного уровня шума в данных. Проблема решается путём удаления выбросов или применения менее агрессивных функций потерь. Во-вторых, жадная стратегия последовательного добавления приводит к построению неоптимального набора базовых алгоритмов.

Для улучшения композиции можно периодически возвращаться к ранее построенным алгоритмам и обучать их заново. В-третьих, AdaBoost требует достаточно длинных обучающих выборок. Другие методы линейной коррекции, в частности бэггинг, способны строить алгоритмы сопоставимого качества по меньшим выборкам данных. В-четвертых, бустинг может приводить к построению громоздких композиций, состоящих из сотен алгоритмов. Такие композиции исключают возможность содержательной интерпретации, требуют больших объёмов памяти для хранения базовых алгоритмов и существенных затрат времени на вычисление классификаций.

Предлагаемый подход

Исходя из того, что каждый из описанных выше алгоритмов имеет свои плюсы и минусы по точности работы, наиболее рациональным решением можно считать сравнение результатов голосования каждого алгоритма и выбор того решения, за которое проголосовало большее количество алгоритмов.

Если за один класс проголосовало два и более алгоритма, то он признается правильным и фиксируется в системе. Если все алгоритмы проголосовали за разные классы, то приоритет отдается алгоритму лес деревьев по той причине, что он показал наибольшую точность на этапе тестирования алгоритмов (рис. 1).

Последовательность функций предложенного алгоритма следующая:

- загрузка и установка необходимых пакетов для обработки данных;
- импорт выборки данных и ее предобработка;
- выполнение кластеризации данных на основе алгоритма k-means;
- выполнение алгоритма дерево решений;
- выполнение алгоритма лес деревьев решений;
- выполнение алгоритма наивного байесовского классификатора;
- выполнение алгоритма голосующего комитета;
- оценка точности работы всех алгоритмов классификации.

Полученные результаты

Для тестирования алгоритмов классификации были использованы данные компании OttoGroup, которые имели обезличенный вид и находились в свободном доступе. Исходный файл с данными содержал 93 атрибута и 61878 объектов. Для формирования обучающей и тестовой выборки исходная выборка разбивалась в процентном соотношении 70:30.

```
#Запуск алгоритма DT
tree <- rpart(km.cluster ~ ., data=traindata, method="class")
DTpredictions <- predict(tree, testdata, type = "class")

#Построения графика DT
prp (tree, extra=7)
fancyRpartPlot(tree, cex=0.5)

#Запуск алгоритма RF
random <- randomForest(km.cluster ~ ., data = traindata, importance=TRUE, ntree=10)
RFpredictions <- predict(random, testdata)

#Определение важности атрибутов RF
check <- as.data.frame(importance(random))
names(check)[names(check)=="MeanDecreaseGini"] <- "MeanDecreaseNodeImpurity"
plot(check$MeanDecreaseNodeImpurity, xlab="Атрибуты", ylab="Относительная важность")
check <- check[order(-check$MeanDecreaseNodeImpurity),]

#Запуск алгоритма NB
naive <- naiveBayes(km.cluster ~ ., data = traindata)
NBpredictions <- predict(naive, testdata, type="class")

#Подготовка данных для голосования
vote <- data.frame(RFlabel = RFpredictions, NBlabel = NBpredictions, DTlabel = DTpredictions)

#Выбор лучшего значения на основе голосования
vote$PriorRF <- ifelse (vote$RFlabel == vote$DTlabel,
                        vote$DTlabel,

                        ifelse (vote$RFlabel == vote$NBlabel,
                                vote$NBlabel,

                                ifelse (vote$DTlabel == vote$NBlabel,
                                        vote$NBlabel,
                                        vote$RFlabel)))
```

Рис. 1. Алгоритм голосований на языке R

Был проведен ряд экспериментов для определения точности классификации алгоритмов на различных объемах данных и для различных классов.

Параметры для тестирования имели следующий вид. Объем выборки составляли малые наборы (100 и 250), средние наборы (500 и 1500), большие наборы (500 и 15000). Количество классов составляло 4, 6 и 8. На основе указанных данных был проведен полнофакторный эксперимент.

Сводный результат по всем экспериментам представлен в табл. 1. Темным фоном выделены ячейки полученных результатов с большей точностью. Используемые обозначения алгоритмов:

- RF – лес деревьев решений;
- NB – наивный байесовский классификатор;
- DT – дерево решений;
- VT – комитет голосующих алгоритмов.

Полученные результаты классификации представлены на рис. 2 и 3. Из визуального представления полученных результатов классификации хорошо видно, что алгоритм лес деревьев решений лучше остальных отработал на большом наборе данных и на малом количестве классов (рис. 2). В свою очередь, комитет голосующих алгоритмов показал лучшие результаты на малых и средних наборах данных, а также на большом количестве классов (рис. 3).

Результаты сравнительного анализа точности полученных решений представлены в табл. 2. Исследование проводилось на выборках объемом 100 и 4 классах.

Таблица 1

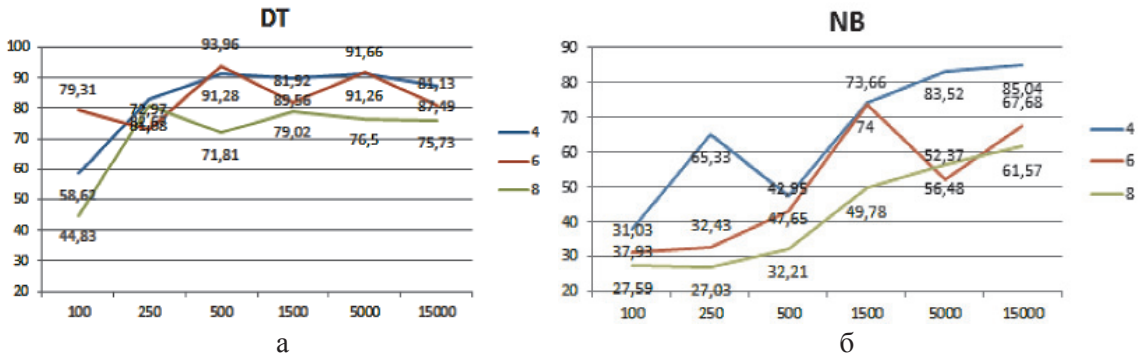
Сводная таблица по всем экспериментам

DT			
Выборка	Классов		
	4	6	8
100	58,62	79,31	44,83
250	82,67	72,97	81,08
500	91,28	93,96	71,81
1500	89,56	81,92	79,02
5000	91,26	91,66	76,5
15000	87,49	81,13	75,73

NB			
Выборка	Классов		
	4	6	8
100	37,93	31,03	27,59
250	65,33	32,43	27,03
500	47,65	42,95	32,21
1500	74	73,66	49,78
5000	83,52	52,37	56,48
15000	85,04	67,68	61,57

RF			
Выборка	Классов		
	4	6	8
100	72,41	72,41	68,97
250	89,33	85,14	81,08
500	94,63	90,6	81,88
1500	93,56	88,39	86,16
5000	94,4	95	87,85
15000	94,09	92,15	89,35

VT			
Выборка	Классов		
	4	6	8
100	62,07	82,76	68,97
250	86,67	86,49	82,43
500	96,64	93,29	81,88
1500	93,33	87,72	87,05
5000	94,13	95,46	87,05
15000	93,98	89,78	88,02



а б

Рис. 2. Результат классификации алгоритма:
а – дерево решений; б – наивный байесовский классификатор

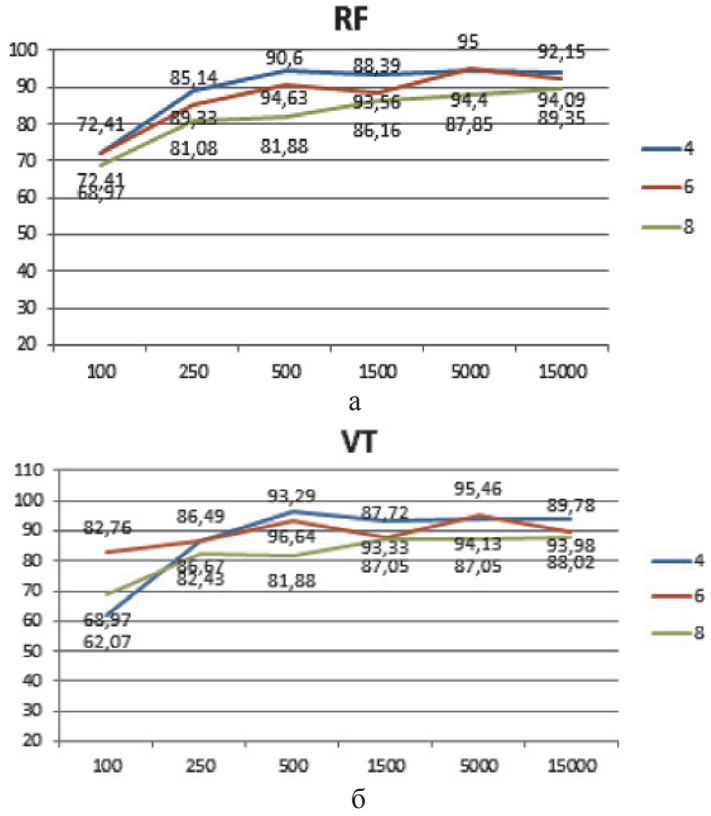


Рис. 3. Результат классификации алгоритма:
а – лес деревьев решений; б – комитет голосующих алгоритмов

Таблица 2

Точность работы алгоритмов на выборки объемом 100 и 4 классов

№ п/п	Алгоритм	Точность (%)
1	RF	85,97
2	NB	71,62
3	DT	82,09
4	VT	90,96

Как видно из представленных выше результатов, в зависимости от объема выборки и количества классов, различные алгоритмы показывают различную точность. Несмотря на то, что хуже всех отработал наивный байесовский классификатор, который сильно отставал от лидера на малых наборах, он существенно улучшил точность на больших выборках, показывая точность до 85 %. Алгоритмы лес случайных деревьев и комитет голосующих алгоритмов показали хорошую точность почти на всех объемах данных, за исключением малых объемов, но почти у всех алгоритмов заметны серьезные проблемы с точностью на данном наборе. Дальнейшее повышение точности возможно при обучении комитета голосующих алгоритмов.

Заключение

Для повышения точности при решении задачи классификации в данной работе было предложено использовать комитет алгоритмов (наивный Байес, случайные деревья, дерево решений), где при голосовании используется модернизированное большинство. Для исследования данного метода было написано программное приложение на языке R, использование которого показало, что на больших выборках метод имеет преимущества перед подходом AdaBoost, так как дает точность 90 % и выше. Причем точность голосующих алгоритмов лежит в диапазоне 71–86 %.

Для дальнейшего усиления алгоритма, по нашему мнению, могут использоваться два направления: обучение комитета алгоритмов и введение весов атрибутов в пространстве признаков, описывающие

классифицируемые объекты. Подобное динамическое изменение весов хорошо согласуется как с природой научных направлений (классифицируемых объектов), так и с моделью жизненного цикла научного направления.

Работа поддержана грантом РФФИ № 15-07-08742.

Список литературы

1. Лычагин М.В., Мкртчян Г.М., Лычагин А.М., Попов И.Ю. Новое исследование инноваций в 2006–2013 годах: библиометрический анализ на основе EconLit // Вестник Новосибирского государственного университета. Серия: Социально-экономические науки. – 2014. – Т. 14, Вып. 3. – С. 150–162.

2. Мазуров В.Д. Метод комитетов в задачах оптимизации и классификации. – М.: Наука, 1990.

3. Несветайлов Г.А. Научные кадры: возраст и творчество // Социологические исследования. – 1998. – № 9.– С. 115–119.

4. Нессонова М.Н. Метод рейтингового голосования комитета алгоритмов в задачах классификации с учителем // Запорожский медицинский журнал. – 2013. – № 1 (76). – С. 101–102

5. Никулин В.Н., Палешева С.А., Зубарева Д.С. Об однородных ансамблях при использовании метода бустинга в приложении к классификации несбалансированных данных // Вестник пермского университета. Серия: Экономика. – 2012. – № 4. – С. 8–14

6. Попов А.К., Трофимов А.Г. Применение комитетов классификаторов в задаче классификации многомерных динамических данных // Новый университет. – 2012. – № 4(10). – С. 34–37.

7. Boosting the margin: a new explanation for the effectiveness of voting methods / R.E. Schapire, Y. Freund, W.S. Lee, P. Bartlett // Annals of Statistics. – 1998. – Vol. 26, № 5. – P. 1651–1686.

8. Breiman, L. 2000. Some infinity theory for predictor ensembles. Technical Report 579, Statistics Dept. UCB.

9. Grove A., Schuurmans D. (1998). Boosting in the limit: Maximizing the margin of learned ensembles // In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98).

References

1. Lychagin M.V., Mkrtchyan G.M., Lychagin A.M., Popov I.Yu. *Vvestnicc Novosibirskogo gosugarstvennogo universiteta. Seria; Sosialno-economicheskie nauki*, 2014, no. 14(3), pp. 150–162.

2. Mazurov V.D. *Metod komitetov v zadachah optimizatsii I klassifikatsii* [Method of committees in optimization and classication]. Moscow, Nauka, 1990.

3. Nesvetailov G.A. *Sotziologicheskie issledovania* [Sociological Studies], 1989, no. 9, pp. 115–119.

4. Nessonova M.N. *Zaporogskiy mediztinskiy jurnal,* 2013, no. 1(76), pp. 101–102.

5. Nikulin V.N., Palesheva S.A., Zubarev D.S. *Vestnic permskogo universiteta. Seria; economika,* 2012, no. 4, pp. 8–14.

6. Popov A.K., Trofimov A.G. *Noviy universitet,* 2012, no. 4(10), pp. 34–37.

7. Schapire R.E., Freund Y., Lee W.S., Bartlet P. *Annals of Statistics*, 1998, vol. 26, no. 5, pp. 1651–1686.

8. Breiman, L. *Technical Report 579, Statistics Dept. UCB,* 2000.

9. Grove A., Schuurmans D. *In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98),* 1998.

Рецензенты:

Путилов А.В., д.т.н., профессор, декан факультета управления и экономики высоких технологий, Национальный исследовательский ядерный университет «МИФИ», г. Москва;

Ромашкова О.Н., д.т.н., профессор, зав. кафедрой прикладной информатики, Московский городской педагогический университет, г. Москва.