Классификация данных в образовательной сфере с применением технологий интеллектуального анализа[1]

Л. А. Демидова, М. М. Егин, Ю. С. Соколова

Рязанский государственный радиотехнический университет
390005, Рязань, ул. Гагарина, 59/1

e-mail: *liliya.demidova@rambler.ru, eginmm@gmail.com, JuliaSokolova62@yandex.ru*

Аннотация. В статье рассматривается задача классификации данных в образовательной сфере в контексте предсказания успешности прохождения итоговой государственной аттестации выпускниками средней школы. Такие данные могут быть несбалансированными. Для решения этой задачи предлагается использовать SVM-классификаторы на основе модифицированного PSO-алгоритма, реализующего одновременный поиск типа функции ядра, значений параметров функции ядра и значения параметра регуляризации. Для восстановления баланса классов предлагается применять стратегии сэмплинга на основе SMOTE-алгоритма. Анализ результатов классификации с использованием SVM-классификаторов на основе модифицированного PSO-алгоритма и стратегий сэмплинга и результатов, полученных в статистических пакетах программ, свидетельствует о целесообразности применения предлагаемого инструментария к решению задачи классификации данных в образовательной сфере.
Ключевые слова: SVM-алгоритм, классификация, алгоритм роя частиц, PSO-алгоритм, SMOTE-алгоритм.

1. Введение

В перечне задач интеллектуального анализа данных особое место занимает задача классификации, решение которой необходимо, например, в сфере кредитного скоринга, в области медицинской диагностики, при осуществлении категоризации текстов, при идентификации изображений лиц и т. п. Адекватное решение этой задачи востребовано и в образовательной сфере. В последние годы студенты высшей школы и ученики (выпускники) средней школы активно привлекаются к участию в разнообразных опросах и тестах, в том числе с привлечением широко апробированных методик, с целью оценки их интеллектуального уровня, индивидуально-психологических особенностей, профилирования по специальностям и пр.

[1] Работа выполнена при поддержке гранта РФФИ № 17-29-02198.

Например, для оценки готовности первокурсников может быть выполнена диагностика мотивационного компонента [1, 2], диагностика когнитивного компонента с помощью теста интеллекта Р. Амтхауэра [3], позволяющего осуществить оценку вербального, математического и пространственного интеллекта, диагностика личностного компонента с использованием пятифакторного личностного опросника, позволяющего осуществить оценку степени выраженности личностных качеств по пяти факторам (интроверсия — экстраверсия; эмоциональная устойчивость — нейротизм; закрытость новому опыту — открытость; несобранность — сознательность; враждебность — доброжелательность) [4]. Еще одной актуальной задачей анализа данных в образовательной сфере является задача диагностики готовности выпускников средней школы к прохождению государственной итоговой аттестации, для решения которой, в частности, привлекаются данные, консолидирующие в себе информацию по самодиагностике выпускника, а также сведения о его ближайшем окружении и комфортности среды обитания. Эта задача может быть рассмотрена как задача прогнозирования, будет ли результат государственной итоговой аттестации высокобалльным.

Очевидно, что результаты опросов и тестов, накопленные в больших объемах, могут быть использованы для извлечения дополнительной скрытой в них информации, в частности, для выявления тех или иных причинно-следственных связей и взаимосвязей в контексте диагностики личности выпускника и для разработки классификаторов.

В настоящее время для решения прикладных задач, требующих анализа данных различной природы, используются разнообразные алгоритмические средства, среди которых наиболее известны линейная и логистическая регрессии, байесовский классификатор, деревья решений, решающие правила, нейронные сети, алгоритм k ближайших соседей (kNN-алгоритм, k Nearest Neighbors Algorithm), алгоритм опорных векторов (SVM-алгоритм, Support Vector Machine Algorithm) и т. п. При этом с точки зрения представляемых возможностей и неоспоримых достоинств, декларируемых в работах научного сообщества, в контексте решении задач анализа данных в образовательной сфере наиболее перспективным представляется использование SVM-алгоритма. Предлагается использовать SVM-классификаторы на основе модифицированного PSO-алгоритма, адаптированные к специфическим особенностям проблемы анализа данных в образовательной сфере.

SVM-алгоритм (Support Vector Machines, SVM) успешно используется для разработки SVM-классификаторов [5, 6]. SVM-классификатор использует функцию ядра для определения гиперплоскости, разделяющей классы данных. Удовлетворительное качество обучения и тестирования разработанного SVM-классификатора позволяет использовать его для классификации новых объектов.

Задача поиска оптимальных значений параметров SVM-классификатора является весьма актуальной. При этом необходимо выбрать тип функции ядра, значения параметров функции ядра и значение параметра регуляризации [5–7]. Невозможно обеспечить высокую точность классификации данных с использованием SVM-классификатора без адекватного решения этой задачи. В простейшем случае решение этой задачи может быть найдено перебором типов функций ядра, значений параметров функции ядра и значения параметра регуляризации, что требует значительных временных затрат. Градиентные методы оптимизации не могут быть использованы для поиска оптимального решения этой задачи, в то время как стохастические алгоритмы оптимизации, в частности генетический алгоритм (genetic algorithm, GA), алгоритм колонии роя пчел (artificial bee colony algorithm, ABC algorithm), алгоритм роя частиц (particle swarm algorithm, PSO algorithm), позволяют решить такую задачу при приемлемых временных затратах.

PSO-алгоритм является простейшим алгоритмом поисковой оптимизации. Традиционный подход к применению PSO-алгоритма при разработке SVM-классификатора заключается в выполнении этого алгоритма для фиксированного типа функции ядра с целью выбора оптимальных значений параметров функции ядра и значения параметра регуляризации с последующим выбором лучшей комбинации этих значений, соответствующей некоторому лучшему ядру. Предлагается использовать модифицированный PSO-алгоритм для одновременного поиска типа функции ядра, значений параметров функции ядра и значения параметра регуляризации.

Результаты опроса, применяемые для формирования наборов данных для обучения и тестирования SVM-классификатора, могут быть существенно несбалансированы. Это может значительно ухудшить качество разработанного SVM-классификатора и снизить значения его показателей качества.

В настоящее время для решения проблемы несбалансированности наборов данных применяются различные стратегии сэмплинга.

В данной статье предлагается использовать алгоритм синтетического сэмплинга — SMOTE-алгоритм (Synthetic Minority Oversampling Technique) для восстановления баланса между классами.

В частности, планируется исследовать возможности таких вариантов этого алгоритма, как «regular», «borderline1», «borderline2» и «SVM».

Следует отметить наличие реализаций SVM-алгоритма в ряде статистических пакетов программ, в частности, в пакетах Statistica StatSoft и IBM SPSS Modeler. При этом в IBM SPSS Modeler имеются и некоторые реализации SMOTE-алгоритма. Однако эти реализации являются недостаточно гибкими и не позволяют

изменять настройки своих параметров так, как это может потребоваться разработчику (исследователю).

Таким образом, задача разработки SVM-классификаторов на основе модифицированного PSO-алгоритма и различных стратегий сэмплинга в контексте предсказания успешности прохождения итоговой государственной аттестации выпускниками средней школы является весьма актуальной, а ее адекватное решение — востребованным.

2. Теоретическая часть

Пусть имеется экспериментальный набор вида: $\{(z_1, y_1), ..., (z_s, y_s)\}$, в котором каждому объекту $z_i \in Z$ поставлено в соответствие число $y_i \in Y = \{-1; +1\}$, принимающее значение −1 или +1, в зависимости от того, какому классу принадлежит объект z_i. При этом предполагается, что каждому объекту z_i поставлен в соответствие q-мерный вектор числовых значений характеристик $z_i = (z_i^1, z_i^2, ..., z_i^q)$ (обычно нормированный значениями из отрезка [0, 1]), где z_i^l — числовое значение l-й характеристики для i-го объекта $(i = \overline{1,s},\ l = \overline{1,q})$ [5–7]. При построении SVM-классификатора с помощью специальной функции $\kappa(z_i, z_\tau)$, называемой ядром, определяется классифицирующая функция (классификатор) $F : Z \to Y$, сопоставляющая классу $Y = \{-1; +1\}$ произвольный объект из Z.

Для обучения SVM-классификатора необходимо определить тип функции ядра $\kappa(z_i, z_\tau)$, значения параметров ядра и значение параметра регуляризации C, позволяющего найти компромисс между максимизацией ширины полосы, разделяющей классы, и минимизацией суммарной ошибки. При этом в качестве функции ядра $\kappa(z_i, z_\tau)$, позволяющей разделить объекты разных классов, обычно используются линейная, полиномиальная, радиальная базисная и сигмоидная функции [5–7].

При разработке SVM-классификатора необходимо реализовать многократное обучение и тестирование на разных случайным образом сформированных обучающем и тестовом наборах, состоящих соответственно из S и $s - S$ элементов ($s > S$), с последующим определением лучшего SVM-классификатора в смысле обеспечения максимально возможного качества классификации, оценка которого может быть выполнена с применением различных показателей качества классификации [8, 9]. Если качество обучения и тестирования является приемлемым, то SVM-классификатор может быть применен для классификации новых объектов.

В результате обучения определяется классифицирующая функция [5–10]:

$$f(z) = \sum_{i=1}^{S} \lambda_i y_i \kappa(z_i, z) + b.$$

Классификационное решение, сопоставляющее объект z классу с меткой «−1» или «+1», принимается в соответствии с правилом [5–10]:

$$F(z) = sign(f(z)) = sign\left(\sum_{i=1}^{S} \lambda_i \cdot y_i \cdot \kappa(z_i, z) + b\right).$$

В результате обучения SVM-классификатора определяются опорные векторы, которые находятся ближе всего к гиперплоскости, разделяющей классы, и несут всю информацию о разделении классов [5–7].

2.1. Модифицированный PSO-алгоритм

Использование модифицированного PSO-алгоритма обеспечивает лучшую точность классификации посредством подбора типа функции ядра, значений параметров функции ядра и значения параметра регуляризации. Кроме того, модифицированный PSO-алгоритм позволяет уменьшить временные затраты на разработку SVM-классификатора [11–13]. Качество SVM-классификатора может быть оценено с применением различных показателей качества классификации [9, 13].

При реализации традиционного PSO-алгоритма n-мерное пространство поиска (n — количество параметров, подлежащих оптимизации) населяется роем из m агентов-частиц (элементарных решений). Положение (позиция) i-й частицы задается вектором $x_i = (x_i^1, x_i^2, ..., x_i^n)$, который определяет некоторый набор значений параметров оптимизации. При этом в аналитической записи целевой функции $u(x)$ алгоритма оптимизации (оптимум — это, например, минимум которой необходимо найти) такие параметры могут как присутствовать в явном виде, так и отсутствовать.

Для каждой позиции n-мерного пространства поиска, в которой побывала i-я частица $(i = \overline{1,m})$, выполняется вычисление значения целевой функции $u(x)$. При этом каждая i-я частица запоминает, какое лучшее значение целевой функции лично она нашла, а также координаты позиции в n-мерном пространстве, соответствующие этому значению целевой функции. Кроме того, каждая i-я частица $(i = \overline{1,m})$ «знает», где расположена позиция, являющаяся лучшей (с точки зрения достижения оптимума целевой функции) среди всех позиций, которые «разведали» (опробовали) частицы — благодаря этому имитируется мгновенный обмен информацией между всеми частицами роя. На каждой итерации частицы корректируют свою скорость, так, чтобы, с одной стороны, быть поближе к лучшей позиции, которую частица нашла сама, и в то же время приблизиться к позиции, которая в данный момент является глобально лучшей (среди совокупности позиций, найденных всеми частицами). Через некоторое количество итераций частицы должны собраться вблизи наиболее хорошей позиции (глобально лучшей по результатам всех итера-

ций). Однако возможно, что часть частиц роя останется где-то в относительно неплохом локальном оптимуме или нескольких таких оптимумах.

Сходимость PSO-алгоритма зависит от того, каким образом выполняется коррекция векторов скоростей частиц. Известны различные подходы к выполнению коррекции вектора скорости v_i для i-й частицы $(i=\overline{1,m})$. В традиционном PSO-алгоритме коррекция j-й координаты вектора скорости $(j=\overline{1,n})$ для i-й частицы $(i=\overline{1,m})$ производится по формуле [9, 13]

$$v_i^j = v_i^j + \hat{\varphi}\hat{r}(\hat{x}_i^j - x_i^j) + \tilde{\varphi}\tilde{r}(\tilde{x}^j - x_i^j), \quad (1)$$

где v_i^j — j-я координата вектора скорости i-й частицы; x_i^j — j-я координата вектора x_i, задающего позицию i-й частицы; $\hat{x}_i^j$ — j-я координата вектора лучшей позиции, найденного i-й частицей за все время ее существования; $\tilde{x}^j$ — j-я координата глобально лучшей позиции всего роя частиц, в которой целевая функция имеет оптимальное значение; $\hat{r}$ и $\tilde{r}$ — случайные числа в интервале (0, 1), которые вносят элемент стохастичности в процесс поиска; $\hat{\varphi}$ и $\tilde{\varphi}$ — личный и глобальный коэффициенты ускорения частиц — они являются константами и определяют поведение и эффективность PSO-алгоритма в целом.

Коррекция каждой j-й координаты вектора скорости $(j=\overline{1,n})$ i-й частицы $(i=\overline{1,m})$ производится в соответствии с формулой [9, 13]:

$$v_i^j = \chi \cdot [v_i^j + \hat{\varphi} \cdot \hat{r} \cdot (\hat{x}_i^j - x_i^j) + \tilde{\varphi} \cdot \tilde{r} \cdot (\tilde{x}_i^j - x_i^j)], \quad (2)$$

где χ — коэффициент сжатия;

$$\chi = 2K / |2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|, \quad (3)$$

$$\varphi = \hat{\varphi} + \tilde{\varphi}, \ (\varphi > 4),$$

K — некоторый масштабирующий коэффициент, принимающий значения из интервала (0, 1).

Пусть коррекция вектора скорости i-й частицы $(i=\overline{1,m})$ выполнена в соответствии с формулой (1) или (2). Тогда коррекция j-й координаты позиции i-й частицы $(i=\overline{1,m})$ выполняется по формуле $x_i^j = x_i^j + v_i^j$.

Далее для каждой i-й частицы $(i=\overline{1,m})$ рассчитывается новое значение целевой функции $u(x_i)$ и выполняется проверка: не стала ли новая позиция с вектором координат x_i лучшей среди всех позиций, в которых i-я частица ранее побывала. Если новая позиция i-й частицы признается лучшей для нее на текущий момент времени, то информация об этой позиции сохраняется в векторе $\hat{x}_i$ $(i=\overline{1,m})$ — с

«запоминанием» значения целевой функции $u(x_i)$ в этой позиции. Затем среди всех новых позиций частиц роя осуществляется проверка на наличие глобально лучшей позиции. Если некоторая новая позиция, соответствующая одной из частиц роя, признается глобально лучшей на текущий момент времени, то информация о ней сохраняется в векторе $\tilde{x}$ — с «запоминанием» значения целевой функции $u(x_i)$ в этой позиции.

В случае использования PSO-алгоритма при разработке SVM-классификатора частицам роя могут быть сопоставлены векторы, описывающие их позиции в пространстве поиска и закодированные параметрами функции ядра и параметром регуляризации: (x_i^1, x_i^2, C_i), где i — номер частицы $(i=\overline{1,m})$; x_i^1, x_i^2 — параметры функции ядра i-й частицы [при этом параметр x_i^1 полагается равным параметрам функций ядра d, σ или k_2 (в зависимости от того, какому типу функции ядра соответствует частица роя); параметр x_i^2 полагается равным параметру функций ядра k_2, если частица роя соответствует сигмоидному типу функции ядра, в противном случае значение этого параметра считается равным нулю]; C_i — параметр регуляризации [9, 13]. В результате для каждого типа T функции ядра, участвующего в поиске, будет определена частица с оптимальной комбинацией значений параметров $(\tilde{x}^1, \tilde{x}^2, \tilde{C})$, обеспечивающая высокое качество классификации [9, 13]. Лучший тип и лучшие значения соответствующих ему параметров определяются по результатам сравнительного анализа лучших частиц, полученные при реализации PSO-алгоритма с фиксированным типом функции ядра.

Наряду с традиционным подходом к применению PSO-алгоритма при разработке SVM-классификатора предлагается применять новый подход, реализующий одновременный поиск лучшего типа функции ядра $\tilde{T}$, значений параметров $\tilde{x}^1$ и $\tilde{x}^2$ функции ядра и значения параметра регуляризации $\tilde{C}$ [9, 13]. При таком подходе каждой i-й частицы роя $(i=\overline{1,m})$ соответствует вектор, описывающий ее позицию в пространстве поиска: (T_i, x_i^1, x_i^2, C_i), где T_i — номер типа функции ядра (например, 1, 2, 3 — для полиномиальной, радиальной базисной и сигмоидной функций соответственно); параметры x_i^1, x_i^2, C_i определяются аналогично предыдущему случаю. При этом возможно «перерождение» частицы — изменение ее координаты T_i на номер того типа функции ядра, частицы которого показывают максимально высокое качество классификации. При «перерождении» возможно изменение значений параметров x_i^1, x_i^2 и C_i так, чтобы они соответствовали новому типу функции ядра (с учетом диапазонов изменения их значений). Частицы, кото-

рые не подверглись «перерождению», осуществляют движение в своем собственном пространстве поиска (некоторой размерности).

Доля частиц, участвующих в «перерождении» определяется перед запуском алгоритма, ее рекомендуется выбирать от 15 до 25%.

Предложенная модификация PSO-алгоритма может быть представлена следующей последовательностью шагов [9, 13].

Шаг 1. Определить параметры PSO-алгоритма: количество частиц в рое m, масштабирующий коэффициент для скорости K, личный и глобальный коэффициенты ускорения $\hat{\varphi}$ и $\tilde{\varphi}$, максимальное количество итераций PSO-алгоритма $N_{\max}$. Определить типы T функций ядра, участвующие в поиске ($T = 1$ — полиномиальная, $T = 2$ — радиальная базисная, $T = 3$ — сигмоидная функция ядра) и границы изменения параметров функции ядра и параметра регуляризации C для выбранных типов функций ядра T: $x_{\min}^{1T}$, $x_{\max}^{1T}$, $x_{\min}^{2T}$, $x_{\max}^{2T}$, $C_{\min}^{T}$, $C_{\max}^{T}$ ($x_{\min}^{2T} = 0$ и $x_{\max}^{2T} = 0$ для $T = 1$ и $T = 2$). Определить процент «перерождения» частиц p.

Шаг 2. Задать равное количество частиц для каждого ядра T, включенного в поиск. Затем для каждой i-й частицы $(i = \overline{1,m})$ инициализировать координату T_i (так, чтобы каждому используемому в процессе поиска типу функции ядра соответствовало одинаковое количество частиц). Остальные координаты i-й частицы $(i = \overline{1,m})$ сгенерировать случайным образом из соответствующих диапазонов: $x_i^1 \in [x_{\min}^{1T}, x_{\max}^{1T}]$, $x_i^2 \in [x_{\min}^{2T}, x_{\max}^{2T}]$, ($x_i^2 = 0$ при $T = 1$ и $T = 2$), $C_i \in [C_{\min}^{T}, C_{\max}^{T}]$. Инициализировать случайный вектор скорости $v_i(v_i^1, v_i^2, v_i^3)$ i-й частицы $(i = \overline{1,m})$ ($v_i^2 = 0$ при $T = 1$ и $T = 2$). Принять начальное положение i-й частицы $(i = \overline{1,m})$ за лучшее ее известное положение $(\hat{T}_i, \hat{x}_i^1, \hat{x}_i^2, \hat{C}_i)$ и определить лучшую частицу с вектором координат $(\tilde{T}, \tilde{x}^1, \tilde{x}^2, \tilde{C})$ среди всех m частиц, а также лучшую частицу для каждого типа функции ядра T, включенного в поиск, с вектором координат $(\overline{T}, \overline{x}^{1T}, \overline{x}^{2T}, \overline{C}^{T})$. При этом количество выполненных итераций полагается равным 1.

Шаг 3. Пока количество выполненных итераций не превышает заданное число $N_{\max}$ выполнять такие операции:

– «перерождение» частиц: из тех частиц, у которых координата $T_i \neq \tilde{T}$ $(i = \overline{1,m})$, выбрать p% частиц, показавших самое низкое качество классификации, и изменить значение координаты T_i с номером типа функции ядра на значение $\tilde{T}$; изменить значения параметров x_i^1, x_i^2, C_i «перерождаемой» частицы так, чтобы они соответствовали новому типу ядра $\tilde{T}$ (т. е. попадали в соответствующие диапазоны);

– выполнить коррекцию вектора скорости $v_i(v_i^1, v_i^2, v_i^3)$ и положения (x_i^1, x_i^2, C_i) i-й частицы $(i=\overline{1,m})$ по формулам:

$$v_i^j = \begin{cases} \chi[v_i^j + \hat{\varphi}\cdot\hat{r}\cdot(\hat{x}_i^j - x_i^j) + \tilde{\varphi}\cdot\tilde{r}\cdot(\bar{x}_i^{jT} - x_i^j)], \ j=1,2, \\ \chi[v_i^j + \hat{\varphi}\cdot\hat{r}\cdot(\hat{C}_i - C_i) + \tilde{\varphi}\cdot\tilde{r}\cdot(\bar{C}^T - C_i)], \ j=3, \end{cases} \quad (4)$$

$$x_i^j = x_i^j + v_i^j \text{ для } j=1,2,$$

$$C_i = C_i + v_i^3,$$

где $\hat{r}$ и $\tilde{r}$ — случайные числа в интервале (0, 1); χ — коэффициент сжатия, рассчитанный по формуле (3); $\bar{x}^{1T}$, $\bar{x}^{2T}$, $\bar{C}^T$ — координаты частицы, лучшей для типа функции ядра $T=T_i$; при этом формула (4) является модификацией формулы (2): вместо значений координат глобально лучшей частицы $\tilde{x}^1$, $\tilde{x}^2$, $\tilde{C}$ используются значения $\bar{x}^{1T}$, $\bar{x}^{2T}$, $\bar{C}^T$;

– произвести расчет точности SVM-классификатора со значениями параметров (T_i, x_i^1, x_i^2, C_i) $(i=\overline{1,m})$ — с целью поиска оптимальной комбинации $(\tilde{T}, \tilde{x}^1, \tilde{x}^2, \tilde{C})$, обеспечивающей высокое качество классификации;

– увеличить количество итераций на «1».

После выполнения данного алгоритма будет определена частица с оптимальной комбинацией значений параметров $(\tilde{T}, \tilde{x}^1, \tilde{x}^2, \tilde{C})$, обеспечивающая высшее качество классификации на включенных в поиск типах функций ядра.

В модифицированном PSO-алгоритме у частиц происходит изменение координаты (значения), отвечающей за номер функции ядра. Поэтому после выполнения данного алгоритма может оказаться, что все частицы будут сосредоточены в пространстве поиска, которое соответствует функции ядра с максимально высоким качеством классификации. При этом остальные пространства поиска окажутся пустыми, так как у всех частиц, принадлежавших этим пространствам изначально, произойдет «перерождение» координаты с номером (значения) типа функции ядра. В ряде случаев (при небольших значениях количества итераций $N_{\max}$ или процента «перерождения» частиц p) также возможно, что у некоторых частиц «перерождение» ядер не произойдет, и они останутся в своем пространстве поиска.

Использование модифицированного PSO-алгоритма в задаче разработки SVM-классификатора позволяет снизить временные затраты на построение искомого SVM-классификатора. Модифицированный PSO-алгоритм может быть использован при разработке ансамблей SVM-классификаторов, SVM-классификаторов с при-

влечением ансамблей алгоритмов кластеризации, каскадных SVM-классификаторов и т. п. [14–16].

2.2. The SMOTE алгоритм и его варианты

Набор данных считается несбалансированным, если представленные в нем классы не равны по числу объектов. SMOTE-алгоритм [17] реализует подход оверсэмплинга к миноритарному классу: он создает искусственные объекты миноритарного класса на основе сходства объектов в пространстве характеристик с помощью алгоритма *k*-ближайших соседей (kNN-алгоритма). При этом искусственно созданные объекты не дублируют объекты миноритарного класса.

В настоящее время наиболее известны такие варианты SMOTE-алгоритма, как «regular» [17], «borderline1», «borderline2» [19] и «SVM» [19].

Вариант «regular» соответствует классическому SMOTE-алгоритму [17]. Варианты «borderline1», «borderline2» и «SVM» определяют шумовые объекты и объекты, лежащие на границе между классами. В таком случае перед генерированием искусственных объектов с использованием kNN-алгоритма реализуется поиск *m* ближайших соседей с целью определения принадлежности объекта к шумовым или лежащим на границе.

Варианты «borderline1» и «borderline2» генерируют искусственные объекты вблизи объектов, лежащих на границе классов. Эти варианты осуществляют поиск объектов, находящихся «в опасности».

В варианте «borderline1» осуществляется поиск m ближайших соседей для каждого объекта миноритарного класса. Миноритарные объекты, для которых все ближайшие *m* соседей являются объектами мажоритарного класса, удаляются, так как считаются шумовыми. Объекты, у которых число ближайших соседей из мажоритарного класса не больше, чем *m*/2, считаются «в безопасности» и не используются для генерирования новых объектов

Объекты, для которых число ближайших соседей из мажоритарного класса больше, чем *m*/2, считаются «в опасности» (вблизи границы) и используются для генерирования искусственных объектов. Искусственно созданные объекты миноритарного класса генерируются вдоль линий, соединяющих исходные объекты-соседи миноритарного класса.

В «borderline2» используется схожий с «borderline1» подход. Различие состоит в том, что в «borderline2» синтетические объекты генерируются из ближайших объектов как миноритарного, так и из мажоритарного классов. При этом искусственные объекты, созданные на основе мажоритарных, располагаются ближе к миноритарным чем те, которые были сгенерированы на основе миноритарных соседей.

Вариант «SVM» применяет SVM-классификатор к набору данных и использует опорный вектор для определения понятия «граница». В варианте «regular» для определения понятия «граница» используется соотношение числа соседей, принадлежащих разным классам.

Вариант «regular» не производит поиск объектов «в опасности», а создает искусственные объекты с помощью kNN-алгоритма без предварительной фильтрации.

3. Экспериментальные исследования

Для разработки SVM-классификаторов были использованы результаты опроса 546 учеников (выпускников) средней школы перед сдачей единого государственного экзамена (ЕГЭ), а также их тестовые баллы ЕГЭ по дисциплинам «Русский язык» и «Математика» [20].

Разделы опросника можно разбить на 3 следующие группы.

1. Общие вопросы (вопросы, касающиеся планов ученика после сдачи экзамена; вопросы, относительно того, что ученик считает важным для поступления; вопросы, позволяющие оценить отношение ученика к сдаваемым предметам; вопросы, позволяющие оценить взаимоотношение ученика с окружающими людьми, материальное положение семьи и т. п.).

2. Вопросы, связанные с дисциплиной «Русский язык».

3. Вопросы, связанные с дисциплиной «Математика».

Система ЕГЭ подразумевает перевод первичных баллов экзамена в тестовые, которые выставляются по стобалльной шкале в результате процедуры шкалирования, учитывающей все статистические материалы, полученные в рамках проведения ЕГЭ данного года. Шкалирование позволяет объективно сравнить и оценить уровень подготовленности выпускников. Именно в тестовых баллах предоставляются результаты для поступления в учебные заведения среднего профессионального и высшего образования. Дисциплины «Русский язык» и «Математика» являются обязательными к сдаче для получения аттестата о среднем образовании. При этом должны быть преодолены соответствующие дисциплинам минимальные пороги, задаваемые в баллах и определяемые заранее.

С учетом вышесказанного подготовка данных для обучающей выборки на основе результатов опроса учеников заключается в выполнении следующих шагов.

1. Выбор дисциплины и установка порога разделения на классы в виде тестового балла ЕГЭ.

2. Формирование вектора характеристик для каждого объекта (ученика) на основе его результатов опроса в контексте выбранной дисциплины.

В итоге для двух дисциплин «русский язык» и «математика» были сформированы две обучающие выборки одинаковой мощности (в 546 объектов), но с разным

числом характеристик: 133 характеристики для дисциплины «Русский язык» и 141 характеристика для дисциплины «Математика», что объясняется разным числом вопросов, имеющих непосредственное или косвенное отношение к соответствующей дисциплине. При этом экспериментально было установлено, что целесообразно установить порогом разделения классов 80 баллов для дисциплины «Русский язык» и 70 баллов для дисциплины «Математика» (хотя высокобалльными принято считать работы, набравшие более 80 баллов). Установление именно таких пороговых значений в проводимых экспериментах может быть объяснено ограниченным объемом данных результатов опросов и традиционно более низкими баллами по дисциплине «Математика» (и, как следствие, отсутствием достаточного числа высокобалльных работ).

Как и ожидалось, при выбранных вариантах разделения объектов (учеников) на классы с метками «+1» и «–1» сами классы являются несбалансированными, т. е. число объектов одного класса (мажоритарного класса с меткой «–1») значительно превышает число объектов второго класса (миноритарного класса с меткой «+1») (табл. 1). Например, миноритарный класс с меткой «+1» описывает учеников с баллами аттестации, равными 80 или выше.

В связи с этим было принято решение об использовании SMOTE-алгоритма с целью снижения несбалансированности классов [13]. Так, «Rus_SVM», «RUS_regular», «Rus_borderline1» и «Rus_borderline2» наборы данных были получены из «Rus_80» набора данных с использованием «SVM», «regular», «borderline1» and «borderline2» вариантов SMOTE-алгоритма соответственно. В результате несбалансированность классов была уменьшена (см. табл. 1).

Затем были выполнены эксперименты по разработке SVM-классификатора с использованием статистических пакетов Statistica StatSoft [21] и IBM SPSS Modeler [22], и авторской программы «Intellectual Classification» (IC). При этом для всех наборов данных размер тестовой выборки составлял 20% от размера экспериментального набора данных. В табл. 1 для каждого набора данных содержится информация о точности классификации на обучающей (Train) и тестовой (Test) выборках, указано число объектов в этих выборках, а также приведена общая точность классификации. В ходе экспериментов использовались полиномиальная и радиальная базисная функции ядра. Во всех случаях радиальная базисная функция ядра показала лучший результат в контексте обеспечения высокого качества классификации. Результаты разработки SVM-классификаторов приведены в табл. 1.

Таблица 1. Результаты разработки SVM-классификаторов в задаче предсказания успешности прохождения государственной итоговой аттестации

Набор *s*×*q*	Среда построения SVM-классификатора	Число объектов в выборках Train и Test (Train/Test)	Параметры ядра		Число опорных векторов	Точность			1 класс (класс с меткой "+")			2 класс (класс с меткой "–")		
			C	σ		Train	Test	Overall	Реальное число объектов в класс	Число ошибок	% от числа объектов в классе	Реальное число объектов в классе	Число ошибок	% от числа объектов в классе
Rus_80 546×133	*STATISTICA*	436/110	1	0.008	102	94.954	93.636	94.689	29	29	100	517	0	0
	SPSS Modeler	436/110	10	0.1	–	100	94.55	98.90		6	20.69		0	0
	IC	437/109	9.88	7.03	217	100	99.08	99.82		1	3.45		0	0
Rus_SVM 814×133	*STATISTICA*	651/163	5	0.008	211	98.310	93.252	97.297	297	4	1.35	517	18	3.48
	SPSS Modeler	649/165	10	0.1	–	100	100	100		0	0		0	0
	IC	652/162	8.86	8.16	166	100	100	100		0	0		0	0
RUS_ regular 1034×133	*STATISTICA*	827/207	8	0.008	215	98.791	96.618	98.356	517	0	0	517	17	3.29
	SPSS Modeler	826/208	10	0.1	–	100	100	100		0	0		0	0
	IC	828/206	3.49	5.75	269	100	100	100		0	0		0	0
Rus_ borderline1 1034×133	*STATISTICA*	827/207	9	0.008	183	98.670	96.135	98.162	517	4	0.77	517	15	2.901
	SPSS Modeler	826/208	10	0.1	–	100	100	100		0	0		0	0
	IC	828/206	7.35	9.11	152	100	99.51	99.9		1	0.19		0	0
Rus_ borderline2 1033×133	*STATISTICA*	826/207	10	0.008	225	98.668	96.135	98.161	516	4	0.78	517	15	2.90
	SPSS Modeler	825/208	10	0.1	–	100	100	100		0	0		0	0
	IC	827/206	9.59	8.68	185	100	99.51	99.90		1	0.19		0	0
Math_70 546×141	*STATISTICA*	436/110	7	0.007	91	95.560	92.727	95.788	38	22	57.89	508	1	0.20
	SPSS Modeler	436/110	10	0.1	–	99.77	100	99.82		1	2.63		0	0
	IC	437/109	5.38	8.20	176	100	97.25	99.45		1	2.63		2	0.39
Math_SVM 1016×141	*STATISTICA*	812/204	5	0.007	182	98.030	95.098	97.441	508	4	0.79	508	22	4.33
	SPSS Modeler	812/204	10	0.1	–	99.88	99.51	99.8		1	0.20		1	0.20
	IC	813/203	7.22	7.90	202	100	99.51	99.9		1	0.20		0	0
Math_ regular 1016×141	*STATISTICA*	812/204	5	0.007	178	98.645	96.078	98.13	508	3	0.59	508	16	3.15
	SPSS Modeler	812/204	10	0.1	–	99.88	100	99.9		0	0		1	0.20
	IC	813/203	6.73	7.96	174	100	100	100		0	0		0	0
Math_ borderline1 1016×141	*STATISTICA*	812/204	7	0.007	149	99.138	96.078	98.524	508	2	0.39	508	13	2.56
	SPSS Modeler	812/204	10	0.1	–	99.88	100	99.9		0	0		1	0.20
	IC	813/203	9.28	9.95	144	100	99.51	99.9		1	0.20		0	0
Math_ borderline2 1016×141	*STATISTICA*	812/204	10	0.007	182	99.138	96.078	98.524	508	4	0.79	508	11	2.17
	SPSS Modeler	812/204	10	0.1	–	99.88	99.51	99.8		1	0.20		1	0.20
	IC	813/203	8.85	6.52	279	100	100	100		0	0		0	0

Из табл. 1 видно, что при отсутствии перебалансировки в Statistica StatSoft все объекты (т. е. 29 объектов) миноритарного класса набора данных «Rus_80» и значительная часть объектов (22 объекта из 38) миноритарного класса набора данных

«Math_70» были классифицированы ошибочно. Для набора данных «Rus_80» было получено 100% и 0% ошибок в классах с метками «+1» и «–1» соответственно, для набора данных «Math_70» было получено 57.89% и 0.20% ошибок в классах с метками «+1» и «–1» соответственно, хотя общая точность классификации является высокой (94.689% и 95.788% соответственно).

Применение IBM SPSS Modeler для этих наборов данных позволило несколько повысить общую точность классификации (см. табл. 1). Использование программы «Intellectual Classification» позволило еще улучшить значения показателей точности классификации. Но во всех случаях почти все ошибки оказались в миноритарном классе, поэтому разработанный на основе несбалансированных наборов SVM-классификатор будет давать неверные прогнозы для новых объектов этого класса (класса с высокобалльными работами).

В пакете Statistica StatSoft значения параметров функции ядра (радиальной базисной функции ядра и полиномиальной функции ядра) были выбраны в соответствии с установленными по умолчанию значениями, а значение параметра регуляризации определялось с использованием процедуры скользящего контроля. Так, параметр σ радиальной базисной функции ядра по умолчанию выбирается в соответствии с размером набора данных (чем больше размер набора данных, тем меньше σ). Лучшие результаты разработки SVM-классификатора были получены для набора данных «RUS_regular» для дисциплины «Русский язык» (0.78 и 3.29% ошибок в классах) и для набора данных «Math_borderline1» для дисциплины «Математика» (0.39 и 2.56% ошибок в классах) с использованием радиальной базисной функции ядра.

В пакете IBM SPSS Modeler отсутствуют средства подбора параметров SVM-классификатора, обеспечивающего максимальную точность классификации, поэтому разработка SVM-классификатора с радиальной базисной и полиномиальной функциями ядра производилась с использованием параметров, заданных по умолчанию. Например, при использовании радиальной базисной функции ядра по умолчанию используются следующие значения: $C = 10$ and $\sigma = 0.1$. Кроме того, в этом пакете невозможно оценить (увидеть) число опорных векторов. Лучшие результаты разработки SVM-классификатора были получены при использовании радиальной базисной функции ядра. Для всех синтезированных наборов данных для дисциплины «Русский язык» удалось достигнуть 100%-ной точности классификации. Для дисциплины «Математика» лучшие результаты были получены для наборов данных «Math_regular» и «Math_borderline1» (0% и 0.20% ошибок в классах).

Авторская программа «Intellectual Classification» содержит модуль автоматического поиска оптимальных значений параметров SVM-классификатора (параметра регуляризации и параметров функции ядра) с использованием модифицированного

PSO-алгоритма. Использование этой программы позволило разработать SVM-классификаторы с минимальным числом ошибок для исходных несбалансированных наборов данных (1 ошибка для набора данных «Rus_80» и 3 ошибки для набора данных «Math_70»), а также свести до нуля число ошибок для сбалансированных наборов данных «Rus_SVM», «Rus_regular», «Math_regular» и «Math_borderline2».

Подтверждение актуальности темы проводимых исследований, связанных с анализом готовности выпускников средней школы к прохождению итоговой государственной аттестации, и целесообразности применения аппарата интеллектуального анализа данных было получено по результатам обзора научных публикаций отечественных и зарубежных авторов. Наиболее близкое по постановке задачи и по применяемому инструментарию анализа данных исследование приведено в работе [23], авторы которой решают задачу предсказания успешности учащихся двух португальских школ. Для разработки 5 типов классификаторов (Neural Networks (NN), SVM, Naive Predictor (NV), Random Forest (RF), Decision Tree (DT) [23]) с целью предсказания успеваемости по 2 дисциплинам — «Португальский язык» и «Математика» — используются 2 набора данных (http://archive.ics.uci.edu/ml/datasets/Student+Performance), содержащих соответствующую консолидированную информацию об учащихся (о составе семьи, уровне образования родителей, отношении к изучаемым дисциплинам, интересах и увлечениях учащихся, их планах на будущее и т. п.)

Упомянутые выше наборы данных были использованы для проведения экспериментов, аналогичных экспериментам, выполненным для наборов данных «Rus_80» и «Math_70». Исходные наборы данных «Portu» («Португальский язык») и «Math» («Математика»), как и ожидалось, оказались несбалансированными (при этом набор данных «Portu» оказался несбалансирован в большей степени, чем набор данных «Math»). В связи с этим было принято решение об использовании SMOTE-алгоритма с целью снижения несбалансированности классов. Результаты экспериментов по разработке SVM-классификаторов с применением радиальной базисной функции ядра приведены в табл. 2. Результаты экспериментов по разработке SVM-классификаторов с применением полиномиальной функции ядра по своей сути оказались аналогичными. Полученные результаты экспериментов свидетельствуют о том, что для обоих несбалансированных наборов данных наблюдается низкое качество классификации данных с применением статистического пакета Statistica StatSoft (со значительным числом ошибок в миноритарном классе). При этом применение авторской программы «Intellectual Classification» позволяет улучшить значения показателей точности классификации и выбрать лучший вариант балансировки данных. Кроме того, программа «Intellectual Classification» обеспечивает поиск оптимального типа функции ядра посредством применения модифици-

рованного PSO-алгоритма, что позволяет минимизировать затраты на разработку искомого SVM-классификатора. Следует отметить, что применение программы «Intellectual Classification» к сбалансированным наборам данных позволяет получить лучшие значения оценок качества классификации, чем в работе [23].

Таблица 2. Результаты разработки SVM-классификаторов в задаче предсказания успешности учеников португальских школ

Набор *s*×*q*	Среда построения SVM-классификатора	Число объектов в выборках Train и Test (Train/Test)	Параметры ядра		Число опорных векторов	Точность			1 класс (класс с меткой "+")			2 класс (класс с меткой "–")		
			C	σ		Train	Test	Overall	Реальное число объектов в класс	Число ошибок	% от числа объектов в классе	Реальное число объектов в классе	Число ошибок	% от числа объектов в классе
Portu 649×30	*STATISTICA*	519/130	8	0.033	201	87.86	86.92	87.67	549	7	1.27	100	73	73
	SPSS Modeler	521/128	10	0.1	–	97.89	85.16	95.38		12	2.19		18	18
	IC	518/129	9.864	4.851	244	99.16	88.37	97.37		6	1.09		11	11
Portu_SVM 1098×30	*STATISTICA*	878/220	10	0.033	345	90.77	90.00	90.62	549	55	10.01	549	48	8.74
	SPSS Modeler	880/218	10	0.1	–	98.41	94.04	97.54		14	2.55		13	2.37
	IC	879/219	2.529	2.768	440	100	96.35	99.27		2	0.36		6	1.09
Portu _regular 1098×30	*STATISTICA*	878/220	10	0.033	382	93.39	89.09	92.53	549	58	10.56	549	24	4.37
	SPSS Modeler	880/215	10	0.1	–	98.30	94.04	97.45		16	2.91		12	2.19
	IC	879/219	7.972	3.241	432	100	97.26	99.45		5	0.91		1	0.18
Portu_borderline1 1098×30	*STATISTICA*	878/220	10	0.033	350	93.96	89.55	93.08	549	49	8.38	549	27	4.92
	SPSS Modeler	880/218	10	0.1	–	98.30	93.12	97.27		16	2.91		14	2.55
	IC	879/219	5.118	3.15	435	100	94.52	98.91		8	1.46		4	0.73
Portu_borderline2 1098×30	*STATISTICA*	878/220	10	0.033	401	93.85	90.46	93.17	549	47	8.56	549	28	5.10
	SPSS Modeler	880/218	10	0.1	–	98.30	93.58	97.36		17	3.10		12	2.19
	IC	879/219	7.269	3.91	385	100	97.26	99.45		1	0.18		5	0.91
Math 395×30	*STATISTICA*	316/79	3	0.033	221	74.68	67.09	73.17	265	8	3.02	130	98	75.38
	SPSS Modeler	315/80	10	0.1	–	94.29	65.00	88.35		24	9.06		22	16.92
	IC	316/79	6.650	6.886	243	91.14	65.82	86.08		30	11.32		25	19.23
Math_SVM 529×30	*STATISTICA*	423/106	10	0.033	296	76.60	69.81	75.24	265	56	21.13	264	75	28.41
	SPSS Modeler	423/106	10	0.1	–	94.56	74.53	90.55		30	11.32		20	7.58
	IC	424/106	8.735	4.560	286	99.76	80.95	96.03		13	4.91		8	3.03
Math_ regular 530×30	*STATISTICA*	424/106	6	0.033	316	78.77	68.87	76.79	265	57	21.51	265	66	24.91
	SPSS Modeler	424/106	10	0.1	–	94.81	73.58	90.57		30	11.32		20	7.55
	IC	424/106	9.736	1.993	398	100	91.51	98.30		7	2.64		2	0.75
Math_borderline1 530×30	*STATISTICA*	424/106	9	0.033	308	81.60	70.76	79.43	265	56	21.13	265	53	20
	SPSS Modeler	424/106	10	0.1	–	94.58	78.30	91.32		29	10.94		17	6.42
	IC	424/106	7.575	2.584	357	100	84.91	96.98		7	2.64		9	3.40
Math_borderline2 529×30	*STATISTICA*	423/106	6	0.033	315	80.14	73.59	78.83	265	47	17.74	264	65	24.62
	SPSS Modeler	423/106	10	0.1	–	94.80	77.36	91.30		26	9.81		20	7.58
	IC	424/106	4.113	3.521	317	99.76	78.10	95.46		15	5.66		9	3.41

Полученные экспериментальные результаты, приведенные в табл. 2, свидетельствуют о целесообразности привлечения дополнительного инструментария интеллектуального анализа данных (например, kNN-алгоритма [16]) с целью снижения доли ошибочно классифицируемых данных.

4. Заключение

Использование SVM-классификаторов на основе модифицированного PSO-алгоритма и различных стратегий сэмплинга в контексте предсказания успеха прохождения окончательной государственной аттестации выпускниками средней школы позволяет обеспечить высокую точность классификации. Результаты экспериментальных исследований подтверждают целесообразность дальнейшего развития предлагаемого подхода к разработке SVM-классификаторов. При этом планируется использовать SVM-алгоритм для разработки регрессионной модели, что позволит прогнозировать результаты аттестации в баллах.

Литература:

[1] *Atkinson J. W.* Studying personality in the context of an advanced motivational psychology // *American Psychologist*, 1981. Vol. 36. p. 117–128.

[2] Issues in the Psychology of Motivation. Ed. by Paula R. ZelickNova. Science Publishers, Inc., 2007. p. 241.

[3] Behavioral Toxicology. Ed. by Bernard Weiss and Victor C. Laties. Springer, 1975, p. 469.

[4] *Bye H. H., Sandal G. M.* Applicant Personality and Procedural Justice Perceptions of Group Selection Interviews // *Journal of Business and Psychology*. 2016. Vol. 31. No. 4. p. 569–582.

[5] *Vapnik V.* Statistical Learning Theory. — New York : John Wiley & Sons, 1998.

[6] *Chapelle O., Vapnik V., Bousquet O., Mukherjee S.* Choosing Multiple Parameters for Support Vector Machine // *Machine Learning.* 2002. Vol. 46. p. 131–159.

[7] *Yu L., Wang S., Lai K. K., Zhou L.* Bio-Inspired Credit Risk Analysis. Computational Intelligence with Support Vector Machines. Springer, 2008.

[8] *Демидова Л. А., Соколова Ю. С.* Аспекты применения алгоритма роя частиц в задаче разработки SVM-классификатора // *Вестник РГРТУ*. 2015. № 53. С. 84–92.

[9] *Демидова Л. А., Никульчев Е. В., Соколова Ю. С.* Классификация больших данных: использование SVM-ансамблей и SVM-классификаторов с модифицированным роевым алгоритмом // *Cloud of Science*. 2016. Т. 3. № 1. С. 5–42.

[10] *Демидова Л. А., Соколова Ю. С.* Разработка ансамбля SVM-классификаторов с использованием декорреляционного алгоритма максимизации // *Информатика и системы управления*. 2016. № 1(47). С. 95–105.

[11] *Demidova L., Nikulchev E., Sokolova Yu.* Use of Fuzzy Clustering Algorithms' Ensemble for SVM classifier Development // *International Review on Modelling and Simulations*. 2015. Vol. 8. No. 4. p. 446–457.

[12] *Demidova L., Sokolova Yu.* Modification Of Particle Swarm Algorithm For The Problem Of The SVM Classifier Development // 2015 International Conference «Stability and Control Processes» in Memory of V. I. Zubov (SCP). IEEE, 2015. p. 623–627.

[13] *Demidova L., Nikulchev E., Sokolova Yu.* The SVM Classifier Based on the Modified Particle Swarm Optimization // *International Journal of Advanced Computer Science and Applications*. 2016. Vol. 7. No. 2. p. 16–24.

[14] *Demidova L., Nikulchev E., Sokolova Y.* Big Data Classification Using the SVM Classifiers with the Modified Particle Swarm Optimization and the SVM Ensembles // *International Journal of Advanced Computer Science and Applications*. 2016. Vol. 7. No. 5. p. 294–312.

[15] *Demidova L., Sokolova Y.* Development of the SVM Classifier Ensemble for the Classification Accuracy Increase // *ITM Web of Conferences*, 2016. Vol. 6. P. 02003.

[16] *Demidova L., Klyueva I., Sokolova Y., Stepanov N., Tyart N.* Intellectual Approaches to Improvement Of the Classification Decisions Quality On the Base Of the SVM Classifier // *Procedia Computer Science*. 2017. Vol. 103. p. 222–230.

[17] *Chawla N., Bowyer K., Hall L., Kegelmeyer W.* SMOTE: Synthetic Minority Over-sampling Technique // *Journal of Artificial Intelligence Research*. 2002. Vol. 16. p. 341–378.

[18] *Han H., Wen-Yuan W., Bing-Huan M.* Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science. Vol. 3644. — Berlin : Heidelberg, Springer, 2005. p. 878–887.

[19] *Nguyen H. M., Cooper E. W., Kamei K.* Borderline over-sampling for imbalanced data classification // *International Journal of Knowledge Engineering and Soft Data Paradigms*. 2001. Vol. 3. No. 1. p. 4–21.

[20] *Демидова Л. А., Егин М. М.* Использование многоцелевого генетического алгоритма в задаче поиска оптимальных значений для набора показателей классификации // Прикладные исследования и технологии. Сб. тр. конф. — М. : МТИ, 2017. С. 123–129.

[21] Statistica Help. Support Vector Machine Example 1 — Classification [Электронный ресурс]. URL: http://documentation.statsoft.com/STATISTICAHelp.aspx?path=MachineLearning/MachineLearning/SupportVectorMachine/SupportVectorMachineExample1Classification

[22] Classifying Cell Samples (SVM). IBM SPSS Modeler Tutorial. IBM Knowledge Center [Электронный ресурс]. URL: https://www.ibm.com/support/knowledgecenter/SS3RA7_18.1.0/modeler_tutorial_ddita/clementine/example_svm_intro.html

[23] *Cortez P., Silva A.*, Using Data Mining to Predict Secondary School Student Performance // Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008). Portugal, Porto, 2008. p. 5–12.

***Авторы*:**
Лилия Анатольевна Демидова — доктор технических наук, профессор, профессор кафедры вычислительной и прикладной математики, Рязанский государственный радиотехнический университет
Максим Михайлович Егин — студент 3-го курса, Рязанский государственный радиотехнический университет
Юлия Сергеевна Соколова — старший преподаватель кафедры вычислительной и прикладной математики, Рязанский государственный радиотехнический университет

Data classification in the sphere of education using intellectual technologies

Liliya Demidova, Maxim Egin, Yulia Sokolova

Ryazan State Radio Engineering University
Gagarin Str., 59/1, Ryazan, Russian Federation, 390005

e-mail: *liliya.demidova@rambler.ru, eginmm@gmail.com, JuliaSokolova62@yandex.ru*

Abstract. The problem of the data classification in the educational sphere in the context of prediction of the passing's success of the final state attestation by the graduates of the secondary school has been considered. Such data can be imbalanced. To solve this problem it is offered to use the SVM classifiers on the base of the modified PSO algorithm, which allows choosing the kernel function type, the values of the kernel function parameters and the value of the regularization parameter simultaneously. The strategies, based on the SMOTE algorithm, can be applied for rebalance the classes in the datasets. Analysis of the classification results using the SVM classifiers based on the modified PSO algorithm and the resampling strategies and the results obtained in statistical programs packages demonstrates the advisability of application of the proposed toolkit to solving the data analysis problem in the sphere of education.
Key words: support vector machine, classification, particle swarm optimization, PSO algorithm, SMOTE algorithm.

References

[1] *Atkinson J. W.* (1981) *American Psychologist*, 36: 117–128.
[2] Issues in the Psychology of Motivation (2007) Science Publishers, Inc.
[3] Behavioral Toxicology (1975) Springer.
[4] *Bye H. H. and Sandal G. M.* (2016) *Journal of Business and Psychology*. **31**(4): 569–582.
[5] *Vapnik V.* (1998) Statistical Learning Theory. New York, John Wiley & Sons.
[6] *Chapelle O., Vapnik V., Bousquet O., Mukherjee S.* (2002) Machine Learning, 46: 131–159.
[7] *Yu L., Wang S., Lai K. K., and Zhou L.* (2008) Bio-Inspired Credit Risk Analysis. Computational Intelligence with Support Vector Machines. Berlin Heidelberg, Springer-Verlag.
[8] *Demidova L. A., Sokolova Ju. S.* (2015) *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta*, 53: 84–92 [In Rus]
[9] *Demidova L. A., Nikulchev E., Sokolova Ju. S.* (2016) *Cloud of Science*, **3**(1):5–42 [In Rus]
[10] *Demidova L. A., Sokolova Ju. S.* (2016) *Informatika i sistemy upravleniya*, **1**(47):95–105 [In Rus]
[11] *Demidova L., Nikulchev E., Sokolova Yu.* (2015) *International Review on Modelling and Simulations*. **8**(4):446–457.

[12] *Demidova L., Sokolova Yu.* (2015) 2015 International Conference «Stability and Control Processes» in Memory of V. I. Zubov (SCP): 623–627.

[13] *Demidova L., Nikulchev E., and Sokolova Yu.* (2016) *International Journal of Advanced Computer Science and Applications*, **7**(2): 16–24.

[14] *Demidova L., Nikulchev E., and Sokolova Yu.* (2016) *International Journal of Advanced Computer Science and Applications*, **7**(5): 294–312.

[15] *Demidova L., Sokolova Yu.* (2016) *ITM Web of Conferences*, 6:02003.

[16] *Demidova L., Klyueva I., Sokolova Y., Stepanov N., and Tyart N.* (2017) *Procedia Computer Science*, 103: 222–230.

[17] *Chawla N., Bowyer K., Hall L., and Kegelmeyer W.* (2002) *Journal of Artificial Intelligence Research*, 16: 341–378.

[18] *Han H., Wen-Yuan W., and Bing-Huan M* (2005) Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science. Vol. 3644. Berlin, Heidelberg, Springer. p. 878–887.

[19] *Nguyen H. M., Cooper E. W., and Kamei K.* (2001) *International Journal of Knowledge Engineering and Soft Data Paradigms*, **3**(1):4–21.

[20] *Demidova L. A., Egin M. M.* (2017) Prikladnyye issledovaniya i tekhnologii ART2017: sbornik trudov mezhdunarodnoi konferentsii, p. 123–129 (In Rus)

[21] http://documentation.statsoft.com/STATISTICAHelp.aspx?path=MachineLearning/MachineLearning/SupportVectorMachine/SupportVectorMachineExample1Classification

[22] https://www.ibm.com/support/knowledgecenter/SS3RA7_18.1.0/modeler_tutorial_ddita/clementine/example_svm_intro.html

[23] *Cortez P., Silva A.* (2008) Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008). Porto, Portugal. p. 5–12.