УДК 621.391:396

А. А. Сирота, д-р техн. наук, проф., зав. каф., e-mail: sir@cs.vsu.ru.,
А. В. Цуриков, аспирант, e-mail: andrew.tsurikov@gmail.com,
Воронежский государственный университет

Модели и алгоритмы классификации фрагментов текста и их применение для создания контентно-зависимых цифровых водяных знаков

Рассматриваются модели и алгоритмы классификации многомерных данных с использованием различных подходов к построению классификатора применительно к задаче создания контентно-зависимых цифровых водяных знаков. Проводится сравнение исследуемых алгоритмов, синтезированных на основе различных методов машинного обучения (нейронные сети, машины опорных векторов, потенциальные функции). Исследуется вероятность ошибки классификации многомерных данных в зависимости от размерности признакового пространства.

Ключевые слова: *цифровой водяной знак, классификация данных, радиально-базисные функции, машина опорных векторов, метод потенциальных функций, нейронные сети*

Введение и постановка задачи

В настоящее время повышенное внимание уделяется задачам автоматического распознавания текстовых документов как с точки зрения понимания его содержания, так и с точки зрения защиты авторских прав на произведения, представленные в текстовом формате, проверки наличия заимствований и т. п. Одной из перспективных информационных технологий, направленных на поддержку организационно-технических мероприятий по защите интеллектуальной собственности, является технология цифровых водяных знаков (ЦВЗ) [1, 2], реализующая в том или ином виде алгоритмы стеганографического скрытия информации (ССИ). При реализации технологий создания ЦВЗ применительно к текстовым документам основной задачей является обеспечение высокой достоверности восстановления привязанных к тексту ЦВЗ и их устойчивости к переформатированию текстовых данных. Известные способы [3], такие как метод выравнивания пробелами, метод чередования маркеров конца строки, двоичных нулей, знаков одинакового начертания и др., этому требованию не удовлетворяют. В силу этих особенностей эффективным способом обеспечения заданных требований, по нашему мнению, является использование ЦВЗ, которые не зависят от конкретной формы представления текста. Такие ЦВЗ являются контентно-зависимыми, так как фактически при их использовании реализуется процедура "узнавания" кодированных определенным образом фрагментов текста для извлечения последовательностей элементов ЦВЗ.

В общем виде модель стеганографического скрытия информации для создания ЦВЗ может быть представлена следующим образом. Пусть Z, D, K есть множества возможных контейнеров, скрываемых сообщений (ЦВЗ) и ключей, тогда процедура встраивания и извлечения сообщения может быть представлена в виде отображений вида

$$F_*\colon Z \times D \times K \to Z,\ \bar{z} = F_*(z, d, k),$$

$$z \in Z,\ \bar{z} \in Z,\ d \in D,\ k \in K,\ \|z - \bar{z}\| \to \min,$$

$$F_{**}\colon Z \times K \to D,\ \tilde{d} = F_{**}(\bar{z}, k),\ \tilde{d} \in D,\ \|d - \tilde{d}\| \to \min,$$

где z, $\bar{z}$ — исходный и заполненный контейнер; d, $\tilde{d}$ — исходное и восстановленное сообщение. В задаче создания контентно-зависимых ЦВЗ встраивание и восстановление данных может быть реализовано одним оператором, и соответствующее отображение нужно искать в виде

$$\tilde{d} = F(d, z),\ \tilde{d} \in D,\ \|\tilde{d} - d\| \to \min,$$

где оператор F реализует одновременно встраивание и извлечение информации, не искажая контейнер. Далее без ограничения общности в качестве сообщения или ЦВЗ будем рассматривать двоичную последовательность $d^{(p)}$, $p = \overline{1, P}$, где $d^{(p)} \in \{-1, +1\}$ — скалярная величина, несущая бит информации.

Пусть $z = (z_1, ..., z_n)^{\mathrm{T}}$ — случайный вектор, представляющий фрагмент текстового файла контейнера. Совокупность реализаций вектора $z \in R^n$: $z^{(p)}$, $p = \overline{1, P}$, представляет текст в целом и сопоставляется с элементами последовательности ЦВЗ. При формализации задачи необходимо представить каждый фрагмент контейнера в виде некоторого многомерного вектора данных $z \in R^n$, однозначно характеризующего этот фрагмент.

Компоненты вектора $z \in R^n$, а также его размерность непосредственно зависят от способа его фор-

мирования на основе исходного текста. К ним можно отнести:

- способы, основанные на кодировании символов текста;
- способы, основанные на подсчете длины каждого слова в тексте, подсчете числа слов в предложении;
- способы, основанные на числовом представлении основных структурных элементов организации текста и т. п.

Каждый из этих способов дает возможность получения некоторых структурных характеристик текста. Таким образом, преобразование текста в последовательность $B = \{z^{(p)}, p = \overline{1,P}\}$ можно осуществить различными способами, которые образуют конечное множество S. Тогда любой фрагмент текстового контейнера может быть представлен в виде вектора $z = z(s)$, где $s \in S$, $P = P(s)$. Используя введенные обозначения, можно свести исходную задачу к задаче нахождения отображения вида

$$\tilde{d}^{(p)} = F(d^{(p)}, z^{(p)}(s)),\ p = \overline{1,P}(s),$$
$$s \in S,\ E_d = \|d - \tilde{d}\| \to \min.$$

Это означает, что задача всегда сводится к задаче классификации произвольного множества точек B в многомерном пространстве на два класса, первый из которых соответствует значениям $d^{(p)} = 1$, а второй — значениям $d^{(p)} = -1$. В итоге задача создания ЦВЗ для текстового контейнера сводится к построению алгоритма классификации многомерных данных, представляющих текст. Такие ЦВЗ, очевидно, являются контентно-зависимыми, так как фактически при их создании реализуется процедура "узнавания" кодированных фрагментов текста для извлечения последовательности элементов ЦВЗ. Как показано в работах [4, 5], данная задача может быть достаточно эффективно решена путем построения классификатора, основанного на использовании различных методов машинного обучения. К ним можно отнести нейронные сети на основе многослойных персептронов (MLP) и радиально-базисных функций (RBF), метод потенциальных функций (PF), а также метод машин опорных векторов (SVM).

Целью настоящей работы является сравнительный анализ различных алгоритмов классификации фрагментов текста на основе эталонной статистической модели и реальных текстовых данных.

Эталонная модель классификации текстовых данных

В работе [5] предложена эталонная статистическая модель классификации текста в интересах создания *контентно-зависимых ЦВЗ*, т. е. меток, привязанных к конкретному тексту. Использование данной модели позволяет провести оценку потенциальной эффективности классификации фрагментов текста при их привязке к элементам двоичной последовательности ЦВЗ с использованием различных методов и алгоритмов классификации. При ее обосновании учитывались следующие соображения. Так как общая задача классификации текста сводится к определению принадлежности каждого его фрагмента одному из двух классов (1 и -1), в модели должно генерироваться два класса случайных векторов, образующих точки в многомерном пространстве произвольной размерности. Учитывая, что в процессе преобразования текст разбивается на фрагменты, каждый из которых содержит одинаковое число структурных элементов (в зависимости от типа разбиения это может быть набор букв, слов, абзацев), размерность пространства можно интерпретировать как число элементов в каждом фрагменте. Эта размерность напрямую влияет на потенциальную разделимость данных при классификации, так как ее увеличение означает увеличение числа признаков, по которым классифицируется каждый фрагмент. Таким образом, для исследования потенциальных возможностей создания контентно-зависимых ЦВЗ для контейнеров текстового типа предлагается следующая модель. В единичном гиперкубе I размерности n в соответствии с равномерным законом распределения случайным образом размещается P точек двух классов:

$$B_+ = \{z^{(p)}, p = \overline{1, P_1}\},\ B_- = \{z^{(p)}, p = \overline{1, P_2}\},$$
$$B = \{z^{(p)}, p = \overline{1,P}\} = B_+ \cup B_-,\ P_1 + P_2 = P;$$

$$D_+ = \{d^{(p)} = 1, p = \overline{1, P_1}\},\ D_- = \{d^{(p)} = 0, p = \overline{1, P_2}\},$$
$$D = \{d^{(p)}, p = \overline{1,P}\} = D_+ \cup D_-.$$

Вероятность появления каждой точки первого класса обозначим Q_1, вероятность появления точек второго класса — $Q_2 = 1 - Q_1$. Для генерации случайного числа точек с заданными вероятностями можно использовать алгоритм генерации биномиального распределения.

Основной задачей являются разработка обучаемого классификатора для совокупности векторов $B = \{z^{(p)}, p = \overline{1,P}\}$ и оценка вероятности ошибки классификации точек двух типов, равномерно распределенных в гиперкубе I. Следует также учитывать, что после создания классификатора предъявляемые для классификации данные могут быть некоторым образом искажены, что также отражается в модели. Для этого вводится вероятность искажения каждого фрагмента P_r, при этом искажение вносится в одну из компонент соответствующего вектора, которая модифицируется по равновероятному закону распределения. Синтезируемый классификатор должен быть устойчивым по отношению к данным искажениям.

Как уже упоминалось выше, в ходе исследования проводился сравнительный анализ возможностей применения различных методов классификации данных. С учетом специфики поставленной задачи при задании структуры и параметров классификаторов использовались определенные приемы, направленные на повышение качества получаемых алгоритмов. Рассмотрим реализованные варианты построения классификаторов подробно.

Нейронные сети с RBF и MLP

При решении задачи с использованием нейронных сетей на основе радиально-базисных функций (сети с RBF) в множестве B выделяются группы точек B_+ и B_-, соответствующие точкам первого и второго класса, после чего строится классификатор вида

$$\tilde{d} = F(d, z) = \operatorname{sign}\Phi(z) = \begin{cases} 1, \Phi(z) \geqslant 0, \\ -1, \Phi(z) < 0, \end{cases}$$

$$\Phi(z) = \sum_{i=1}^{K} w_i \varphi_i(z), \qquad (1)$$

$$\varphi_i(z) = \exp\left[-\frac{\|z - u_i\|^2}{2\sigma_i^2}\right],$$

где φ_i — радиально-базисные функции; u_i — центр i-й радиально-базисной функции; σ_i — параметр влияния i-й радиально-базисной функции; w_i — соответствующий весовой коэффициент этой функции; K — число используемых функций.

При фиксированном K классификатор зависит от трех групп параметров: центров радиальных функций $U = (u_1, ..., u_K)^{\mathrm{T}}$, параметров влияния $\Sigma = \{\sigma_1, ..., \sigma_K)^{\mathrm{T}}$, а также весовых коэффициентов $W = (w_1, ..., w_K)^{\mathrm{T}}$. При обосновании алгоритма настройки этих параметров в процессе обучения классификатора с учетом специфики решаемой задачи учитывались следующие соображения.

Для упрощения структуры и повышения эффективности нейронной сети число радиально-базисных функций должно быть меньше общего числа точек ($K < P$). При уменьшении числа радиально-базисных функций начинает расти ошибка классификации, поэтому для ее минимизации к каждому из множеств B_+, B_- независимо применяется алгоритм кластеризации точек k-средних, который выделяет заданное число кластеров, основываясь на определении центров масс векторов (центроидов) и поиске наиболее близких к каждому центроиду элементов по критерию $\sum_{z \in B_{+,-}^{(l)}} (z^{(p,\,l)} - u_l)^2 \to \min$.

Эти центроиды назначаются в качестве центров радиальных функций:

$$u_l^{(+)} = \sum_{z \in B_+^{(l)}} z^{(p,\,l)},\ l = \overline{1, P_1},$$

$$u_l^{(-)} = \sum_{z \in B_-^{(l)}} z^{(p,\,l)},\ l = \overline{1, P_2}.$$

В итоге формируется разбиение общего гиперкуба, соответственно, на K_1 и K_2 ячеек; при этом в пространстве данных образуется многомерная "решетка", получаемая при разбиении на кластеры точек первого класса. Поверх нее независимо накладывается решетка, получаемая при разбиении точек второго класса. Так как каждая ячейка будет классифицироваться своей функцией RBF, то очевидно, что наложение ячеек с точками разных классов будет приводить к ошибкам классификации.

Помимо центров радиальных функций необходимо назначить параметры влияния для RBF-функций. В предлагаемом алгоритме рассматриваются два способа задания этого параметра. Первый способ предполагает задание общего значения σ для всех функций путем последовательного перебора в заданном диапазоне от 0 до 1. Второй способ основан на определении радиуса каждого полученного кластера (расстояния от центра до наиболее удаленной точки) и задании параметра влияния как

$$\sigma_l^{(+)} = \rho R^{(p,l)},\ R^{(p,l)} = \max_{z \in B_+^{(l)}} (\|z^{(p,\,l)} - u_l\|),\ l = \overline{1, P_1},$$

$$\sigma_l^{(-)} = \rho R^{(p,l)},\ R^{(p,l)} = \max_{z \in B_-^{(l)}} (\|z^{(p,\,l)} - u_l\|),\ l = P_2,$$

где ρ — эмпирически подбираемый коэффициент, обычно равный 0,3...0,5.

На заключительном этапе построения классификатора необходимо вычисление весовых коэффициентов на основе решения системы линейных алгебраических уравнений (СЛАУ) [6]. В данном случае ($K < P$) СЛАУ для нахождения весовых коэффициентов является переопределенной. Для ее решения могут быть использованы различные методы, в том числе метод псевдоинверсии Мура — Пенроуза (нормальное псевдорешение), метод, основанный на разложении SVD (Singular Value Decomposition), метод регуляризации по А. Н. Тихонову [6]. Как показали исследования, последний вариант позволяет сформировать устойчивые решения в виде

$$GW = d,\ w = w^{(a)} + (G^{\mathrm{T}}G + \alpha I)^{-1} G^{\mathrm{T}}(d - Gw^{(a)}),$$

$$G = \|g_{p,\,i}\|,\ g_{p,\,i} = \|\varphi_i(z^{(p)})\|,\ p = \overline{1, P},\ i = \overline{1, K},\ K < P,$$

где G — матрица Грина [6], являющаяся в данном случае квадратной; $d = (d^{(1)}, ..., d^{(p)})^{\mathrm{T}}$ — целевой вектор, определяемый исходя из исходного множества тре-

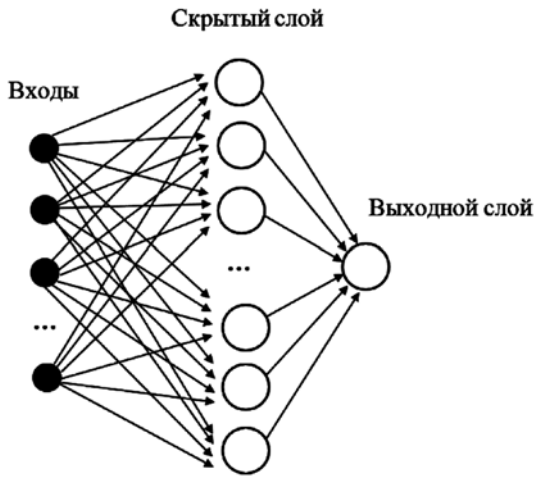

Рис. 1. Архитектура нейронной сети типа MLP

буемой классификации данных $D = \{d^{(p)}, p = \overline{1, N}\} = D_+ \cup D_-$; $w^{(a)}$ — априорное решение, которое в данном случае целесообразно выбрать следующим образом: $w_l^{(a)} = 1$, $l = \overline{1, K_1}$, $w_l^{(a)} = -1$, $l = \overline{K_1 + 1, K_2}$; α — параметр регуляризации, выбираемый одним из стандартных методов.

При построении классификатора на основе многослойных нейронных сетей персептронного типа (сети типа MLP) так же, как и в случае с сетями на основе радиально-базисных функций, возникает проблема определения оптимальной конфигурации сети, усложняющаяся в силу того, что MLP, в отличие от RBF, могут содержать более одного скрытого слоя. В работе [7] рассматривается *теорема об универсальной аппроксимации* для нелинейного отображения "вход—выход", которая определяет математические обоснования возможности аппроксимации любой непрерывной функции. Теорема утверждает, что многослойного персептрона с одним скрытым слоем достаточно для построения равномерной аппроксимации с точностью ε для любого обучающего множества, представленного набором входов $z^{(1)}, z^{(2)}, ..., z^{(P)}$ и желаемых откликов $f(z^{(1)}, ..., z^{(P)})$ [3]. При проведении исследования далее рассматривается сеть с одним скрытым слоем, содержащим K нелинейных нейронов, и выходным слоем, содержащим один линейный нейрон. Общий вид такой сети представлен на рис. 1.

Метод потенциальных функций (PF)

Подход к построению классификатора, основанный на методе потенциальных функций [7], также позволяет достаточно эффективно проводить разделение произвольной выборки данных на классы. Общая идея заключается в представлении элементов выборки в виде точек пространства, в которые помещается электрический заряд $+q$, если точка принадлежит элементу 1-го класса, и $-q_i$, если точка принадлежит элементу 2-го класса. Учитывая, что в рамках метода потенциальных функций можно рассматривать каждый элемент $z = (z_1, ..., z_n)^T$, $z \in R^n$, как единичный заряд, конечный вид классификатора, представляющего собой кумулятивный потенциал, создаваемый P зарядами в точке z, будет определяться следующим выражением:

$$g(z) = \sum_{i=1}^{P} q_i \Phi(z, z_i) = \Phi_N(z), \qquad (2)$$

где в качестве функции Φ выступает функция $\Phi(z, z_k) = \exp[-\|z - z_k\|^2 / 2\sigma^2]$, а коэффициенты q_i определяются соотношением

$$q_i = \begin{cases} 0, & z_i \in B_+, \ \Phi_{i-1}(z_i) > 0; \\ 0, & z_i \in B_-, \ \Phi_{i-1}(z_i) \leqslant 0; \\ 1, & z_i \in B_+, \ \Phi_{i-1}(z_i) \leqslant 0; \\ -1, & z_i \in B_-, \ \Phi_{i-1}(z_i) > 0. \end{cases}$$

Стоит отметить, что число ненулевых коэффициентов q_i определяет общее число нелинейных классифицирующих элементов K, составляющих классификатор, которое в данном случае зависит от обучающей выборки.

Машина опорных векторов (SVM)

Суть работы алгоритма, основанного на машине опорных векторов, состоит в построении гиперплоскости, максимально разделяющей элементы разных классов в многомерном пространстве [9—11]. В общем виде линейный классификатор для задачи классификации данных на два непересекающихся класса может быть представлен в виде

$$a(z) = \operatorname{sign}\left(\sum_{j=1}^{n} w_j z_j - w_0 \right), \qquad (3)$$

где $z = (z_1, ..., z_n)^T$ — фрагмент текста, вектор $w = (w_1, ..., w_n) \in R^n$ и скалярный порог $w_0 \in R$ являются параметрами алгоритма. Задача нахождения этих параметров сводится к эквивалентной двойственной задаче поиска седловой точки функции Лагранжа.

При построении нелинейного классификатора данных используются функции, обеспечивающие нелинейное преобразование исходного пространства, при этом классификатор ищется в виде [9]

$$a(z) = \operatorname{sign}\left(\sum_{i=1}^{l} \lambda_i d_i \Phi(z_i, z) - w_0 \right),$$

$$\Phi(z_i, z) = \exp[-(\|z_i - z\|)^2 / 2\sigma_i^2], \qquad (4)$$

где λ_i — компоненты вектора двойственных переменных; d_i — компоненты целевого вектора; σ_i — параметр влияния i-й функции.

Стоит отметить, что при решении задачи классификации фрагментов текста на основе машины опорных векторов классификатор можно считать

эффективным, если число используемых опорных векторов $l < P$ существенно меньше, чем число классифицируемых точек, т. е. на один опорный вектор приходится несколько классифицируемых точек (фрагментов текста).

Тогда настройка параметров классификатора (4) сводится к решению задачи квадратичного программирования вида

$$\max_{\lambda} W(\lambda) = \sum_{i=1}^{l} \lambda_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} d_i d_j \varphi(z_i, z_j) \lambda_i \lambda_j;$$

$$\sum_{i=1}^{l} d_i \lambda_i = 0, \ \forall i, \ 0 \leqslant \lambda_i \leqslant C.$$

Существует несколько различных алгоритмов решения данной задачи, среди которых непосредственное решение задачи квадратичного программирования (QP), использование алгоритма последовательной минимальной оптимизации (SMO) [9] и метод "бюджетированых" SVM, позволяющий выдержать заданный "бюджет" числа опорных векторов [10, 11]. Последний также предоставляет интерес для решения поставленной задачи.

В отличие от непосредственного решения задачи квадратичного программирования алгоритм SMO выбирает решение наименьшей возможной проблемы оптимизации на каждом шаге, что позволяет существенно ускорить и оптимизировать вычисления. Метод "бюджетированных" SVM основан на применении стохастического градиентного спуска при решении задачи построения классификатора, но при этом реализуется задача сохранения заданного "бюджета" опорных векторов. Для реализации подобного подхода на каждом шаге обучения алгоритм проверяет, не превышает ли число найденных опорных векторов (ОВ) заданное ограничение ($|I_{t+1}| < T$). В случае превышения число ОВ уменьшается на 1. При использовании данного метода эффективность классификации данных может снижаться по сравнению с обычными градиентными методами. В работе [10] показывается, что ошибка классификации, накладываемая введением в алгоритм стратегии поддержания опорных векторов на заданном уровне, ограничена сверху значением средней ошибки градиента

$$E = \sum_{t=1}^{P} \|E_t\|/P,$$

при этом $E_t = \Delta_t/\eta_t$, где Δ_t — разница между значениями параметров классификатора до начала процедуры сохранения ограничений и после; η_t — скорость обучения.

С учетом того, что влияние процедуры сохранения бюджета на ошибку классификации должно быть минимальным, алгоритм старается уменьшить величину E путем минимизации $\|E_t\|$ на каждом шаге, что сводится к минимизации $\|\Delta_t\|^2$ [10].

При реализации ограничений могут быть использованы различные стратегии сохранения числа опорных векторов: удаление избыточных опорных векторов, проецирование новых векторов на уже найденные и слияние векторов. В первых двух случаях вычислительная сложность алгоритмов достаточно высока, в связи с этим наиболее оптимальным представляется последний подход.

Необходимо отметить, что одним из главных условий эффективности всех описанных выше способов построения классификаторов для решения рассматриваемой задачи является минимизация общего числа нелинейных классифицирующих элементов, составляющих каждый классификатор (для RBF — это число радиально оазисных функций, для MLP — число персептронов в скрытом слое, для PF — число функций Ф, для SVM — число опорных векторов). Поэтому наряду с вероятностью общей ошибки классификации предлагается рассматривать универсальную величину K, которая является *"бюджетом" нелинейных классифицирующих элементов*, в качестве критерия эффективности для произвольного классификатора. Кроме того, для корректности сравнения классифицирующих способностей при различных подходах можно также рассматривать обратную величину *"бюджетного" отношения* $C = P/K$, которое показывает число классифицируемых элементов, приходящихся на единицу бюджета классифицирующих элементов. Для классификаторов на основе SVM, RBF и MLP существует возможность управлять бюджетом, в то время как для метода потенциальных функций ограничения такой возможности нет. Поэтому в данном случае проводилось усреднение бюджета, получаемого для нескольких реализаций процесса обучения классификатора.

Исследование алгоритмов на основе эталонной статистической модели

Представляется важным исследование потенциальных возможностей различных подходов к созданию классификатора фрагментов текста на основе эталонной статистической модели данных. В процессе моделирования в среде MATLAB проводилось сравнение вероятности ошибки классификации многомерных данных для классификаторов на основе RBF, нейронной сети MLP, машин опорных векторов, а также для классификатора, основанного на методе потенциальных функций. Для SVM использовалось три разных метода настройки параметров: SMO, непосредственное решение задачи квадратичного программирования (QP) и алгоритм "бюджетированного" стохастического градиентного спуска [10], причем в качестве эталонной была выбрана стандартная реализация алгоритма SMO [8] в MATLAB.

На рис. 2 показана зависимость вероятности ошибки классификации для классификаторов на

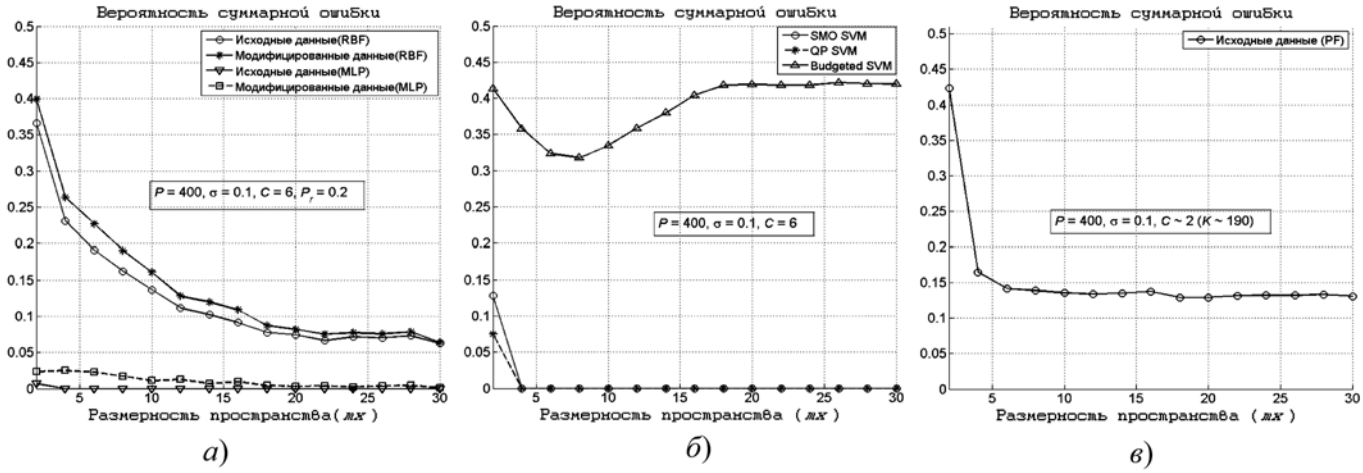

Рис. 2. Зависимость вероятности ошибки классификации для классификаторов на основе RBF, MLP, SVM и PF

основе RBF, MLP, SVM и PF от размерности *mx* признакового пространства при фиксированном параметре влияния RBF σ = 0,1 и "бюджетном" отношении $C = 6$. Так как для метода потенциальных функций невозможно установление ограничения на величину K, то на рисунке приведено фактическое значение параметра, получившееся в результате работы алгоритма. Для обеспечения приемлемой точности данные брались как усредненное значение результатов 100 реализаций процесса обучения на каждом шаге. Кроме того, график на рис. 2, *а* содержит результаты тестирования классификаторов RBF и MLP на модифицированных данных (каждая компонента всех элементов модифицируется с вероятностью $P_r = 0{,}2$).

Из графиков следует, что в целом наиболее эффективным классификатором является MLP, а наиболее эффективным алгоритмом настройки параметров SVM при малых значениях σ является SMO, который сохраняет показатели при увеличении размерности пространства. В то же время можно сказать, что несмотря на самую высокую эффективность классификации данных в целом, классификатор MLP является менее устойчивым к модификации исходных данных, чем RBF. Также стоит отметить, что классификатор, реализованный на основе метода потенциальных функций, является сопоставимым по величине ошибки классификации с другими алгоритмами, но гораздо менее эффективным с точки зрения величины бюджета нелинейных классифицирующих элементов.

При изменении значения параметра влияния σ эффективность алгоритма SMO остается на прежнем высоком уровне, в то время как метод, основанный на сохранении "бюджета" опорных векторов, показывает снижение ошибки классификации только при увеличении значения параметра σ и очень больших размерностях пространства.

График на рис. 3 показывает вероятность суммарной ошибки метода "бюджетированного" стохастического градиентного спуска для SVM при изменении ограничений на "бюджетное" соотношение C.

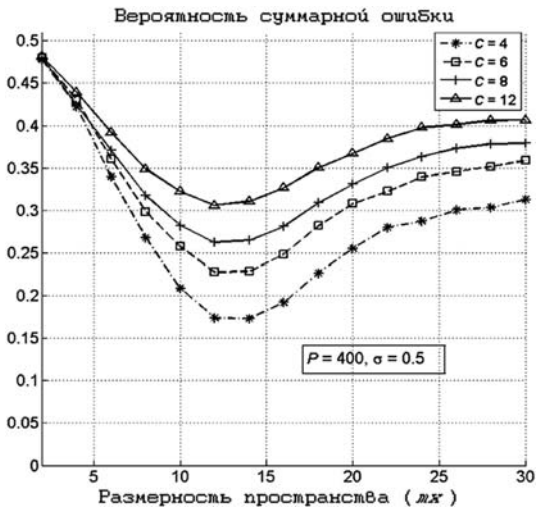

Рис. 3. Вероятность ошибки классификации для классификатора SVM при различных ограничениях на "бюджет" векторов

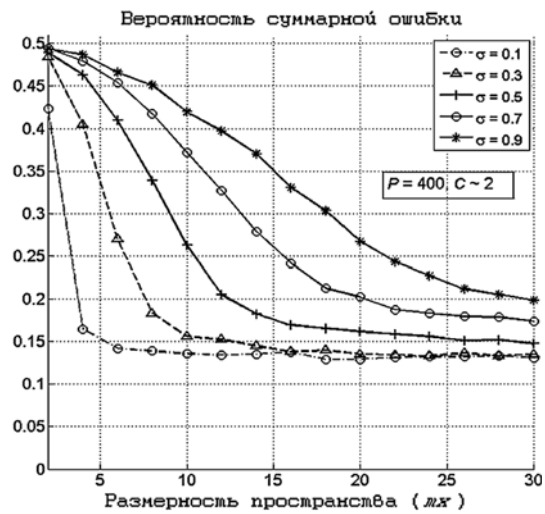

Рис. 4. Вероятность ошибки классификации для метода потенциальных функций при различных значениях параметра влияния

Учитывая, что для машины опорных векторов данная величина показывает отношение числа классифицируемых элементов к числу опорных векторов, зависимости ожидаемо показывают, что при уменьшении числа опорных векторов (соответственно, при увеличении числа элементов, приходящихся на один опорный вектор) вероятность ошибки классификации существенно возрастает, причем увеличение размерности пространства позволяет снизить эту величину лишь до некоторого предела. В частности, как можно видеть на графиках, минимальное значение вероятности ошибки наблюдается при выборе размерности пространства в диапазоне 12...14.

Рис. 4 показывает зависимость вероятности ошибки классификации от размерности пространства для классификатора на основе метода потенциальных функций при использовании разных значений параметра влияния σ. Очевидно, что увеличение параметра σ приводит к увеличению ошибки классификации, так как области влияния каждого нелинейного классифицирующего элемента начинают пересекаться все сильнее. В случае применения метода потенциальных функций оптимальным представляется использование наименьшей величины σ.

Необходимо отметить, что данные результаты были получены при наиболее плохом для алгоритма "бюджетированных" SVM случае распределения данных — равномерном. В случае даже незначительной неравномерности распределения данных показатели "бюджетированных" SVM улучшаются.

Тестирование алгоритмов на реальных данных

Для проверки работоспособности алгоритма при использовании реальных данных был выбран текст объемом около 3000 слов, который был закодирован несколькими способами. В частности, в качестве кодирования рассчитывались длины слов и брались коды символов. В отличие от эталонной модели, где число распределяемых в гиперкубе элементов было строго фиксированным и задаваемым в начале моделирования, при моделировании работы алгоритма на реальных данных число элементов, закодированных и нормализованных выбранным способом, будет меняться в силу того, что исходный текст представляется в виде последовательности $B = \{z^{(p)},\ p = \overline{1, P}\}$, где величина P, характеризующая число векторов, образующих последовательность B, зависит от размерности гиперкуба (фактически — от выбранной размерности векторов $z^{(p)} = (z_1, ..., z_n)^{\mathrm{T}}$). Кроме того, реальные данные имеют гораздо большую неравномерность распределения в силу естественных ограничений, накладываемых на способы кодирования. На рис. 5 приведен пример распределения нормализованных данных в двумерном гиперкубе для эталонной модели и реального текста соответ-

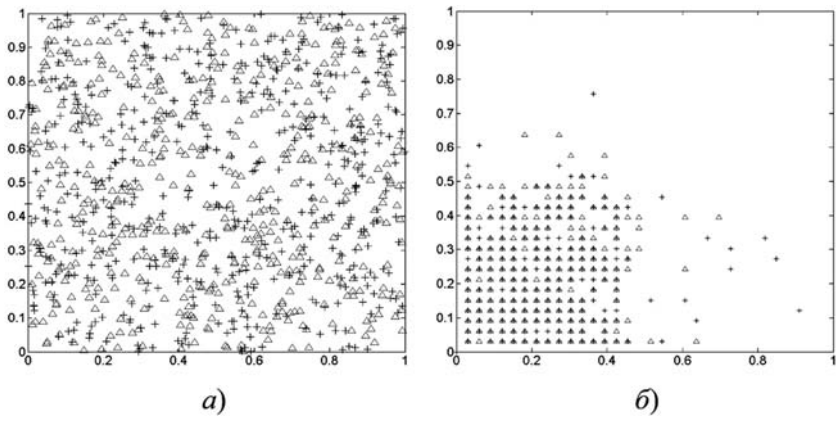

Рис. 5. Распределение элементов в двумерном гиперкубе для данных, полученных на основе эталонной модели, и для реального текста

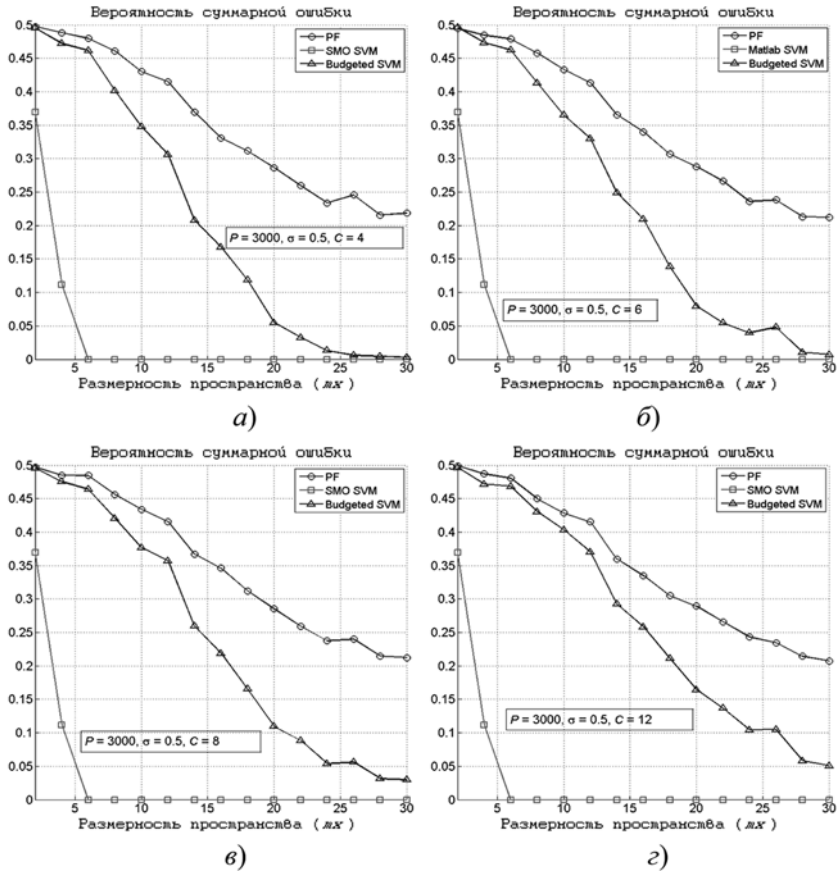

Рис. 6. Вероятность ошибки классификации для классификатора SVM при различных ограничениях на бюджет векторов (реальный текст)

ственно в случае, когда в качестве ЦВЗ использовалась псевдослучайная двоичная последовательность, а кодирование текста проводилось на основе подсчета длин слов.

Наличие сильной неравномерности распределения реальных данных существенно меняет результаты классификации. Как видно на рис. 6, наибольшую эффективность классификации по-прежнему сохраняет алгоритм SMO, но теперь алгоритм "бюджетированных" SVM позволяет достичь практически таких же результатов для размерности классифицируемых векторов больше 20.

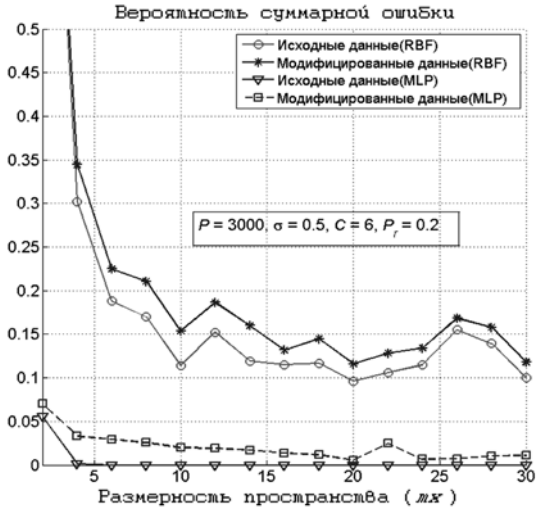

Рис. 7. Вероятность ошибки классификации для классификатора на основе RBF и MLP (реальный текст)

Таблица 1

Вероятности ошибки классификации в зависимости от размерности признакового пространства для эталонной модели

Размерность признакового пространства (*mx*)	SVM		RBF	MLP	PF
	SMO	Budgeted SVM			
2	0,3930	0,5028	0,2965	0,175	0,4885
4	0,1198	0,4810	0,2288	0,08	0,4638
8	0	0,4138	0,1624	0,0025	0,3303
16	0	0,3023	0,0960	0	0,1758
20	0	0,2665	0,0790	0	0,1540
24	0	0,2808	0,0679	0	0,1623
30	0	0,3163	0,0629	0	0,1475

Таблица 2

Вероятности ошибки классификации в зависимости от размерности признакового пространства для реального текста

Размерность признакового пространства (*mx*)	SVM		RBF	MLP	PF
	SMO	Budgeted SVM			
2	0,3691	0,4907	1	0,2913	0,4949
4	0,1117	0,4769	0,3667	0,1193	0,4913
8	0	0,4170	0,1538	0	0,4659
16	0	0,1280	0,1220	0	0,3227
20	0	0,0274	0,1151	0	0,3142
24	0	0,0102	0,0920	0	0,2364
30	0	0,0014	0,0861	0	0,2097

Для сравнения на рис. 7 показана вероятность ошибки классификации данных для реального текста при использовании нейронной сети на основе RBF и MLP при бюджетном отношении $C = 6$. На данном графике также представлен результат распознавания искаженных данных. При подобном искажении отдельные компоненты векторов $z^{(p)}$ модифицируются случайным образом с некоторой вероятностью P_r.

Можно заметить, что результаты работы алгоритма как с использованием нейронных сетей, так и с использованием машины опорных векторов, показывают достаточную для успешного разделения данных на два класса эффективность. В табл. 1 и 2 представлены общие результаты работы алгоритмов с использованием различных классификаторов для эталонной статистической модели и реального текста, соответственно.

Заключение

Результаты показывают, что использование машины опорных векторов позволяет достичь сопоставимых с классификатором на основе RBF показателей. Кроме того, стоит отметить, что кодирование реального текста накладывает естественные ограничения, непосредственно связанные с лингвистикой, которые сказываются на конечном распределении закодированных элементов. Наиболее эффективным алгоритмом с точки зрения вероятности ошибки классификации является нейронная сеть MLP, хотя данная вероятность падает при искажении текста. Сопоставимым с ней является алгоритм SMO SVM, который фактически обучается классифицировать уникально каждый элемент текста по аналогии с сетью на основе RBF с числом радиально-базисных функций, равным размеру обучающей выборки. Вместе с тем, в отличие от эталонной статистической модели, где метод бюджетированных SVM не показывал достаточно высокую ошибку классификации, при использовании реальных данных он становится более эффективным, чем классификатор на основе RBF, и успешно приближается в плане эффективности классификации к SMO, сохраняя при этом возможности по управлению ограничениями на число опорных векторов. Таким образом, можно сделать вывод, что классификатор на основе машины опорных векторов является альтернативой нейронным сетям и обладает сопоставимым качеством классификации.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 13-01-97507 р_центр_а.

Список литературы

1. **Мельников Ю. П., Теренин А. В., Погуляев В. Г.** Цифровые водяные знаки — новые методы защиты информации // Компьютерная неделя. 2007. № 48 (606). URL: http://www.pcweek.ru/security/article/detail.php?ID=105054 (дата обращения 08.06.2014).
2. **Барсуков В. С., Шувалов А. В.** Еще раз о стенографии — самой современной из древнейших наук // Специальная техника. 2004. № 2. URL: http://www.ess.ru/sites/default/files/files/articles/2004/02/2004 02 04.pdf (дата обращения 08.06.2014).
3. **Текин В.** Текстовая стеганография // МирПК. URL: http://www.osp.ru/pcworld/2004/11/169154/ (дата обращения 17.04.2014).
4. **Сирота А. А., Цуриков А. В.** Модели и алгоритмы классификации многомерных данных на основе нейронных сетей с радиально-базисными функциями // Вестник ВГУ. Сер. "Системный анализ и информационные технологии". Воронеж, 2013. № 1. С. 154—161.
5. **Сирота А. А., Цуриков А. В., Дрюченко М. А.** Модели и алгоритмы классификации фрагментов текста на основе нейронных сетей с радиально-базисными функциями // Нейрокомпьютеры: разработка, применение. 2013. № 5. С. 26—37.
6. **Хайкин С.** Нейронные сети: полный курс. 2-е изд., испр. М.: ООО "И. Д. Вильямс", 2006. 1104 с.
7. **Осовский С.** Нейронные сети для обработки информации / Пер с польского И. Д. Рудинского. М.: Финансы и статистика, 2002. 344 с.
8. **Вапник В. Н., Червоненкис А. Я.** Теория распознавания образов (статистические проблемы обучения). М.: Наука, 1974. 416 с.
9. **Platt J.** Fast Training of Support Vector Machines Using Sequential Minimal Optimization // Advances in Kernel Methods. — Support Vector Learning, 1998. P. 185—208.
10. **Wang Z., Crammer K., Vucetic S.** Breaking the Curse of Kernelization: Budgeted Stochastic Gradient Descent for Large-Scale SVM Training // Journal of Machine Learning Research. 2012. N. 3. P. 3103—3131.
11. **Dekel O., Singer Y.** Support Vector Machines on a Budget // Advances in Neural Information Processing Systems. 2006. N. 19. P. 345—352.

A. A. Sirota, Professor, **A. V. Tsurikov,** Postgraduate Student, e-mail: andrew.tsurikov@gmail.com,
Voronezh State University

Creating Content-Dependent Digital Watermarks Using their Structural Patterns: Models and Algorithms of Text Fragments Classification

The paper introduces approaches to creating content-dependent digital watermarks for the text data containers. It proposes ways of encoding text for further implementation of the matching procedure between the encoded text fragments and binary encoded digital watermark elements. The paper shows that this matching procedure is fully equivalent to the classification of the high-dimensional data. It also outlines approaches to creating high-dimensional data classifiers using various techniques for developing content-dependent digital watermarks, and compares them with other classification algorithms (neural networks, support vector machines and potential functions). Along with common algorithms of machine learning, it considers the ways of making them more efficient by limiting the number of the elements that take part in the classification. It also examines the dependency between the data classification errors and the space dimension size for different classifiers using simulated and real text data.

Keywords: *content-dependent digital watermark, data classification, radial-basis function, support vector machine, potential functions method, neural networks*

References

1. **Mel'nikov Ju. P., Terenin A. V., Poguljaev V. G.** Cifrovye vodjanye znaki — novye metody zashhity informacii. *Komp'juternaja nedelja*. 2007. N. 48 (606). URL: http://www.pcweek.ru/security/article/detail.php?ID=105054
2. **Barsukov V. S., Shuvalov A. V.** Eshhe raz o stenografii — samoj sovremennoj iz drevnejshih nauk. *Special'naja tehnika*. 2004. N. 2. URL: http://www.ess.ru/sites/default/files/files/articles/2004/02/2004 02 04.pdf
3. **Tekin V.** Tekstovaja steganografija. *MirPK*. URL: http://www.osp.ru/pcworld/2004/l 1/169154/
4. **Sirota A. A., Curikov A. V.** Modeli i algoritmy klassifikacii mnogomernyh dannyh na osnove nejronnyh setej s radial'no-bazisnymi funkcijami. *Vestnik VGU, Ser.: "Sistemnyj analiz i informacionnye tehnologii"*. Voronezh, 2013. N. 1. P. 154—161.
5. **Sirota A. A., Curikov A. V., Drjuchenko M. A.** Modeli i algoritmy klassifikacii fragmentov teksta na osnove nejronnyh setej s radial'no-bazisnymi funkcijami. *Neirokomp'jutery: razrabotka, primenenie*. 2013. N. 5. P. 26—37.
6. **Hajkin S.** Nejronnye seti: polnyj kurs. 2-e izd., ispr. M.: OOO "I. D. Vil'jams", 2006. 1104 p.
7. **Osovskij S.** *Nejronnye seti dlja obrabotki informacii* / Per s pol'skogo I. D. Rudinskogo. M.: Finansy i statistika, 2002. 344 p.
8. **Vapnik V. N., Chervonenkis A. Ja.** *Teorija raspoznavanija obrazov (statisticheskie problemy obuchenija)*. M.: Nauka, 1974. 416 p.
9. **Platt J.** Fast Training of Support Vector Machines Using Sequential Minimal Optimization. *Advances in Kernel Methods*. — Support Vector Learning, 1998. P. 185—208.
10. **Wang Z., Crammer K., Vucetic S.** Breaking the Curse of Kernelization: Budgeted Stochastic Gradient Descent for Large-Scale SVM Training. *Journal of Machine Learning Research*. 2012. N. 13. P. 3103—3131.
11. **Dekel O., Singer Y.** Support Vector Machines on a Budget. *Advances in Neural Information Processing Systems*. 2006. N. 19. P. 345—352.