УДК 681.3.06

КЛАССИФИКАЦИЯ ТЕКСТОВ НА ОСНОВЕ ОЦЕНКИ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ ТЕРМИНОВ

Нгуен Ба Нгок, А.Ф. Тузовский

Томский политехнический университет
E-mail: nguyen_bn@hotmail.com

Рассматривается способ увеличения точности классификации текстов по алгоритму kNN путем применения оценки семантической близости на основе матрицы совместной встречаемости терминов. Предлагается метод уменьшения размера матрицы совместной встречаемости путем фильтрации терминов по частям речи. Выполнена проверка влияния метода фильтрации на точность классификации.

Ключевые слова:
Классификация текстов, kNN, семантическая близость, матрица совместной встречаемости, аннотация по частям речи.

Key words:
Text classification, kNN, semantic similarity, co-occurrence matrix, parts of speech annotation.

Введение

Классификация текстов по тематическим категориям является актуальной задачей, которая требуется в различных информационных системах. Общая задача классификации данных рассматривается в книге [1]. В данной статье под термином *классификатор* понимается алгоритм решения задачи классификации текстов по тематическим категориям.

Среди существующих классификаторов, таких, как *Naïve Bayes* [1, 2], *Rocchio* [1, 3], *SVM* [1, 4], *kNN* [1] и т. д., самым быстрым и интуитивно понятным является классификатор *kNN* (*k-Nearest Neighbor* – ближайший из *k* соседей).

По алгоритму *kNN* результирующим тематическим классом классификатора является класс, имеющий максимальную оценку близости

$$kNN(D) = K_{\max},$$

при этом $K_{\max}$ определяется по условию:

$$sim(D, K_{\max}) = \max_{i=1\cdots n}(sim(D, K_i)),$$

где $sim(D,K_i)$ – оценка близости документа D корпусу K_i.

Целью данной работы является повышение точности работы алгоритма *kNN* с помощью применения оценки семантической близости на основе матрицы совместной встречаемости терминов в тексте. Предложенный алгоритм сравнивается по точности с классическим текстовым классификатором *kNN*.

Постановка задачи

Заданы k тематических классов, при этом, каждый i-й тематический класс определяется коллекцией (корпусом) документов K_i, которая состоится из множества текстовых документов по i-й теме. Каждая коллекция K_i также называется обучающей выборкой i-го тематического класса:

$$K_i = \{D_1, D_2, \cdots, D_{n_i}\},$$

где каждый документ D_j состоится из множества терминов:

$$D_j = \{t_1, t_2, \cdots, t_{m_j}\}.$$

Для любого текстового документа D, который не входит в обучающую выборку K_i, требуется определить наиболее подходящий тематический класс.

1. Классификатор *kNN* в векторном пространстве

Классический алгоритм *kNN* для классификации текстов базируется на оценки близости в векторном пространстве. Для этого выполняется представление документа в виде вектора терминов.

1.1. Представление документов в векторном пространстве

Пусть T – множество всех уникальных терминов системы:

$$T = \{t_1, t_2, \cdots, t_n\},$$

которое определяет n-мерное пространство, где каждый термин t_i соответствует одной размерности пространства.

В данном n-мерном пространстве каждый документ D коллекции K соответствует точке, которая определяется вектором v_D – представление документа D:

$$v_D = \langle (t_1, \omega_1), (t_2, \omega_2), \cdots, (t_n, \omega_n) \rangle,$$

где ω_i – весовой коэффициент термина t_i.

Одним из наиболее эффективных методов вычисления весовых коэффициентов является метод *tf.idf*, по которому значение ω_i определяется следующим образом:

$$\omega_i = \begin{cases} tf.idf_{D,K}(t_i), \text{ если } t_i \in D; \\ 0, \text{ в противном случае.} \end{cases}$$

1.2. Схема вычисления весовых коэффициентов *tf.idf*

Весовой коэффициент термина по схеме *tf.idf* вычисляется как произведение нормализованной частоты встречаемости термина *tf* (*term frequency*) и обратного значения нормализованной частоты документов *idf* (*inverted document frequency*).

Коэффициент *tf* термина *t* документа *D* определяется по следующей формуле:

$$tf_D(t) = \frac{freq_D(t)}{|D|},$$

где $fred_D(t)$ – частота встречаемости термина *t* в документе *D*; $|D|$ – количество всех терминов документа *D* (размер документа *D*).

Нормализованная частота документов *df* (*document frequency*) термина *t* в корпусе *K* определяется по формуле:

$$df_K(t) = \frac{|\{D_i \mid D_i \in K, t \in D_i\}|}{|K|},$$

где $|K|$ – количество документов коллекции *K*; обратное значение нормализованной частоты документов вычисляется следующим образом:

$$idf_K(t) = \log\left(\frac{1}{df_K(t)}\right).$$

Весовой коэффициент ω_i термина t_i документа *D* в корпусе *K* вычисляется по следующей формуле:

$$tf \cdot idf_{D,K}(t) = \frac{freq_D(t)}{|D|} \cdot \log\left(\frac{|K|}{|\{D_i \mid t \in D_i, D_i \in K\}|}\right).$$

1.3. Обучение классификатора

Обучение классификатора *kNN* заключается в вычислении центральных векторов обучающих выборок. Под центральным вектором C_K обучающей выборки *K* понимается следующий вектор:

$$C_K = \frac{\sum_{D \in K} v_D}{|K|} = \langle (t_1, c_1), (t_2, c_2), \cdots, (t_n, c_n) \rangle,$$

где v_D – вектор представления документа *D*; t_i – термины; c_i – весовой коэффициент, который определяется следующим образом:

$$c_i = \frac{\sum_{D \in K} \omega_{i,D}}{|K|},$$

где $\omega_{i,D}$ – весовой коэффициент термина t_i документа *D*.

1.4. Классический алгоритм *kNN* для классификации текстов

Оценка близости классифицируемого документа *D* и обучающей выборки *K* определяется как близость их векторов представления в векторном пространстве

$$sim(D, K) = sim_{\text{vector}}(v_D, C_K).$$

На основе данной оценки близости результат классификатора *kNN* определяется следующим образом:

$$kNN(D) = K_{\max},$$

при этом $K_{\max}$ определяется по условию:

$$sim_{\text{vector}}(v_D, C_{K_{\max}}) = \max_{K \in K_i}(sim_{\text{vector}}(v_D, C_K)),$$

где K_i – множество обучающих выборок;

1.5. Оценки близости векторов

Оценка близости векторов *a* и *b* в векторном пространстве обозначается $sim_{\text{vector}}(a,b)$. Методы вычисления близости векторов представляются в таблице.

2. Классификатор *kNN* с использованием оценки семантической близости

По предлагаемой схеме для вычисления оценки семантической близости, сначала требуется определение матрицы совместной встречаемости терминов. При этом отличаются два типа оценки семантической близости: 1) терминов; 2) коллекций.

2.1. Матрица совместной встречаемости терминов

Матрица совместной встречаемости M_K коллекции *K* определяется в множестве уникальных терминов *T* следующим образом:

$$M_K(i, j) = M_K(t_i, t_j) = freq_K(t_i, t_j),$$

где $t_i, t_j \in T$; $freq_K(t_i, t_j)$ – частота совместной встречаемости, т. е. частота, по которой термин t_i встречается вместе с термином t_j в коллекции *K*.

Считается, что термин t_i встречается вместе с термином t_j, если расстояние между ними в тексте не превышает предела *z*: $|i-j| \le z$.

Таблица. *Методы вычисления оценки близости между векторами*

Название метода	Формула
Inv. Sq. City-block – обратное значения расстояния по Сити-Блок метрики (city-block distance) в квадрате	$sim_{inv.sq.city-block}(a,b) = \frac{1}{(\sum_i \mid a_i - b_i \mid)^2 + 1}$
Inv. Sq. Euclidean – обратное значение евклидова расстояния в квадрате	$sim_{inv.sq.euclidean}(a,b) = \frac{1}{\sum_i (a_i - b_i)^2 + 1}$
Cosine – косинус угла между векторами	$sim_{cosine}(a,b) = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2 \sum_i b_i^2}}$
Correlation – корреляция между векторами	$sim_{correlation}(a,b) = \frac{\sum_i (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_i (a_i - \bar{a})^2 \sum_i (b_i - \bar{b})^2}}$

Имеется следующий псевдокод алгоритма определения матрицы совместной встречаемости M_K коллекции документов K:

M_K=[0] // в начале алгоритма, все элементы матрицы равны нулю
for $D \in K$ **do**
  for $t_i \in D$ **do**
    for $j=i-z$ to $i+z$ **do**
      if i ! = j **then**
        $M_K(t_j,t_i)$++ // увеличение на единицу

Главной проблемой создания матрицы совместной встречаемости является большой требуемый объем оперативной памяти для её сохранения. В следующем разделе представляется решение данной проблемы путем фильтрации терминов.

2.2. Метод фильтрации терминов

Целью применения метода фильтрация терминов является уменьшение размера матрицы совместной встречаемости путем снижения количества терминов коллекции, при этом поддерживается минимальная потеря точности классификации.

В данной статье предлагается метод фильтрации терминов фильтрации терминов по частям речи, согласно которому из коллекции документов удаляются все термины, относящиеся к части речи, которая является исключением. Для чего, сначала требуется выполнение аннотирования текстов по частям речи (*parts of speech tagging* [5]).

Одним из эффективных инструментов для выполнения аннотирования текстов на английском языке является библиотека *Stanford CoreNLP* (*http://nlp.stanford.edu/ software/corenlp.shtml*), аналогом которой для текстов на русском языке является библиотека *MyStem* (*http://company.yandex.ru/technologies/ mystem*).

2.3. Оценка семантической близости терминов

На основе матрицы совместной встречаемости терминов M_K, значение термина t_i определяется i-й строкой матрицы M_K следующим образом:

$$mean_K(t_i) = \langle (t_1,\omega_1),(t_2,\omega_2),\cdots,(t_n,\omega_n) \rangle,$$

где $\omega_j = M_K(t_i,t_j)$ – частота совместной встречаемости терминов t_i и t_j.

Семантическая близость терминов определяется как близость между их значениями, представленными в виде векторов. Семантическая близость термина t_i коллекции K_1 и термина t_j коллекции K_2 определяется следующим образом:

$$sim_{sem}(t_{i_{K_1}},t_{j_{K_2}}) = sim_{vector}(mean_{K_1}(t_i),mean_{K_2}(t_j)).$$

Другие методы определения семантической близости терминов на основе матрицы совместной встречаемости представляются в работах [6, 7, 8].

2.4. Оценка семантической близости коллекции документов

Оценка семантической близости коллекций документов K_1 и K_2 определяется как сумма оценки семантической близости терминов следующим образом:

$$sim_{\text{sem}}(K_1,K_2) = \sum_{t\in T} sim_{\text{vector}}(mean_{K_1}(t),mean_{K_2}(t)).$$

Идентично, оценка семантической близости документа D и коллекции K вычисляется по следующей формуле:

$$sim_{\text{sem}}(D,K) = \sum_{t\in T} sim_{\text{vector}}(mean_{D}(t),mean_{K}(t)).$$

2.5. Алгоритм *kNN* с использованием оценки семантической близости

С использованием оценки семантической близости предлагается следующая комбинация (*combination*) оценок близости документа D и коллекции K:

$$sim_{comb}(D,K) = sim_{\text{vector}}(v_D,C_K) + sim_{sem}(D,K),$$

результат классификатора *kNN* определяется следующим образом:

$$kNN(D) = K_{\max},$$

при этом, $K_{\max}$ определяется по условию:

$$sim_{comb}(D,K_{\max}) = \max_{K\in K_i}(sim_{comb}(D,K)).$$

3. Эксперимент

В качестве тестовых данных используется коллекция *20Newsgroups* (*http://people.csail.mit.edu/jrennie/20Newsgroups*), которая предназначена для тестирования метода классификации. Коллекция *20Newsgroups* содержит 20 тысяч текстовых сообщений, которые разделяются на 20 тематических групп новостей. Каждая группа имеет приблизительно 1000 сообщений содержащихся 400 документов для тестирования и 600 документов для обучения классификатора.

С использованием данной коллекции были выполнены три эксперимента для проверки точности предлагаемого метода классификации и размера матрицы совместной встречаемости.

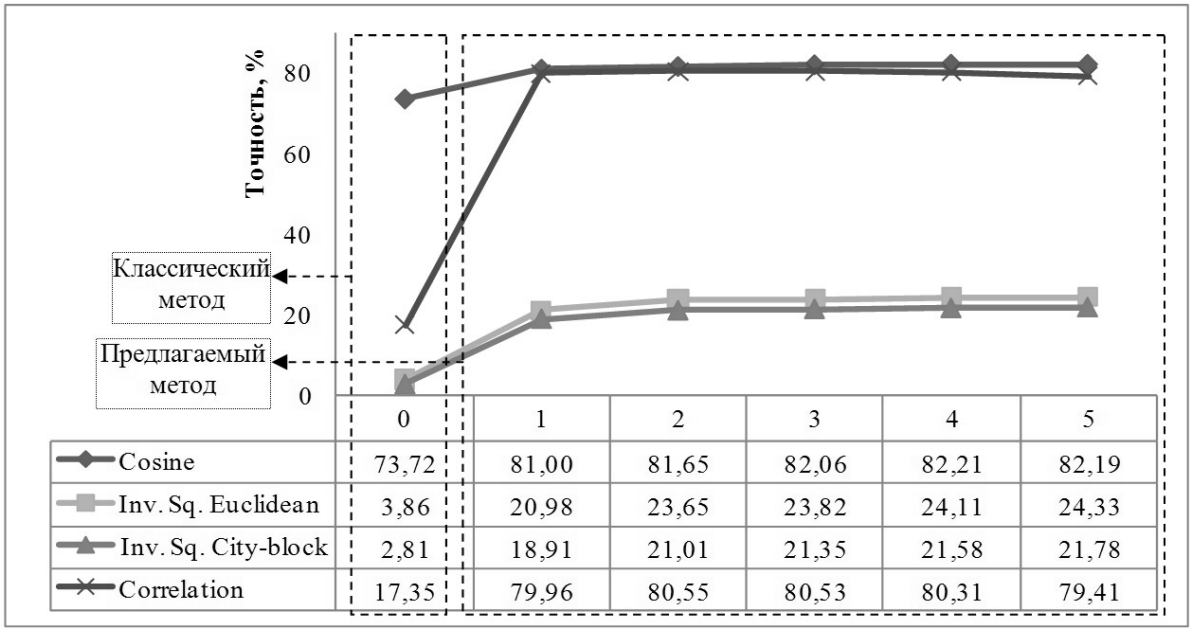
Оценка точности классификации в одной группе новостей определяется как отношение между количеством правильных результатов и общим количеством документов группы. Итоговая оценка точности классификации вычисляется как среднее значение оценок точности классификации в отдельных группах.

В данной статье предлагается определение размера матрицы совместной встречаемости M как количество её ненулевых элементов:

$$size(M) = \left|\{(t_i,t_j) \mid t_i,t_j \in T, M(t_i,t_j) > 0\}\right|.$$

3.1. Методы вычисления близости векторов

Значения точности предлагаемого алгоритма и классического алгоритма в зависимости от используемого метода вычисления близости векторов (см. таблица) и коэффициента z (см. раздел 2.1) представлены на рис. 1.



	0	1	2	3	4	5
Cosine	73,72	81,00	81,65	82,06	82,21	82,19
Inv. Sq. Euclidean	3,86	20,98	23,65	23,82	24,11	24,33
Inv. Sq. City-block	2,81	18,91	21,01	21,35	21,58	21,78
Correlation	17,35	79,96	80,55	80,53	80,31	79,41

Рис. 1. *Оценки точности классификации*

Горизонтальная ось представляет собой значения коэффициента z, а вертикальная – значения точности классификации. Четыре серии данных соответствуют значениям точности классификации при использовании разных оценок близости векторов sim_{vector}: *cosine* – косинус; *correlation* – значение корреляции; *inv.sq.eucledian* – обратное значение евклидова расстояния в квадрате; *inv.sq.city-block* – обратное значение расстояния по сити-блок метрики в квадрате.

Исходя из полученных данных, для всех методов вычисления близости векторов можно отметить существенное увеличение точности классификации в результате использовании оценки семантической близости по сравнению с классическим методом, т. е. точность классификации в случаях $z>0$ значительно выше, чем точность классификации в случае $z=0$ (разница >7 %).

Наивысшая точность для классического метода классификации (≈74 %) достигается при использовании метода вычисления близости векторов по косинусу (*cosine*), $z=0$.

По сравнению с классическим методом наивысшая точность для предлагаемого метода классификация (≈82 %) достигается при использовании метода вычисления близости векторов по косинусу и $z=4$. Кроме того, если $z>0$, то идентичные результаты также получаются с применением значения корреляции между векторами.

Самая низкая точность получается при использовании метода вычисления близости на основе сити-блок-метрики и метода вычисления близости на основе евклидова расстояния (<30 %).

3.2. Размер матрицы совместной встречаемости

Зависимость размера матрицы совместной встречаемости от метода фильтрации терминов и значения расстояния z представлены на рис. 2.

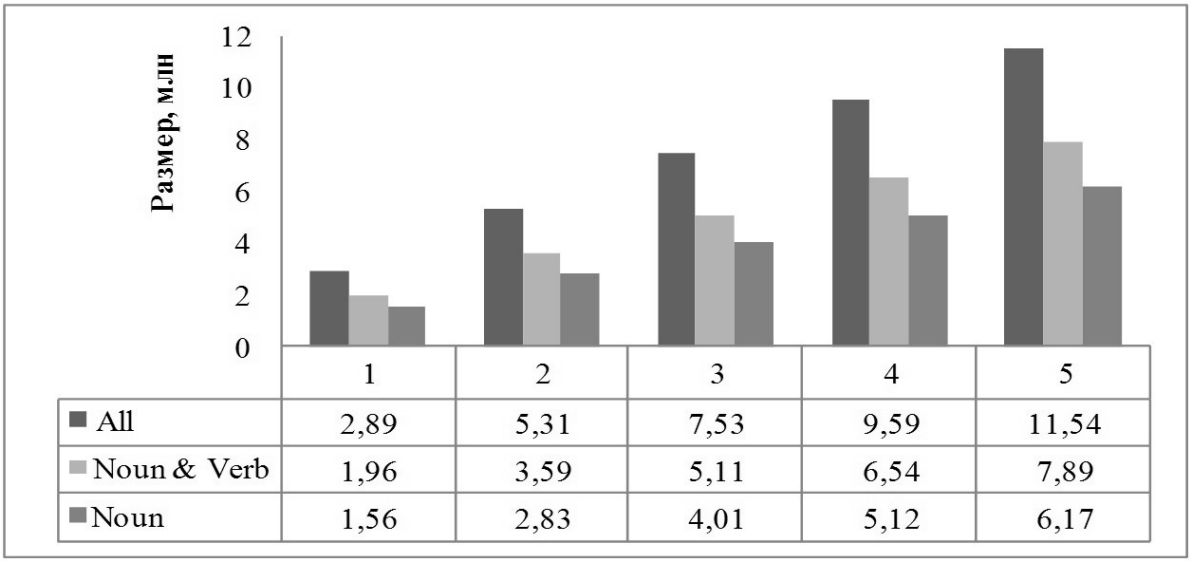
Горизонтальная ось представляет собой значения коэффициента z, а вертикальная ось – значения размера матрицы совместной встречаемости. Три серии данных соответствуют значениям размера матрицы, полученным при использовании разных методов фильтрации терминов, в каждых сохраняются: *All* – все термины; *Noun & Verb* – только существительные и глаголы; *Noun* – только существительные.

Исходя из полученных данных, видно, что в результате фильтрации терминов значительно уменьшается размер матрицы совместной встречаемости по сравнению с случаем использования всех терминов: приблизительно 30 % в случае использовании только существительных и глаголов и приблизительно 50 % в случае использования только существительных.

Кроме того, также видно, что с условием фиксированного метода фильтрации, если увеличивается расстояние z, то резко увеличивается размер матрицы совместной встречаемости (рис. 2), однако точность классификации увеличивается незначительно (рис. 3).

3.3. Точность классификатора с использованием метода фильтрации терминов

Зависимость точности классификатора от используемого метода фильтрации терминов и значения расстояния z представлено на рис. 3. В каче-



	1	2	3	4	5
All	2,89	5,31	7,53	9,59	11,54
Noun & Verb	1,96	3,59	5,11	6,54	7,89
Noun	1,56	2,83	4,01	5,12	6,17

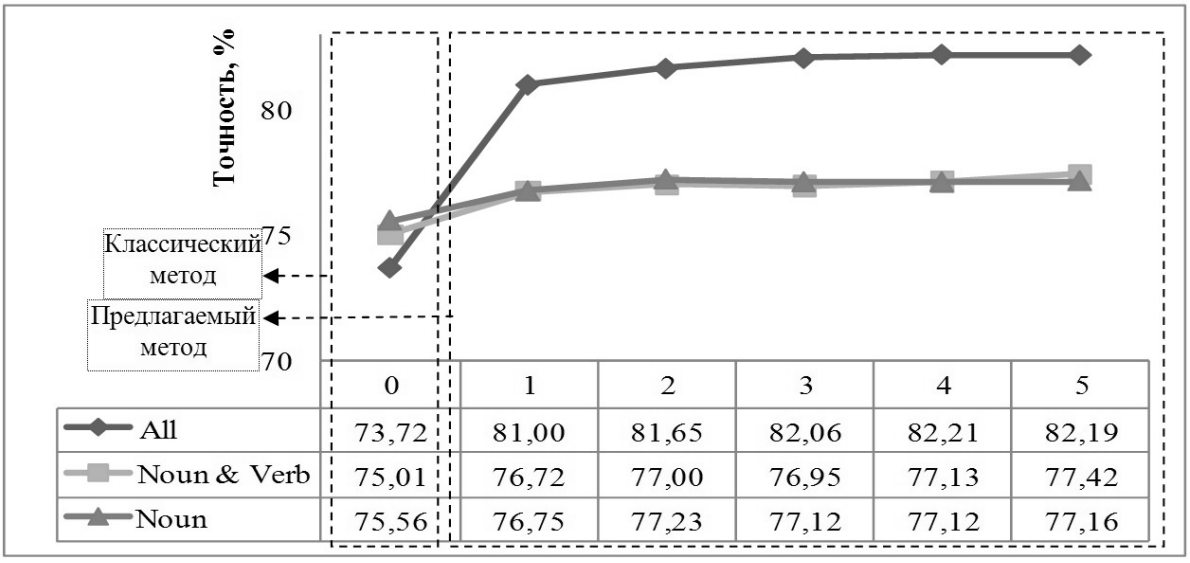
Рис. 2. *Размер матрицы совместной встречаемости*

стве оценки близости векторов используется оценка близости по косинуса, так как, исходя из первого эксперимента, она является лучшим методом определения близости векторов.

Горизонтальная ось представляет собой значения коэффициента *z*, а вертикальная ось – значения точности классификации. Три серии данных соответствуют значениям точности, полученным при использовании разных методов фильтрации терминов: *All* – используются все терминов; *Noun & Verb* – используются только существительные и глаголы; *Noun* – используются только существительные.

Исходя из полученных данных видно, что в случае z=0 (классический алгоритм *kNN*), максимальная точность (75,56 %) получается при сохранения в коллекции документов только существительных.

Однако в остальных случаях z>0 (используется оценка семантической близости), наивысшая точность (≈82 %) получается при использовании всех терминов, и если применяется метод фильтрация, то снижается точность классификации. Кроме того, в случае использования метода фильтрации терминов при увеличении z значение точности увеличивается незначительно в пределах 3...4 %, по сравнению с 7 % в случае без использования метода фильтрации (*All*).



	0	1	2	3	4	5
All	73,72	81,00	81,65	82,06	82,21	82,19
Noun & Verb	75,01	76,72	77,00	76,95	77,13	77,42
Noun	75,56	76,75	77,23	77,12	77,12	77,16

Рис. 3. *Влияние метода фильтрации на точность классификатора*

Выводы

Показано, что: 1) лучшим методом оценки векторной близости (среди рассмотренных) является оценка близости по косинусу; 2) метод фильтрации существительных имеет большое практическое значение для классического алгоритма *kNN* в векторном пространстве, так как, при сохранении только существительных из коллекции документов, точность классификации является максимальной, одновременно количество используемых терминов для классификации значительно сокращается; 3) с использованием оценки семантической близости точность классификатора *kNN* значительно увеличивается по сравнению с классической реализацией (>7 %), однако требуется дополнительные затраты для вычисления матрицы совместной встречаемости и оценки семантической близости.

СПИСОК ЛИТЕРАТУРЫ

1. Nisbet R., Elder J., Miner G. Handbook of statistical analysis and data mining applications. – Amsterdam: Academic press, 2009. – 824 p.
2. McCallum A., Nigam K. A comparison of event models for naïve bayes text classification // multinomial-aaaiws98. 1998. URL: http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf (дата обращения: 10.01.2012).
3. Joachims T. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization // Joachims_97a. 1997. URL: http://www.cs.cornell.edu/People/tj/ publications/joachims_97a.pdf (дата обращения: 10.01.2012).
4. Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features // Joachims_98a. 1998. URL: http://www.cs.cornell.edu/People/tj/ publications/joachims_98a.pdf (дата обращения: 10.01.2012).
5. Handbook of natural language processing / Ed. by N. Indurkhya, F.J. Damerau. – London: Chapman & Hall/CRC, 2010. – 692 p.
6. Douglas L.T.R., Laura M.G., David C.P. An improved method for deriving word meaning from lexical co-occurance // Cognitive Science. – 2009. – V. 7. – № 2. – P. 573–605.
7. Foltz P.W., Kintsch W., Landauer T.K. The measurement of textual coherence with Latent Semantic Analysis // Discourse Processes. – 1998. – V. 25. – № 2. – P. 285–307.
8. Lund K., Burgess C. Producing high-dimensional semantic spaces from lexical co-occurrence // Behavior Research Methods, Instruments and Computers. – 1996. – V. 28. – № 2. – P. 203–208.

Поступила 26.01.2012 г.

УДК 004.931

АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РАСПОЗНАВАНИЯ ФОРМЫ РУКИ В РЕАЛЬНОМ ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ *SURF*-ДЕСКРИПТОРОВ И НЕЙРОННОЙ СЕТИ

Нгуен Тоан Тханг, В.Г. Спицын

Томский политехнический университет
E-mail: thangngt.cntt@gmail.com; spvg@tpu.ru

Разработан оригинальный алгоритм распознавания формы руки в реальном времени на основе SURF-дескрипторов и нейронной сети. Предложен новый метод генерации дескрипторов для нейронной сети. Создано программное обеспечение для распознавания формы руки в режиме реального времени на основе предложенного алгоритма. Численные эксперименты по распознаванию формы руки на видеопоследовательности в реальном времени показали, что средняя точность распознавания составляет 92 %.

Ключевые слова:
Распознавание руки, SURF-дескриптор, обнаружение признаков, выделение признаков, описание признаков, кластеризация, многослойная нейронная сеть.

Key words:
Hand recognition, SURF-descriptor, feature detection, feature extraction, feature description, clustering, multilayer neural network.

Введение

Распознавание позы и жестов является одной из центральных задач в человеко-машинном взаимодействии и привлекает внимание многих исследователей. Для решения этой проблемы были предложены различные методы и алгоритмы. Они могут быть сгруппированы в две категории: методы на основе внешнего вида руки (*Vision-based approach*), и методы на основе 3*D* модели руки (3*D hand model based approach*) [1]. Методы на основе внешности руки используют двумерные признаки изображения для моделирования визуальной внешности руки и сравнивают эти параметры с теми же признаками, выделенными из входного изображения. В методах на основе 3*D* модели применяют 3*D* кинематические модели руки, чтобы оценить параметры руки, сравнивая эти параметры с двумерными проекциями 3*D* моделей. В первую