



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 69 за 2007 г.



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

М. Б. Гавриков, Н.В. Пестрякова,
А.В. Усков, В.В. Фарсоби́на

О некоторых свойствах
метода распознавания
символов, основанного на
полиномиальной регрессии

Статья доступна по лицензии
[Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/)



Рекомендуемая форма библиографической ссылки: О некоторых свойствах метода распознавания символов, основанного на полиномиальной регрессии / М. Б. Гавриков [и др.] // Препринты ИПМ им. М.В.Келдыша. 2007. № 69. 21 с.
<https://library.keldysh.ru/preprint.asp?id=2007-69>

Введение

Реализация оптического ввода бумажных документов основывается на превращении графического образа страницы в текст. Это типичная задача искусственного интеллекта, включающая распознавание символов, словарную обработку, элементы языковой семантики. При этом требуется высокое качество и надежность распознавания отдельных символов.

Распознавание символов начинается после завершения этапов сканирования, бинаризации и сегментации образа документа.

Системы распознавания ориентируются на печатные и рукописные образы. Рукописная информация может являться как действительно таковой, например, в банковских чеках, так и документами, заполняемыми от руки, но с ориентацией на стандартное начертание цифр и букв, сходное с большими печатными буквами определенного языка – так называемые рукопечатные символы. В настоящей работе рассматриваются именно рукопечатные цифры.

Успех в решении задачи распознавания в последние годы определяется развитием современных точных методов, одному из которых [1, 2] посвящена настоящая работа. Он основан на линейном регрессионном анализе [3 - 11]. Практика использования данного метода и сравнение его с другими способами распознавания подтвердили, что он удовлетворяет высоким требованиям по качеству распознавания, быстродействию, монотонности оценок [2]. В настоящей работе продемонстрированы некоторые свойства данного алгоритма.

Показана нелинейная зависимость оценки распознавания символа от «расстояния» между его изображением и среднестатистическим растром этого символа по базе, которая используется как для обучения, так и для распознавания.

Исследована зависимость точности и оценки распознавания от степени различия между базами обучения и распознавания. Изучение проведено для баз распознавания, которые получены при помощи четырех моделей «разрушения» обучающей базы.

1. Описание метода распознавания

Постановка задачи. Задача распознавания символов состоит в разработке алгоритма, позволяющего по данному растру изображения (рис.1) определить, какому символу из некоторого конечного множества с K элементами он соответствует. Представлением символа является растр, состоящий из $N=N_1 \times N_2$ серых или черных пикселей. Перенумеровав все пиксели растра, запоминаем в i -ой компоненте ($1 \leq i \leq N$) вектора $\mathbf{v} \in \mathbf{R}^N$ состояние i -го пикселя, а именно, 0 или 1 в случае черно-белого растра и значение на отрезке $[0,1]$ для серого растра. Пусть $\mathbf{V}=\{\mathbf{v}\}$ – совокупность всевозможных растров. Очевидно, $\mathbf{V} \subseteq \mathbf{R}^N$, причем если пиксели черно-белые, то $\mathbf{V}=\{0,1\}^N$ – конечное множество, элементами которого являются последовательности из нулей и единиц длины N . Если пиксели серые, то $\mathbf{V}=[0,1]^N$ – N -мерный единичный куб в $\mathbf{R}^N$.

Математическая постановка задачи распознавания состоит в следующем. Пусть для некоторого растра $\mathbf{v} \in \mathbf{V}$ можно найти $p_k(\mathbf{v})$ – вероятность того, что растр изображает символ с

$$p_{k_0}(\mathbf{v}) = \max_k p_k(\mathbf{v}), \quad 1 \leq k \leq K$$

Для решения задачи следует вычислить вектор вероятностей $(p_1(\mathbf{v}), p_2(\mathbf{v}), \dots, p_K(\mathbf{v}))$. Он может быть найден на основе метода наименьших квадратов.

Метод наименьших квадратов. Отождествим k -й символ с базисным вектором $\mathbf{e}_k = (0 \dots 1 \dots 0)$ (1 на k -м месте, $1 \leq k \leq K$) из $\mathbf{R}^K$, причем $\mathbf{Y} = \{\mathbf{e}_1, \dots, \mathbf{e}_K\}$. Пусть $p(\mathbf{v}, \mathbf{y})$ – вероятность наступления события $(\mathbf{v}, \mathbf{y})$, $\mathbf{v} \in \mathbf{V}$, $\mathbf{y} \in \mathbf{Y}$ (если $\mathbf{V}$ непрерывно, то плотность вероятности). Наступление события $(\mathbf{v}, \mathbf{y})$ означает, что выпадает растр $\mathbf{v}$, и этот растр изображает символ $\mathbf{y}$. Тогда вероятность $p_k(\mathbf{v})$ – это условная вероятность $p_k(\mathbf{v}) = p(\mathbf{e}_k | \mathbf{v}) = p(\mathbf{v}, \mathbf{e}_k) / \sum_{s=1}^K p(\mathbf{v}, \mathbf{e}_s)$. С другой стороны, имеет место следующий результат [6,12]:

Теорема 1. Экстремальная задача

$$E\{||\mathbf{d}(\mathbf{v})-\mathbf{y}||^2\} \xrightarrow{\mathbf{d}(\mathbf{v})} \min \quad (1)$$

достигает минимума на векторе $\mathbf{d}(\mathbf{v})=(p_1(\mathbf{v}), \dots, p_K(\mathbf{v}))$, причем $\min$ берется по всем непрерывным $\mathbf{d}: V \rightarrow \mathbf{R}^K$.

В **Теореме 1** $\|\cdot\|$ - евклидова норма в $\mathbf{R}^K$ и для любой функции $f: \mathbf{V} \times \mathbf{Y} \rightarrow \mathbf{R}$ через $E\{f(\mathbf{v}, \mathbf{y})\}$ обозначено математическое ожидание случайной величины f :

$$E\{f(\mathbf{v}, \mathbf{y})\} \stackrel{\text{def}}{=} \int_{\mathbf{V} \times \mathbf{Y}} f(\mathbf{v}, \mathbf{y}) p(\mathbf{v}, \mathbf{y}) d\mathbf{v} d\mathbf{y}$$

(в случае конечного $\mathbf{V}$ или $\mathbf{Y}$ интегрирование по соответствующей переменной заменяется на сумму). Итак, требуемый вектор вероятностей $(p_1(\mathbf{v}), \dots, p_k(\mathbf{v}))$ ищется как решение экстремальной задачи (1). Приближенное решение задачи (1) может быть найдено **методом полиномиальной регрессии**.

Метод полиномиальной регрессии. Приближенные значения компонент вектора $(p_1(\mathbf{v}), \dots, p_K(\mathbf{v}))$ будем искать в виде многочленов от координат $\mathbf{v}=(v_1, \dots, v_N)$:

$$\begin{aligned} \mathbf{p}_1(\mathbf{v}) &\cong \mathbf{c}_0^{(1)} + \sum_{i=1}^N \mathbf{c}_i^{(1)} \mathbf{v}_i + \sum_{i,j=1}^N \mathbf{c}_{i,j}^{(1)} \mathbf{v}_i \mathbf{v}_j + \dots \\ \mathbf{p}_2(\mathbf{v}) &\cong \mathbf{c}_0^{(2)} + \sum_{i=1}^N \mathbf{c}_i^{(2)} \mathbf{v}_i + \sum_{i,j=1}^N \mathbf{c}_{i,j}^{(2)} \mathbf{v}_i \mathbf{v}_j + \dots \\ &\dots\dots\dots \\ \mathbf{p}_K(\mathbf{v}) &\cong \mathbf{c}_0^{(K)} + \sum_{i=1}^N \mathbf{c}_i^{(K)} \mathbf{v}_i + \sum_{i,j=1}^N \mathbf{c}_{i,j}^{(K)} \mathbf{v}_i \mathbf{v}_j + \dots \end{aligned} \quad (2)$$

$$p_2(\mathbf{v}) \cong c_0^{(2)} + \sum_{i=1}^N c_i^{(2)} v_i + \sum_{i,j=1}^N c_{i,j}^{(2)} v_i v_j + \dots \quad (2)$$

.....

$$p_K(\mathbf{v}) \cong c_0^{(K)} + \sum_{i=1}^N c_i^{(K)} v_i + \sum_{i,j=1}^N c_{i,j}^{(K)} v_i v_j + \dots$$

Суммы в правых частях равенств (2) конечные и определяются выбором базисных мономов. А именно, если

$$\mathbf{x}(\mathbf{v}) = (1, v_1, \dots, v_N, \dots)^T$$

конечный вектор размерности L из выбранных и приведенных в (2) базисных мономов, упорядоченных определенным образом, то в векторном виде соотношения (2) можно записать так:

$$\mathbf{p}(\mathbf{v}) = (p_1(\mathbf{v}), \dots, p_K(\mathbf{v})) \cong \mathbf{A}^T \mathbf{x}(\mathbf{v}) \quad (2')$$

где $\mathbf{A}$ – матрица размера $L \times K$, столбцами которой являются векторы $\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(K)}$.

Каждый такой вектор составлен из коэффициентов при мономах соответствующей строки (2) (с совпадающим верхним индексом), упорядоченных так же, как в векторе $\mathbf{x}(\mathbf{v})$. Следовательно, приближенный поиск вектора вероятностей $\mathbf{p}(\mathbf{v})$ сводится к нахождению матрицы $\mathbf{A}$. Учитывая результат *Теоремы 1*, следует понимать, что матрица $\mathbf{A}$ является решением следующей экстремальной задачи:

$$E\{\|\mathbf{A}^T \mathbf{x}(\mathbf{v}) - \mathbf{y}\|^2\} \xrightarrow{\mathbf{A}} \min \quad (3)$$

где $\min$ берется по всевозможным матрицам $\mathbf{A}$ размера $L \times K$.

Имеет место следующий результат [4].

Теорема 2. Матрица $\mathbf{A}$ размера $L \times K$ доставляет решение экстремальной задаче (3) тогда и только тогда, когда $\mathbf{A}$ является решением матричного уравнения

$$E\{\mathbf{x}(\mathbf{v}) \mathbf{x}(\mathbf{v})^T\} \mathbf{A} = E\{\mathbf{x}(\mathbf{v}) \mathbf{y}^T\} \quad (4)$$

Итак, согласно методу полиномиальной регрессии

$$\mathbf{p}(\mathbf{v}) \cong \mathbf{A}^T \mathbf{x}(\mathbf{v}), \quad \mathbf{A} = E\{\mathbf{x}(\mathbf{v}) \mathbf{x}(\mathbf{v})^T\}^{-1} E\{\mathbf{x}(\mathbf{v}) \mathbf{y}^T\}$$

Заметим, что если $\mathbf{x}(\mathbf{v}) = (1, v_1, \dots, v_N)^T$, то метод полиномиальной регрессии называется методом **линейной регрессии** [11].

Вычисление матриц $E\{\mathbf{x}(\mathbf{v}) \mathbf{x}(\mathbf{v})^T\}$, $E\{\mathbf{x}(\mathbf{v}) \mathbf{y}^T\}$ основано на следующем результате математической статистики. Пусть имеется датчик случайных векторов, распределенных по неизвестному нам закону $p(\mathbf{v}, \mathbf{y})$:

$$[\mathbf{v}^{(1)}, \mathbf{y}^{(1)}], [\mathbf{v}^{(2)}, \mathbf{y}^{(2)}], \dots$$

Тогда математическое ожидание любой случайной величины $f(\mathbf{v}, \mathbf{y})$ может быть вычислено из предельного равенства:

$$E\{f(\mathbf{v}, \mathbf{y})\} = \lim_{J \rightarrow \infty} \frac{f(\mathbf{v}^{(1)}, \mathbf{y}^{(1)}) + f(\mathbf{v}^{(2)}, \mathbf{y}^{(2)}) + \dots + f(\mathbf{v}^{(J)}, \mathbf{y}^{(J)})}{J}$$

Если J достаточно большое, то верно приближенное равенство:

$$E\{f(\mathbf{v}, \mathbf{y})\} \cong \frac{f(\mathbf{v}^{(1)}, \mathbf{y}^{(1)}) + f(\mathbf{v}^{(2)}, \mathbf{y}^{(2)}) + \dots + f(\mathbf{v}^{(J)}, \mathbf{y}^{(J)})}{J} \quad (5)$$

Практически набор $[\mathbf{v}^{(1)}, \mathbf{y}^{(1)}], \dots, [\mathbf{v}^{(J)}, \mathbf{y}^{(J)}]$ реализуется некоторой базой данных, а вычисления по формуле (5) называются обучением. В данном случае $f(\mathbf{v}, \mathbf{y}) = x_i(\mathbf{v})x_j(\mathbf{v})$ или $f(\mathbf{v}, \mathbf{y}) = x_i(\mathbf{v})y_k$ и по определению

$$\begin{aligned} E\{\mathbf{x}(\mathbf{v}) \mathbf{x}(\mathbf{v})^T\} &= [E\{x_i(\mathbf{v})x_j(\mathbf{v})\}]_{1 \leq i, j \leq L} = \left[\sum_{s=1}^K \int_{\mathbf{v}} x_i(\mathbf{v})x_j(\mathbf{v})p(\mathbf{v}, \mathbf{e}_s) d\mathbf{v} \right]_{1 \leq i, j \leq L} \\ E\{\mathbf{x}(\mathbf{v}) \mathbf{y}^T\} &= [E\{x_i(\mathbf{v})y_k\}]_{1 \leq i \leq L, 1 \leq k \leq K} = \left[\sum_{s=1}^K \int_{\mathbf{v}} x_i(\mathbf{v})(\mathbf{e}_s)_k p(\mathbf{v}, \mathbf{e}_s) d\mathbf{v} \right]_{1 \leq i \leq L, 1 \leq k \leq K} = \\ &= \left[\int_{\mathbf{v}} x_i(\mathbf{v})p(\mathbf{v}, \mathbf{e}_k) d\mathbf{v} \right]_{1 \leq i \leq L, 1 \leq k \leq K} \end{aligned}$$

А согласно формуле (5), получим выражение для практического вычисления:

$$E\{\mathbf{x}(\mathbf{v}) \mathbf{x}(\mathbf{v})^T\} \cong \frac{1}{J} \sum_{j=1}^J \mathbf{x}^{(j)} (\mathbf{x}^{(j)})^T, \quad E\{\mathbf{x}(\mathbf{v}) \mathbf{y}^T\} \cong \frac{1}{J} \sum_{j=1}^J \mathbf{x}^{(j)} (\mathbf{y}^{(j)})^T \quad (6)$$

где $\mathbf{x}^{(j)} = \mathbf{x}(\mathbf{v}^{(j)})$, $1 \leq j \leq J$

Практическое нахождение матрицы А. Согласно (4) и (6), имеем следующее приближенное значение для А:

$$\mathbf{A} \cong \left(\frac{1}{J} \sum_{j=1}^J \mathbf{x}^{(j)} (\mathbf{x}^{(j)})^T \right)^{-1} \left(\frac{1}{J} \sum_{j=1}^J \mathbf{x}^{(j)} (\mathbf{y}^{(j)})^T \right) \quad (7)$$

В [6] показано, что правую часть (7) можно вычислить по следующей рекуррентной процедуре:

$$\begin{aligned} \mathbf{A}_j &= \mathbf{A}_{j-1} - \alpha_j \mathbf{G}_j \mathbf{x}^{(j)} [\mathbf{A}_{j-1}^T \mathbf{x}^{(j)} - \mathbf{y}^{(j)}]^T, \quad \alpha_j = 1/J \\ \mathbf{G}_j &= \frac{1}{1 - \alpha_j} \left[\mathbf{G}_{j-1} - \alpha_j \frac{\mathbf{G}_{j-1} \mathbf{x}^{(j)} (\mathbf{x}^{(j)})^T \mathbf{G}_{j-1}}{1 + \alpha_j ((\mathbf{x}^{(j)})^T \mathbf{G}_{j-1} \mathbf{x}^{(j)} - 1)} \right] \quad 1 \leq j \leq J \end{aligned} \quad (8)$$

где $\mathbf{A}_0$ и $\mathbf{G}_0$ заданы.

Введение вспомогательной матрицы $\mathbf{G}_j$ размера $L \times L$ помогает избежать процедуры обращения матрицы в (7). Реально выбор параметров α_j производится экспериментально. Вообще на практике используются следующие две упрощенные модификации процедуры (8).

Модификация А. $\mathbf{G}_j \equiv \mathbf{E}$

$$\mathbf{A}_j = \mathbf{A}_{j-1} - \frac{1}{J} \mathbf{x}^{(j)} [\mathbf{A}_{j-1}^T \mathbf{x}^{(j)} - \mathbf{y}^{(j)}]^T \quad (9)$$

$$\text{Модификация Б. } G_j \equiv D^{-1}, D = \text{diag}(E\{x_1^2\}, E\{x_2^2\}, \dots, E\{x_L^2\}) \quad (10)$$

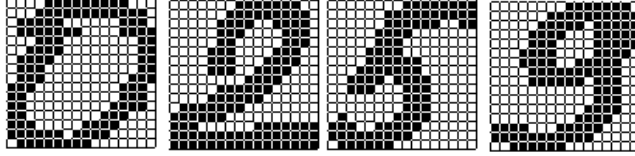


Рис. 1. Образы 16x16 рукопечатных цифр

2. Развитие и практическая реализация метода полиномиальной регрессии

Следует сразу отметить, что ниже рассматривается только **Модификация Б**, так как, в отличие от **Модификации А**, именно в этом случае получены приемлемые практические результаты. Поэтому не исследовалась постановка задачи с уравнениями, записанными в общем виде, и вследствие этого с существенно более медленными алгоритмами обучения и распознавания. Мы будем использовать серые растры размера $N=256=16 \times 16$. Масштабирование образов до размера 16x16 сохраняет особенности геометрии исходных символов (рис.1).

Построение вектора x . Используются два варианта вектора x : короткий и длинный. Для более длинного вектора больше размеры таблицы (матрицы) распознавания, медленнее осуществляется обучение и распознавание, но при этом качество распознавания выше. Длинный вектор строится следующим образом:

$$\begin{aligned} x = & (1, \{v_i\}, \{v_i^2\}, \{(\delta v_i)_r\}, \{(\delta v_i)_r^2\}, \{(\delta v_i)_y\}, \{(\delta v_i)_y^2\}, \\ & \{(\delta v_i)_r^4\}, \{(\delta v_i)_y^4\}, \{(\delta v_i)_r(\delta v_i)_y\}, \{(\delta v_i)_r^2(\delta v_i)_y^2\}, \{(\delta v_i)_r^4(\delta v_i)_y^4\}, \\ & \{(\delta v_i)_r((\delta v_i)_r)_L\}, \{(\delta v_i)_y((\delta v_i)_y)_L\}, \{(\delta v_i)_r((\delta v_i)_y)_L\}, \\ & \{(\delta v_i)_y((\delta v_i)_r)_L\}, \{(\delta v_i)_r((\delta v_i)_r)_D\}, \{(\delta v_i)_y((\delta v_i)_y)_D\}, \\ & \{(\delta v_i)_r((\delta v_i)_y)_D\}, \{(\delta v_i)_y((\delta v_i)_r)_D\}) \end{aligned} \quad (11)$$

Короткий вектор составлен из элементов длинного вектора, записанных в первой строке (11):

$$x = (1, \{v_i\}, \{v_i^2\}, \{(\delta v_i)_r\}, \{(\delta v_i)_r^2\}, \{(\delta v_i)_y\}, \{(\delta v_i)_y^2\}) \quad (12)$$

В (11) и (12) выражения в фигурных скобках соответствуют цепочкам элементов вектора, вычисляемым по всем пикселям раstra (за исключением указанных ниже случаев). Через $(\delta v_i)_r$ и $(\delta v_i)_y$ обозначены конечные центральные разности величин v_i по ортогональным направлениям ориентации раstra – нижние индексы r и y соответственно. Если имеется нижний индекс L (left) или D (down), то это означает, что соответствующие величины отно-

сются к пикселю слева или снизу от рассматриваемого. Компоненты вектора $\mathbf{x}$, не имеющие индекса L или D, вычисляются для всех пикселей раstra; с индексом L – кроме левых граничных, с индексом D – кроме нижних граничных пикселей. Вне раstra считаем, что $v_i=0$ (используется при вычислении конечных разностей на границе раstra). Для длинного вектора $\mathbf{x}$ к перечисленному в (11) добавляются компоненты, являющиеся средним арифметическим значений v_i (по восьми пикселям, окружающим данный), а также квадраты этих компонент. Отметим, что набор компонент вектора $\mathbf{x}$ подобран в процессе численных экспериментов. А именно, среди множества рассмотренных вариантов оставлены те, которые делают заметный вклад в улучшение распознавания.

Алгоритмы обучения и распознавания (модификация Б). При обучении элементы матрицы $\mathbf{D}$ (10) вычисляются следующим образом. Для каждого j -го элемента базы символов, последовательно, начиная с первого и заканчивая J -м (последним), строится вектор $\mathbf{x}^j$ согласно (11) или (12). Попутно рассчитываются значения компонент вспомогательного вектора $\mathbf{m}^j$ по рекуррентной формуле:

$$m_p^j = (1 - \beta_j) m_p^{j-1} + \beta_j (x_p^j)^2, \quad j=1, \dots, J, \quad p=1, \dots, L, \quad (13)$$

$$\beta_j = 1/j$$

По окончании этой процедуры для последнего элемента имеем согласно (10):

$$\mathbf{G}_J \equiv \mathbf{D}^{-1} = \text{diag} (1/m_1^J, 1/m_2^J, \dots, 1/m_L^J) \quad (14)$$

После того, как завершено вычисление $\mathbf{G}_J$, элементы матрицы $\mathbf{A}_j$ (8) рассчитываются следующим образом. Выполняется еще один проход по базе и для каждого j -го элемента базы символов, начиная с первого и заканчивая J -м, строится вектор $\mathbf{x}^j$ согласно (11) или (12). Попутно вычисляется $\mathbf{A}_j$:

$$a_j^{pk} = a_{j-1}^{pk} - \alpha_j x_k^j \left(\sum_{i=1}^L a_{j-1}^{ik} x_i^j - y_k^j \right) / m_p^j, \quad \alpha_j = 1/J \quad (15)$$

$$\mathbf{A}_j = [a_j^{pk}], \quad j = 1, \dots, J, \quad j=1, \dots, J, \quad p = 1, \dots, L, \quad k = 1, \dots, K$$

На этом этап обучения закончен: матрица $\mathbf{A}=\mathbf{A}_J$ получена.

Распознавание осуществляется следующим образом. Для серого изображения размером 16x16 пикселей строится вектор $\mathbf{x}$ согласно (11) или (12). После этого по формуле (2'), используя $\mathbf{A}=\mathbf{A}_J$ (15), вычисляются оценки, соответствующие каждому из возможных символов. Затем выбирается символ с максимальной оценкой. Получаемые оценки могут выходить за рамки отрезка $[0,1]$ из-за того, что используемый метод является приближенным. Мы искусственно обнуляли отрицательные значения и делали равными 1 те, которые были больше этой величины. Практика распознавания показала приемлемость такого довольно грубого способа коррекции оценок.

3. Особенности распознавания символов обучающей базы

При анализе программной реализации метода распознавания имеет смысл проводить как обучение, так и распознавание на одной и той же графической базе данных, содержащей изображения и коды символов,. Это служит своего рода гарантом «чистоты эксперимента», поскольку результат распознавания на символьных последовательностях, «посторонних» для обучающей базы, может сильно отличаться для разных последовательностей. Причиной, видимо, является большая или меньшая степень схожести базы обучения и распознавания. Данная проблема будет рассматриваться в п.4.

В настоящей работе при обучении и распознавании использовалась модификация длинного вектора x – см. (11). После многократного обучения по базе в 174 778 элементов полученная матрица на той же базе обеспечивает распознавание 99,5% элементов (881 символ распознан неверно).

Результатом распознавания образа является код символа и его целочисленная оценка, лежащая в диапазоне $[0,255]$ (оценка 255 является наилучшей). Эта новая оценка получается следующим образом. В результате умножения оценки на 255 старый диапазон оценок $[0,1]$ (см. пп.1, 2) переходит в новый $[0,255]$, после чего проводится дискретизация, а именно, $[0,1] \rightarrow 1, (1,2] \rightarrow 2, \dots, (254,255] \rightarrow 255$.

На рис.2а-11а представлены графики зависимости средней оценки распознавания рукопечатного символа (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) от величины отклонения между его изображением и «среднестатистическим» растром этого символа по базе, которая используется как для обучения, так и для распознавания.

«Среднестатистический» растр конкретного символа получаем следующим образом. Значение в каждом пикселе, имеющем номер i , равно среднему арифметическому значений i -х пикселей по всем имеющимся в базе растрам рассматриваемого символа. Расстояние между двумя растрами $\mathbf{v}=(v_1, \dots, v_N)$ и $\mathbf{u}=(u_1, \dots, u_N)$ определяем так: вычисляем модуль разности значений в i -х пикселях, затем суммируем по всем пикселям.

$$\|\mathbf{v}-\mathbf{u}\| = \sum_{i=1}^N |v_i - u_i| \quad (16)$$

Диапазон отклонений между изображением распознанного верно символа и среднестатистическим растром этого символа по рассматриваемой базе меняется от минимального true_min до максимального true_max . В таблице 1 приведены значения этих величин для каждого из символов 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Делим отрезок $[\text{true_min}, \text{true_max}]$ (оси абсцисс на рис.2а-11а) на 20 равных по длине частей – отрезок и 19 полуинтервалов: $[\text{true_min}, \text{true_min} + dv]$, $(\text{true_min} + dv, \text{true_min} + 2dv]$, $\dots$, $(\text{true_min} + 19dv, \text{true_min} + 20dv]$, где $dv = (\text{true_max} - \text{true_min})/20$. Затем для совокупности растров, попадающих в каждый такой участок, вычисляем среднюю оценку распознавания (оси ординат на рис. 2а-11а). На этих рисунках видно, что средняя оценка распознавания ни для одного из рассматриваемых символов не убывает монотонно по мере «удаления» растра от «среднестатистического». а для «1» принимает максимальное значение 255 на наибольшем удалении от среднестатистического образа.

На рис.26-116 приведен «дискретный» аналог функции распределения для распознанных верно растров каждого из символов 0, 1, ... , 9. А именно, ось абсцисс такая же, как указано ранее, а по оси ординат отложено количество правильно распознанных растров, попавших в каждую вышеописанную двадцатую часть отрезка [true_min, true_max].

Для неправильно распознанных символов диапазон отклонений между изображением символа и среднестатистическим растром этого символа по рассматриваемой базе находится от минимального false_min до максимального false_max. В таблице 1 приведены значения этих величин для каждого из символов 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, а также средняя оценка неправильного распознавания. Это оценка, с которой один из перечисленных символов принимается за какой-то другой. Следовательно, оценка, с которой он принимается сам за себя, еще ниже. Но даже приведенные оценки неправильного распознавания намного меньше (приблизительно в два раза), чем оценки правильного распознавания. Отметим, что диапазон [true_min, true_max] отличается от [false_min, false_max] для каждого из рассматриваемых символов, но не очень существенно. Следовательно, поскольку доля неправильно распознанных символов весьма незначительна, «дискретный» аналог функции распределения для всех растров (распознанных как верно, так и неверно) каждого из символов 0, 1, ... , 9 мало отличается от приведенных на рис.26-116.

Таблица 1

символ	true_min	true_max	false_min	false_max	средняя оценка неправильного распознавания
0	35,41	113,59	56,34	101,09	106,61
1	42,58	173,80	52,22	131,38	128,17
2	38,33	105,57	61,62	109,43	120,19
3	39,92	103,10	55,63	95,53	118,84
4	50,76	106,34	56,02	123,47	131,78
5	36,28	130,66	52,68	98,26	126,65
6	44,60	115,07	55,17	103,11	105,26
7	40,56	101,70	53,45	93,89	114,43
8	50,15	119,80	57,23	115,19	121,27
9	47,36	120,58	54,28	117,37	127,45

Были проведены расчеты и получены средние оценки распознавания и дискретные аналоги функции распределения при делении отрезка [true_min, true_max] на 50 равных частей. Результаты аналогичны приведенным на рис.2-11.

Итак, можно сделать следующие выводы. В среднем оценка распознавания не зависит от удаленности раstra символа от «среднестатистического» раstra данного символа по рассматриваемой базе обучения и распознавания. Средняя оценка правильного распознавания существенно выше (приблизительно в два раза), чем средняя оценка неправильного распознавания.

4. Анализ качества распознавания на различных базах символов

Интуитивно понятно, что любая база распознавания – это каким-либо образом «испорченная» база обучения. На практике бывает трудно определить, существует ли некая закономерность преобразования базы обучения в базу распознавания. В настоящей работе мы сами предлагаем четыре модели «разрушения» базы обучения, и в рамках каждой из них исследуем зависимость качества распознавания от степени «искажения» обучающей базы.

Будем анализировать поведение таких характеристик качества распознавания символов, как оценка распознавания (средняя по всем верно распознанным растрам базы), а также число неправильно распознанных образов по базе.

Скорость (абсолютная, средняя) между двумя точками на графике равна модулю изменения величины, откладываемой по оси ординат, отнесенному к модулю изменения величины, откладываемой по оси абсцисс.

Модель «наихудшего» разрушения. В обучающей базе на этапе распознавания для каждого растра выбирается случайным образом определенное количество пикселей, в которых v_i , лежащие в полуинтервале $0 \leq v_i < 0,5$, заменяются значением 1: $[0; 0,5) \rightarrow 1$, а при $0,5 \leq v_i \leq 1$ величиной 0: $[0,5; 1] \rightarrow 0$. Число пикселей, в которых производятся такие модификации, варьируется от 1 до некоторого значения, в данном случае 10, при котором преодолевается «порог» количества неправильно распознанных растров (мы устанавливаем его равным удвоенному числу изображений, нераспознанных на самой базе обучения, а именно, $881 \cdot 2 = 1762$). При каждом фиксированном количестве k ($1 \leq k \leq 10$) заменяемых пикселей осуществляется распознавание по всей базе. При построении графиков формально добавляется значение числа измененных пикселей, равное 0, что соответствует распознаванию обучающей базы.

Модель «случайного» разрушения. В обучающей базе на этапе распознавания для каждого растра выбирается случайным образом определенное количество пикселей, в которых v_i , лежащее на отрезке $0 \leq v_i \leq 1$, заменяется на случайное значение, также принадлежащее отрезку $[0,1]$. Число пикселей, в которых производятся такие модификации, варьируется от 1 до 256, поскольку в данном случае не преодолевается «порог» количества неправильно распознанных растров ($881 \cdot 2 = 1762$).

Здесь и выше используется генератор псевдослучайных чисел. Так в каждом текущем растре для случайного выбора k различных пикселей из имеющихся перенумерованных от 0 до 255 (т.е. всего 256 пикселей), выполняются следующие действия. Для натурального числа, предлагаемого генератором псевдослучайных чисел, находим остаток от деления на 256. Он является целым и принадлежит требуемому отрезку $[0,255]$. Дополнительно проводится проверка отличия нового номера от полученных прежде.

Аналогично, при выборе случайного значения v_i , принадлежащего отрезку $[0,1]$, для натурального числа, предлагаемого генератором псевдослучайных чисел, находим остаток от деления на 101. Он является целым и принадлежит отрезку $[0,100]$. Делим его на 100. Результат принадлежит отрезку $[0,1]$.

На рис.12а,б для сравнения приводятся результаты, полученные при «наихудшем» и «случайном» разрушении для числа пикселей, в которых производятся такие модификации, изменяющемся от 1 до 10. Также добавлен 0, что соответствует отсутствию модификаций.

На рис.12а на отрезке для числа «испорченных» пикселей $[0, 10]$ показано линейное падение средней оценки распознавания для «наихудшего» разрушения; для «случайного» разрушения убывание почти линейное, но средняя скорость его приблизительно в два раза меньше. В обоих случаях зависимости являются строго монотонными.

На рис.12б можно видеть монотонное нарастание количества неправильно распознанных растров при увеличении числа «испорченных» «наихудшем» образом пикселей. Когда их количество становится равным 10, число неправильно распознанных растров преодолевает «пороговое» значение и становится равным 1835. Для «случайного» разрушения график является почти монотонно нарастающим (наблюдаются небольшие колебания) и близким к линейному. При 10 модифицированных пикселях количество нераспознанных растров увеличивается незначительно, до значения 994. Т.е. на рассматриваемом отрезке для числа «испорченных» пикселей $[0, 10]$ средняя скорость увеличения числа неправильно распознанных растров для «наихудшего» разрушения более чем в семьдесят три раза превышает соответствующий показатель при «случайном» разрушении.

Как уже было отмечено ранее, при «случайном» разрушении пикселей «пороговое» количество нераспознанных растров не достигается даже при модификации всех пикселей. На рис. 13а видно, что средняя оценка распознавания монотонно убывает на максимально возможном отрезке для числа «испорченных» пикселей $[0, 256]$, но скорость падения этой величины на участке $[16, 40]$ существенно замедляется и далее остается очень незначительной. Количество нераспознанных растров увеличивается немонотонным образом, но в среднем на участке $[0, 16]$ увеличивается с большой скоростью, средняя скорость нарастания этой величины на участке $[16, 40]$ существенно замедляется и далее становится равной нулю, но при этом наблюдаются колебания около оценки 205,7.

Анализ поведения оценки распознавания и числа нераспознанных растров на моделях «наихудшего» и случайного разрушения позволяют нам делать выводы о поведении аналогичных характеристик для моделей разрушения, лежащих «между» двумя изученными. Очевидно, что с увеличением числа разрушенных пикселей средняя оценка распознавания будет монотонно падать, а число нераспознанных символов увеличиваться, но в среднем, немонотонно. Сделанные выводы позволяют строить следующие модели разрушения уже для всех пикселей раstra, без дополнительного анализа влияния увеличения их числа.

Модели затемнения (засветления). На этапе распознавания все пиксели раstra постепенно затемняются: $v_i \rightarrow v_i + 0,01 \cdot n$, где $n=0,1,2,...,n_{\text{тсмн}}$. Если для каких-то пикселей начиная с некоторого n имеем: $v_i \geq 1$, то считаем, что $v_i=1$. При засветлении аналогично $v_i \rightarrow v_i - 0,01 \cdot n$, где $n=0,1,2,...,n_{\text{свтл}}$. Если для каких-то пикселей начиная с некоторого n имеем: $v_i \leq 0$, то считаем, что $v_i=0$. При $n=n_{\text{тсмн}}=12$ и аналогично при $n=n_{\text{свтл}}=20$ число неправильно распознанных растров преодолевает «пороговое» значение ($881 \cdot 2 = 1762$).

На рис.14а,б для сравнения приводятся результаты распознавания, полученные при затемнении и засветлении растров. По оси абсцисс откладывается число n , определяющее степень затемнения (засветления). Распознаванию обучающей базы соответствует $n=0$.

На рис.14а показано почти линейное падение оценки распознавания для затемнения и засветления, причем на отрезке $[0, 12]$ средняя скорость второго в 4,2 раза меньше. В обоих случаях зависимости являются строго монотонными.

На рис.14б можно видеть немонотонное изменение количества неправильно распознанных растров при затемнении. Дополнительные расчеты показали, что при $n=2,5$ достигается

минимальное число 823 неправильно распознанных символов. При $n = n_{\text{темн}} = 12$ число неправильно распознанных растров преодолевает «пороговое» значение и становится равным 1783. Для засветления график является монотонным. При $n = n_{\text{светл}} = 20$ число неправильно распознанных растров преодолевает «пороговое» значение и становится равным 1747.

Итак, затемнение и засветление растров оказывают различное влияние на распознавание. Оказывается, при небольшом затемнении (до $n = 6$) количество нераспознанных символов даже меньше, чем в немодифицированной базе. Однако, в результате наблюдающегося затем стремительного роста количества нераспознанных символов при затемнении, увеличение их числа по сравнению с распознаванием исходной базы для $n = 12$ при затемнении приблизительно в 4 раза больше, чем при засветлении. Оценка распознавания как для затемнения, так и для засветления, монотонно уменьшается с ростом степени изменения исходных растров, причем при $0 \leq n \leq 12$ для второго процесса в 4,2 раза медленнее, чем для первого.

Модель «дискретизации». В рассматриваемых «серых» растрах для каждого пикселя $0 \leq v_i \leq 1$. Поделим отрезок $[0,1]$ на 256 равных по длине частей – отрезок и 255 полуинтервалов: $[0, dv]$, $(dv, 2 \cdot dv]$, ..., $(255 \cdot dv, 256 \cdot dv]$, где $dv = 1/256$. Осуществим следующее преобразование для всех пикселей растра. Если $0 \leq v_i \leq dv$, то $v_i \rightarrow dv/2$ (иначе $[0, dv] \rightarrow dv/2$); в полуинтервале $k \cdot dv < v_i \leq (k+1) \cdot dv$, где $k=1, \dots, 255$, производится замена: $v_i \rightarrow (k+1/2) \cdot dv$ (иначе $(k \cdot dv, (k+1) \cdot dv] \rightarrow (k+1/2) \cdot dv$). Мы осуществили дискретизацию бесконечного множества значений $0 \leq v_i \leq 1$, в результате которой v_i будет принимать только 256 значений: $\{dv/2, (1+1/2) \cdot dv, \dots, (255+1/2) \cdot dv\}$. Выполним распознавание полученной базы символов, которая, как нетрудно понять, очень незначительно отличается от исходной базы.

Произведем аналогичную дискретизацию с делением отрезка $[0,1]$ на 128 частей, затем на 64 части, далее на 32 части, на 16, на 8 и наконец на 4. От базы к базе количество отрезков дискретизации уменьшалось в 2 раза. Каждая последующая база все больше отличается от исходной.

На рис.15а,б приводятся результаты, полученные при «дискретизации». По оси абсцисс откладывается число отрезков дискретизации, а именно 256, 128, 64, 32, 16, 8. Распознаванию обучающей базы соответствует число 256.

На рис.15а показано строго монотонное падение оценки распознавания при уменьшении количества отрезков дискретизации.

На рис.15б изображен график немонотонного изменения количества неправильно распознанных растров. Дополнительные расчеты при 12 и 24 отрезках дискретизации подтвердили, что на 16 отрезках дискретизации достигается минимальное число 815 неправильно распознанных символов. Их количество увеличивается (не достигая порогового значения) при дальнейшем уменьшении числа отрезков дискретизации вплоть до 8.

На рис.16а,б также приводятся результаты, полученные при дискретизации, но уже для числа отрезков дискретизации 256, 128, 64, 32, 16, 8, 4 (добавилось число 4).

На рис.16а по-прежнему строго монотонное падение оценки распознавания при уменьшении количества отрезков дискретизации.

На рис.16б изображен график немонотонного изменения количества неправильно распознанных растров. При 4 отрезках дискретизации число неправильно распознанных растров преодолевает «пороговое» значение и становится равным 2806. Скорость роста на отрезке $[8, 4]$ по оси абсцисс существенно выше (в 40 раз), чем на отрезке $[16, 8]$.

Итак, анализ результатов распознавания, полученных с использованием модели «дискретизации», показывает, что оценка распознавания монотонно уменьшается по мере «удаления» базы распознавания от базы обучения, соответствующего убыванию числа отрезков дискретизации. Количество нераспознанных символов ведет себя немонотонным образом. Вначале при уменьшении числа отрезков дискретизации оно уменьшается и достигает минимума при количестве таких отрезков, равном 16. Затем число нераспознанных символов увеличивается (причем с нарастающей скоростью) и преодолевает пороговое значение.

5. Выводы

Изучение «поведения» описанного метода распознавания, основанного на полиномиальной регрессии, при распознавании рукопечатных цифр по базе, совпадающей с базой обучения, для каждого из рассматриваемых символов показало следующее. В среднем оценка распознавания не зависит от «удаленности» раstra символа от «среднестатистического» раstra данного символа по рассматриваемой базе. Средняя оценка правильного распознавания символа существенно выше (приблизительно в два раза), чем средняя оценка неправильного распознавания.

«Превращение» базы обучения в базу распознавания формализовано в виде четырех моделей. Для каждой из них получено, что по мере увеличения степени «искажения» обучающей базы оценка распознавания монотонно уменьшается, а количество нераспознанных символов в целом увеличивается, но этот процесс может протекать существенно немонотонным образом.

При реальном распознавании можно использовать полученные наработки для улучшения качества распознавания, но уже в «обратном» порядке. Разумеется, имеются в виду только те модели, что применяются для всех пикселей раstra, а не для некоторых случайным образом выбранных. Предложенные способы преобразования (затемнение, засветление, «дискретизацию») можно применить к распознаваемой последовательности символов. Из вышеописанного следует, что результатом может быть либо улучшение оценок распознавания, либо уменьшение числа нераспознанных символов, либо и то, и другое. Возможно комбинированное использование затемнения (засветления) и «дискретизации».

Литература

- [1] *Гавриков М.Б., Пестрякова Н. В.* "Метод полиномиальной регрессии в задачах распознавания печатных и рукопечатных символов", //Препринт ИПМатем. АН СССР, М., 2004, №22, 12 стр.
- [2] *Гавриков М.Б., Мисюрёв А.В., Пестрякова Н.В., Славин О.А.* Развитие метода полиномиальной регрессии и практическое применение в задаче распознавания символов. Автоматика и Телемеханика. 2006, №3, С. 119-134.
- [3] *Sebestyen G.S.* Decision Making Processes in Pattern Recognition, MacMillan, New York, 1962.
- [4] *Nilson N. J.* Learning Machines, McGraw-Hill, New York, 1965.
- [5] *Schürmann J.* Polynomklassifikatoren, Oldenbourg, München, 1977.
- [6] *Schürmann J.* Pattern Classification, John Wiley&Sons, Inc., 1996.
- [7] *Albert A.E. and Gardner L.A.* Stochastic Approximation and Nonlinear Regression // Research Monograph 42. MIT Press, Cambridge, MA, 1966.
- [8] *Becker D. and Schürmann J.* Zur verstärkten Berücksichtigung schlecht erkennbarer Zeichen in der Lernstichprobe // Wissenschaftliche Berichte AEG-Telefunken **45**, 1972, pp. 97 – 105.
- [9] *Pao Y.-H.* The Functional Link Net: Basis for an Integrated Neural-Net Computing Environment // in Yoh-Han Pao (ed.) Adaptive Pattern Recognition and Neural Networks, Addison-Wesley, Reading, MA, 1989, pp. 197-222.
- [10] *Franke J.* On the Functional Classifier, in Association Francaise pour la Cybernetique Economique et Technique (AFCET), Paris // Proceedings of the First International Conference on Document Analysis and Recognition, St. Malo, 1991, pp.481-489.
- [11] Дж.Себер. Линейный регрессионный анализ. М.:”Мир”, 1980.
- [12] Ю.В.Линник. Метод наименьших квадратов и основы математико - статистической теории обработки наблюдений. М.:”Физматлит”, 1958.

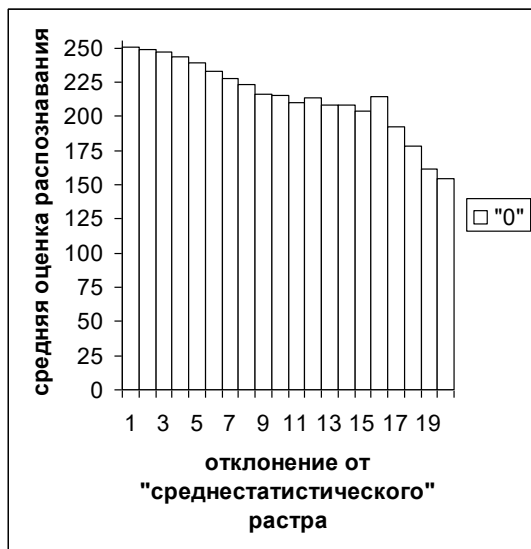


Рис. 2а

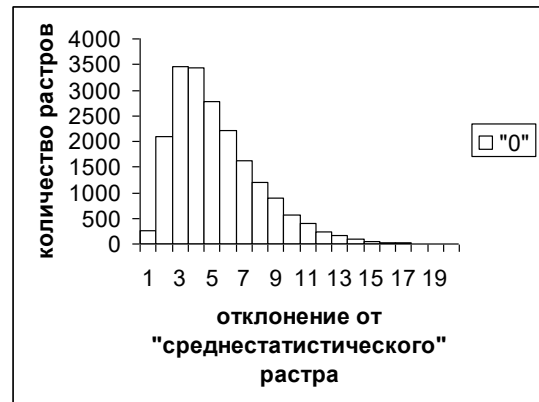


Рис. 2б

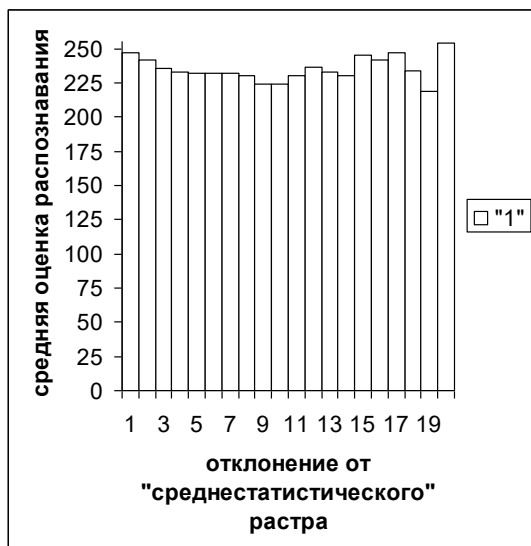


Рис. 3а

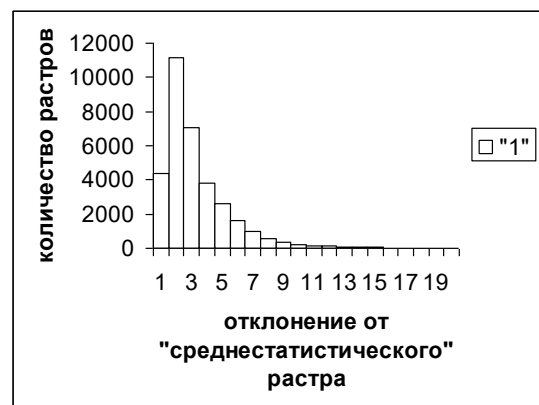


Рис. 3б

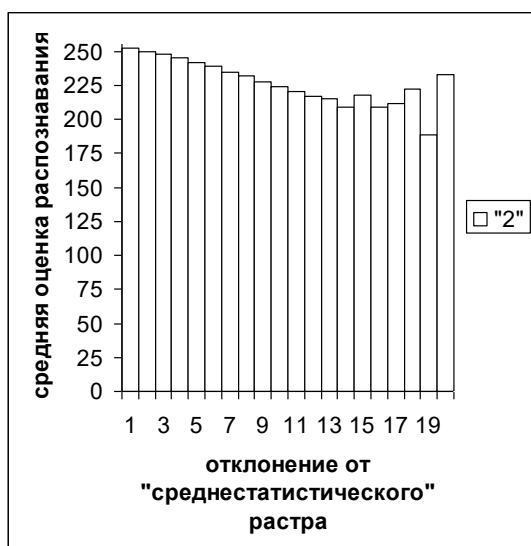


Рис. 4а

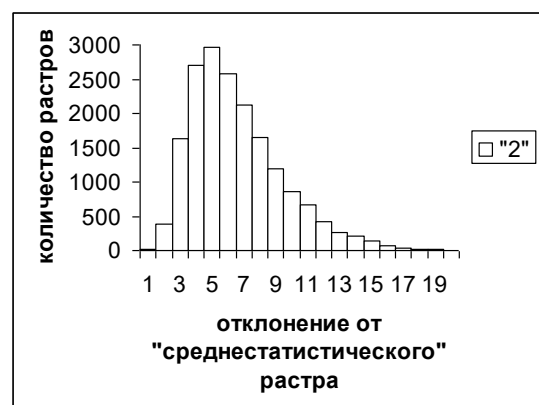


Рис.4б

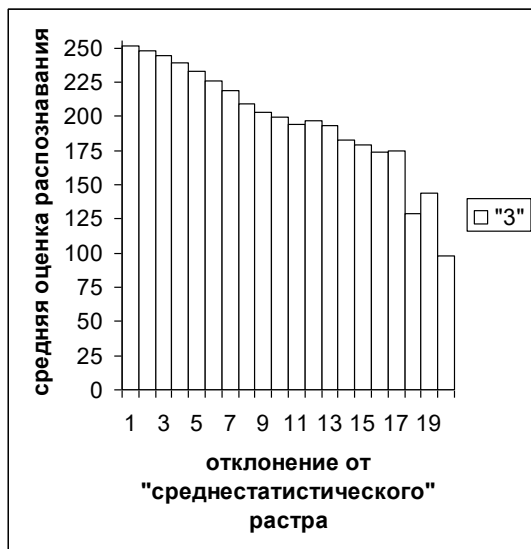


Рис. 5а

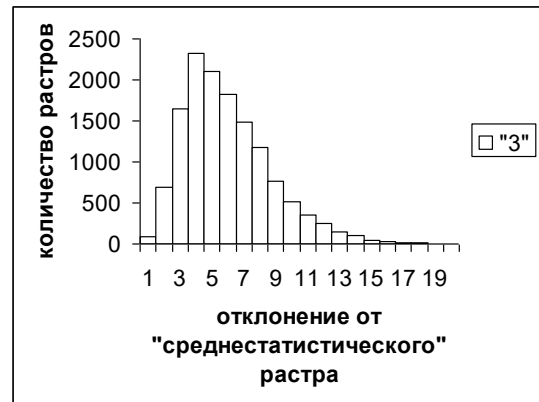


Рис. 5б

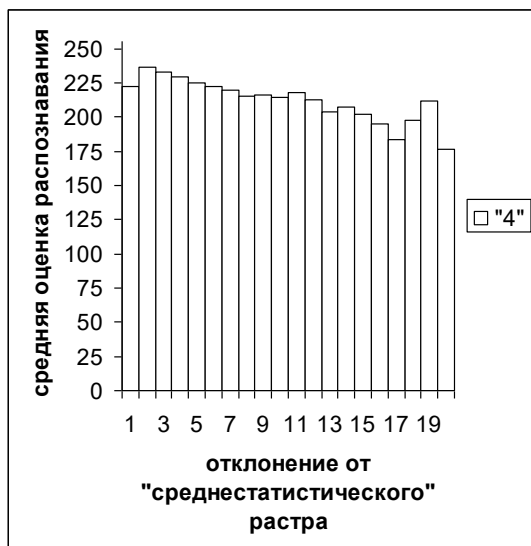


Рис. 6а

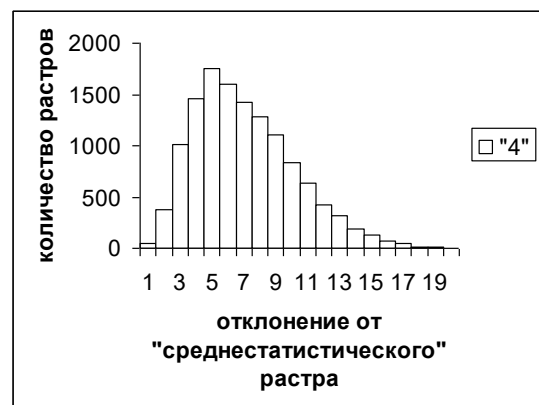


Рис. 6б

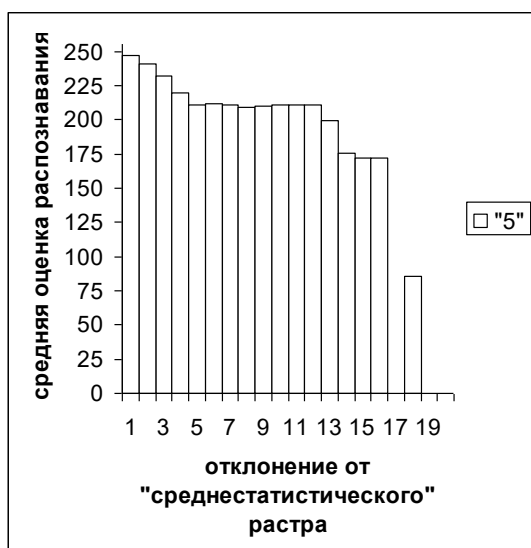


Рис. 7а

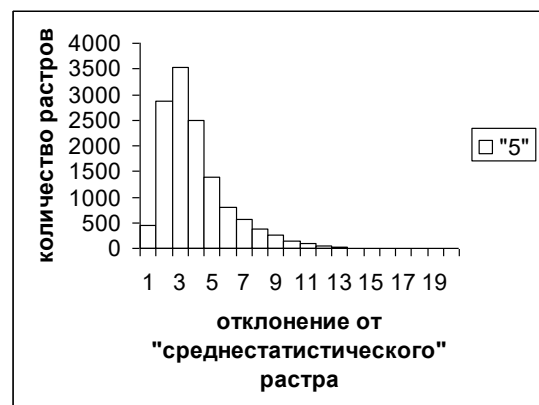


Рис. 7б

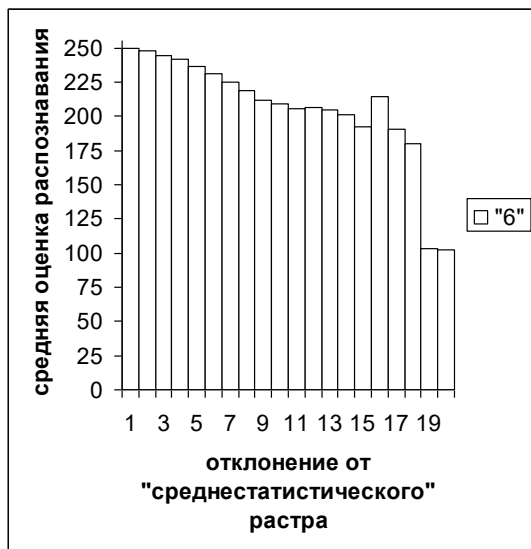


Рис. 8а

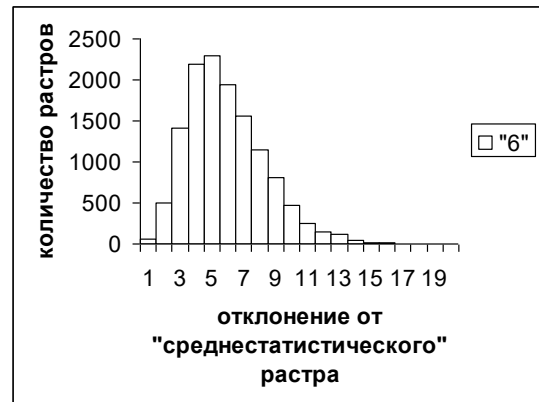


Рис. 8б

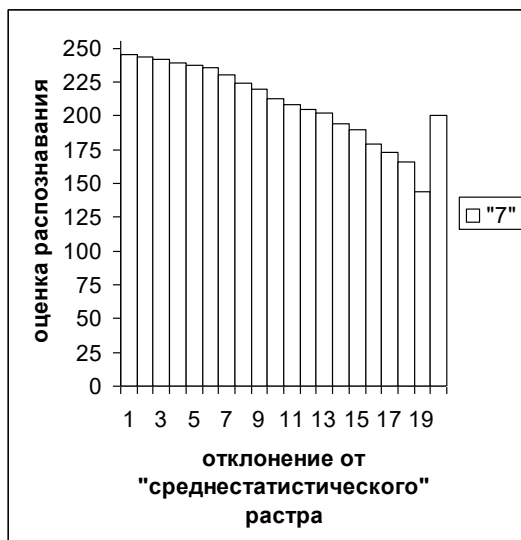


Рис. 9а

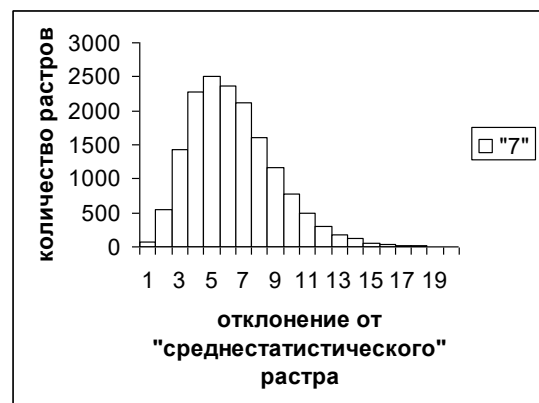


Рис. 9б

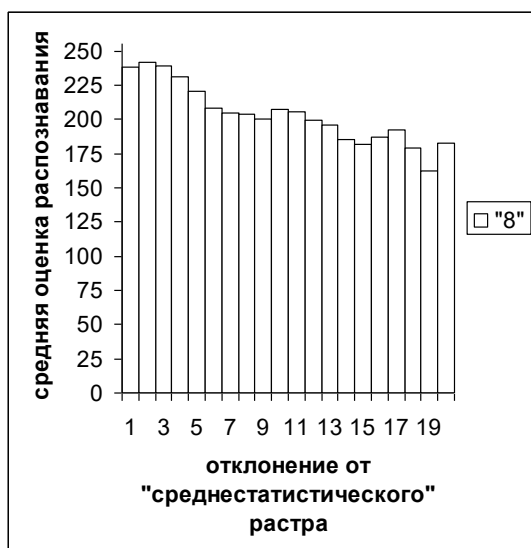


Рис. 10а

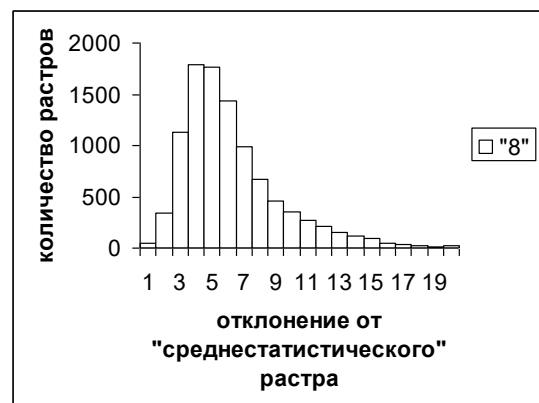


Рис. 10б

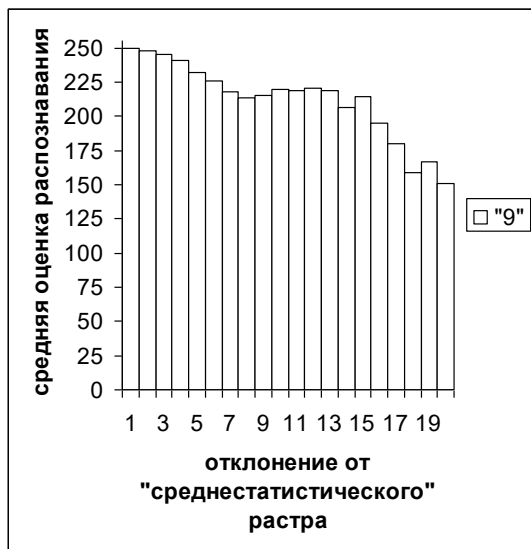


Рис. 11а

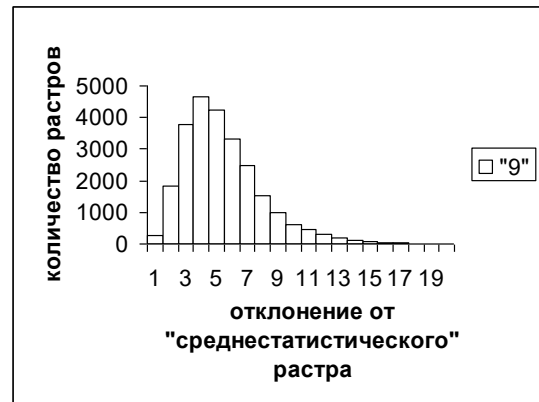


Рис. 11б

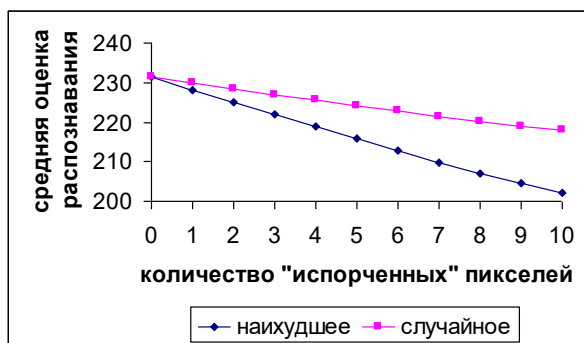


Рис. 12а

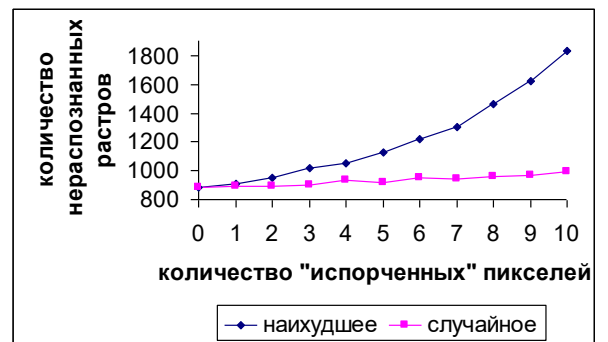


Рис. 12б

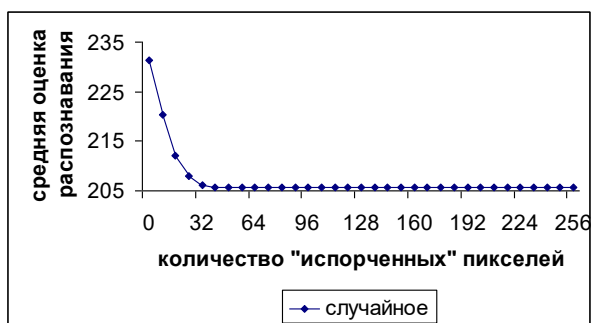


Рис. 13а

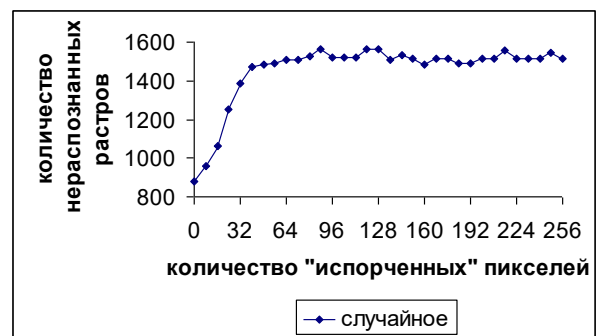


Рис. 13б

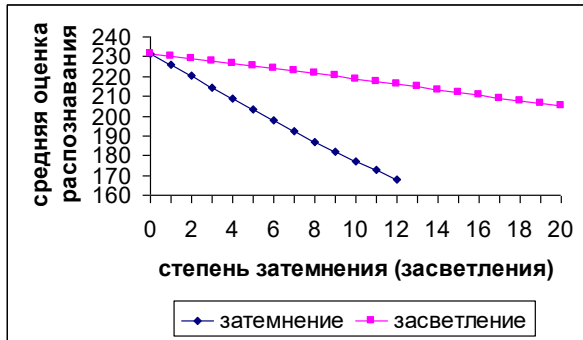


Рис. 14а

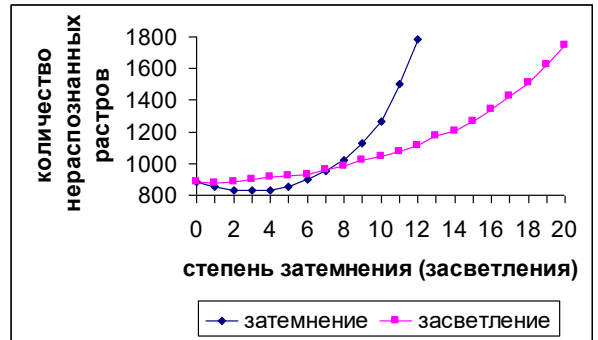


Рис. 14б

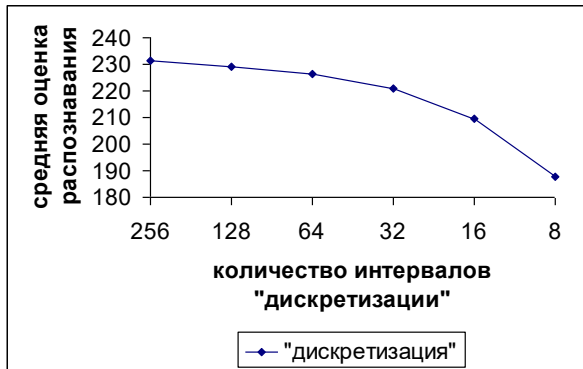


Рис. 15а

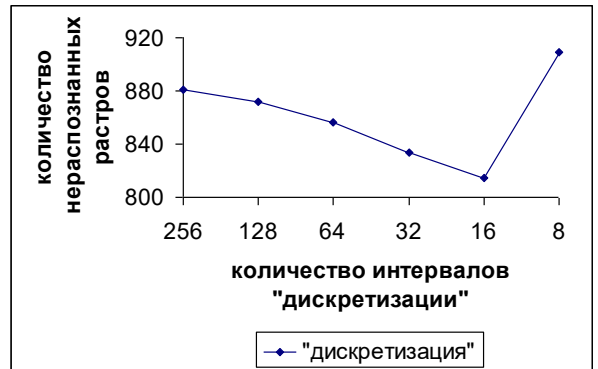


Рис. 15б

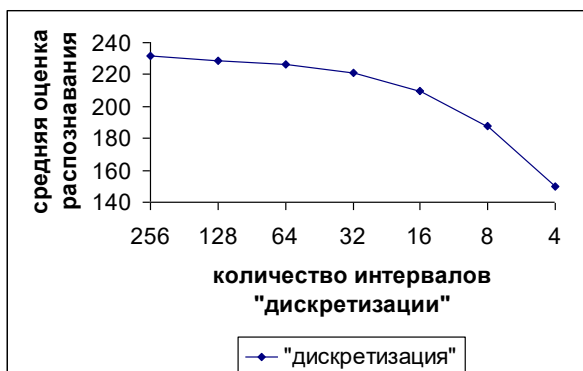


Рис. 16а

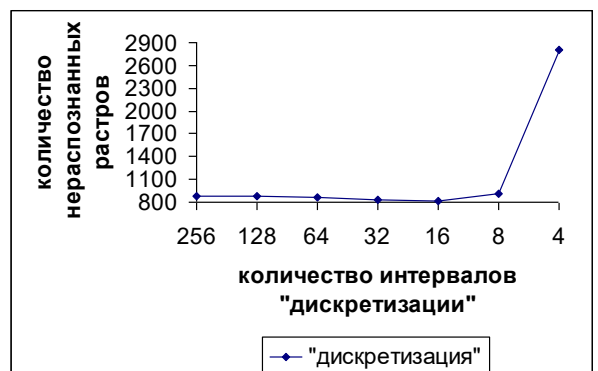


Рис. 16б