



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

О некоторых свойствах
оценки метода
распознавания символов,
основанного на
полиномиальной регрессии

Рекомендуемая форма библиографической ссылки: О некоторых свойствах оценки метода распознавания символов, основанного на полиномиальной регрессии / М.Б.Гавриков [и др.] // Препринты ИПМ им. М.В.Келдыша. 2008. № 7. 29 с. URL: <http://library.keldysh.ru/preprint.asp?id=2008-7>

Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В. Келдыша
Российской Академии наук

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

**О некоторых свойствах оценки метода
распознавания символов, основанного на
полиномиальной регрессии**

Москва - 2008

М.Б. Гавриков , Н.В. Пестрякова, А.В.Усков, В.В. Фарсобина. О некоторых свойствах оценки метода распознавания символов, основанного на полиномиальной регрессии. Препринт Института прикладной математики им. М.В. Келдыша РАН, 2008, 29 страниц, 11 рисунков, библиография: 14 наименований.

Показано, что при распознавании рукопечатных цифр по базе, совпадающей с базой обучения, средняя оценка распознавания для каждого из рассматриваемых символов убывает монотонно по мере «удаления» от «среднестатистического» полиномиального вектора, а для «1» сначала монотонно убывает, а затем монотонно увеличивается и принимает максимальное значение 255. Уровень шумов в этих зависимостях существенно ниже, чем в зависимостях при отклонении от «среднестатистического» раstra.

В результате численного эксперимента удалось показать, что база единичных символов, состоит из двух подбаз, каждая из которых имеет свой «среднестатистический» вектор (и растр). Исследуемый метод распознавания, обученный на совокупности всех изображений «1», выставляет оценки распознавания, поведение которых при удалении от каждого из этих двух «среднестатистических» векторов не отличается от имеющего место для баз остальных цифр одним «среднестатистическим» вектором (и растром).

M.B.Gavrikov, N.V.Pestryakova, A.V.Uskov, V.V.Farsobina. About some properties of estimation of method of character recognition, based on polynomial regression. Preprint, Inst. Appl. Mathem., Russian Academy of Sciences, 2008, 29 Pages, 11 Figures, 14 References.

Shown that at recognition of handprinted figures on base, coinciding with base of education, average estimation of recognition for each of considered symbols decreases monotonously on measure 'removing' from 'average' polynomial vector, and for '1' first monotonously decrease, and then monotonously increase and take maximum rating 255. Level of noises in these dependences greatly lower than in dependences at deviation from 'average' raster.

In result numerical research it was possible to show that base of single symbols, consist of two subbase, each of which has own 'average' vector and (raster). Researched recognition method, trained on sets of all images '1', expose estimations of recognition, behaviour of which at removing from each of these two 'average' vectors are not distinguished from having place for bases of other figures with one 'average' vector and (raster).

Введение

Настоящая работа является очередной в серии [1 - 4], посвященной одному из современных точных методов распознавания символов. Он основан на линейном регрессионном анализе [5 - 13]. Практика использования данного метода и сравнение его с другими способами распознавания подтвердили, что он удовлетворяет высоким требованиям по качеству распознавания, быстрдействию, монотонности оценок [2, 3]. Заметим, что результаты работы [4] получены с использованием новой модификации метода, имеющей еще более высокое качество распознавания.

При исследовании характеристик рассматриваемого метода возник вопрос, существуют ли закономерности в выставлении оценок распознавания? Появилось предположение о наличии некоторых векторных аналогов образов символов, при отклонении от которых оценка распознавания падала бы монотонно. Нетрудно догадаться, что если возможно построить такие аналоги, то они должны зависеть от базы обучения.

Дальнейшее изложение относится к базе рукопечатных цифр, которая используется как для обучения, так и для распознавания. Заметим, что рукопечатные цифры пишутся от руки, но имеют сходство с некоторым стандартным начертанием (подобно рукопечатным и печатным буквам).

В [4] в качестве такого аналога исследовался «среднестатистический» растр для каждого из символов. Было показано, что при удалении от него оценка распознавания не убывает монотонно. Некоторые диаграммы имели общую тенденцию к убыванию, но при этом уровень шумов был весьма велик.

Здесь изучается поведение оценки распознавания при увеличении «расстояния» между полиномиальным вектором, построенным по растру символа, и «среднестатистическим» вектором этого символа по базе. Проводится сравнение с результатами работы [4]. В частности, показано, что уровень шумов в зависимостях, полученных в настоящей работе, существенно ниже, а, следовательно, степень монотонности выше, чем в зависимостях, указанных в предыдущем абзаце.

Для «1» оценка распознавания вела себя совершенно отлично от имевшего место для других цифр. При увеличении отклонения как от «среднестатистического» растра, так и от вектора сначала наблюдалось падение, а затем значительный рост (с сохранением описанных различий по шумам). Возникла гипотеза о том, что база единиц составлена из двух подбаз. При помощи численного эксперимента удалось разделить исходную базу на эти подбазы. Для каждой из

выделенных подбаз оценки распознавания и функции распределения ведут себя аналогично диаграммам, полученным для других символов.

1. Метод 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 черно-белых 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 $V = \{\mathbf{v}\}$ – совокупность всевозможных растров. Очевидно, $V \subseteq \mathbf{R}^N$, причем если пиксели черно-белые, то $V = \{0,1\}^N$ – конечное множество, элементами которого являются последовательности из нулей и единиц длины N . Если пиксели серые, то $V = [0,1]^N$ – N -мерный единичный куб в \mathbf{R}^N .

Математическая постановка задачи распознавания состоит в следующем. Пусть для некоторого растра $\mathbf{v} \in V$ можно найти $p_k(\mathbf{v})$ – вероятность того, что растр изображает символ с порядковым номером k , $1 \leq k \leq K$. Тогда распознанным считается символ с порядковым номером k_0 , где

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

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 $Y = \{\mathbf{e}_1, \dots, \mathbf{e}_K\}$. Пусть $p(\mathbf{v}, \mathbf{y})$ – вероятность наступления события (\mathbf{v}, \mathbf{y}) , $\mathbf{v} \in V$, $\mathbf{y} \in Y$ (если V континуально, 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

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

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 $\mathbf{v} = (v_1, \dots, v_N)$:

$$\begin{aligned} p_1(\mathbf{v}) &\cong c_0^{(1)} + \sum_{i=1}^N c_i^{(1)} v_i + \sum_{i,j=1}^N c_{i,j}^{(1)} v_i v_j + \dots \\ p_2(\mathbf{v}) &\cong c_0^{(2)} + \sum_{i=1}^N c_i^{(2)} v_i + \sum_{i,j=1}^N c_{i,j}^{(2)} v_i v_j + \dots \\ &\dots\dots\dots \\ p_K(\mathbf{v}) &\cong c_0^{(K)} + \sum_{i=1}^N c_i^{(K)} v_i + \sum_{i,j=1}^N c_{i,j}^{(K)} v_i v_j + \dots \end{aligned} \tag{2}$$

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:

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

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:

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

Набор $[\mathbf{v}^{(1)}, \mathbf{y}^{(1)}], \dots, [\mathbf{v}^{(J)}, \mathbf{y}^{(J)}]$ практически 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

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

$$E\{\mathbf{x}(\mathbf{v}) \mathbf{y}^T\} = [E\{x_i(\mathbf{v})y_k\}]_{1 \leq i \leq L, 1 \leq k \leq K} =$$

$$\left[\sum_{s=1}^K \int_{\mathbf{v}} x_i(\mathbf{v})(\mathbf{e}_s)_k p(\mathbf{v}, \mathbf{e}_s) d\mathbf{v} \right]_{1 \leq i \leq L, 1 \leq k \leq K} = \left[\int_{\mathbf{v}} x_i(\mathbf{v})p(\mathbf{v}, \mathbf{e}_k) d\mathbf{v} \right]_{1 \leq i \leq L, 1 \leq k \leq K}$$

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, где \mathbf{A}_0 и \mathbf{G}_0 заданы:

$$\mathbf{A}_j = \mathbf{A}_{j-1} - \alpha_j \mathbf{G}_j \mathbf{x}^{(j)} [\mathbf{A}_{j-1}^T \mathbf{x}^{(j)} - \mathbf{y}^{(j)}]^T, \quad \alpha_j = 1/J$$

$$\mathbf{G}_j = \frac{1}{1 - \alpha_j} \left[\mathbf{G}_{j-1} - \alpha_j \frac{\mathbf{G}_{j-1} \mathbf{x}^{(j)} (\mathbf{x}^{(j)})^T \mathbf{G}_{j-1}}{1 + \alpha_j ((\mathbf{x}^{(j)})^T \mathbf{G}_{j-1} \mathbf{x}^{(j)} - 1)} \right] \quad 1 \leq j \leq J \quad (8)$$

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

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

Модификация Б. $\mathbf{G}_j \equiv \mathbf{D}^{-1}$,

$$\mathbf{D}=\text{diag}(E\{x_1^2\},E\{x_2^2\},\dots,E\{x_L^2\}) \quad (10)$$

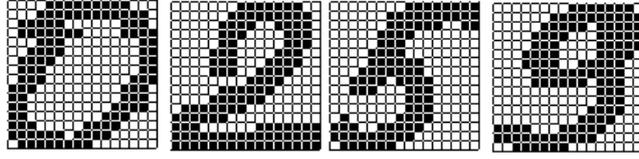


Рис. 1. Образы 16x16 рукопечатных цифр

2. Развитие и практическая реализация метода распознавания

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 \mathbf{x} . Используются два варианта вектора \mathbf{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 так:

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

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

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

В (11) и (12) выражения в фигурных скобках соответствуют цепочкам элементов вектора, вычисляемым по всем пикселям растра (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 растра – нижние индексы r и y соответственно. Если имеется нижний индекс L (left) или D (down), то это означает, что соответствующие величины относятся к пикселю

слева или снизу от рассматриваемого. Компоненты вектора \mathbf{x} , не имеющие индекса L или D , вычисляются для всех пикселей растра; с индексом L – кроме левых граничных; с индексом D – кроме нижних граничных пикселей. Вне растра считаем, что $v_i=0$ (используется при вычислении конечных разностей на границе растра). 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 так. 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:

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

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 :

$$\begin{aligned} a_j^{pk} &= a_{j-1}^{pk} - \alpha_j x_k^j \left(\sum_{i=1}^L a_{j-1}^{ik} x_i^j - y_k^j \right) / m_p^j, \quad \alpha_j = 1/J \\ \mathbf{A}_j &= [a_j^{pk}], \quad j=1,\dots,J, \quad p=1,\dots,L, \quad k=1,\dots,K \end{aligned} \quad (15)$$

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

Распознавание осуществляется следующим образом. Для серого изображения размером 16×16 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 мак-

симальной 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. Распознавание 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 является большая или меньшая степень схожести базы обучения и распознавания. Данная проблема рассматривалась в [4].

В настоящей работе при обучении и распознавании использовалась модификация длинного вектора \mathbf{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 диаграммы зависимости средней оценки распознавания рукопечатного символа (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) от величины отклонения между его растрами 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 N пикселям:

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

Диапазон отклонений между растрами распознанных верно изображений символа и «среднестатистическим» растром этого символа по рассматриваемой базе находится от минимального v_true_min до максимального v_true_max . В таблице 1 приведены значения этих величин для каждого из десяти цифр.

Делим отрезок $[v_true_min, v_true_max]$ (оси абсцисс на рис.2а-11а) на 20 равных по длине частей – отрезок и 19 полуинтервалов: $[v_true_min, v_true_min + dv]$, $(v_true_min + dv, v_true_min + 2dv]$, ... , $(v_true_min + 19dv, v_true_min + 20dv]$, где $dv = (v_true_max - v_true_min)/20$. Затем для совокупности изображений, попадающих в каждый такой участок, вычисляем среднюю оценку распознавания (оси ординат на рис. 2а-11а). На этих диаграммах видно, что средняя оценка распознавания ни для одного из рассматриваемых символов на соответствующем этому символу отрезке $[v_true_min, v_true_max]$ не убывает монотонно по мере «удаления» растров от «среднестатистического» растра, а для «1» принимает максимальное значение 255 на наибольшем удалении от «среднестатистического» образа. Некоторые диаграммы имеют общую тенденцию к убыванию, но на эту закономерность накладывается высокий уровень шумов.

На рис.2б-11б приведены диаграммы - «дискретный» аналог функции распределения для распознанных верно изображений каждого из символов 0, 1, ... , 9. А именно, ось абсцисс такая же, как указано в предыдущем абзаце, а по оси ординат отложено количество правильно распознанных изображений, попавших в каждую двадцатую часть отрезка $[v_true_min, v_true_max]$.

Для неправильно распознанных образов символа диапазон отклонений между его растрами и «среднестатистическим» растром этого символа по рассматриваемой базе находится от минимального v_false_min до максимального v_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 P_{false} . 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 P_{false} намного меньше (приблизительно в два раза), чем оценки правильного распознавания. Отметим, что для каждого из рассматриваемых символов $v_true_min < v_false_min$. Тем не менее, диапазон $[v_true_min, v_true_max]$ отличается от $[v_false_min, v_false_max]$ не очень существенно. Следовательно, поскольку доля неправильно распознанных символов весьма незначительна, «дискретный» аналог функции распределения для всех изображений (распознанных как верно, так и неверно) каждого из символов 0, 1, ... , 9 мало отличается от приведенных на рис.2б-11б.

Таблица 1

сим- вол	v_true_min	v_true_max	v_true_max/ v_true_min	v_false_min	v_false_max	P _{false}
0	35,41	113,59	3,21	56,34	101,09	106,61
1	42,58	173,80	4,08	52,22	131,38	128,17
2	38,33	105,57	2,75	61,62	109,43	120,19
3	39,92	103,10	2,58	55,63	95,53	118,84
4	50,76	106,34	2,09	56,02	123,47	131,78
5	36,28	130,66	3,60	52,68	98,26	126,65
6	44,60	115,07	2,58	55,17	103,11	105,26
7	40,56	101,70	2,51	53,45	93,89	114,43
8	50,15	119,80	2,39	57,23	115,19	121,27
9	47,36	120,58	2,55	54,28	117,37	127,45

На рис.2в-11в представлены диаграммы зависимости средней оценки распознавания рукопечатного символа (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) от величины отклонения между полиномиальным вектором x , построенным по его растру, и «среднестатистическим» полиномиальным вектором этого символа по базе, которая используется как для обучения, так и для распознавания.

«Среднестатистический» полиномиальный вектор конкретного символа получаем следующим образом. Значение в каждой компоненте вектора, имеющей номер i , равно среднему арифметическому значений i -х компонент по всем имеющимся в базе растрам рассматриваемого символа. Расстояние между двумя векторами $\mathbf{v}=(v_1, \dots, v_L)$ и $\mathbf{u}=(u_1, \dots, u_L)$ определяем так: вычисляем модуль разности значений в i -х компонентах, затем суммируем по всем L компонентам:

$$\|\mathbf{v}-\mathbf{u}\| = \sum_{i=1}^L |v_i - u_i| \quad (17)$$

Диапазон отклонений между полиномиальным вектором распознанного верно изображения символа и «среднестатистическим» вектором этого символа по рассматриваемой базе лежит от минимального x_true_min до максимального x_true_max . В таблице 2 приведены значения этих величин для каждого из символов 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Делим отрезок $[x_true_min, x_true_max]$ (оси абсцисс на рис.2в-11в) на 20 равных по длине частей – отрезок и 19 полуинтервалов: $[x_true_min, x_true_min$

+ x_{dv}], $(x_true_min + x_{dv}, x_true_min + 2x_{dv}]$, ... , $(x_true_min + 19x_{dv}, x_true_min + 20x_{dv}]$, где $x_{dv} = (x_true_max - x_true_min)/20$. Затем для совокупности изображений, имеющих полиномиальные векторы, попадающие в каждый такой участок, вычисляем среднюю оценку распознавания (оси ординат на рис. 2в-11в). На этих рисунках видно, что средняя оценка распознавания для каждого из рассматриваемых символов на соответствующем этому символу отрезке $[x_true_min, x_true_max]$ убывает монотонно (с некоторыми шумовыми погрешностями) по мере «удаления» от «среднестатистического» вектора, а для «1» сначала монотонно убывает, а затем монотонно увеличивается и принимает максимальное значение 255 на предпоследнем интервале удаления от «среднестатистического» вектора (также с некоторыми погрешностями). Уровень шумов в этих зависимостях существенно ниже, а, следовательно, степень монотонности выше, чем в аналогичных зависимостях для средней оценки распознавания при отклонении от «среднестатистического» растра.

На рис.2г-11г приведены диаграммы - «дискретный» аналог функции распределения для распознанных верно изображений каждого из символов 0, 1, ... , 9. А именно, ось абсцисс такая же, как указано в предыдущем абзаце для рис.2в-11в, а по оси ординат отложено количество правильно распознанных изображений, попавших в каждую вышеописанную двадцатую часть отрезка $[x_true_min, x_true_max]$.

Диапазон отклонений между полиномиальным вектором неправильно распознанного изображения символа и «среднестатистическим» вектором этого символа по рассматриваемой базе находится от минимального x_false_min до максимального x_false_max . В таблице 2 приведены значения этих величин для каждого из символов 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Средние оценки неправильного распознавания не приводятся, поскольку они, как нетрудно догадаться, совпадают с указанными в таблице 1. Отметим, что для каждого из рассматриваемых символов $x_true_min < x_false_min$. Тем не менее, диапазон $[x_true_min, x_true_max]$ отличается от $[x_false_min, x_false_max]$ 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.2г-11г.

Выше описаны результаты двух типов: в терминах растров и полиномиальных векторов. Возникает вопрос, как их сравнивать? Для каждого символа одни и те же правильно распознанные изображения находятся в первом случае на отрезке $[v_true_min, v_true_max]$, а во втором на $[x_true_min, x_true_max]$. Предлагается идеология, основанная на преобразовании соответствующей оси

абсцисс координаты посредством выравнивания по длине диапазонов отклонения растров / векторов правильно распознанных изображений каждого из символов от «среднестатистических» растра / вектора данного символа, а именно $[v_true_min, v_true_max]$ и $[x_true_min, x_true_max]$. В этих целях вводятся «отнормированные» (по величине соответствующего диапазона) координаты.

Таблица 2

символ	x_true_min	x_true_max	x_true_max/x_true_min	x_false_min	x_false_max
0	2004	5290	3,21	3002	5119
1	2416	7917	4,08	3046	6437
2	2237	5265	2,75	3491	5523
3	2276	4954	2,58	3026	4936
4	2798	5158	2,09	3135	5619
5	2104	6300	3,60	3142	4909
6	2416	5161	2,58	3375	4966
7	2324	5276	2,51	2913	5621
8	2679	5505	2,39	3335	5188
9	2559	5482	2,55	3076	5313

В таблице 1 приводятся значения величины v_true_max/v_true_min , а в таблице 2 – соответственно x_true_max/x_true_min . Сравнение показывает, что для каждого из символов $v_true_max/v_true_min > x_true_max/x_true_min$. Следовательно, как нетрудно показать, $v_true_max/(v_true_max-v_true_min) < x_true_max/(x_true_max-x_true_min)$, а также $v_true_min/(v_true_max-v_true_min) < x_true_min/(x_true_max-x_true_min)$. Эти два неравенства означают, что если на одном луче отложить «отнормированные» координаты $v_true/(v_true_max-v_true_min)$ и $x_true/(x_true_max-x_true_min)$, то $v_true_max/(v_true_max-v_true_min)$ будет находиться левее (ближе к началу луча), чем $x_true_max/(x_true_max-x_true_min)$; аналогично, $v_true_min/(v_true_max-v_true_min)$ будет левее (ближе к началу луча), чем $x_true_min/(x_true_max-x_true_min)$. Точка 0 (начало луча) соответствует «среднестатистическому» растру / вектору. Следовательно, в описанных «отнормированных» координатах для каждого из символов имеет место следующее: величина удаления отрезка, на котором находятся растры правильно распознанных изображений, от «среднестатистического» растра меньше, чем

максимальное удаление отрезка векторов правильно распознанных изображений от «среднестатистического» вектора.

Сравним поведение оценки распознавания при отклонении от «среднестатистического» растра (рис.2а-11а) и вектора (рис.2в-11в). Как уже отмечалось, уровень шумов в диаграммах с вектором существенно ниже, а, следовательно, степень монотонности выше, чем в диаграммах с растром.

Для более детального сопоставления совместим отрезки между минимальным и максимальным отклонениями от «среднестатистических» растра $[v_true_min, v_true_max]$ и полиномиального вектора $[x_true_min, x_true_max]$. Математически это можно представить так, что точке с координатой v_true на первом отрезке соответствует точка с координатой $x_true = x_true_min + (v_true - v_true_min) \cdot (x_true_max - x_true_min) / (v_true_max - v_true_min)$. На рис.2д-11д точка 0 соответствует минимальному отклонению от «среднестатистического» растра / вектора; здесь приведены графики, построенные по соответствующим диаграммам (рис.2а-11а, рис.2в-11в). Вблизи точки 0 и до половины или до третьей части (в зависимости от символа) величины максимального отклонения от 0 средняя оценка распознавания для вектора выше, чем для растра. Для более отдаленных участков ситуация меняется на противоположную (в некоторых случаях для максимально удаленных участках указанная закономерность не выполнялось, но это неважно, т.к. количество находящихся там символов ничтожно, как видно на рис.2б-11б, рис.2г-11г). Описанное поведение оценки распознавания наблюдалось для всех символов, кроме «1».

Для «1» при увеличении отклонения от 0 сначала наблюдалось уменьшение оценки, а затем ее рост до максимальной оценки 255 на наиболее удаленных участках. Возникла гипотеза о том, что база единиц составлена из двух подбаз.

Дальнейшие действия были проделаны для «среднестатистических» векторов, поскольку в использующих их зависимостях уровень шумов существенно ниже, чем для «среднестатистических» растров. Найдя первоначальное «среднестатистическое» значение полиномиального вектора x_0 , мы отделили ту часть изображений, полиномиальные векторы которых удалены от соответствующего x_true_min более чем на $2/3$ величины $x_true_max - x_true_min$ и построили для них «среднестатистический» вектор x_1 . Оказалось, что для изображений, векторы которых к x_1 ближе, чем к x_0 , при удалении от x_1 оценка монотонно падает (рис.3з). Всего оказалось 714 таких изображений. Для оставшихся 32388 изображений построили новый среднестатистический вектор x_2 , при отклонении от которого имела место аналогичная закономерность (рис.3е). Для каждой из выделенных подбаз построили функции распределения (соответственно рис.3и, рис.3ж), которые оказались схожи с функциями распределения

других символов. Дополнительные итерации, несомненно, улучшили бы степень разделения подбаз.

Несмотря на то, что база изображений «1», как удалось показать в результате численного эксперимента, состоит из двух подбаз, каждая из которых имеет свой «среднестатистический» вектор (также и растр, как нетрудно догадаться), тем не менее, исследуемый метод распознавания, обученный на совокупности всех изображений «1», выставляет оценки распознавания, поведение которых при удалении от каждого из этих двух «среднестатистических» векторов не отличается от имеющего место для баз цифр 2, 3, 4, 5, 6, 7, 8, 9, обладающих одним «среднестатистическим» вектором (и растром).

4. Выводы

Изучение метода распознавания, основанного на полиномиальной регрессии, при распознавании рукопечатных цифр по базе, совпадающей с базой обучения, для каждого из рассматриваемых символов показало следующее.

Средняя оценка распознавания для каждого из рассматриваемых символов убывает монотонно (с некоторыми шумовыми погрешностями) по мере «удаления» от «среднестатистического» вектора, а для «1» сначала монотонно убывает, а затем монотонно увеличивается и принимает максимальное значение 255 на предпоследнем интервале удаления от «среднестатистического» вектора (также с некоторыми погрешностями). Уровень шумов в этих зависимостях существенно ниже, а, следовательно, степень монотонности выше, чем в зависимостях при отклонении от «среднестатистического» растра.

Численный эксперимент показал, что база единичных символов состоит из двух подбаз, каждая из которых имеет свой «среднестатистический» вектор (и растр). Исследуемый метод распознавания, обученный на совокупности всех изображений «1», выставляет оценки распознавания, поведение которых при удалении от каждого из этих двух «среднестатистических» векторов не отличается от имеющего место для баз остальных цифр, обладающих одним «среднестатистическим» вектором (и растром).

Для каждого из рассматриваемых символов средние оценки неправильного распознавания намного меньше (приблизительно в два раза), чем оценки правильного распознавания. Кроме того, минимальная по рассматриваемой базе величина отклонения между изображением символа и «среднестатистическим» растром (вектором) этого символа для правильно распознанных растров меньше, чем для неправильно распознанных.

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., Пестрякова Н. В., Славин О.А, Фарсобина В.В.* "Развитие метода полиномиальной регрессии и практическое применение в задаче распознавания", //Препринт ИПМатем. АН СССР, М., 2006, №25, 21 стр.
- [3] *Гавриков М.Б., Мисюрев 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.
- [4] *Гавриков М.Б., Пестрякова Н. В., Усков А.В., Фарсобина В.В.* "О некоторых свойствах метода распознавания символов, основанного на полиномиальной регрессии", //Препринт ИПМатем. АН СССР, М., 2004, №22, 20 стр.
- [5] *Sebestyen G.S.* Decision Making Processes in Pattern Recognition, MacMillan, New York, 1962.
- [6] *Nilson N. J.* Learning Machines, McGraw-Hill, New York, 1965.
- [7] *Schürmann J.* Polynomklassifikatoren, Oldenbourg, München, 1977.
- [8] *Schürmann J.* Pattern Classification, John Wiley&Sons, Inc., 1996.
- [9] *Albert A.E. and Gardner L.A.* Stochastic Approximation and Nonlinear Regression // Research Monograph 42. MIT Press, Cambridge, MA, 1966.
- [10] *Becker D. and Schürmann J.* Zur verstärkten Berücksichtigung schlecht erkennbarer Zeichen in der Lernstichprobe // Wissenschaftliche Berichte AEG-Telefunken **45**, 1972, pp. 97 – 105.
- [11] *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.
- [12] *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.
- [13] Дж.Себер. Линейный регрессионный анализ. М.:”Мир”, 1980.
- [14] Ю.В.Линник. Метод наименьших квадратов и основы математико - статистической теории обработки наблюдений. М.:”Физматлит”, 1958.

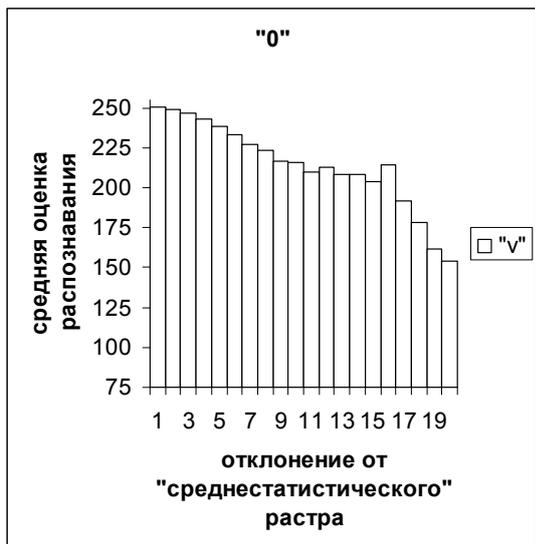


Рис. 2а

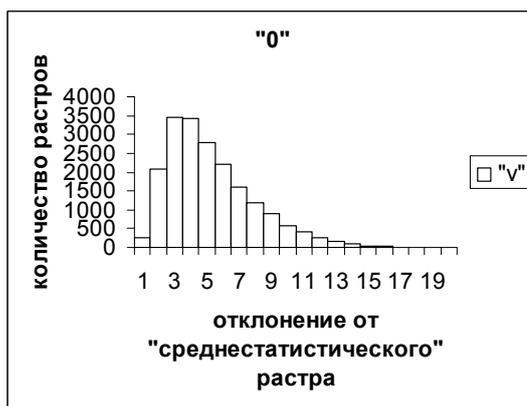


Рис. 2б

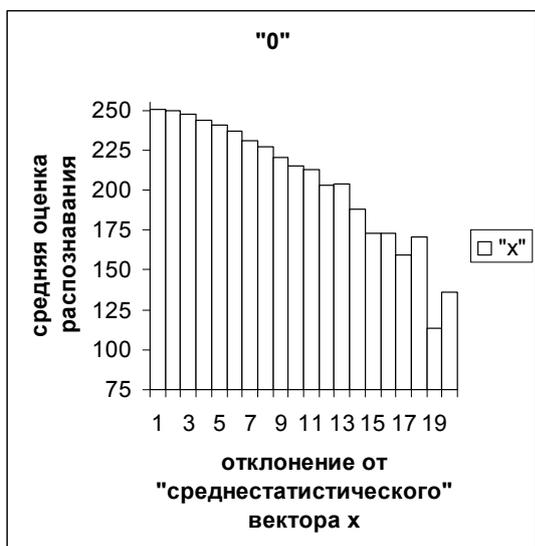


Рис. 2в

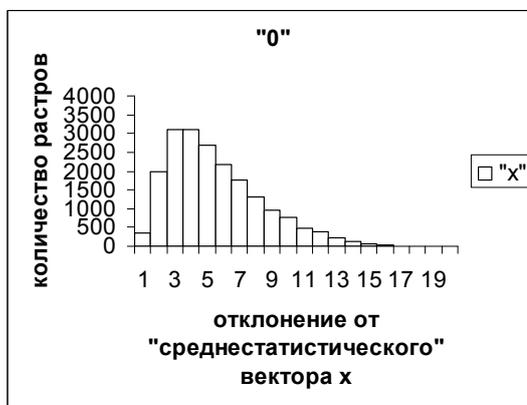


Рис. 2г

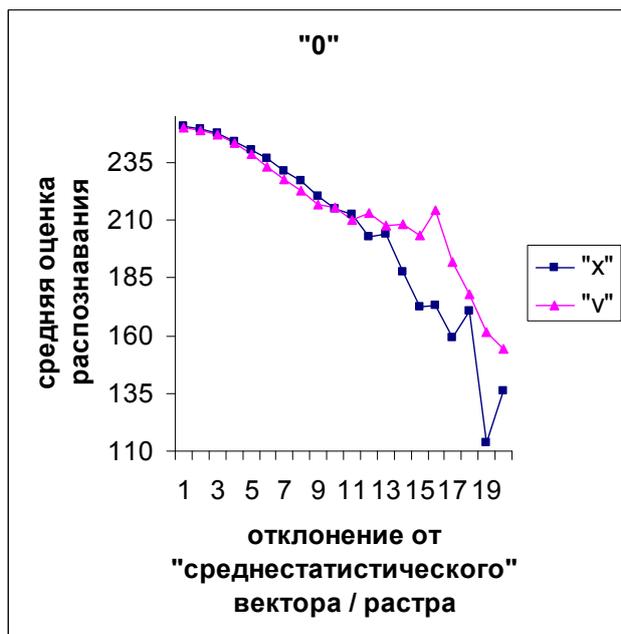


Рис. 2д

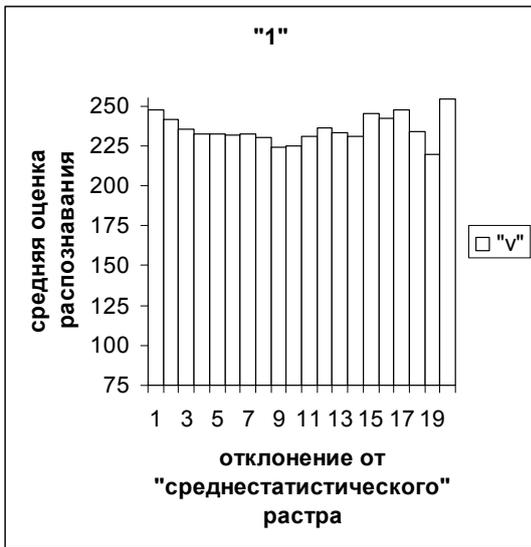


Рис. 3а

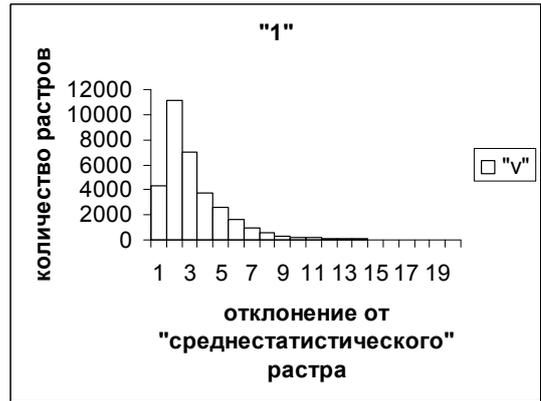


Рис. 3б

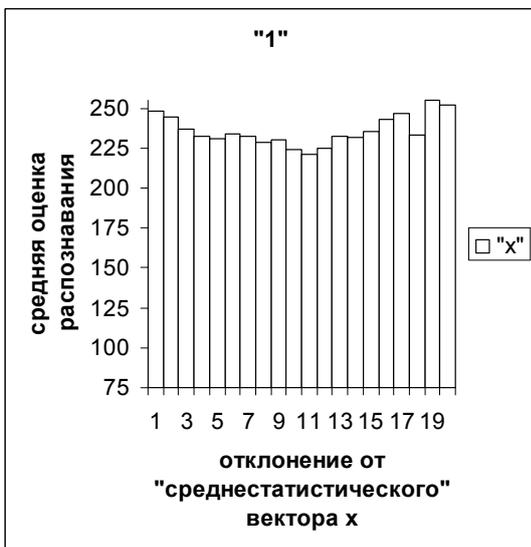


Рис. 3в

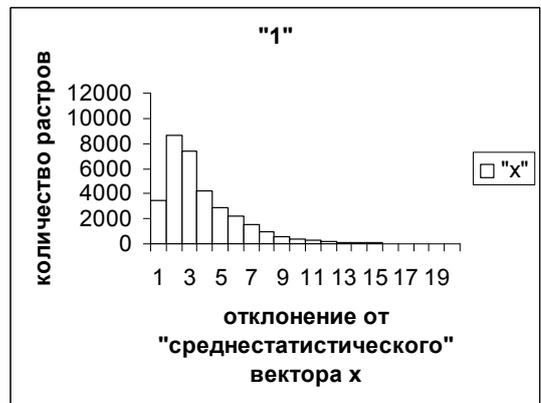


Рис. 3г

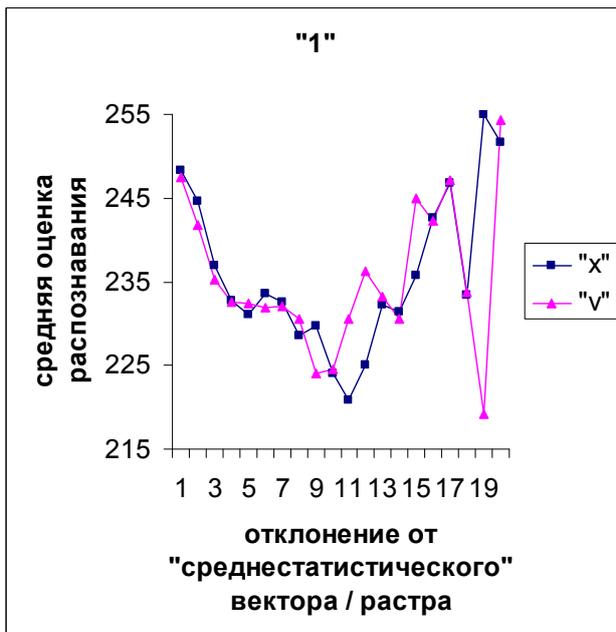


Рис. 3д

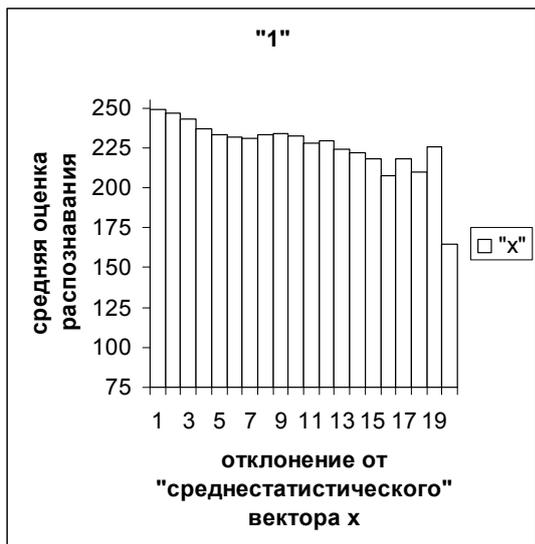


Рис. 3е

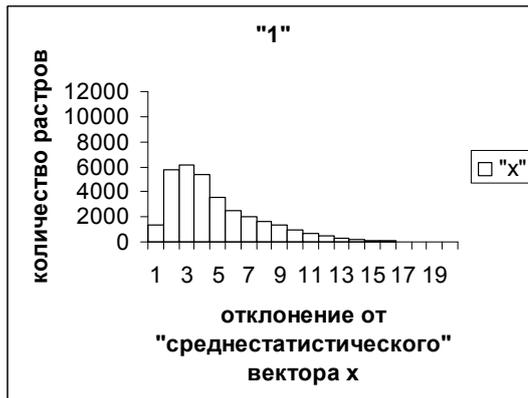


Рис. 3ж

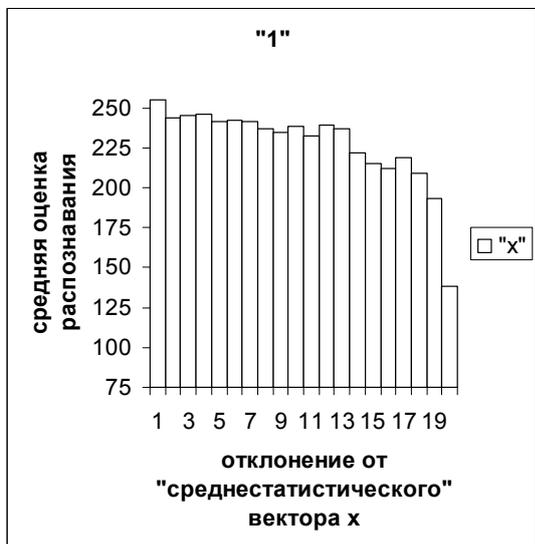


Рис. 3з

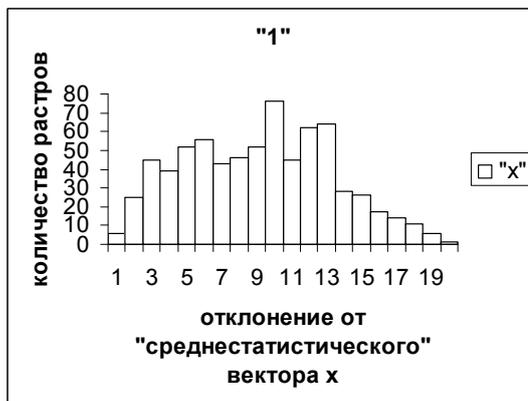


Рис. 3и

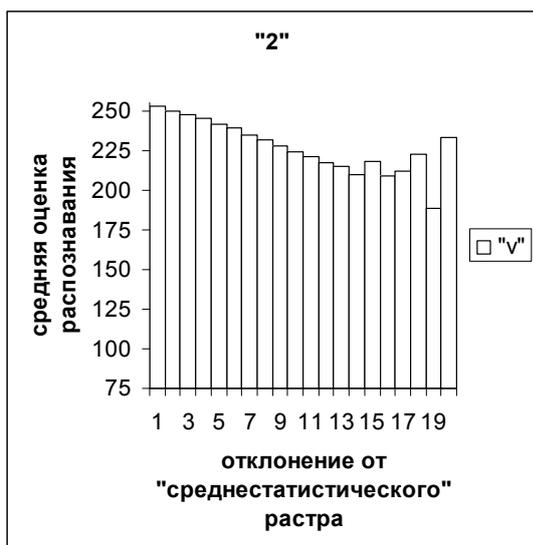


Рис. 4а

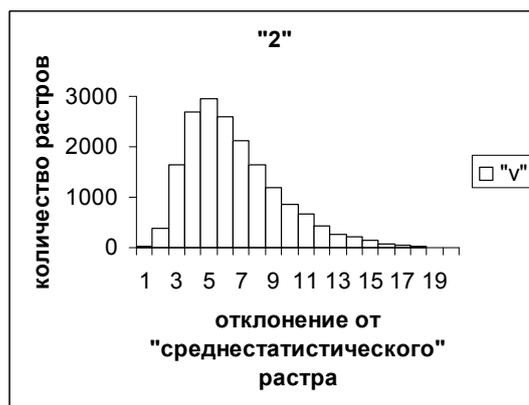


Рис. 4б

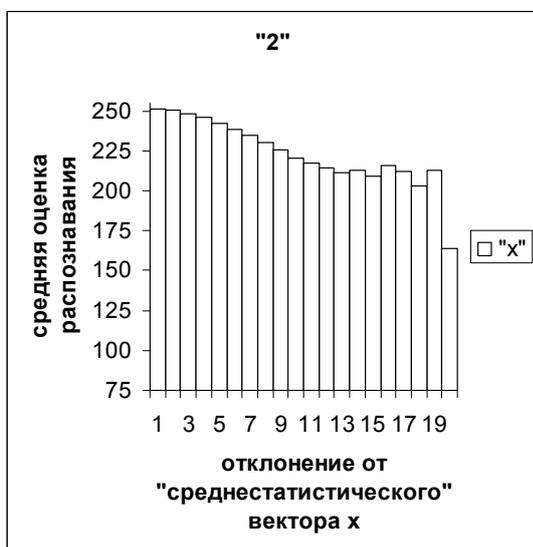


Рис. 4в

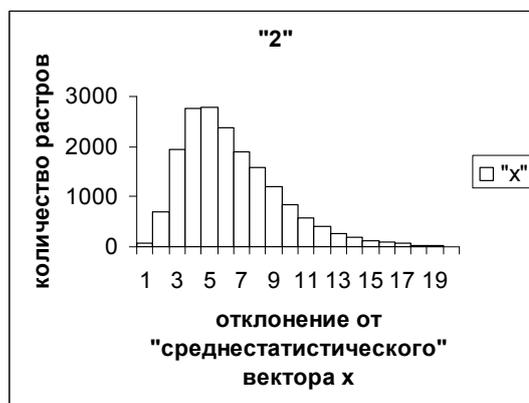


Рис. 4г

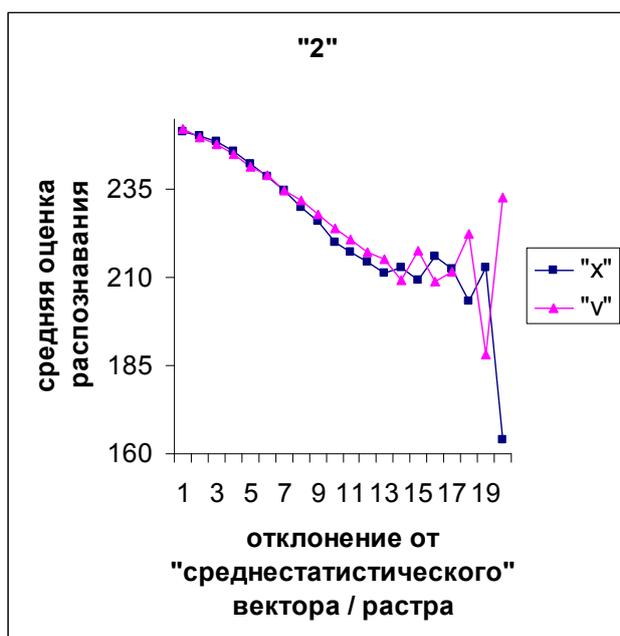


Рис. 4д

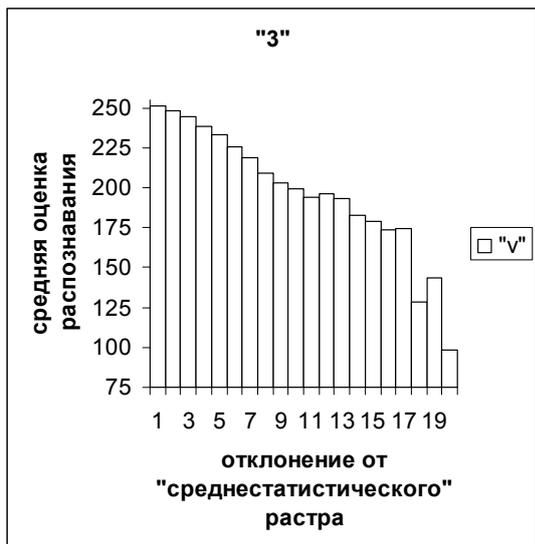


Рис. 5а

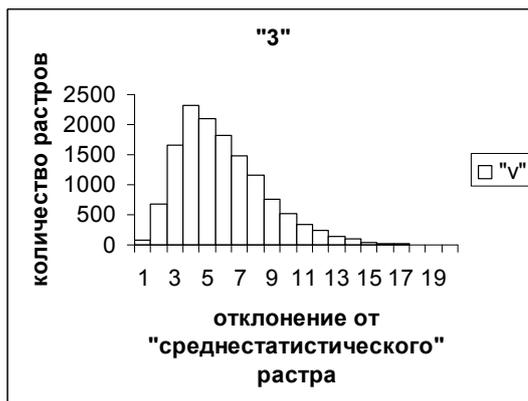


Рис. 5б

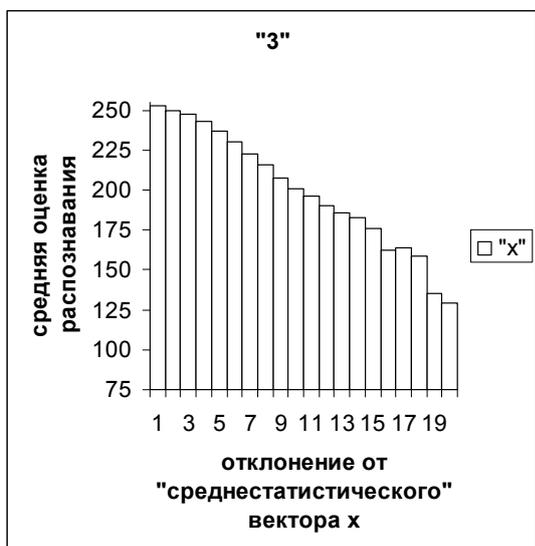


Рис. 5в

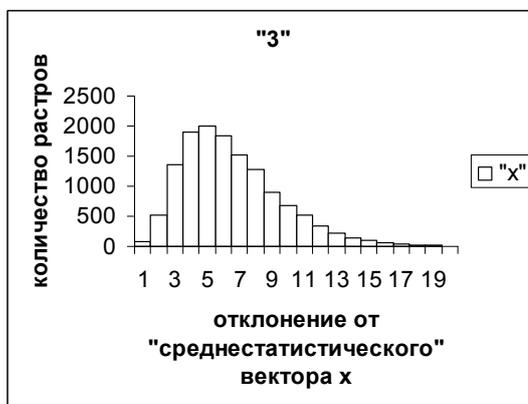


Рис. 5г

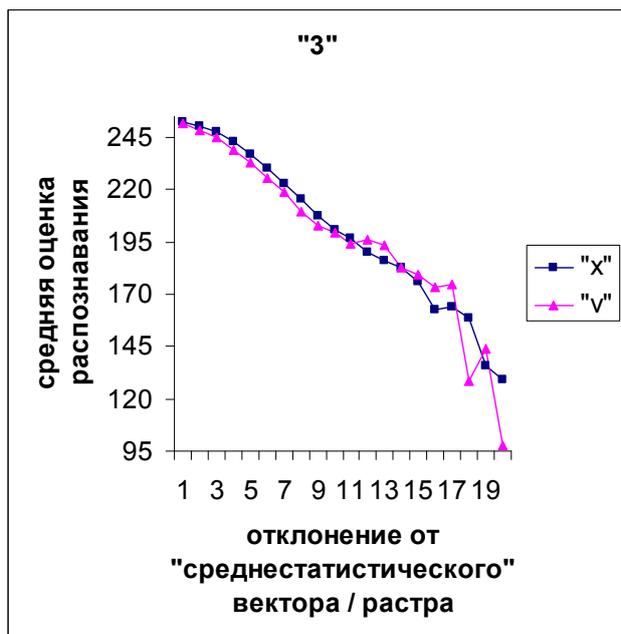


Рис. 5д

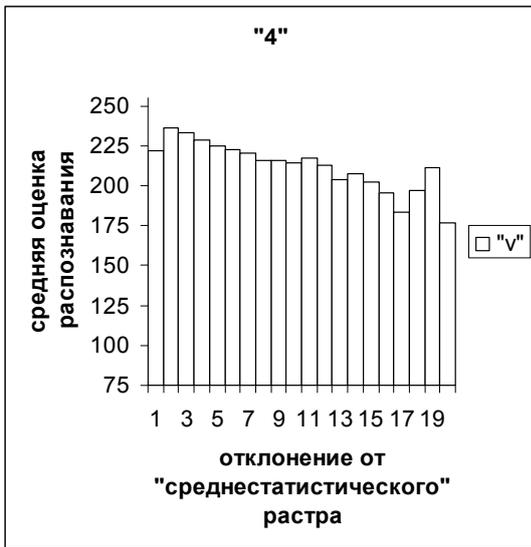


Рис. 6а

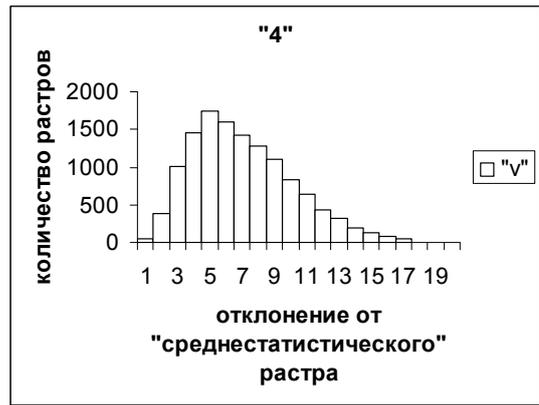


Рис. 6б

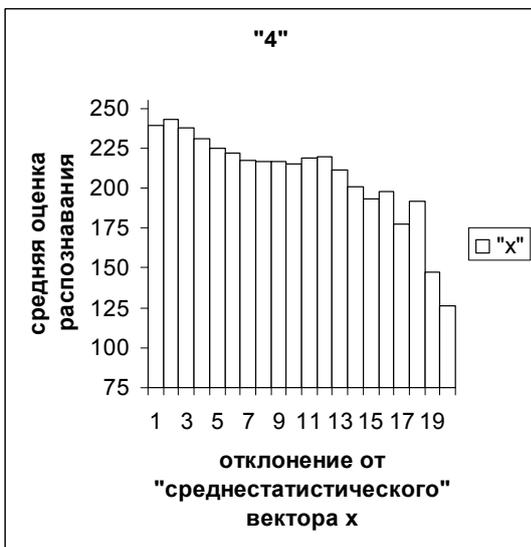


Рис. 6в

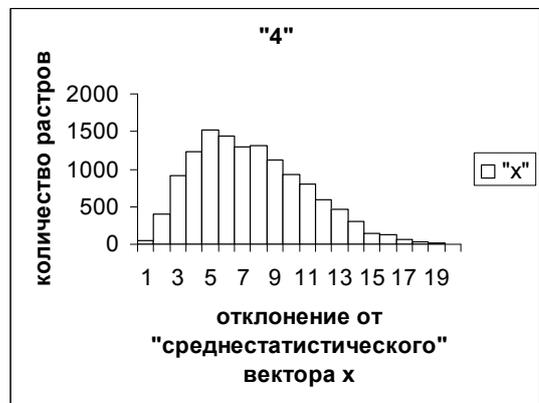


Рис. 6г

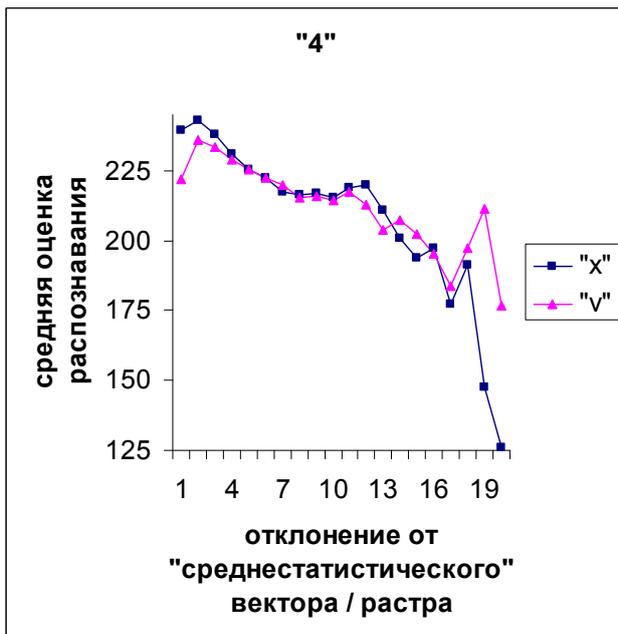


Рис. 6д

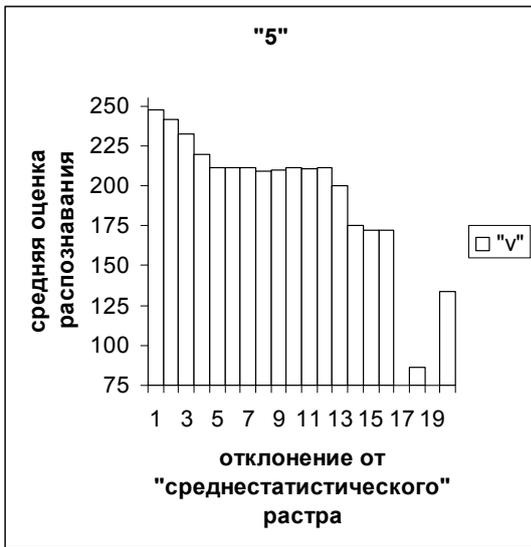


Рис. 7а

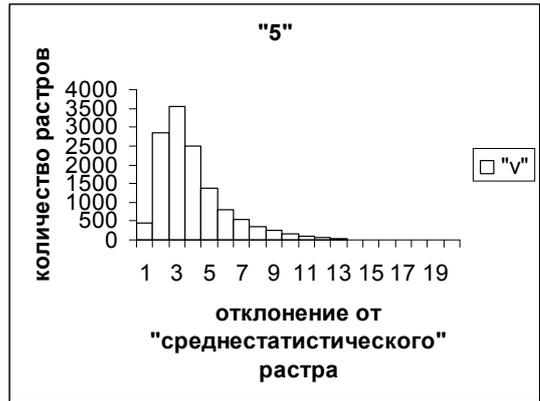


Рис. 7б

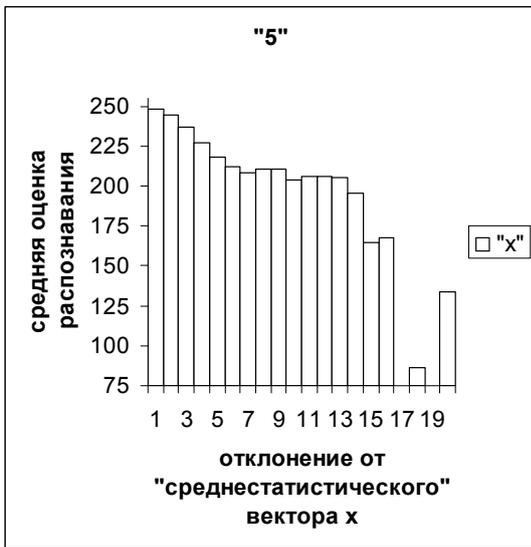


Рис. 7в

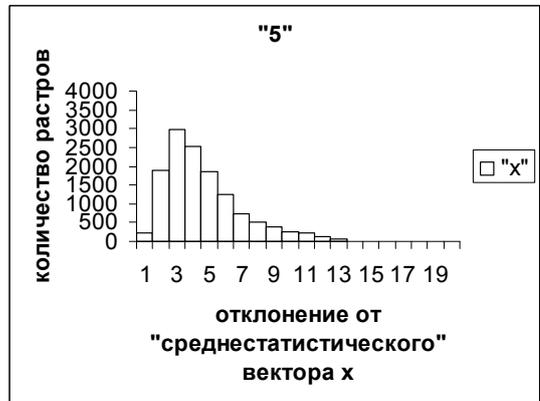


Рис. 7г

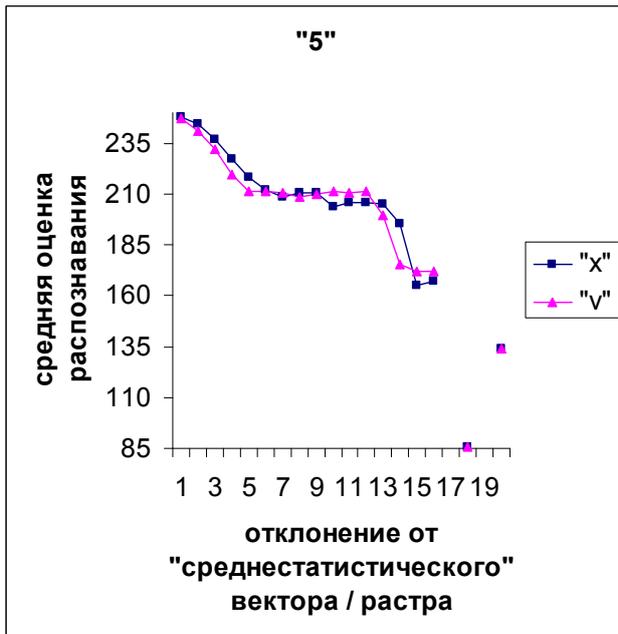


Рис. 7д

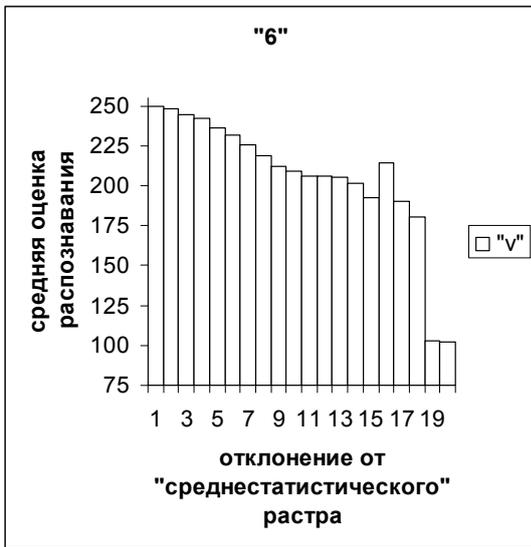


Рис. 8а

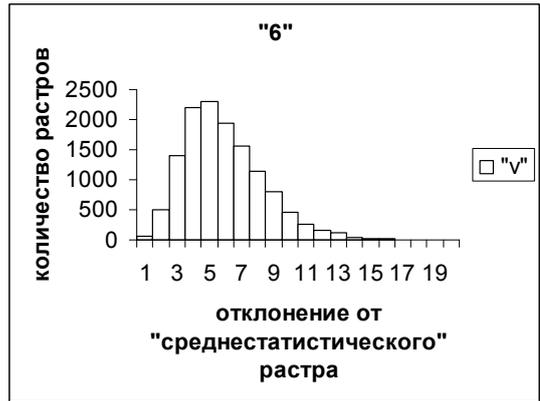


Рис. 8б

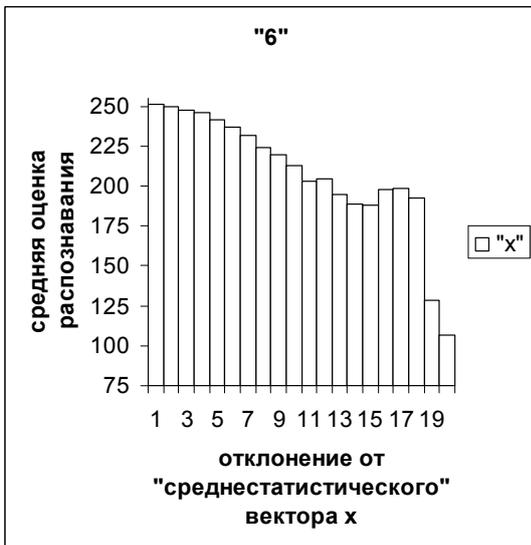


Рис. 8в

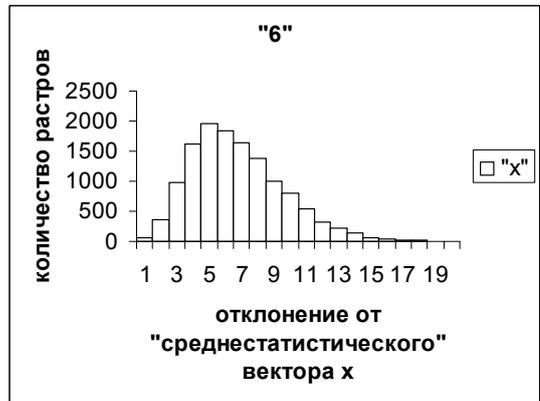


Рис. 8г

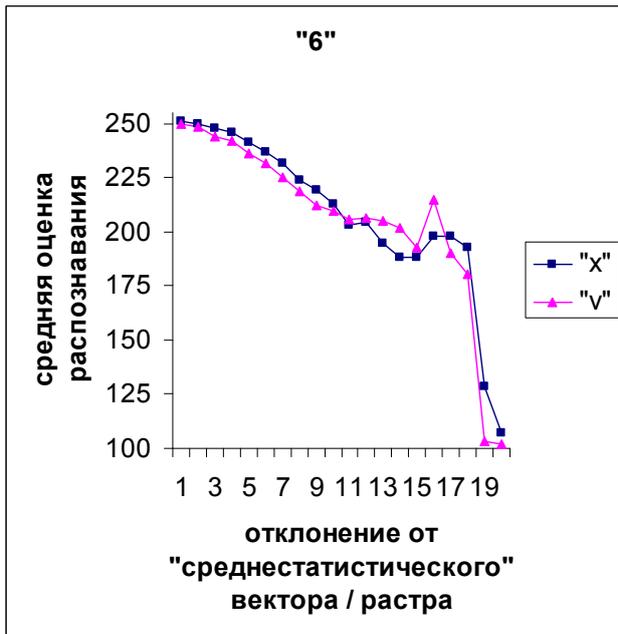


Рис. 8д

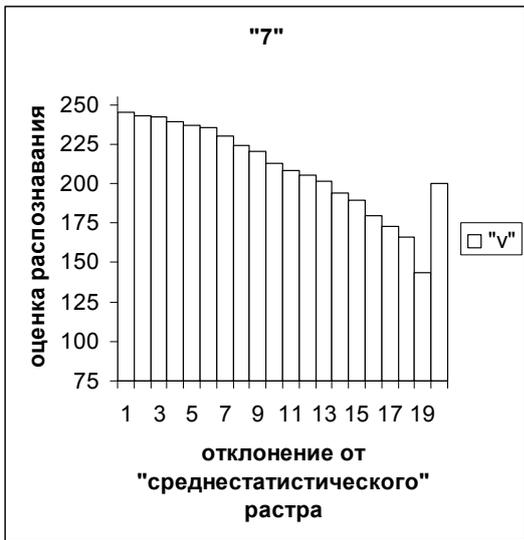


Рис. 9а

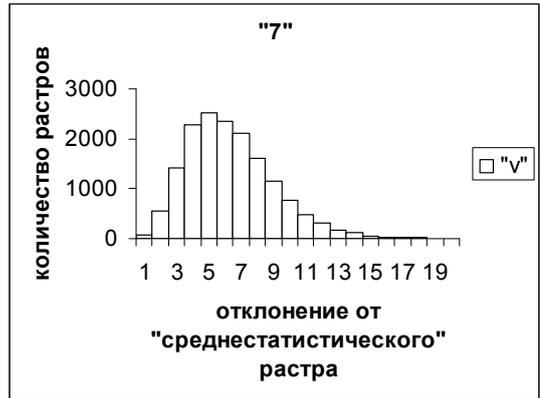


Рис. 9б

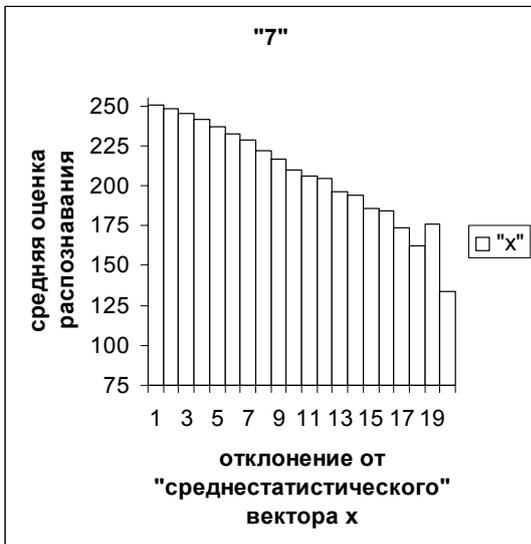


Рис. 9в

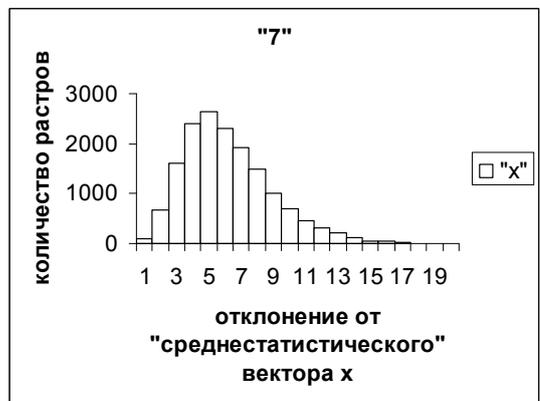


Рис. 9г

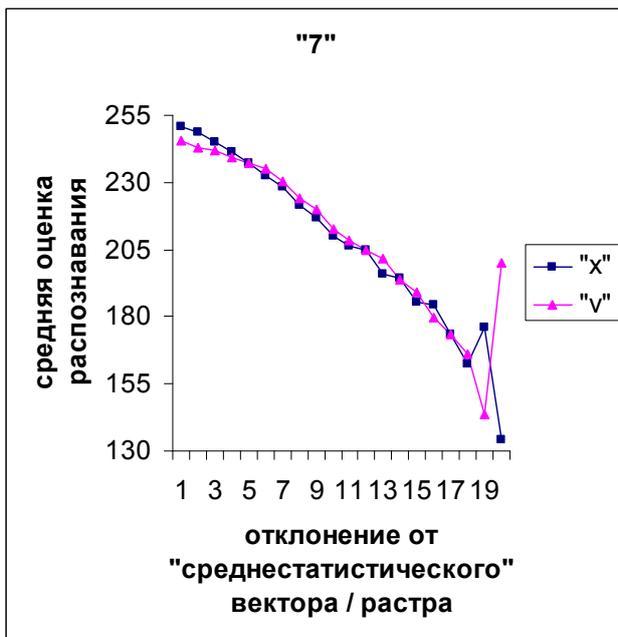


Рис. 9д

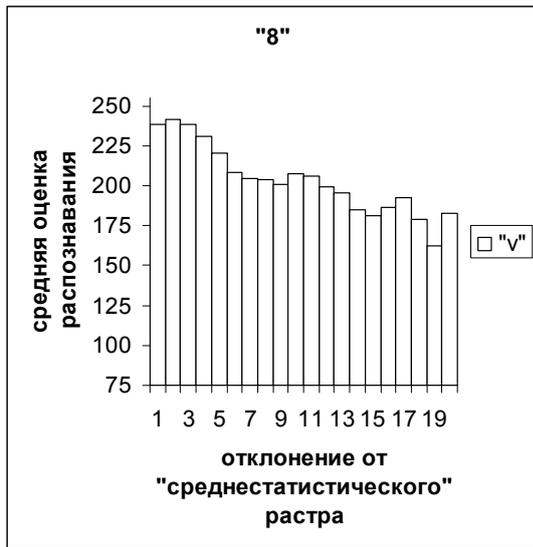


Рис. 10а

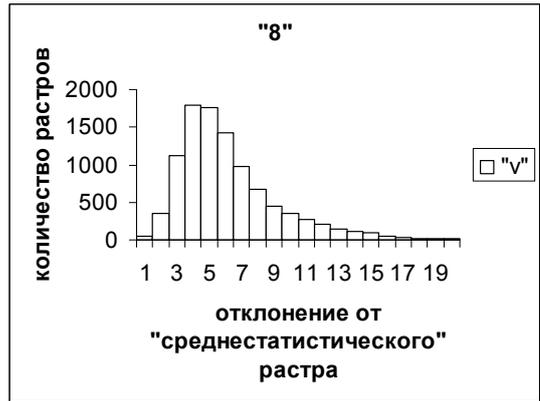


Рис. 10б

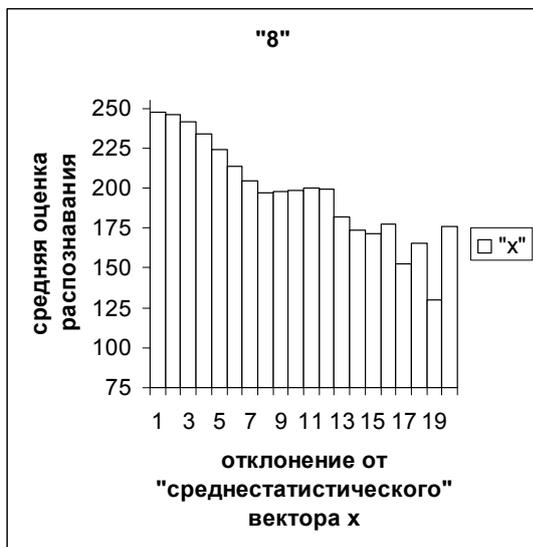


Рис. 10в

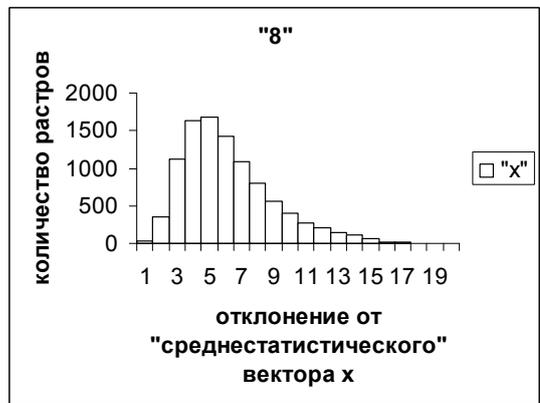


Рис. 10г

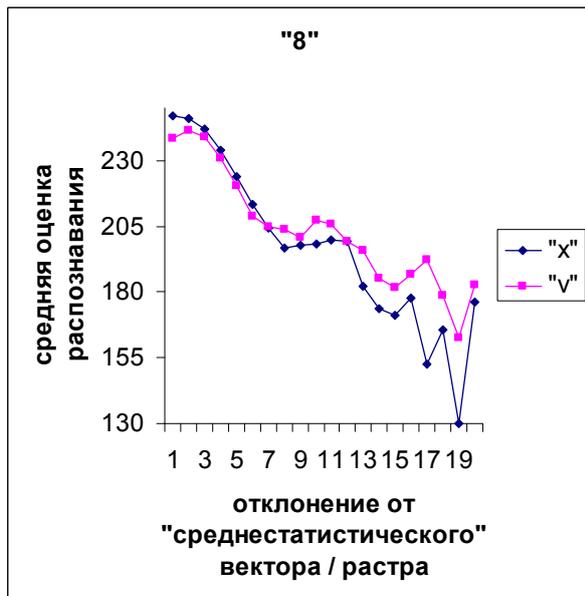


Рис. 10д

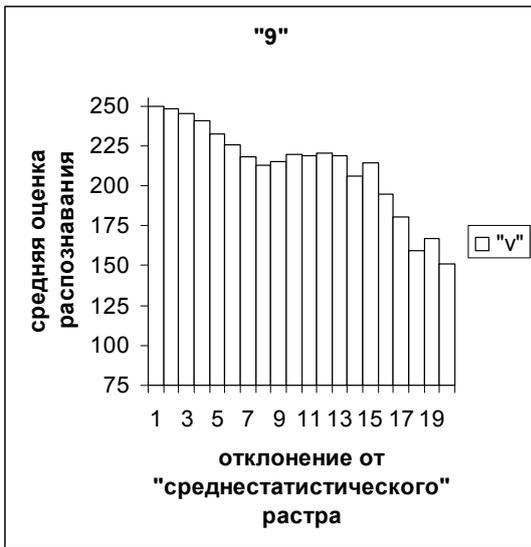


Рис. 11а

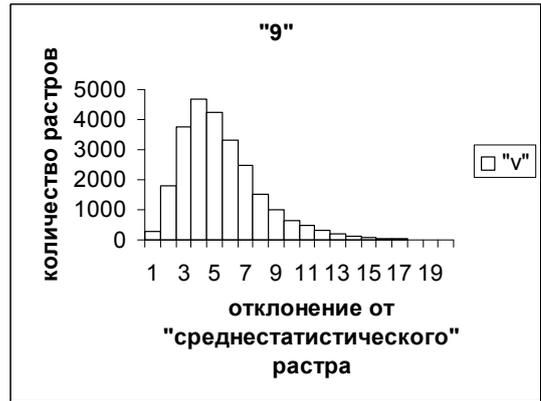


Рис. 11б

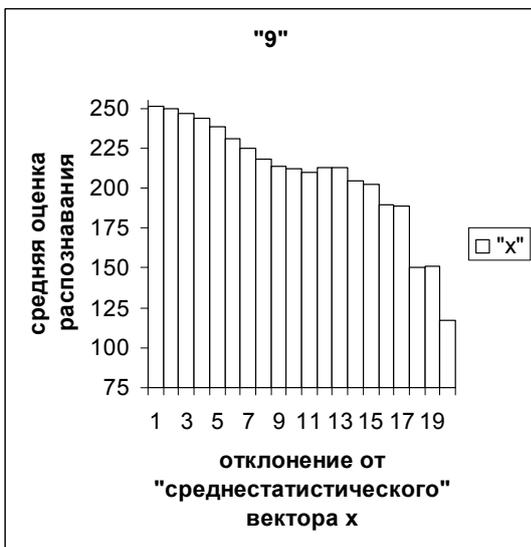


Рис. 11в

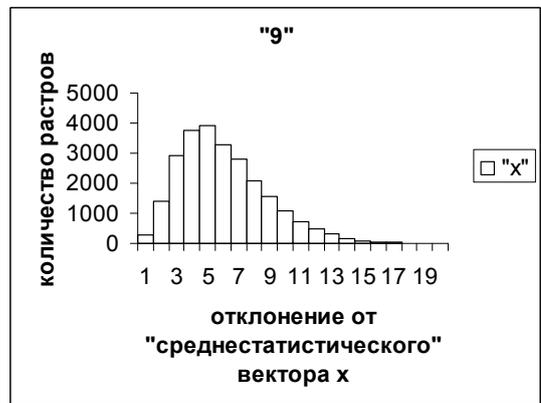


Рис. 11г

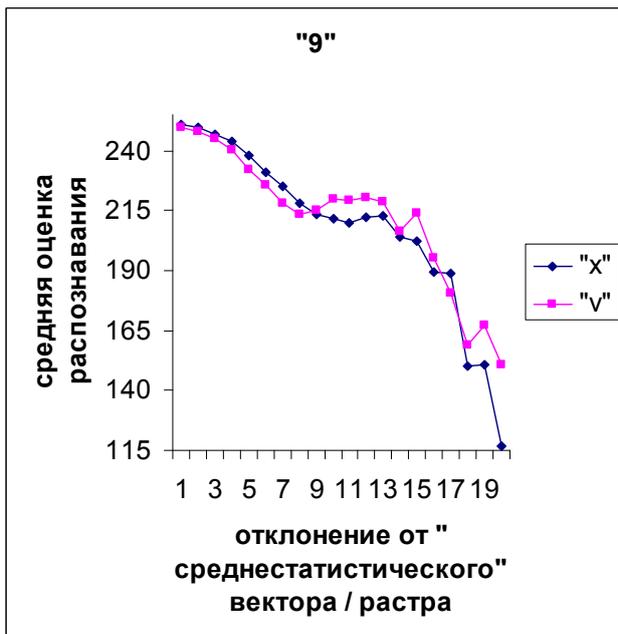


Рис. 11д