



Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Камалбеков Тимур Арманович

Модели комбинирования функций сходства

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

к.ф-м.н.

Майсурадзе Арчил Ивериевич

Москва, 2018

Содержание

1	Введение	2
1.1	Структура работы	2
1.2	Постановка задачи генерации метрик	3
1.3	Задача распознавания подписей	4
1.4	Оценка качества верификации	5
2	Первичные метрики на описаниях динамических подписей	7
2.1	Dynamic Time Warping (DTW)	7
2.2	Расстояние Вассерштейна (Earth Mover’s Distance, EMD)	8
2.3	«Расширенный» коэффициент детерминации (ER^2)	9
2.4	Коэффициент детерминации для временных рядов разной длины ($ER^2 + DTW$)	10
3	Модели комбинирования метрик и методы их обучения	11
3.1	Основные подходы генерации метрик	11
3.2	Расширения SAME	12
3.3	Способы обучения	13
3.4	Генерация множеств \mathcal{S} и \mathcal{D}	14
3.5	Этап применения	14
4	Эксперименты	15
4.1	Описание данных	15
4.2	Предобработка	15
4.3	Базовые метрики	16
4.4	Разбиение выборки	17
4.5	Измерение качества верификации	17
4.6	Результаты и выводы	18
5	Заключение	21
	Список литературы	22

1 Введение

В силу интерпретируемости и простоты реализации, метрические методы машинного обучения часто используются на практике. Одним из важнейших параметров метрических методов является функция расстояния. Однако, редки случаи, когда известна «хорошая» функция расстояния; в этом случае стоит проблема выбора метрики. Если объекты описываются числовыми векторами, то зачастую выбирают евклидову метрику. Однако, этот выбор, как правило, ничем не обоснован.

Существует и другая ситуация: мы уже знаем «хорошие» функции расстояния. Но если их несколько, то непонятно, каким образом их агрегировать, как получить итоговую метрику.

Возникает закономерный вопрос: а почему бы нам не обучить метрику? Тогда в первом случае (когда нам не даны «хорошие» метрики) говорят о задаче обучения метрики (Metric Learning), а во втором случае (когда какие-то метрики нам даны, и мы хотим получить новую метрику, являющуюся некоторой комбинацией исходных метрик) – о задаче извлечения метрики (Metric Extraction).

В данной работе рассматриваются методы извлечения метрик в применении к задаче верификации подписи.

1.1 Структура работы

В разделе 1.2 вводится общая постановка задачи генерации метрики, в разделе 1.3 вводится постановка задачи распознавания подписи. В разделе 2 вводятся первичные метрики между описаниями динамических подписей.

В разделе 3.1 описываются основные методы, решающие задачу генерации метрики, в разделе 3.2 приводятся возможные расширения метода генерации метрики SAME, в разделе 3.5 приводятся дальнейшие пути применения метрики для верификации.

В разделе 4 описаны проведённые эксперименты.

1.2 Постановка задачи генерации метрик

Введём следующие обозначения:

- \mathcal{S} – множество пар объектов, которые должны быть «близкими» по данной метрике;
- \mathcal{D} – множество пар объектов, которые должны быть «непохожими» по данной метрике;
- \mathcal{R} – множество троек объектов (x, y, z) , таких, что для них должно выполняться неравенство $\rho(x, y) \leq \rho(x, z)$.

Тогда под задачей генерации метрики обычно подразумевается следующая оптимизационная задача [2]:

$$\min_{\rho \in \mathcal{M}} \sum_{i=1}^k \mathcal{L}_i(\rho, \mathcal{S}, \mathcal{D}, \mathcal{R}) + \lambda R(\rho), \quad (1)$$

где R – некоторый регуляризатор, \mathcal{M} – множество допустимых метрик, λ – некоторая неотрицательная константа, \mathcal{L}_i – набор из k функционалов соответствия.

Также можно рассматривать задачу с ограничениями:

$$\begin{aligned} \min_{\rho \in \mathcal{M}} \quad & \mathcal{L}_0(\rho, \mathcal{S}, \mathcal{D}, \mathcal{R}) \\ \text{s. t.} \quad & \mathcal{L}_i(\rho, \mathcal{S}, \mathcal{D}, \mathcal{R}) \leq 0, \quad i = 1, \dots, m. \end{aligned} \quad (2)$$

Задача генерации метрики в свою очередь делится на две:

1. задача обучения метрики (Metric Learning). Здесь даны признаковые описания объектов. Цель – построить метрику как функцию от признаковых описаниях двух объектов. Обычно при решении данной задачи в качестве множества \mathcal{M} берётся множество обобщённых евклидовых метрик, имеющих в качестве параметра матрицу A – неотрицательно определённую матрицу; тогда итоговая метрика ищется в виде $\rho_A^2(x, y) = (x - y)^T A (x - y)$, где $A \in \mathcal{M}$;
2. задача извлечения метрики (Metric Extraction). Здесь нам даны какие-то функции расстояния ρ_1, \dots, ρ_K , цель задачи – построить метрику, которая является функцией от заданных нам. В большинстве работ итоговая метрика ищется как линейная комбинация заданных: $\forall x, y \quad \rho(x, y) = \sum_{i=1}^K w_i \rho_i(x, y)$, где $w_i \geq 0$.

Подходы к генерации метрик подробнее рассматриваются в главе 3.

1.3 Задача распознавания подписей

Известным полигоном для генерации метрики является задача распознавания подписей.

Под задачей распознавания подписи обычно подразумевают одну из следующих задач:

- задача идентификации пользователя – в этом случае нам на вход подаётся подпись S , и необходимо установить, какому пользователю она принадлежит;
- задача верификации подписи – в данной задаче нам на вход подаётся пара пользователь-подпись (U, S) ; нам необходимо ответить, действительно ли подпись S принадлежит пользователю U .

Задачу распознавания подписи можно разделить на несколько этапов. Сначала происходит регистрация эталона подписи. Человеку предлагается несколько раз ввести подпись для сбора статистики. Затем сравниваются характеристики эталонов и образца, происходит оценка, насколько они совпадают. При достаточной степени совпадения образец подписи считается подлинным. В противном случае, образец считается подделкой.

В зависимости от входных данных задача распознавания подписей делится на два основных вида:

- задача статической верификации (offline). В этом случае нам на вход подаётся изображение, содержащее подпись;
- задача динамической верификации (online). В данной задаче необходимо регистрирующее устройство (например, планшет). Здесь подписи представлены в виде троек (x_i, y_i, t_i) – позиции стилуса в разные моменты времени.

В данной работе решается задача динамической верификации подписей.

1.4 Оценка качества верификации

В задаче верификации подписи для оценки решающего правила основными являются следующие показатели качества [3]:

- FAR (False Acceptance Rate) – вероятность того, что поддельная подпись будет принята за оригинал;
- FRR (False Rejection Rate) – вероятность того, что оригинал будет принят за подделку;
- TPR (True Positive Rate) – вероятность того, что оригинал будет верно распознан.

На основании данных показателей качества принято вводить показатель, оценивающий качество семейства решающих правил – EER (Equal Error Rate). Пусть алгоритм $a(s)$ выдал числовые оценки подлинности подписей. Тогда порог t задаёт верификатор $y(s) = [a(s) > t]$. Данный верификатор имеет показатели качества $FAR(t)$ и $FRR(t)$. Пусть данные кривые в зависимости от t пересекаются в точке $t = t_0$. Тогда $EER := FAR(t_0) = FRR(t_0)$ (см. рис. 1).

ROC-кривая – функция $TPR(FAR)$. С помощью данной кривой можно определить EER как значение FAR : $FAR = 1 - TPR$. Данное определение проиллюстрировано на рис. 2.

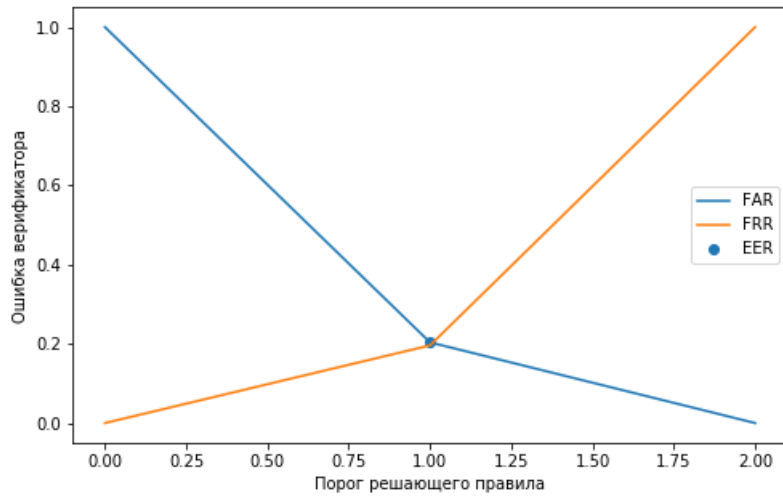


Рис. 1: Определение EER как ошибки в точке равенства FRR и FAR

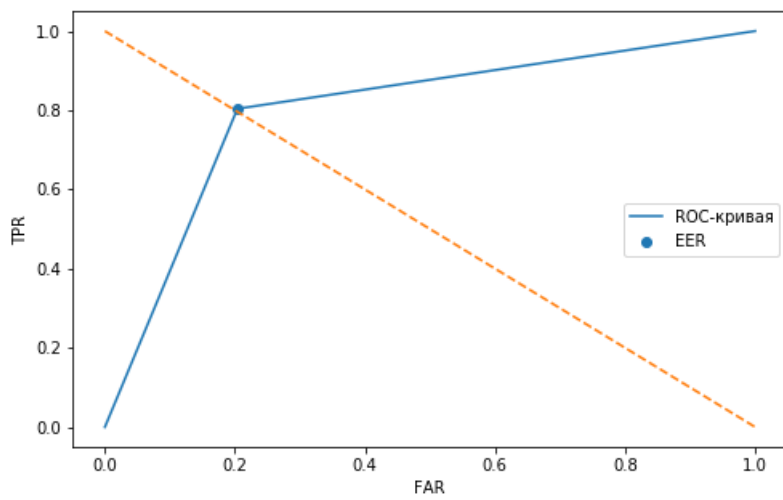


Рис. 2: Определение EER на ROC-кривой

2 Первичные метрики на описаниях динамических подписей

Напомним, что под динамической подписью подразумевается набор троек (x_i, y_i, t_i) – позиции стилуса в разные моменты времени. Из заданных позиций можно извлечь также следующие описания:

$$v_i = \sqrt{\frac{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}{(t_{i+1} - t_i)^2}}, \quad (3)$$

$$vx_i = \frac{x_{i+1} - x_i}{v_i(t_{i+1} - t_i)}, \quad (4)$$

$$vy_i = \frac{y_{i+1} - y_i}{v_i(t_{i+1} - t_i)}, \quad (5)$$

$$\theta_i = \arctan \frac{vy_i}{vx_i}. \quad (6)$$

Далее между заданными описаниями подписей можно считать различные метрики и строить решающие правила на основании метрик. Также на основе базовых метрик можно строить новую метрику, являющуюся комбинацией заданных.

2.1 Dynamic Time Warping (DTW)

Пусть даны две последовательности:

$$S = s_1, s_2, \dots, s_n,$$

$$T = t_1, t_2, \dots, t_m$$

Построим путь трансформации $W = w_1, \dots, w_k$, где w_l соответствует некоторой паре индексов (i_l, j_l) . Данный путь должен минимизировать расстояние $d_{total} = \sum_{l=1}^k dist(s_{i_l}, t_{j_l})$ для заданной функции расстояния $dist$.

Данный путь также должен удовлетворять следующим ограничениям [8]:

- монотонность: $i_{l-1} \leq i_l, j_{l-1} \leq j_l$;
- непрерывность: $i_l - i_{l-1} \leq 1, j_l - j_{l-1} \leq 1$;
- граничные условия: $i_1 = 1, j_1 = 1, i_k = n, j_k = m$.

Тогда

$$DTW(S, T) := \frac{\sum_l dist(s_{i_l}, t_{j_l})}{n + m}, \quad (7)$$

где $\{(i_l, j_l)_{l=1}^k\}$ – путь трансформации.

Данное расстояние можно ввести иным способом:

$$\begin{aligned} D(1, 1) &= dist(S_1, T_1), \\ D(i, j) &= \min\{D(i - 1, j), D(i, j - 1), D(i - 1, j - 1)\} + dist(s_i, t_j), \\ DTW(S, T) &:= \frac{D(n, m)}{n + m}. \end{aligned} \quad (8)$$

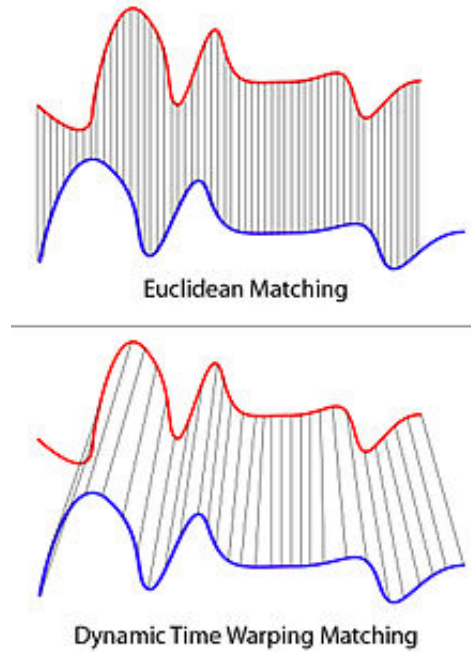


Рис. 3: DTW

В процессе построения пути трансформации по сути происходит сопоставление между точками двух временных рядов. На рис. 3 проиллюстрировано отличие полученного сопоставления от сопоставления точек при использовании евклидовой метрики.

2.2 Расстояние Вассерштейна (Earth Mover's Distance, EMD)

Пусть даны две гистограммы:

$$x = \{(x_1, w_1), \dots, (x_n, w_n)\},$$

$$y = \{(y_1, u_1), \dots, (y_m, u_m)\},$$

где x_i, y_j – веса соответствующих кластеров, w_i, u_j – их веса.

Пусть также задано расстояние $\{(d_{ij})_{i=1\dots n, j=1\dots m}\}$ между кластерами x_i и y_j .

Для заданных распределений x и y поток f – матрица размера $n \times m$. По смыслу, f_{ij} – количество массы первого распределения в точке x_i , сопоставляемое второму распределению в точке y_j . Введём множество допустимых потоков \mathcal{F} , как множество потоков f , удовлетворяющих следующим требованиям [9]:

- $f_{ij} \geq 0$;
- $\sum_j f_{ij} \leq w_i$;
- $\sum_i f_{ij} \leq u_j$;
- $\sum_i \sum_j f_{ij} = \min(\sum_i w_i, \sum_j u_j)$;

Тогда

$$EMD(x, y) := \min_{f \in \mathcal{F}} \frac{\sum_i \sum_j f_{ij} d_{ij}}{\sum_i \sum_j f_{ij}}. \quad (9)$$

2.3 «Расширенный» коэффициент детерминации (ER^2)

Для заданных кривых $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_n)$ коэффициент детерминации определяется следующим образом [7]:

$$R^2 = \frac{(\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y}))^2}{\sum_{i=1}^n (x_i - \bar{X})^2 \sum_{i=1}^n (y_i - \bar{Y})^2} \quad (10)$$

Коэффициент детерминации обладает следующими свойствами:

- рефлексивность: $R^2(x, x) = 1$;
- симметричность;
- $R^2 \in [0, 1]$.

В работе [7] по аналогии предлагается мера ER^2 для многомерных временных рядов:

$$ER^2 = \frac{(\sum_{j=1}^K \sum_{i=1}^n (x_{ji} - \bar{X}_j)(y_{ji} - \bar{Y}_j))^2}{\sum_{j=1}^K \sum_{i=1}^n (x_{ji} - \bar{X}_j)^2 \sum_{j=1}^K \sum_{i=1}^n (y_{ji} - \bar{Y}_j)^2} \quad (11)$$

ER^2 обладает теми же свойствами, что и коэффициент детерминации.

После введения меры сходства $ER_{measure}^2$ можно ввести «метрику» $ER^2 := 1 - ER_{measure}^2$.

2.4 Коэффициент детерминации для временных рядов разной длины ($ER^2 + DTW$)

С помощью ER^2 можно измерить сходство только между временными рядами равной длины. В той же статье [7] предлагается объединить DTW с ER^2 : пусть $\{(i_l, j_l)_{l=1\dots k}\}$ – путь трансформации, полученный во время подсчёта DTW . Тогда составим новые временные ряды: $\hat{X} = \{(X_{i_l})_{l=1\dots k}\}$, $\hat{Y} = \{(Y_{j_l})_{l=1\dots k}\}$. Далее между полученными рядами есть возможность посчитать ER^2 .

3 Модели комбинирования метрик и методы их обучения

В данной работе рассматривается задача извлечения метрики, при этом метрика ищется в виде линейной комбинации заданных:

$$\rho(x, y) = \sum_{i=1}^K w_i \rho_i(x, y), \quad (12)$$

где $\mathcal{M} = \{\mathbf{w} | w_i \geq 0, i = 1 \dots K\}$.

Утверждение 1 [11] Если ρ_1, \dots, ρ_k – метрики, то и их комбинация $\sum_{i=1}^k w_i \rho_i$, где $w_i \geq 0$, является метрикой.

3.1 Основные подходы генерации метрик

Рассмотрим некоторые подходы генерации метрик в общем виде. В частном случае, определяя множество допустимых метрик \mathcal{M} , можно решать либо задачу обучения метрики, либо задачу извлечения метрики.

- MMC [1]:

$$\begin{aligned} \max_{\rho \in \mathcal{M}} \quad & \sum_{(x_i, x_j) \in \mathcal{D}} \rho(x_i, x_j) \\ \text{s. t.} \quad & \frac{1}{|\mathcal{S}|} \sum_{(x_i, x_j) \in \mathcal{S}} \rho(x_i, x_j) \leq 1. \end{aligned} \quad (13)$$

- SAME [10]:

$$\begin{aligned} \min_{\rho \in \mathcal{M}} \quad & \sum_{(x_i, x_j) \in \mathcal{S}} \rho(x_i, x_j) \\ \text{s. t.} \quad & \rho(x_i, x_j) \geq 1 \quad \forall (x_i, x_j) \in \mathcal{D}. \end{aligned} \quad (14)$$

Утверждение 2 [10] Если не существует пары объектов $(x, y) \in \mathcal{D}$: $\rho_i(x, y) = 0 \quad \forall i = 1 \dots K$, то задача 14 имеет решение.

- LMNN [2].

Интуиция данного метода в следующем: объекты, находящиеся рядом, должны быть из одного класса, в то время как объекты разных классов должны находиться далеко друг от друга.

В LMNN для каждого объекта x_i вводятся следующие понятия:

- целевые соседи (target neighbors) – объекты из того же класса, которые должны быть максимально близки к x_i ;
- «самозванцы» (impostors) – объекты, которые принадлежат другому классу, но при этом являются ближайшими соседями x_i .

Тогда множество \mathcal{R} представляет собой множество троек (x, y, z) , таких что y – целевой сосед для x , z – "самозванец" для x .

Сама задача оптимизации следующая:

$$\min_{\rho \in \mathcal{M}} \sum_{(i,j) \in \mathcal{S}} \rho(x_i, x_j) + \lambda \sum_{(i,j,k) \in \mathcal{R}} [1 + \rho(x_i, x_j) - \rho(x_i, x_k)]_+ \quad (15)$$

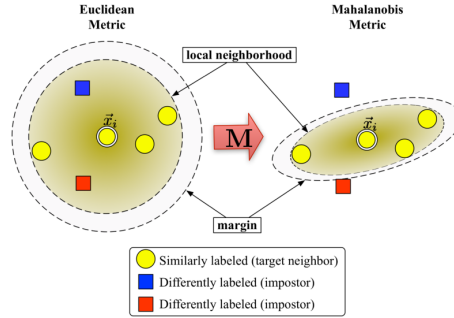


Рис. 4: Иллюстрация работы метода LMNN

В данной работе метод LMNN не рассматривается.

3.2 Расширения SAME

Заметим, что по аналогии с (14) можно поставить другие задачи:

1. Двойственная задача. Вместо минимизации суммы расстояний между похожими объектами можно максимизировать сумму расстояний между непохожими:

$$\begin{aligned} \max_{\rho \in \mathcal{M}} \quad & \sum_{(x_i, x_j) \in \mathcal{D}} \rho(x_i, x_j) \\ \text{s. t.} \quad & \rho(x_i, x_j) < 1 \quad \forall (x_i, x_j) \in \mathcal{S}. \end{aligned} \tag{16}$$

2. с целью борьбы с переобучением можно ввести регуляризацию $R(\rho)$. Минус данного метода в том, что при решении задачи извлечения, в отличие от исходного метода (14), он не способен автоматически шкалировать входные метрики.

$$\begin{aligned} \min_{\rho \in \mathcal{M}} \quad & \sum_{(x_i, x_j) \in \mathcal{S}} \rho(x_i, x_j) + \lambda R(\rho) \\ \text{s. t.} \quad & \rho(x_i, x_j) \geq 1 \quad \forall (x_i, x_j) \in \mathcal{D}. \end{aligned} \tag{17}$$

3. чтобы не сильно затачиваться под отдельно взятую подпись, можно позволить расстоянию между непохожими объектами быть меньше единицы, но ввести штрафные переменные:

$$\begin{aligned} \min_{\rho \in \mathcal{M}} \quad & \sum_{(x_i, x_j) \in \mathcal{S}} \rho(x_i, x_j) + \lambda \sum_{(i, j): (x_i, x_j) \in \mathcal{D}} \xi_{i, j} \\ \text{s. t.} \quad & \rho(x_i, x_j) \geq 1 - \xi_{i, j} \quad \forall (x_i, x_j) \in \mathcal{D}, \\ & \xi_{i, j} \geq 0, \forall i, j : (x_i, x_j) \in \mathcal{D}. \end{aligned} \tag{18}$$

3.3 Способы обучения

Заметим, что сгенерировать метрику можно двумя способами:

1. для каждого пользователя вектор весов \mathbf{w} обучается отдельно;
2. вектор весов \mathbf{w} обучается для всех пользователей сразу.

Второй способ не требует большого объёма выборки для нового пользователя: после получения первого эталона уже можно применять методы верификации. Зато при наличии достаточного объёма выборки первый способ позволяет извлечь особенности конкретного человека.

3.4 Генерация множеств \mathcal{S} и \mathcal{D}

Рассмотрим, как можно генерировать множества, возникающие в задаче генерации метрики, при решении задачи верификации подписи.

Пусть G_U – множество подлинных подписей данного пользователя U , F_U – множество подделок подписей пользователя U .

Тогда множество $\mathcal{S} := G_U \times G_U$ – декартов квадрат множества подлинных подписей, $\mathcal{D} := G_U \times F_U$ – декартово произведение множества подлинных подписей на множество подделок.

3.5 Этап применения

На данном этапе мы уже сгенерировали метрику и требуется по паре (U, S) определить, принадлежит ли подпись S пользователю U . На этом этапе можно действовать двумя способами:

- для данного пользователя U мы имеем множество подлинных подписей \mathcal{G} и множество \mathcal{F} подделок. Задачу верификации можно рассматривать как задачу двухклассовой классификации и решать с помощью, например, метода ближайших соседей. Минус данного подхода в том, что он предполагает близость между собой поддельных подписей;
- другой подход заключается в нахождении расстояний от подписи S до всех эталонных подписей из множества \mathcal{G} . Далее находится минимальное расстояние $d(S) = \min_{g \in \mathcal{G}} \rho(S, g)$ и пороговый классификатор строится в виде $a_t(S) = \mathbb{I}[d(S) < t]$. К примеру, при использовании метода SAME естественно будет взять в качестве $t = 1$.

4 Эксперименты

4.1 Описание данных

Для тестирования методов были использованы наборы данных *SVC2004* [4] и *MOBISIG*[5]. Оба набора содержат подписи в виде троек (x_i, y_i, t_i) .

В *SVC2004* содержатся подписи 80 пользователей. На каждого пользователя дано 20 подлинных подписей и 20 подделок (skilled).

В *MOBISIG* содержатся подписи 83 пользователей. На каждого пользователя дано 45 подлинных подписей и 20 подделок (skilled).



Рис. 5: SVC, Примеры подлинных подписей

Рис. 6: SVC, Примеры поддельных подписей

4.2 Предобработка

Для повышения качества верификации предварительно предобработаем все подписи:

1. Поворот. Введём обозначения:

$$\overline{x^2} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (19)$$

$$\overline{xy} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad (20)$$

$$A = \begin{bmatrix} \overline{x^2} & \overline{xy} \\ \overline{xy} & \overline{y^2} \end{bmatrix} \quad (21)$$

Тогда воспользуемся методом, предложенным в статье [6]: повернём все точки кривой (x_i, y_i) на угол, обратный углу, соответствующему собственному вектору, который соответствует максимальному собственному значению матрицы A .

2. Нормализация. Отобразим координаты на отрезок $[0, 1]$:

$$x_i^{new} = \frac{(x_i^{old} - x_{min})}{(x_{max} - x_{min})}, \quad (22)$$

$$y_i^{new} = \frac{(y_i^{old} - y_{min})}{(y_{max} - y_{min})}. \quad (23)$$

4.3 Базовые метрики

В качестве первичных метрик для последующей агрегации взяты следующие функции:

1. DTW между координатами $(x, y) - DTW_{xy}$;
2. DTW между скоростями $(v_x, v_y) - DTW_v$;
3. ER^2 между координатами $(x, y) - ER_{xy}^2$;
4. ER^2 между скоростями $(v_x, v_y) - ER_v^2$;
5. EMD между гистограммами $\theta - EMD_\theta$;
6. EMD между гистограммами $v_x - EMD_{vx}$;
7. EMD между гистограммами $v_y - EMD_{vy}$;
- 8.

$$SumMin_{xy}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^n \min_j dist(x_i, y_j) + \sum_{j=1}^m \min_i dist(x_i, y_j)}{2} \quad (24)$$

между координатами (x, y) .

При проведении экспериментов использовались открытые реализации подсчёта метрик EMD (ссылка: <https://github.com/wmayner/pyemd>) и DTW (ссылка: <https://github.com/pierre-rouanet/dtw>).

4.4 Разбиение выборки

Подписи каждого человека разбивались на 2 части: обучение и контроль. Контроль для каждого пользователя U состоял из 5 подлинных (S_1^U, \dots, S_5^U) и 5 поддельных (S_6^U, \dots, S_{10}^U) подписей, случайно взятых из полного набора. Далее метрика обучалась одним из следующих способов:

- для всех пользователей как одно общее решение SAME (14);
- для всех пользователей как одно общее решение MMC (13);
- отдельно для каждого пользователя решение SAME (14);
- отдельно для каждого пользователя как решение MMC (13);
- отдельно для каждого пользователя как решение двойственной к SAME (16);
- отдельно для каждого пользователя как решение SAME с L_1 -регуляризацией $(R(w) = \sum_i |w_i|)$ (17);
- отдельно для каждого пользователя как решение SAME со штрафами (18), где $\lambda = 1$.

4.5 Измерение качества верификации

Для измерения качества верификации был использован второй подход, описанный в разделе 3.5: для каждого пользователя U для подписи S из контрольной выборки считается расстояние до всех подлинных эталонных подписей из множества \mathcal{G}_U . Далее оценка подлинности подписи вводится как $d(S) = \min_{g \in \mathcal{G}_U} \rho(S, g)$.

Далее по заданным оценкам подлинности $d(S_1^U), \dots, d(S_{10}^U)$ подписей из контроля для пользователя U считается показатель EER.

В качестве финального показателя качества взят EER, усреднённый по пользователям.

4.6 Результаты и выводы

В таблицах 1 и 2 указаны средние EER при построении решающего правила на основании лишь одной базовой метрики.

В таблицах 3 и 4 указаны средние EER при построении решающего правила на основе комбинированной метрики, полученной в результате решения задачи Metric Extraction.

Наилучшего качества верификации при построении верификатора на основе одной функции расстояния получилось достичь с помощью метрики ER^2 между координатами (x, y) на наборе данных SVC2004. На наборе данных MOBISIG лучшее качество при построении верификатора на основе одной функции расстояния показала метрика DTW_{xy} .

Лучшее качество верификации было достигнуто при использовании метрики, полученной с помощью стандартного метода SAME (14), где веса \mathbf{w} обучались отдельно для каждого пользователя.

Заметим, что почти всегда функция расстояния, получаемая при решении задачи извлечения метрики обеспечивала лучший верификатор, чем одиночная функция расстояния. Традиционно плохо работал метод ММС.

Таблица 1: усреднённый по пользователям EER при использовании базовых метрик на SVC2004

Метрика	EER
DTW_{xy}	8%
DTW_v	10%
ER_{xy}^2	6.75%
ER_v^2	15%
EMD_θ	16.75%
EMD_{vx}	20.5%
EMD_{vy}	26.25%
$SumMin_{xy}$	12.5%

Таблица 2: усреднённый по пользователям EER при использовании базовых метрик на MOBISIG

Метрика	EER
DTW_{xy}	7.23%
DTW_v	8.43%
ER_{xy}^2	8.67%
ER_v^2	9.88%
EMD_{θ}	14.94%
EMD_{vx}	25.06%
EMD_{vy}	22.65%
$SumMin_{xy}$	13.49%

Таблица 3: усреднённый по пользователям EER в зависимости от используемого метода на наборе данных SVC2004

Метод	EER
SAME (14), для каждого пользователя	5%
двойств. к SAME(16), для каждого пользователя	6.25%
SAME со штрафами(18), для каждого пользователя	5.25%
SAME с L1-регуляризацией(17), для каждого пользователя	6.25%
SAME (14), для всех	7.25%
ММС, для каждого пользователя	9%
ММС, для всех	6.5%

Таблица 4: усреднённый по пользователям EER в зависимости от используемого метода на наборе данных MOBISIG

Метод	EER
SAME (14), для каждого пользователя	4.8%
двойств. к SAME(16), для каждого пользователя	7%
SAME со штрафами(18), для каждого пользователя	6.5%
SAME с L1-регуляризацией(17), для каждого пользователя	5.54%
SAME (14), для всех	5.1%
ММС, для каждого пользователя	8.43%
ММС, для всех	8.67%

5 Заключение

В данной работе были исследованы разные подходы для решения задачи генерации метрики. Были рассмотрены базовые функции, оценивающие сходство между двумя подписями, и различные модели их комбинирования.

Были применены методы извлечения метрик применительно к задаче верификации подписей. Было показано, что метод SAME позволяет достичь лучшего качества верификации, чем метод MMC и чем верификатор, основанный на использовании любой одиночной функции расстояния.

Были предложены модификации метода SAME, в частности использующие регуляризацию, а также использующие штрафные переменные для предотвращения настройки на конкретную подпись. Было проведено сравнение предложенных модификаций со стандартным SAME; с помощью модификаций не получилось достигнуть лучшего качества, чем при использовании стандартного метода SAME.

Список литературы

- [1] Bellet A., Habrard A., Sebban M. A survey on metric learning for feature vectors and structured data //arXiv preprint arXiv:1306.6709. – 2013.
- [2] Kulis B. et al. Metric learning: A survey //Foundations and Trends® in Machine Learning. – 2013. – Т. 5. – №. 4. – С. 287-364.
- [3] Сущенко О. А. Оценка эффективности работы биометрических систем //Системы обработки информации. – 2011. – №. 4. – С. 79-81.
- [4] Yeung D. Y. et al. SVC2004: First international signature verification competition //Biometric Authentication. – Springer, Berlin, Heidelberg, 2004. – С. 16-22.
- [5] Antal M., Szabó L. Z., Tordai T. Online Signature Verification on MOBISIG Finger-Drawn Signature Corpus //Mobile Information Systems. – 2018. – Т. 2018.
- [6] Kalera M. K., Srihari S., Xu A. Offline signature verification and identification using distance statistics //International Journal of Pattern Recognition and Artificial Intelligence. – 2004. – Т. 18. – №. 07. – С. 1339-1360.
- [7] Lei H., Palla S., Govindaraju V. ER/sup 2: an intuitive similarity measure for on-line signature verification //Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on. – IEEE, 2004. – С. 191-195.
- [8] Berndt D. J., Clifford J. Using dynamic time warping to find patterns in time series //KDD workshop. – 1994. – Т. 10. – №. 16. – С. 359-370.
- [9] Rubner Y., Tomasi C., Guibas L. J. A metric for distributions with applications to image databases //Computer Vision, 1998. Sixth International Conference on. – IEEE, 1998. – С. 59-66.
- [10] Maysuradze A., Shekar B. H., Suvorov M. Supervised Asymmetric Metric Extraction: An Approach to Combine Distances //International Conference on Mining Intelligence and Knowledge Exploration. – Springer, Cham, 2017. – С. 123-132.
- [11] Wiercioch M. et al. Mixture of metrics optimization for machine learning problems //Schedae Informaticae. – 2016. – Т. 2015. – №. Volume 24. – С. 83-92.