Санкт-Петербургский политехнический университет
Петра Великого

Институт прикладной математики и механики
Кафедра «Прикладная математика»

КУРСОВАЯ РАБОТА
ПО ДИСЦИПЛИНЕ «МЕТОДЫ ОПТИМИЗАЦИИ»
«СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ ПРИ ИСПОЛЬЗОВАНИИ SVM В ЗАДАЧАХ РАСПОЗНОВАНИЯ»

Выполнили
студенты группы 3630102/80201

Деркаченко А. О.
Классический метод машины опорных векторов, Методы формирования вектора признаков изображения лица для определении атрибутов личности, Сравнительный анализ
Хрипунков Д. В.
Введение, Постановка задачи, Практическое задание
Войнова А. Н.
Методы квадратичного программирования, Сравнительный анализ

Руководитель
к. ф.-м. н., доц.

Родионова Елена Александровна

Санкт-Петербург
2021

Содержание

1 Введение

Увеличение объёмов информации в современном мире привело к тому, что ручная обработка такой информации стала невозможной и возникла небходимость в создании систем и алгоритмов, которые автоматизируют эту работу. Одной из задач обработки информации оказалась задача распознавания - определение признаков, которые отличают один набор данных от других.

Обработка неструктурированных данных (фото, видео, аудио) является достаточно сложной, и самым популярным способом решения такой задачи является машинное обучение, которое пытается воссоздать процесс человеческого обучения на компьютере - группировку объектов в классы по некоторым признакам. Выделяются два вида машинного обучения: "с учителем"и "без учителя"[1, 19]:

- "Обучение с учителем"предполагает наличие некой исходной выборки, заведомо разделённой на классы учителем, а системе предлагается обнаружить общие признаки, которые будут описывать класс. В дальнейшем, система сможет на основе этих признаков распределять неразмеченные данные по классам. Такое обучение решает задачу классификации, когда количество классов заранее известно, и от системы требуется лишь отнести данные к одному из них [1, 19].

- "Обучение без учителя"может предполагать неизвестное число классов, а входные данные изначально не являются размеченными. Система должна сама определить правила, которые различают предложенные объекты, и на их основе создать классы. Классы могут быть как уже известные, так и созданые новые в процессе распознавания. Такое обучение уже решает задачу распознавания - формирование правила, которое разделяет объекты разных классов [1, 19].

Самым распространённым видом информации в современном мире является фотография, пользователи социальных сетей активно делятся ими, а качество систем фото- и видеонаблюдения неуклонно растёт. Такое широкое распространение фотографии неуклонно приводит к тому, что задача распознавания ставится и в этом поле - распознавание информации с фотоснимков. Распознавание снимков чаще всего ставит цель в распознавании объектов, будь то буквы, цифры, дома, животные или люди. В этой работе мы рассмотрим частный случай - задачу распознавания лиц со снимков.

Тем не менее, задача распознавания может быть поставлена и в других сферах. Например, также крайне популярным и развивающимся полем является распознавание в речи, которое ставит сразу множество задач, таких как преобразование речи в текст, синтез речи из текста и определение дикторов и относящихся к ним фраз.

Распознавание объектов на изображениях можно разбить на ряд подзадач [1, 20]:

- Сопоставление

- Поиск

- Восстановление

- Классификация

На примере распознавания лиц эти пункты можно описать как:

- Анализ набора изображений для определения принадлежности к одному и тому же классу
- Поиск на изображении фрагмента для распознавания
- Восстановление пропущенных фрагментов по контексту
- Определение класса, к которому относится изображение

Для того чтобы распознавать человеческие лица, выделим три основных аттрибута: возраст, расу и пол. Практически такое разделение может быть применено во многих сферах: поисковая выдача, оценка аудитории, реклама, обучение, возрастные и половые ограничения и многие другие [1, 22]. Существуют также и другие признаки, по которым возможно разделение личностей, но в данной работе будут рассматриваться указанные выше атрибуты.

Задача распознавания по этим признакам интересна ещё и тем, что каждый из них имеет разные категории: числовую, бинарную и множественную нечисловую. А задача определения возраста усложняется ещё и тем, что признаки старения у разных людей проявляются по-разному [1, 25].

2 Постановка задачи

2.1 Постановка задачи квадратичного программирования

Задача квадратичного программирования - задача минимизации квадратичной функции на выпуклом многогранном множестве Ω [3, 5]:

$$F(x) = \frac{1}{2} < Dx, x > + < c, x > \to \min_{x \in \Omega}, \tag{1}$$

где $N = \{1, ..., n\}$, $c \in \mathbb{R}^n$, матрица $D[N, N]$ симметрична и положительно определена, а

$$\Omega = \left\{ x[N] \;\middle|\; \begin{array}{c} A[M_1, N] * x[N] \geq b[M_1] \\ A[M_2, N] * x[N] = b[M_2] \\ x[N_1] \geq 0, N_1 \subset N \end{array} \right\}. \tag{2}$$

Оптимальный план задачи - вектор $x^* \in \Omega : F(x^*) \leq F(x) \forall x \in \Omega$.

Двойственная задача квадратичного программирования:

$$G(y) = \frac{1}{2} < Qy, y > + < d, y > -\frac{1}{2} < D^{-1}c, c > \to \max_{y \in \Omega}, \tag{3}$$

где $c \geq 0, Q = -AD^{-1}A^T, d = -b - AD^{-1}c$ [5, 13].

Теорема 1: Задача (1) разрешима $\leftrightarrow \Omega \neq \emptyset$ и целевая функция $F(x)$ ограничена снизу на множестве планов [3, 13].

Теорема 2: Пусть $\Omega \neq \emptyset$. Задача (1) разрешима, если выполнено хотя бы одно из условий [3, 13]:

1. множество Ω ограничено
2. матрица D положительно определена
3. линейная часть целевой функции $< c, x >$ ограничена снизу на Ω

Лемма: вектор $x^* \in \Omega$ - решение задачи (1) $\leftrightarrow < F'(x^*), (x - x^*) > \geq 0 \forall x \in \Omega$ [3, 16-17].

Теорема Куна-Такера: вектор $x^*[N]$ - решение задачи (1) $\leftrightarrow \exists u^*[M], y^*[N], v^*[N]$, удовлетворяющих условиям:

$$\begin{cases} D[N,N] * x^*[N] - A^T[N,M] * u^*[M] - y^*[N] = -c[N] \\ A[M,N] * x^*[N] - v^*[M] = b[M] \\ y^*[N_2] = 0 \\ v^*[M_2] = 0 \\ x^*[N_1] \geq 0 \\ u^*[M_1] \geq 0 \\ y^*[N_1] \geq 0 \\ v^*[M_1] \geq 0 \\ x^*[N_1] * y^*[N_1] = 0 \\ u^*[M_1] * v^*[M_1] = 0 \end{cases} \tag{4}$$

При этом $F(x^*) = \frac{1}{2}(< c, x^* > + < b, u^* >)$.

Таким образом, решение задачи (1) равносильно разрешению системы теоремы Куна-Такера. Если данная система несовместна, то исходная задача неразрешима, и наоборот [3, 17-21].

2.2 Общая постановка задачи распознавания

Рассмотрим общую постановку задачи на примере распознавания символов. Предположим, система распознавания получила на вход некоторый символ (паттерн) X, который нужно распознать.

Система может считывать скорость изменения закрашенной поверхности как функцию от времени X(t), называемую представлением символа X. Альтернативным вариантом можно считывать сигнал в дискретные моменты времени, в результате чего получается вектор $\bar{X}$. Также возможны переходы из представления в виде функции в векторное [2, 11].

Предположим, что есть некоторое множество непересекающихся классов $\Omega = \{\bar{\omega_1}, ..., \bar{\omega_m}\}$, где каждое ω_i отвечает некоторому символу. Системе распознавания нужно отнести входящий символ X к какому-то из классов ω_i. Для этого предпринимаются следующие шаги [2, 11]:

1. Символ X считывается в представление X(t)

2. Представление X(t) преобразуется в векторную форму $\bar{\mathrm{X}}$

3. Из вектора $\bar{\mathrm{X}}$ извлекаются информативные признаки и образуется вектор $\hat{\mathrm{X}}$

4. Классификатор определяет, к какому классу относятся признаки $\hat{\mathrm{X}}$

Для финального соотнесения к некоторому классу используется классификатор - набор правил для определения класса. После классификации символ может быть определён к одному из существующих классов или как неотносящийся ни к одному из них. Классификация может выполняться, например, посредством вычисления расстояния между классом и вектором признаков $\hat{\mathrm{X}}$ [2, 11].

Также в схеме распознавания может присутствовать блок обучения. Он выбирает учебные образы, которые заведомо распределены по классам. С их помощью можно сформировать правила классификации или определить наиболее информативные признаки [2, 11].

Декомпозируя задачу распознавания на подзадачи, получается следующий набор [2, 12]:

1. *Математически описать образ*
   Такое описание удобнее всего проводить в векторной форме. Образу X сопоставляется некоторый вектор $\bar{X} = (x_1, x_2, ...)^T$, где каждое x_i - некоторый признак, а $\bar{X}$ - элемент конечномерного метрического векторного пространства $\mathbb{X}$

2. *Выбрать информативные признаки*
   Не все признаки символа могут быть одинаково полезны при распознавании. Задача состоит в том, чтобы определить минимально необходимый набор признаков, достаточных для распознавания символа. Этот набор система должна определить сама.

3. *Описать классы*
   Необходимо задать границы классов. Это может быть проделано на этапе разработки или самой системой в ходе её работы.

4. *Определить методы классификации*
   Нужно определить методы, по которым образы будут соотноситься некоторым классам.

5. *Определить оценку достоверности распознавания*
   Оценка нужна для того, чтобы иметь возможность определить величину потерь при неправильной классификации.

Математически задачу распознавания можно поставить так:

Дано множество образов U, отдельный образ обозначим $x \in U$. Из (возможно, несчётного) множества признаков образов x нужно выбрать конечное подмножество - пространство признаков. Пространство признаков конечномерное, линейное или метрическое, обозначим как X. Каждому образу соответствует элемент $\bar{x} \in U$ и оператор $P : U \longrightarrow X$ отображения образа в пространство признаков [2, 15].

Введём конечное множество классов $\Omega = \{\bar{\omega_1}, ..., \bar{\omega_m}\}$, дял которого верно, что $\cup_{i=1}^m \omega_i = U$ и $\omega_i \cap \omega_j = \varnothing, \forall i \neq j$. Для классификации образа $x \in U$ по классам из Ω нужно найти индикаторную функцию $g : U \longrightarrow Y$, $Y = \{y1, ..., y_m\}$, где Y - множество меток класса. То есть $g(x) = y_i$, если $x \in \omega_i$. Поскольку в реальности мы работаем не с самим образами, а с их признаками, то нужно найти решающую функцию $\tilde{g} : X \longrightarrow Y$ для $\bar{x} = P * x \in X$, то есть $\tilde{g}(\bar{x}) = y_i$, если $\bar{x} = P * x \in \omega_i$ [2, 15].

Поскольку множество $P^{-1}\bar{x}$, $\bar{x} \in X$ может иметь непустые пересечения с разными классами ω_i, то функция $\tilde{g}(x)$ будет неоднозначной, тогда из неё нужно выделить однозначную ветвь, удовлетворяющую условиям оптимальности, например, минимальность ошибки неправильной классификации. На этапе обучения система известны некоторые пары $(\bar{x}_j, y_j)$, называемые прецендентами и множество $H = \{\bar{x_1}, ..., \bar{x_N}\}$, называемое обучающей выборкой. По множеству прецендентов (H, Y) нужно построить решающую функцию $\tilde{g}(x)$, которая будет осуществлять классификацию [2, 16].

2.3 Постановка задачи по определению атрибутов личности по изображению лица

Задача определения атрибутов личности по изображению лица - частный случай задачи обучения по прецендентам, описанной выше, где множество U - множество нормализованных изображений [1, 28].

Для каждого из выбранных нами атрибутов будет выбрана соответствующая задача классификации в зависимости от множества меток классов Y [1, 31]:

1. Для атрибута "пол"будет бинарная классификация $Y = \{-1, 1\}$
2. Для атрибута "раса"множественная классификация $Y = \{-1, 0, 1\}$ для рас "европеоидная "монголоидная"и "негроидная"
3. Для атрибута "возраст"восстановление регрессии $Y = [5, 100]$

Эти задачи связаны друг с другом, так, например, задача множественной классификации может быть разложена на несколько задача бинарной классификации, а задача восстановления регрессии формулируется исходя из решения задачи множественной классификации [1, 31].

Задача определения атрибутов личности по изображению лица сводится к тому, что нужно определить метод формирования вектора признаков $P : U \longrightarrow X$ и решающую функцию $\tilde{g} : X \longrightarrow Y$ так, чтобы оценка достоверности распознавания была максимальной. При этом считается, что изображения уже нормализованы, то есть приведены к некоторому унифицированному виду, допускающему применение единых методов его обработки на последующих этапах.

Решение такой задачи получается сравнением решений из конечного множества частных оптимизационных задач, в которых выбраны разные способы формирования вектора признаков и решающей функции [1, 32].

3 Классический метод машины опорных векторов

3.1 Краткое описание метода [1, 57-63]

Метод опорный векторов (SVM) разработан в 60-е годы коллективом советских математиков под руководством В.Н.Вапника и рассчитан на классифицирование объектов по двум классам.

Пусть имеется обучающая выборка $G^l, |G^l| = n$, заданная множеством пар прецедентов $(\bar{x}_i, y_i), i = \overline{1, n}, \bar{x}_i \in \mathbb{R}^m, y_i \in \{-1, +1\}$.

3.1.1 Разделяющая гиперплоскость

Множество F_{SVM}, из которого выбираются решающие функции по методу опорных векторов, образовано функциями вида:

$$f(\bar{x}) = sign(< \bar{w}, \bar{x} > +w_0), \tag{5}$$

где $<, >$ - скалярное произведение векторов, $\bar{w}$ - ортонормированный вектор к разделяющей классы гиперплоскости, w_0 - вспомогательный параметр (сдвиг гиперплоскости).

Так как любая гиперплоскость может быть задана в виде $< \bar{w}, \bar{x} > +w_0 = 0$, то объекты с $f(\bar{x}) \leq -1$ попадут в один класс, а объекты с $f(\bar{x}) \geq +1$ - в другой.

Базовая идея метода: найти такие $\bar{w}, w_0$, которые максимизируют расстояние между классами, что приводит к более уверенной классификации объектов. При этом считается, что нормировка параметров уже произведена.

То есть условие $-1 << \bar{w}, \bar{x} > +w_0 < +1$ задает полосу, разделяющую классы. При этом ни одна точка из множества X^l не должна лежать внутри полосы, а границами полосы являются две параллельные гиперплоскости, проходящие через точки (объекты), ближайшие к разделяющей гиперплоскости, которая находится по середине данной полосы. И объекты, через которые проходят границы полосы, называются *опорными векторами.*

Ширина полосы будет равна $\frac{2}{||w||}$, и она максимальна, когда норма вектора w минимальна [4, 3].

3.1.2 Линейно разделимый случай выборки

Проблема нахождения максимума расстояния сводится к нахождению минимума $||\bar{w}||^2$, которая является стандартной прямой задачей квадратичного программирования:

$$\begin{cases} \frac{1}{2}||\bar{w}||^2 \to \min \\ y_i < \bar{w}, \bar{x}_i + w_0 >\geq 1 \end{cases} \tag{6}$$

и решается методом множителей Лагранжа.

Задача квадратичного программирования, содержащая только двойственные переменные метода множителей Лагранжа λ_i, имеет вид:

$$\begin{cases} -\sum_{i=1}^{n} \lambda_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j < \bar{x}_i, \bar{x}_j > \to \min_{\lambda} \\ \sum_{i=1}^{n} \lambda_i y_i = 0 \\ \lambda_i \geq 0 \end{cases} \tag{7}$$

В результате рещающая функция приобретает вид:

$$f(\bar{x}) = sign(\sum_{i=1}^{n} \lambda_i y_i < \bar{x}_i, \bar{x} > + w_0), \tag{8}$$

где параметр $w_0 = med\{< \bar{w}, \bar{x}_i > - y_i\}, \lambda_i \neq 0$.

3.1.3 Случай линейно неразделимой выборки

Вышеуказанные рассуждения справедливы для линейно разделимой обучающей выборки. Но на практике встречаются случае линейной неразделимости и решающей функции позволяют допускать ошибки на обучающей выборке, но эти ошибки минимизируют и используют управляющую константу C как компромисс между максимизацией ширины разделяющей полосы и минимизацией суммарной ошибки $\xi_i \geq 0$. Тогда вводят ограничение сверху $0 \leq \lambda_i \leq C$ и такой алгоритм называют SVM с "мягким зазором"(soft-margin SVM), иначе имеется "жесткий"зазор.

Тогда прямая задача квадратичного программирования имеет вид:

$$\begin{cases} \frac{1}{2}||\bar{w}||^2 + C\sum_{i=1}^{n} \xi_i \to \min \\ y_i < \bar{w}, \bar{x}_i + w_0 > \geq 1 - \xi_i \end{cases} \tag{9}$$

А двойственная задача квадратичного программирования находится как:

$$\begin{cases} -\sum_{i=1}^{n} \lambda_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j < \bar{x}_i, \bar{x}_j > \to \min_\lambda \\ \sum_{i=1}^{n} \lambda_i y_i = 0 \\ 0 \leq \lambda_i \leq C \end{cases} \tag{10}$$

3.1.4 Ядра

Если признаки x_i заданы в виде функции $\theta(x_i)$, то решающая функция строится аналогично. Тогда функция $K(u, v) =< \theta(u), \theta(v) >$ - *ядро*, если она симметрична и положительно определена. Для решения практических задач классификации изображений по атрибуту "пол"используют RFB-ядро вида:

$$K(\bar{x}_i, \bar{x}) = \exp(-\gamma||\bar{x}_i - \bar{x}||^2), \tag{11}$$

вычисляющее оценку близости вектора $\bar{x}$ к опорному вектору $\bar{x}_i$, где γ - некоторый параметр.

Тогда решающая функция принимает вид:

$$f(\bar{x}) = sign(\sum_{i=1}^{n} \lambda_i y_i K(\bar{x}_i, \bar{x}) + w_0), \tag{12}$$

3.2 Множественная классификация по атрибуту

Классификация по атрибуту "раса"является типичной задачей множественной классификации. Для решения такой задачи используется подход *"один против всех"*, реализующий сведение задачи множественной классификации к последовательному применению бинарных классификаторов. В рамках данного подхода строится бинарное дерево решающих функций $f \in F$, каждая из которых выделяет только один класс объектов.

В случае классификации по атрибуту "раса"на первом шаге объекты разделяются решающей фукцией f_1 на два класса: "европеоиды"и "все остальные". Если объект не попал в класс "европеоиды то на втором шаге другая решающая функция f_2 производит разделение на класс "монголоиды"и "все остальные"("негроиды").

Данный подход позволяет использовать большую часть разработок в области бинарной классификации для решения задач множественной классификации.

Общая формулировка прямой задачи имеет вид:

$$\begin{cases} \frac{1}{2}||\bar{w}_k||^2 + C\sum_{i=1}^{n}\xi_i \to \min \\ < \bar{w}_{k_i}, \bar{x}_i > - < \bar{w}_k, \bar{x}_i > \geq 1 - \delta k_i, y - \xi_i \end{cases} \tag{13}$$

где $k \in K$ - номера классов атрибута.

А двойственная задача квадратичного программирования находится как:

$$\begin{cases} \frac{1}{2}\sum_{k \in K}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{k_i}\lambda_{k_j} < \bar{x}_i, \bar{x}_j > + \sum_{k \in K}\sum_{i=1}^{n}(1 - \delta k_i, k)\lambda_{k_i} \to \min_{\lambda} \\ 0 \leq \lambda_{k_i} \leq C \end{cases} \tag{14}$$

В результате рещающая функция приобретает вид:

$$f(\bar{x}) = argmax_{k \in K}(< \bar{w}_k, \bar{x} > + w_{0_k}), \tag{15}$$

4 Методы квадратичного программирования

Рассматривается задача математического программирования [6, 1]:

$$f(x) \longrightarrow min, \quad F(x) = 0, \quad G(x) \leq 0, \tag{16}$$

где $f : R^n \longrightarrow R$ -гладкая функция, а $F : R^n \longleftrightarrow R^l$ и $G : R^n \longrightarrow R^m$ - гладкие отображения.

4.1 Метод внутренней точки (IP)

Метод внутренней точки - это метод позволяющий решать задачи выпуклой оптимизации с условиями, заданными в виде неравенств [8, 1].

Согласно методам внутренней точки, исходную для поиска точку можно выбирать только внутри допустимой области.

Выбор начальной точки поиска осуществляется в зависимости от формулировки задачи. При отсутствии ограничений или их преобразовании к функциям штрафа с внешней точкой начальная точка выбирается произвольно. При наличии ограничений или их преобразовании к функциям штрафа с внутренней точкой начальная точка выбирается внутри допустимой области.

При этом множество точек делится на допустимые и недопустимые в зависимости от ограничений. В свою очередь множество допустимых точек в зависимости от ограничений также делится на граничные и внутренние

4.1.1 Прямо-двойственный метод внутренней точки

Прямо-двойственный метод внутренней точки) оптимизирует прямые и двойственные переменные (x, λ, μ) путем решения линеаризованной возмущенной системы Куна-Таккера [8, 6]:

$$\begin{bmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \nabla^2 f_i(x) & \nabla f(x)^T & A^T \\ diag(\lambda) \nabla f(x) & diag(f(x)) & O \\ A & O & O \end{bmatrix} \begin{bmatrix} d_x \\ d_\lambda \\ d_\mu \end{bmatrix} = - \begin{bmatrix} \nabla f_0(x) + \nabla f(x)^T \lambda + A^T \mu \\ diag(\lambda) f(x) + \frac{1}{\tau}\epsilon \\ Ax - b \end{bmatrix} = \tag{17}$$

$$= - \begin{bmatrix} r_{dual}(x, \lambda, \mu) \\ r_{center}(x, \lambda, \mu) \\ r_{primal}(x, \lambda, \mu) \end{bmatrix} = -r(x, \lambda, \mu). \tag{18}$$

Здесь через $diag(\lambda)$ обозначена диагональная матрица, в которой на диагонали стоит вектор λ, через $f(x)$ – вектор $[f_1(x), ..., f_m(x)]^T$, через $\nabla f(x)$ – матрица производных, в которой в позиции (ij) стоит $\frac{\partial}{\partial x_i} f_i(x)$, а через e – вектор из единиц.

В данном случае величина $||r(x, \lambda, \mu)||$ отражает прогресс итерационного процесса, и на каждой итерации вдоль найденного направления (d_x, d_λ, d_μ) решается задача одномерной минимизации $||r||$. При ее решении с помощью стратегии backtracking значение α уменьшается до выполнения условия [8, 7]:

$$||r(x + \alpha d_x, \lambda + \alpha d_\lambda, \mu + \alpha d_\mu)|| < (1 - \alpha\rho)||(x, \lambda, \mu)||, \tag{19}$$

где ρ – параметр, задаваемый пользователем. Таким образом, получаем следующую общую схему прямодвойственного метода внутренней точки [8, 7]:

1. Выбирается строго допустимая точка x, положительный вектор λ и произвольный вектор μ. Также выбирается точность оптимизации ϵ, ϵ_{feas}, начальное значение $\tau = \tau_0$, мультипликатор $\upsilon > 1$ и параметры стратегии backtracking;

2. Находится решение (d_x, d_λ, d_μ) СЛАУ (14) для текущего значения τ;

3. Решается задача одномерной минимизации $||r(x + \alpha d_x, \lambda + \alpha d_\lambda, \mu + \alpha d_\mu)|| \longrightarrow min, \alpha \geq 0$ с помощью backtracking;

4. Если для нового набора (x, λ, μ) выполнено условие $||r_{dual}(x, \lambda, \mu) < \epsilon_{feas}||$, $||r_{primal}(x, \lambda, \mu) < \epsilon_{feas}||$ и $-\lambda^T f(x)$, то алгоритм заканчивает работу;

5. Значение τ устанавливается как $min(\frac{m}{\epsilon}, \tau(\frac{m}{-\lambda^T f(x)}))$, переход в шагу 2.

Для прямо-двойственного метода внутренней точки доказана теоретическая сходимость за $O(\sqrt{n}\log(\frac{1}{\epsilon}))$ [8, 8] итераций. Эта теоретическая оценка худшего числа итераций является наилучшей на сегодняшний день оценкой среди всех методов решения задач условной оптимизации. На практике метод внутренней точки сходится за константное число итераций, практически не зависящее от размерности задачи (обычно 30–40 итераций). Это обстоятельство делает метод особенно привлекательным для использования в случае данных большого объема. Метод внутренней точки основан на итерациях Ньютона, обладающих квадратичной скоростью сходимости. Поэтому метод внутренней точки позволяет находить решение задачи квадратичного программирования, в том числе, для высокой точности ϵ.

Этот момент является актуальным, например, для разреженных линейных моделей классификации/регрессии, в которых обнуление максимального количества компонент позволяет получать компактные и легко интерпретируемые решающие правила, предъявляющие минимальные требования к вычисляемому набору признаков, необходимому для принятия решений.

4.2 Последовательное квадратичное программирование (SQP)

Последовательное квадратичное программирование (англ. Sequential quadratic programming (SQP)) — один из наиболее распространённых и эффективных оптимизационных алгоритмов общего назначения, основной идеей которого является последовательное решение задач квадратичного программирования, аппроксимирующих данную задачу оптимизации [6, 2].

Для оптимизационных задач без ограничений алгоритм SQP преобразуется в метод Ньютона поиска точки, в которой градиент целевой функции обращается в ноль. Для решения исходной задачи с ограничениями-равенствами метод SQP преобразуется в специальную реализацию ньютоновских методов решения системы Лагранжа.

SQP методы генерируют траекторию $\left\{x^k\right\} \subset R^n$ следующим образом: по текущему приближению x^k очередное приближение x^{k+1} ищется как локальное решение (или как стационарная точка) задачи квадратичного программирования [6, 3]:

$$\left\langle f'(x^k), x - x^k \right\rangle + \frac{1}{2}\left\langle H_k(x - x^k), x - x^k \right\rangle \longrightarrow min, \tag{20}$$

$$F(x^k) + F(x^k)(x - x^k) = 0, G(x^k) + G'(x^k)(x - x^k) \leq 0, \tag{21}$$

где H_k - симметрическая $n \times n$-матрица, которая в некотором смысле аппроксимирует $\frac{\partial^2 L}{\partial x^2}(\bar{x}, \bar{\lambda}, \bar{\mu})$ при $k \longrightarrow \infty$
Например, можно полагать

$$H_k = \frac{\partial^2 L}{\partial x^2}(x, \lambda, \mu) \tag{22}$$

если параллельно с прямой траекторией x^k генерировать двойственную траекторию (λ^k, μ^k), например, следующим образом: по текущим λ^k, μ^k очередная пара $(\lambda^{k+1}, \mu^{k+1})$ определяется как пара множителей Лагранжа, отвечающих стационарной точке x^{k+1} задачи 20, 21.

4.2.1 Последовательное линейно-квадратичное программирование (SLQP)

Последовательное линейно-квадратичное программирование (SLQP) - это итерационный метод для задач нелинейной оптимизации, в котором целевая функция и ограничения дважды непрерывно дифференцируемы [6, 3]. Подобно последовательному квадратичному программированию (SQP), SLQP выполняется путем решения последовательности подзадач оптимизации.

Разница между двумя подходами заключается в том, что [6, 3]:

- в SQP каждая подзадача представляет собой квадратичную программу с квадратичной моделью цели, подлежащей линеаризации ограничений.

- в SLQP на каждом шаге решаются две подзадачи: линейная программа (LP), используемая для определения активного набора , за которой следует квадратичная программа с ограничениями равенства (EQP), используемая для вычисления общего шага

Эта декомпозиция делает SLQP подходящим для крупномасштабных задач оптимизации, для которых доступны эффективные решатели LP и EQP, причем эти задачи легче масштабировать, чем полноценные квадратичные программы.

Фаза LP [7, 161] В LP-фазе SLQP решается следующая линейная программа:

$$\begin{aligned} & min f(x_k) + \bigtriangledown f(x_k)^T d \\ s.t. \quad & b(x_k) + \bigtriangledown b(x_k)^T d \geq 0 \\ & c(x_k) + \bigtriangledown c(x_k)^T d = 0 \end{aligned}$$

Пусть $\mathcal{A}_k$ обозначает активный набор в оптимуме d^*_{LP} этой проблемы, то есть набор ограничений, которые равны нулю в d^*_{LP}. Обозначим через $b_{\mathcal{A}_k}$ и $c_{\mathcal{A}_k}$ подвекторы b и c, соответствующие элементам $\mathcal{A}_k$.

Фаза EQP [7, 162] На этапе EQP SLQP направление поиска d_k шага получается путем решения следующей квадратичной программы:

$$\begin{aligned} & min f(x_k) + \bigtriangledown f(x_k)^T dc + \frac{1}{2} d^T \bigtriangledown^2_{xx} \mathcal{L}(x_k, \lambda_k, \sigma_k) d \\ s.t. \quad & b_{\mathcal{A}_k}(x_k) + \bigtriangledown b_{\mathcal{A}_k}(x_k)^T d = 0 \\ & c_{\mathcal{A}_k}(x_k) + \bigtriangledown c_{\mathcal{A}_k}(x_k)^T d = 0 \end{aligned}$$

Стоит обратить внимание, что член $f(x_k)$ в приведенных выше целевых функциях может не учитываться для задач минимизации, поскольку он постоянен.

4.2.2 Условия локальной сходимости

Локальное поведение методов ньютоновского типа для задачи 16 обычно исследуют предполагая выполнение в искомом локальном решении этой задачи тех или иных условий регулярности ограничений и достаточных условий второго порядка [6, 4].

Важнейшим условием регулярности ограничений является **условие Мангасариана-Фромовица MFCQ (от английского Mangasarian-Fromovitz constraint qualification)** [6, 4]:

$$rank F(\bar{x}) = l, \exists \bar{\xi} \in ker F'(\bar{x}) : G'_{l(\bar{x})}(\bar{x}) \bar{\xi} \leq 0$$

Выполнение MFCQ в стационарной точке $\bar{x}$ задачи 16 равносильно ограниченности полиэдра. Единственности множителей Лагранжа это условие, вообще говоря, не гарантирует. Комбинация MFCQ и требование единственности отвечающих $\bar{x}$ множителей Лагранжа $\bar{\lambda}$ и $\bar{\mu}$ называется строгим условием регулярности Мангасариана-Фромовица SMFCQ (от английского Strict Mangasarian-Fromovitz constraint qualification). Это условие можно записать в виде [6, 4]:

$$\begin{pmatrix} F'(x) \\ G'_{l_+(\bar{x},\bar{\mu})}(\bar{x}) \end{pmatrix} = l + |I_+(\bar{x},\bar{\mu})|, \exists\xi \in kerF'(\bar{x}) : G'_{l_+(\bar{x},\bar{\mu})}(\bar{x})\bar{xi} = 0, G'_{0+(\bar{x},\bar{\mu})}(\bar{x})\bar{xi} < 0, \quad (23)$$

где $I_+(\bar{x},\bar{xi}) = i \in I(\bar{x})|\bar{\mu}_i > 0$, $I_0(\bar{x},\bar{xi}) = I(\bar{x})\ I_+(\bar{x},\bar{xi})$

Для $(\bar{\lambda},\bar{\mu})$ достаточное условие второго порядка SOSC (от английского Second-order sufficient condition) имеет вид [6, 5]:

$$\frac{\partial^2 L}{\partial x^2}(\bar{x},\bar{\lambda},\bar{\mu}\xi,\xi) > 0 \forall\xi \in C(\bar{x}) \quad (24)$$

где $C(\bar{x}) = \xi \in F'(\bar{x})|G'_{I(\bar{x})}(\bar{x})\xi \leq 0), < f'(\bar{x}), \xi \leq 0 >$ есть критический конус задачи 16 в точке $\bar{x}$.

Локальная сходимость методов SQP и SQLP со сверхлинейной скоростью может быть доказана при выполнении SMFCQ и SOSC. Этот результат был получен в [7]. При этом предполагается, что в качестве очередного прямодвойственного приближения $(x^{k+1}, \lambda^{k+1}, \mu^{k+1})$ берется ближайшее (или, во всяком случае, достаточно близкое) к (x^k, λ^k, μ^k) решение системы ККТ задачи 20, 21.

4.3 Последовательная минимальная Оптимизация (SMO)

Sequential minimal optimization (SMO) - алгоритм для решения задачи квадратичного программирования (QP). SMO широко используется для обучения опорных векторных машин. Публикация алгоритма SMO в 1998 году вызвала большой ажиотаж в сообществе SVM, поскольку ранее доступные методы обучения SVM были намного сложнее и требовали дорогостоящих сторонних решателей QP.

SMO - итерационный алгоритм для решения задачи оптимизации [14]:

$$\begin{aligned} \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j K(x_i,x_j)\alpha_i\alpha_j, \\ \text{при условии:} \qquad 0 \leq \alpha_i \leq C, \quad \text{для} \quad i = 1,2,\ldots,n, \\ \sum_{i=1}^{n} y_i\alpha_i = 0 \end{aligned}$$

где C - гиперпараметр SVM, $K(x_i, x_j)$ - функция ядра, а переменные α_i являются множителями Лагранжа.

SMO разбивает проблему (4.3) на серию минимально возможных подзадач, которые затем решаются аналитически. Из-за ограничения линейного равенства, включающего множители Лагранжа α_i, наименьшая возможная проблема включает два таких множителя. Тогда для любых двух множителей α_1 и α_2 , ограничения сокращаются до [14]:

$$0 \leq \alpha_1, \alpha_2 \leq C,$$

$$y_1\alpha_1 + y_2\alpha_2 = k,$$

и эта сокращенная задача может быть решена аналитически: нужно найти минимум одномерной квадратичной функции. k - отрицательное значение суммы остальных членов в ограничении равенства, которое фиксируется на каждой итерации.

Алгоритм работает следующим образом [14]:

- Найти множитель Лагранжа α_1, который нарушает Условия Каруша — Куна — Таккера (ККТ) для задачи оптимизации.
- Выбрать второй множитель α_2 и оптимизировать пару (α_1, α_2).
- Повторить шаги 1 и 2 до сходимости.

Когда все множители Лагранжа удовлетворяют условиям ККТ (в пределах заданного пользователем допуска) проблема решена. Хотя этот алгоритм гарантирует сходимость, эвристика используется для выбора пары множителей, чтобы ускорить скорость сходимости. Это очень важно для больших наборов данных, поскольку существует $n(n-1)/2$ возможных вариантов для α_i и α_j.

5 Методы формирования вектора признаков изображения лица для определении атрибутов личности

Различают две группы методов Ф формирования вектора признаков X' [1, 34]:

- группа методов Φ_t, основанная на использовании текстуры (значений интенсивности пикселей) изображения
- группа методов Φ_a, основанная на выделении антропометрических точек на изображении лица человека с последующим выделением информации о расстоянии между этими точками и их взаимном расположении

5.1 Группа методов Φ_t

К это группе относятся методы, в которых используются либо непосредственно значения интенсивности пикселей нормализованных изображений, либо результаты применения некоторых операторов к значениям интенсивностей. При этом прямое использование пикселей изображения подходит только для изображений малой размерности и на практике используется для выделения лиц на общем изображении.

Вектор признаков в таком случае представляет собой гистограмму значений интенсивносетй или их модификаций.

5.1.1 Метод локальных бинарных шаблонов (LBP)

Локальный бинарный шаблон - определенный вид признака, представляющий собой описание окрестности пикселя изображения в двоичном представлении.

Базовый оператор LBP, применяемый к пикселю изображения, использует восемь пикселей окрестности и принимает значение интенсивности центрального пикселя в качестве порога. Пиксели, имеющие значение, большее или равное указанному, принимают значение "1 остальные - "0". Таким образом, результатом применения базового оператора LBP к пикселю изображения является восьмиразрядный бинарный код, описывающий окрестность пикселя по часовой стрелке, начиная с верхнего левого пикселя окрестности. Далее этот код рассматривается, как двоичная запись некоторого числа, сопоставленная данному пикселю.

После получения указанных значений строится их гистограмма, то есть формируется набор $\Phi_{LBP}(i) = \bar{\mathbf{x}}_i = (\bar{0}_i, \bar{1}_i, ..., \bar{x}_i, ..., \bar{255}_i)$ частот $\bar{x}_i$ появления бинарных шаблонов, имеющих десятичные числовые эквиваленты $x, x \in \{0, 1, ..., 255\}$ в изображении $i : \bar{x}_i \in \mathbb{R}, 0 \leq \bar{x}_i < 1$. При этом коллекция I' из m элементов обучающей и k элементов тестовой выборок представляются матрицами размером $m \times 256$ и $k \times 256$:

$$\Phi_{LBP}(I^l) = X^l = \begin{pmatrix} \bar{0}_1 & \bar{1}_1 & ... & \bar{x}_1 & ... & \bar{255}_1 \\ ... & ... & ... & ... & ... & ... \\ \bar{0}_i & \bar{1}_i & ... & \bar{x}_i & ... & \bar{255}_i \\ ... & ... & ... & ... & ... & ... \\ \bar{0}_m & \bar{1}_m & ... & \bar{x}_m & ... & \bar{255}_m \end{pmatrix} \tag{25}$$

$$\Phi_{LBP}(I^t) = X^t = \begin{pmatrix} \bar{0}_1 & \bar{1}_1 & ... & \bar{x}_1 & ... & \bar{255}_1 \\ ... & ... & ... & ... & ... & ... \\ \bar{0}_i & \bar{1}_i & ... & \bar{x}_i & ... & \bar{255}_i \\ ... & ... & ... & ... & ... & ... \\ \bar{0}_k & \bar{1}_k & ... & \bar{x}_k & ... & \bar{255}_k \end{pmatrix} \tag{26}$$

Соответствующими векторами заменяются элементы i в описаниях прецедентов (i, y_i) в обучающей $\tilde{G}^l$ и тестовой $\tilde{G}^t$ выборках.

Локальные бинарные шаблоны характеризуют локальные особенности изображения, а частоты этих особенностей можно рассматривать как обобщенную модель лица. При этом шаблоны не характеризуют расположение данных особенностей на изображении. Для этих целей нормализованное изображение разбивается на s регионов, для каждого из которых вычисляется своя гистограмма $\Phi_{LBP}(i_j)$. Тогда результирующим описанием изображения i является вектор значений $\bar{x}_i$, сформированный как конкатенация s гистограмм, полученных по s регионам исходного изображения:

$$\Phi_{LBP}(i) = \bar{\mathbf{x}}_i = (\bar{0}_{i_1}, \bar{1}_{i_1}, ..., \bar{x}_{i_1}, ..., \bar{255}_{i_1}, ..., \bar{0}_{i_j}, \bar{1}_{i_j}, ..., \bar{x}_{i_j}, ..., \bar{255}_{i_j}, ..., \bar{0}_{i_k}, \bar{1}_{i_k}, ..., \bar{x}_{i_s}, ..., \bar{255}_{i_s}) \tag{27}$$

Данное описание уже не является гистограммой, так как сумма значений его элементов равна s.

Исходя из соотношений, характерных для элементов лица человека, нормализованное изображение традиционно разбивается на $k = 6 \times 7 = 42$ региона. При более крупном разбиении теряется информация о локальном расположении особенностей, а при более мелком - увеличивается размерность вектора признаков, что усложняет процесс классификации.

Не все шаблоны обладают одинаковой информативностью. Поэтому выделяют *равномерные шаблоны* - бинарные комбинации, содержащие не более трех серий "0"и "1". Равномерные LBP определяют только важные локальные особенности изображения: концы линий, грани, углы и пятна. Использование только равномерных шаблонов приводит к существенному сокращению размерности вектора признаков, в этом случае используется только $p(p-1)+2$ шаблона [1, 34-39].

5.1.2 Метод построения гистограммы направленных градиентов (HOG) и гистограмм направления края изображения (EOH)

Метод построения гистограммы направленных градиентов (HOG) основан на вычислении градиента изменения интенсивности для каждого пикселя изображения и формировании гистограммы данных градиентов для различных участков изображения. При этом выделяются восемь базовых направлений: "север "северо-восток"и т.д. В итоге формируется пространство признаков, близкое по своим характеристикам к методу LBP [1, 39].

Данный метод имеет свой аналог в виде метода формирования гистограмм направления края изображения EOH. Эти методы нашли основное применение при решении задач выделения заданного объекта на изображении, а не при определении атрибутов личности.

5.2 Группа методов Φ_a

Методы данной группы основаны на воспроизведении процесса описания лица, характерном для человека: фиксируются особые точки и их расположение на лице (границы глаз, носа, рта, бровей, подбородка и т.д.)

5.2.1 Метод активной модели формы (ASM)

Главной предпосылкой для построения ASM является наблюдение, что между расположением антропометрических точек есть зависимости. Для моделирования этих зависимостей по обучающей выборке $\tilde{G}^l$ строится статическая модель положения важных антропометрических точек на изображении лица. При этом в качестве признакового описания изображения используются координаты выделенных точек на нормализованных изображениях, то есть формируется набор $\Phi_{ASM}(i) = \bar{\mathbf{x}}_i = (x_1, x_2, ..., x_u, y_1, y_2, ..., y_u)^T$, а вся совокупность из n прецедентов обучающей выборки $\tilde{G}^l$ образует матрицу $\Phi_{ASM}(\tilde{G}^l) = X = (\bar{\mathbf{x}}_1, ..., \bar{\mathbf{x}}_i, ..., \bar{\mathbf{x}}_n)$.

После выделения главных компонент указанной матрицы получаем выражение для синтезированной формы:

$$\begin{gathered} X = \bar{X} + P_{sh}B_{sh} \\ B_{sh} = P_{sh}^T(X - \bar{X}) \end{gathered} \tag{28}$$

где $\bar{X}$ - форма изображения, усредненная по всем реализациям обучающей выборки (базовая форма), P_{sh} - матрица собственных векторов, B_{sh} - вектор параметров формы.

Приведенное выражение означает, что форма любого изображения X может быть выражена как сумма базовой формы $\bar{X}$ и линейной комбинации собственных форм, содержащихся в матрице P_{sh}. Меняя значение B_{sh} можно синтезировать различные формы изображений [1, 40-43].

5.2.2 Метод активной модели внешности (AAM)

Данная модель является развитием активной модели формы (ASM), путем добавления в нее модели текстуры.

На первом этапе происходит построение активной модели формы (ASM) путем разметки нормализованного изображения лица - выделения на нем некоторых пронумерованных характерных антропометрических точек. Качественное описание изображения достигается при наличии $u = 60 - 70$ таких точек.

Автоматическое определение важных антропометрических точек основано на выделении характерных особенностей изображения: Т-образные пересечения краев, углы, легко различимые биологические особенности, границы изображение и пр.

На втором этапе построения модели AAM учитывается информация о текстуре (значениях интенсивности пикселей) изображения. Для этого изображение разбивается на непересекающиеся треугольники, вершинами которых являются выделенные антропометрические точки активной модели формы. Далее для каждого треугольника вычисляется среднее значение интенсивности, то есть изображение описывается вектором:

$$\Phi_{AAM}(i) = \bar{\mathbf{z}} = (z_1, z_2, ..., z_v), \tag{29}$$

где v - количество треугольников в изображении. А вся совокупность из n прецедентов обучающей выборки $\tilde{G}^l$ образует матрицу:

$$\Phi_{AAM}(\tilde{G}^l) = Z = (\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2, ..., \bar{\mathbf{z}}_i, ..., \bar{\mathbf{z}}_n) \tag{30}$$

Аналогично построению активной модели формы по методу главных компонент строится приближенная модель текстуры изображения:

$$\begin{array}{c} Z = \bar{Z} + P_t B_t \\ B_t = P_t^T (Z - \bar{Z}) \end{array} \tag{31}$$

Как и в модели формы, текстура любого изображения Z может быть синтезирована суммой базовой текстуры $\bar{Z}$ и линейной комбинации собственных форм, содержащихся в матрице P_t. Меняя значение B_t можно синтезировать различные текстуры изображений.

Общее признаковое описание изображения i по методу AAM представляет собой конкатенацию векторов $\bar{x}_i$ и $\bar{z}_i$.

Метод AAM ориентирован на решение другой задачи - идентификации личности по изображению лица, так как в этом случае разметку эталонного изображения осуществляет человек, выделяя наиболее важные антропометрические точки. Построенная "вручную"модель внешности затем используется для поиска требуемой личности на других изображениях или определения выражения эмоций [1, 40-45].

5.3 Применимость методов формирования векторов признаков

Постановка задачи для решения методом опорных векторов удовлетворяет условиям теорем 1, 2 о разрешимости задачи квадратичного программирования.

Использование для формирования вектора признаков изображения методами группы Φ_t строят положительно определенную матрицу. Данную матрицу можно считать после небольших преобразований симметричной, так как лицо имеет достаточно симметричное строение.

При анализе применимости методов квадратичного программирования важным условием является гладкость функции цели. Данное свойство выполняется, что очевидно из постановки задачи метода опорных векторов. Дополнительно метод последовательного квадратичного программирования накладывает условие дважды дифференцируемой функции цели, что так же выполняется. А для метода последовательной минимальной оптимизации необходимо, чтобы накладываемые ограничения были линейными. Таким образом, указанные методы квадратичного программирования применимы к задачам распознавания, поставленных в указанных условиях и имеющих вектор признаков изображения, сформированный методами, основанными на использовании значений интенсивностей пикселей.

Методы из группы Φ_a не гарантируют корректное применение методов квадратичного программирования, так как при формировании вектора признаков строится матрица, не являющаяся симметричной и положительно определенной. А если приводить указанную матрицу в требуемый вид дополнительными преобразованиями, то теряется часть информации о взаимном расположении антропометрических точек.

6 Практическое задание

6.1 Описание практического задания

В качестве практического задания рассмотрим распознавание изображений, содержащих цифры. При решении данной задачи будут использоваться различные классификаторы SVM, каждый из которых заключает в себе отличный от других метод решения задачи квадратичного программирования при построении оптимальной гиперплоскости. Таким образом, решая поставленную задачу с помощью классификаторов с разными методами, можно приблизительно сравнить их эффективность.

Создание разных SVM классификаторов с нуля - достаточно трудоёмкая задача, не рассматриваемая в данной работе. Параметры классификаторов были подобраны так, чтобы обеспечить их сравниваемость, однако это не может в полной мере уберечь от того, что разные классификаторы могут показывать разное поведение, которое обусловлено не только использованием различных методов квадратичного программирования. Для полной чистоты эксперимента необходимо с нуля написать классификаторы SVM и реализации методов квадратичного программирования, а так же составить собственный набор размеченных данных, что выходит за рамки данной работы.

В поставленной задаче использовались следующие классификаторы:

1. *Классификатор SVC из пакета svm python-библиотеки sklearn* [9]

    - Данный классификатор использует наиболее распространённый в программных реализациях SVM метод решения задачи квадратичного программирования - последовательную минимальную оптимизацию (Sequential Minimal Optimization - SMO)
    - Программная реализация SMO берётся из известной си-библиотеки libsvm, что положительно влияет на скорость работы классификатора
    - Классификатор является высокомодифицируемым и крайне распространенным как реализация SVM на python, что говорит о его высоком качестве и возможном превосходстве над менее популярными классификаторами

2. *Классификатор SupportVectorMachine из библиотеки Machine Learning From Scratch* [10]

    - Данный классификатор использует метод внутренней точки (Interior Point - IP), реализация которого берётся из python-библиотеки CVXOPT
    - Библиотека CVXOPT является одной из самых популярных библиотек оптимизации для python. Классификатор при этом является, скорее, учебным примером, чем быть реально используемым в сложных научных задачах, и может показывать результаты хуже указанных классификаторов

3. *Классификатор MaxMarginClassifier из статьи журнала Towards Data Science* [11]

    - Данный классификатор использует метод последовательного линейно-квадратичного программирования (Sequential Linear-Quadratic Programming - SLQP)
    - Программная реализация SLQP берётся из python-библиотеки scipy, при этом наблюдаются недостатки, аналогичные использованию предыдущего классификатора

В качестве тестовых данных используется размеченный набор digits из пакета sklearn.datasets, который содержит 10 классов (цифры от 0 до 9 включительно) примерно по 180 образцов на каждый класс. Суммарно получается набор данных из порядка 1800 образцов [12].

Помимо этого, активно используются утилиты из разных пакетов библиотеки sklearn, например, для сбора данных и подсчёта различных метрик. Также в каждом из рассматриваемых классификаторов вектор признаков формируется посредством метода локальных бинарных шаблонов LBP.

В качестве метрики качества предсказаний используется accuracy score из пакета sklearn.metrics, который считает отношение числа верных предсказаний к общему числу всех предсказаний [13].

Поскольку SVM в первую очередь является методом бинарной классификации, а набор данных содержит в себе 10 размеченных классов, то в качестве дополнительного эксперимента проводится переразметка данных под 2 класса и распознавание по ним. Создаются два класса цифр: $x \leq 4$ и $x \geq 5$, и все существующие метки классов $[0, 4]$ превращаются в метку класса 0, а метки классов $[5, 9]$ в метку класса 1. Изображения в наборе данных при этом не изменяются. Обучение и предсказание происходит на основе уже переразмеченных классов.

Эксперимент проводится последовательно по всем трём методам (SMO, IP, SLQP) с разными наборами меток классов (10 классов или 2 класса). Для надёжности каждый эксперимент повторяется 5 раз. Таким образом, получается $3 * 2 * 5 = 30$ независимых экспериментов.

По итогам практического эксперимента, составляется таблица результатов. Дополнительно по значениям из таблицы считаются среднее, медианное и максимальное значения accuracy score для каждого метода по каждому из наборов меток.

6.2 Результаты практического задания

Ниже представлены таблицы с результатами:

	10 classes	2 classes
1	0.988889	0.995556
2	0.993333	0.995556
3	0.991111	0.991111
4	0.993333	1
5	0.986667	0.991111
Mean	0.990667	0.994667
Median	0.991111	0.995556
Max	0.993333	1

Таблица 1: Метод SMO

	10 classes	2 classes
1	0.0644444	0.488889
2	0.0866667	0.504444
3	0.0866667	0.451111
4	0.108889	0.48
5	0.1	0.506667
Mean	0.0893333	0.486222
Median	0.0866667	0.488889
Max	0.108889	0.506667

Таблица 2: Метод IP

	10 classes	2 classes
1	0.111111	0.473333
2	0.0911111	0.542222
3	0	0.513333
4	0.0911111	0.522222
5	0.122222	0.482222
Mean	0.0831111	0.506667
Median	0.0911111	0.513333
Max	0.122222	0.542222

Таблица 3: Метод SLQP

По результатам эксперимента видно, что лучшее качество на рассмотренной задаче показывает метод SMO с точностью, близкой к 1, а классификаторы на основе методов IP и SLQP показывают качество заметно ниже. Сравнивая методы IP и SLQP, классификатор на основе последнего демонстрирует точность несколько лучше по всем параметрам нежели первый.

Такое сильное различие в точности может быть обосновано тем, что классификатор на основе SMO является программным продуктом высокого качества, используемым в серьёзных научных задачах, а классификаторы, основанные на методах IP и SLQP являются учебными примерами и не предназначены для настоящих задач. Помимо этого, выбранный набор данных содержит изображения очень низкого качества - 8x8 пикселей, а также изображения часто зашумлены - имеют много бесполезной информации вокруг объекта распознавания. Всё это в совокупности может иметь сильное негативное влияние на качество распознавания менее профессиональными классификаторами.

7 Сравнительный анализ

7.1 Метод опорных векторов (SVM)

Преимущества SVM:

- SVM имеет свойство *разреженности*, то есть можно исключить из рассмотрения нулевые λ_i и построить компактный классификатор (рещающую функцию) [1, 60]
- Метод имеет модификацию преобразования множественной классификации в последовательность бинарных классификаций
- SVM позволяет работать с линейно неразделимыми обучающими выборками
- Данный метод показывает один из наилучших на данный момент результатов по точности классификации в сочетании с формированием вектора признаков изображения на основе метода LBP [1, 78]

Недостатком SVM является его ориентированность на классификацию по двум классам, которую обобщить на несколько классов достаточно проблематично.

7.2 Методы формирования вектора признаков изображения

7.2.1 Метод локальных бинарных шаблонов (LBP)

Преимущества LBP:

- Метод наиболее информативен с точки зрения формирования вектора признаков по сравнению с другими методами признакового описания
- Локальные бинарные шаблоны инварианты к небольшим изменениям в условиях освещенности и небольшим поворотам классифицируемого изображения, что обуславливает их широкое распространение для решения задач определения таких атрибутов личности, как "пол "раса"и "возрастная группа"[1, 37]
- Важным достоинством меттода LBP является простота реализации LBP, что позволяет использовать его в задачах обработки изображений в реальном времени
- LBP позволяет сформировать пространство признаков большой размерности (порядка нескольких тысяч), обеспечивая высокую концентрацию информации об исходном изображении, и создает предпосылки для более точной классификации по возрасту
- Метод дает возможность использовать только те шаблоны, которые хранят больше информации о локальных особенностях изображения [1, 11]

Недостатки LBP:

- При небольшом количестве разбиений изображения или его отсутствии теряется информация об расположении локальных особенностей изображения
- При формировании вектора признаков изображения получается пространство большой размерности (более 2000), поэтому необходимо вводить модификации для снижения данной размерности и учитывать симметричность лица и различную информативность отдельных участков изображения лица [1, 46]
- Особое значение в рассмотрении метода равномерных шаблонов выявлено чисто эмпирически, поэтому существует необходимость описать более формально данный подход и сформировать унифицированное функциональное представление изображений

7.2.2 Метод построения гистограммы направленных градиентов (HOG)

Данные методы похожи на метод локальных бинарных шаблонов, поэтому они наследуют часть преимуществ и недостатков последнего.

Недостатки:

- Размерность пространства признаков при использовании метода HOG плохо поддается сокращению, а процедура вычисления градиента интенсивности заметно сложнее процедуры формирования LBP [1, 39]

- Методы более пригодны для решения задач выделения заданного объекта на изображении, а не для определении атрибутов личности

7.2.3 Метод построения активной модели формы (ASM) и активной модели внешности (AAM)

Преимущества:

- Методы позволяют получать хорошие результаты при определении таких атрибутов личности, как "пол"и "раса так как именно взаимное расположение важных антропометрических точек совместно с текстурой изображения позволяют определить принадлежность личности к определенному полу и расе [1, 44]

Недостатки [1, 45]:

- Данный подход формирования вектора по признаку "возраст"может использоваться только для определения возрастной группы, а не для вычисления прогнозируемого возраста, так как расположение важный антропометрических точек и расстояние между ними практически не меняются при изменениях в возрасте на несколько лет
- Проблемой данного метода является сложность автоматического выделения на изображении характерных антропометрических точек, так как для их выделения изображение анализируется на уровне текстуры (интенсивности пикселей) ввиду отсутствия другой информации об изображении

7.3 Методы квадратичного программирования

7.3.1 Метод внутренней точки

Преимущества [8, 7]:

- Теоретическая сходимость за $O(\sqrt{n}\log(\frac{1}{\epsilon}))$ итераций. Эта теоретическая оценка худшего числа итераций является наилучшей на сегодняшний день оценкой среди всех методов решения задач условной оптимизации.
- На практике метод внутренней точки сходится за константное число итераций, практически не зависящее от размерности задачи.
- Отлично подходит для использования в случае данных большого объема.
- Позволяет находить решение задачи квадратичного программирования, в том числе, для высокой точности ϵ.

Недостатки [8, 7]:

- Методы внутренней точки требуют нескольких итераций, но итерации дороги: каждая по сути является шагом Ньютона, и каждая должна выполняться с нуля.
- Теряет свои преимущества на небольших задачах

7.3.2 Последовательное квадратичное программирование

Преимущества [6, 2]:

- Не требует предварительного решения линейной задачи.
- Учитывается погрешность, возникшая на предыдущих итерациях
- Требует меньше времени по сравнению с другимы методами квадратичного программирования

Недостатки [6, 3]:

- Не гарантирует, что решение удовлетворяет вашим ограничениям в конце каждой итерации. Он только гарантирует, что оптимальный план удовлетворяет вашим ограничения. Это означает, что если SQP когда-либо не удастся найти оптимальный план, возможно, не будет доступных промежуточных решений, которые будут лучше первоначального приближение.
- Дает преимущество, только когда значительное количество переменных не меняются в ходе итерации

7.3.3 Последовательная минимальна оптимизация(SMO)

Преимущества [14]:

- Простой алгоритм, который быстро решает проблемы SVM QP без дополнительного хранения матрицы и без привлечения итерационного численного порядка подготовки для каждой позадачи
- Решение для двух множителей Лагранжа может быть выполнено аналитически
- Высокий уровень точности

Недостатком можно назвать, что хотя этот алгоритм гарантирует сходимость, но ускорить сходимость можно только эврестически подбирая параметры.

8 Краткий вывод

Использование метода опорных векторов является существенно менее трудоемким и дает большую точность классификации при определении атрибута "пол"на неподготовленных изображениях. Наиболее перспективным методом формирования вектора признаков является метод локальных бинарных шаблонов (LBP) как для задачи определения атрибутов "пол"и "раса так и для атрибута "возраст". Их сочетание дает точность для атрибута "пол"порядка 87%, для атрибута "раса около 81.3%, а при прогнозировании возраста личности можно достичь точности в 53% [1, 124-128].

Существующие на сегодня методы классификации изображений лиц по атрибуту "пол"в большинстве случаев позволяют правильно классифицировать примерно две трети реальных изображений, что дает предпосылку к созданию модификаций с целью повышения качества классификации.

Касательно методов квадратичного программирования, метод последовательной минимальной оптимизации при использовании машины опорных векторов лучше всего подходит для решения указанной в практической части задачи распознавания. На втором месте по точности среди рассматриваемых методов стоит метод последовательного квадратичного программирования наименьших квадрат, а наименее точным методом оказался метод внутренних точек.

9 Источники

1. Рыбинцев А.В. Исследование, модификация и разработка методов компьютерного зрения для задач определения атрибутов личности по изображению лица: диссертация на соискание ученой степени кандидата технических наук. - М.: НИУ МЭИ, 2018.

2. Лепский А.Е., Броневич А.Г. Математические методы распознавания образов: курс лекций. - Таганрог: Технологический институт Южного федерального университета, 2009.

3. Даугавет В.А. Численные методы квадратичного программирования: учебное пособие. - С.-П.: СПБГУ, 2004.

4. Воронцов К.В. Лекции по методу опорных векторов. - М.: ВЦ РАН, 2007.

5. Николенко С.И. Метод опорных векторов. SVM и задача линейной классификации: - С.-П.: TRA Robotics, 2018.

6. Голишников М.М., Измаилов А.Ф. Ньютоновские методы для задач условной оптимизации с нерегулярными ограничениями //Вычислительная математика и математическая физика. - М.: МГУ, ф-т ВМиК. - 2006. - том 46. - № 8. - с. 1369–1391.

7. Bonnans J.F. Local analysis of Newton-type methods for variational inequalities and nonlinear programming // Appl. Math. Optimizat. - 1994. - V. 29. - P. 161-186.

8. Воронцов К.В. Машинное обучение: курс лекций [Электронный ресурс]: Методы внутренней точки.
- URL: http://www.machinelearning.ru/wiki/images/8/81/MOMO12_ipm.pdf (дата обращения: 12.05.2021).

9. Scikit-learn [Электронный ресурс]: C-Support Vector Classification.
- URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (дата обращения: 12.05.2021).

10. Erik Linder-Norén. github [Электронный ресурс]: ML-From-Scratch
- URL: https://Github.com/eriklindernoren/ML-From-Scratch (дата обращения: 12.05.2021).

11. Antoine Hue. Towards data science [Электронный ресурс]: Kernel Support Vector Machines from scratch.
- URL: https://towardsdatascience.com/support-vector-machines-learning-data-scien
(дата обращения: 12.05.2021).

12. Scikit-learn [Электронный ресурс]: Load and return the digits dataset (classification).
- URL: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html (дата обращения: 12.05.2021).

13. Scikit-learn [Электронный ресурс]: Accuracy classification score.
- URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

14. John C. Platt. Быстрая подготовка метода опорных векторов последовательной минимальной оптимизации / Перевод Богдан Е.Ю. [Электронный ресурс]
- URL: http://masters.donntu.org/2013/fknt/bogdan/library/perevod.htm (дата обращения: 12.05.2021).