Дискриминантный анализ Фишера с kernel trick

Каплоухая Нина

Аягоз Мусабаева. "Fisher Discriminant analysis with Kernels" by S. Mika

Reading group ИППИ

Содержание I

1. DA. Вступление
2. FDA
   - Принцип работы FDA
   - Алгоритм FDA
   - Пример работы алгоритма
   - Основной код для реализации FDA
   - Более сложный пример
3. Другие методы DA. Общие принципы
   - Сравнение FDA с другими линейными классификаторами
4. Ядро. Kernel Trick
   - FDA не справляется. Пример
   - Переход в пространство большей размерности
   - Понятие kernel trick
   - Типы ядер
5. KFDA
   - Переход от FDA к KFDA
   - Алгоритм KFDA
   - Пример работы алгоритма

Содержание II

- Сравнение с линейными классификаторами
- Сравнение с другими реализациями KFDA и KSVM

6. Параметры ядра

7. KFDA для многоклассовой задачи
   - Алгоритм многоклассового KFDA
   - Пример работы алгоритма

8. Общая сравнительная таблица

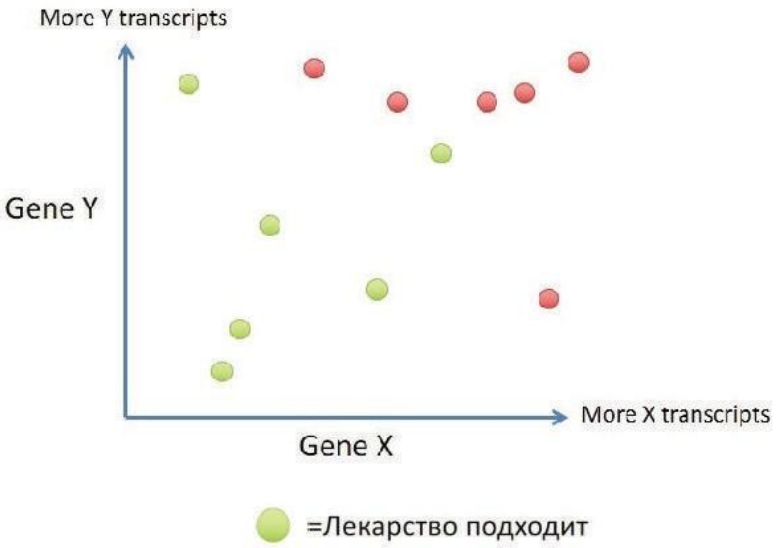
9. Применение

10. Приложения

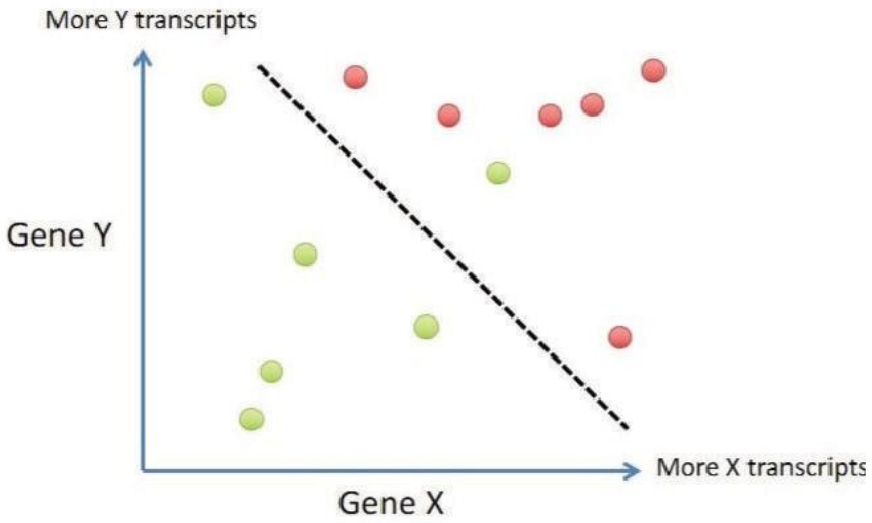
11. Список литературы

Дискриминантный анализ. Пример

Дано: некоторое кол-во данных о том, подошло ли людям лекарство, и количество генов X и Y в ДНК этих людей.
Наблюдение: Присутствует зависмость совместимости с лекарством от количества данных генов.
Задача: Для всевозможных наборов генов X и Y предсказать, подойдет ли лекарство для человека с данным набором в ДНК.



Данные:



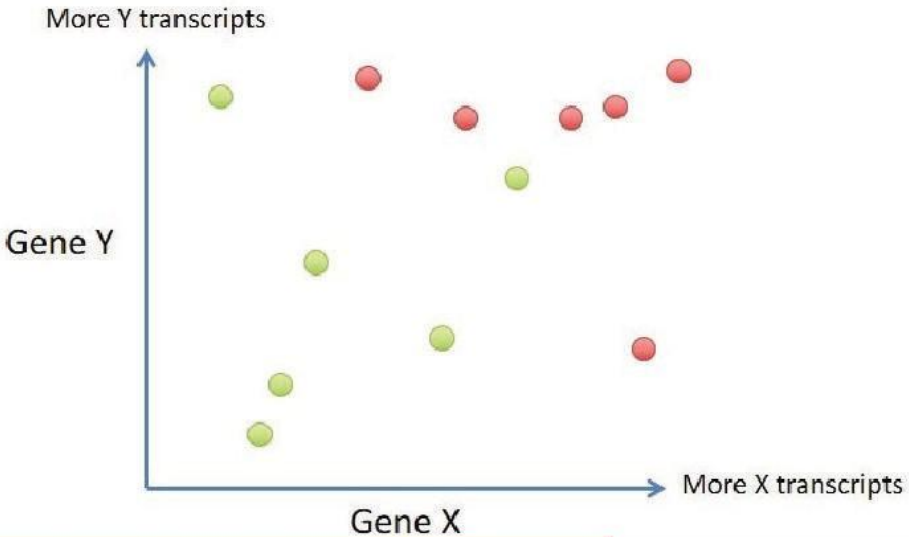
Decision Boundary

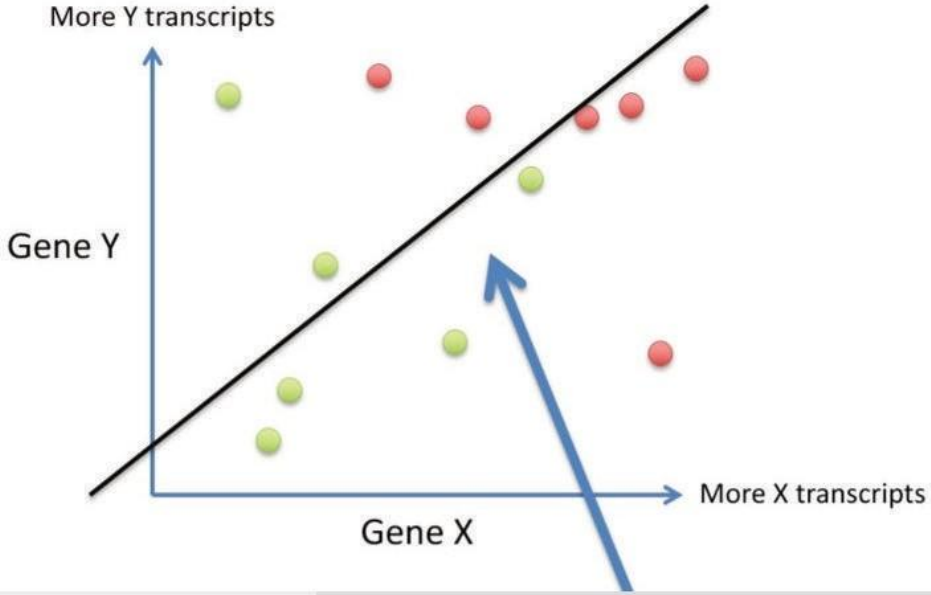
Принцип работы FDA

Линейный дискриминантный анализ и линейный дискриминант Фишера(LDA и FDA)

Метод статистики и машинного обучения, применяемый для нахождения линейных комбинаций признаков, наилучшим образом разделяющих два или более класса объектов или событий. Полученная комбинация может быть использована в качестве линейного классификатора или для сокращения размерности пространства признаков перед последующей классификацией.

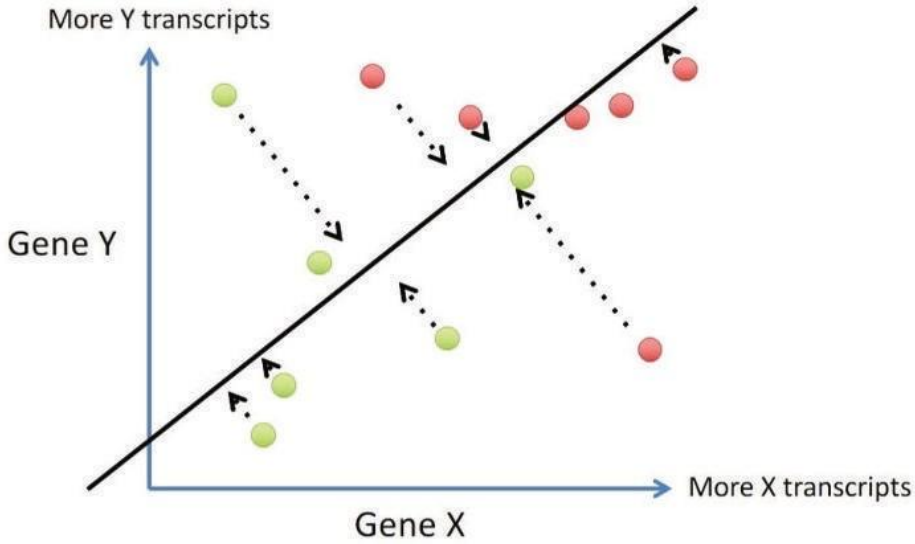
Проведем вектор, соотвествующей данной линейной комбинации:

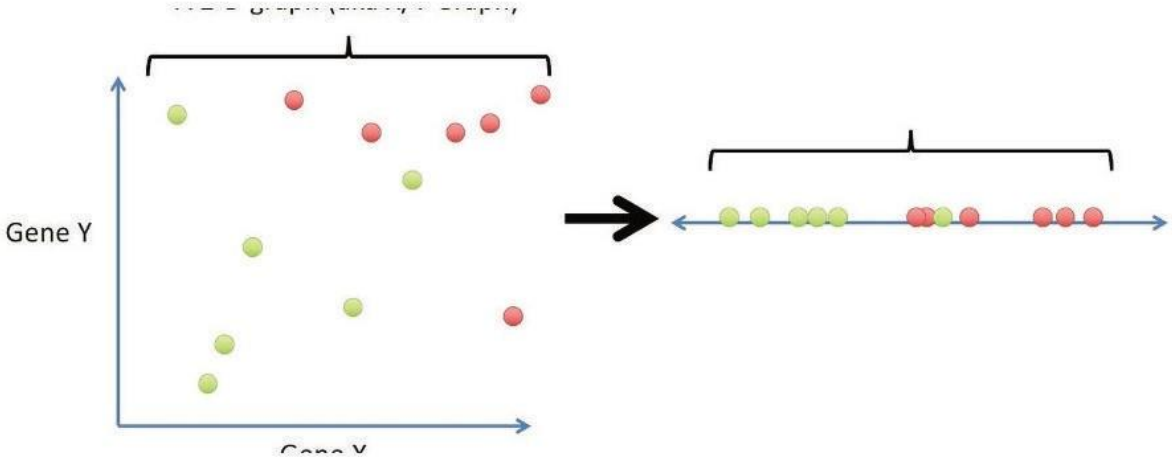




Принцип работы FDA

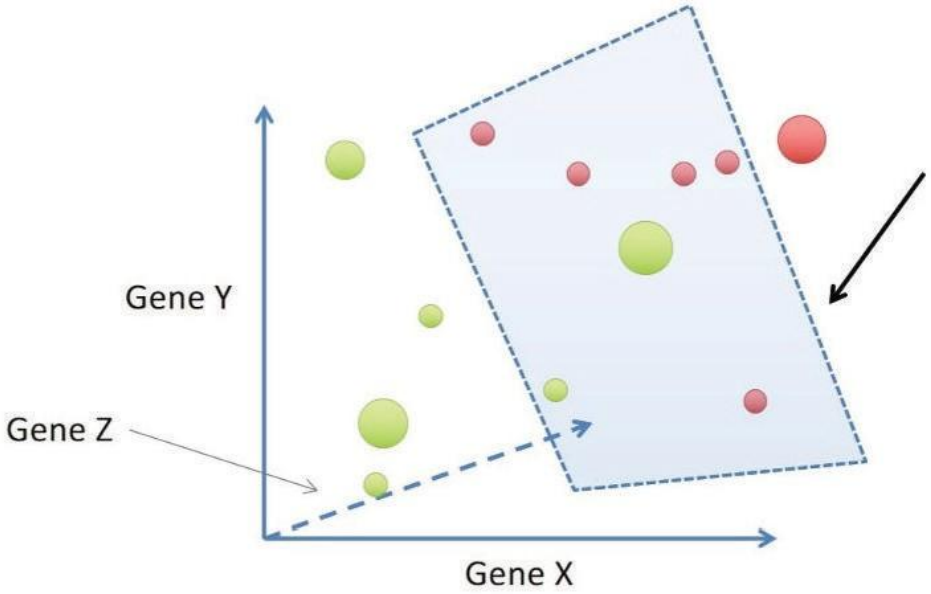
Спроектируем данные на этот вектор:





Переход из 2D в 1D

Если предикторов много, то снижение размерности особенно актульно:



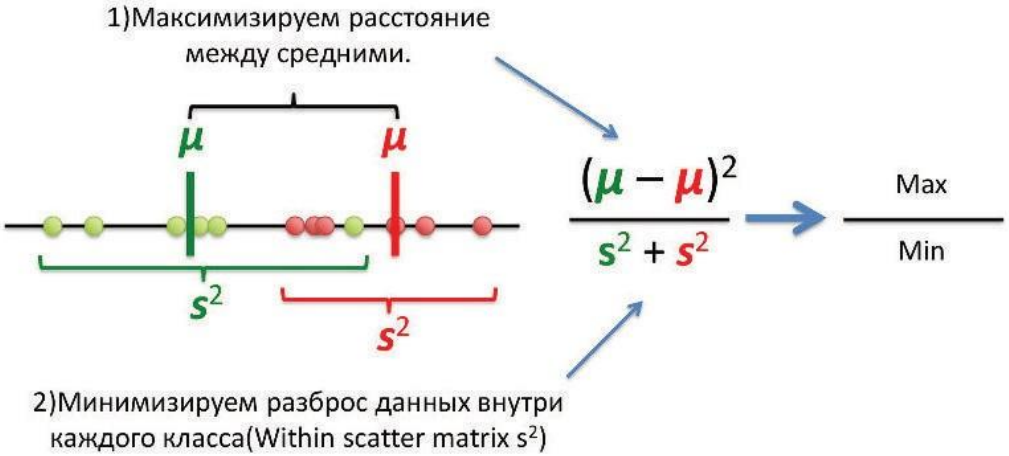
3D случай

Метод FDA основывается на нахождение вектора, в проекции на который данные разделимы наилучшим образом. После того как вектор найден, спроектированная дата может быть классифицирована различными способами.

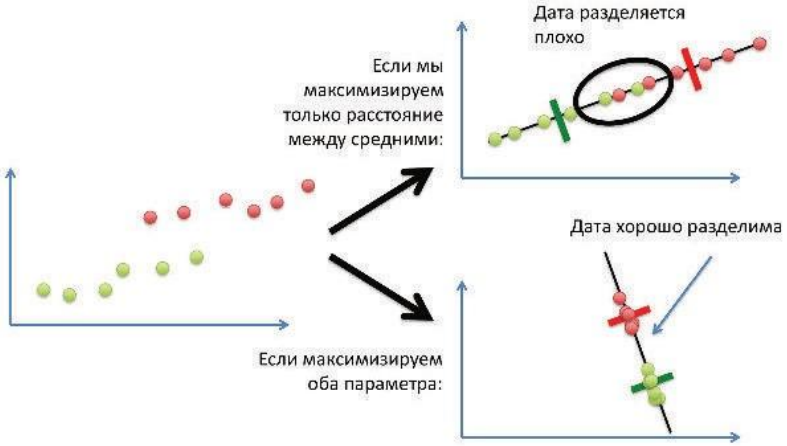
1

[1] Рис. с https://statquest.org/.

Принцип работы FDA
Как найти наилучший вектор для проекции данных?



Оптимизировать только одну величину недостаточно:



Принцип работы FDA

Задача: найти вектор ω, в проекции на который максимально отношение

$$I = \frac{(\mu_1' - \mu_2')^2}{s_1^2 + s_2^2} = \frac{S_b'}{S_w'}$$

- Обозначим еще не спроектированные образцы двух классов как:

$$X_1 = \begin{bmatrix} x_1^1 \\ x_2^1 \\ \dots \\ x_{l_1}^1 \end{bmatrix} = \begin{bmatrix} x_{1_1}^1 & x_{1_2}^1 & \dots & x_{1_n}^1 \\ x_{2_1}^1 & x_{2_2}^1 & \dots & x_{2_n}^1 \\ \dots & & & \\ x_{l_1 1}^1 & x_{l_1 2}^1 & \dots & x_{l_1 n}^1 \end{bmatrix}, X_2 = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \dots \\ x_{l_1}^2 \end{bmatrix}$$

- $$\mu_1 = \left[\frac{1}{l_1} \sum_{i=1}^{l_1} x_{i1}^1, \quad \dots, \frac{1}{l_1} \sum_{i=1}^{l_1} x_{in}^1 \right]$$
  $$\mu_2 = \dots$$

- Значение scatter matrix S_w для исходных данных:

$$S_w = S_1 + S_2 = \sum_{x \in X_1} (x - \mu_1)^T (x - \mu_1) + \sum_{x \in X_2} (x - \mu_2)^T (x - \mu_2) =$$

$$\begin{bmatrix} x^1_{1_1} - \mu_{11} & x^1_{1_2} - \mu_{12} & \dots & x^1_{1_n} \mu_{1n} \\ \dots & & & \\ x^1_{1_1} - \mu_{11} & x^1_{1_2} - \mu_{12} & \dots & x^1_{1_n} \mu_{1n} \end{bmatrix}^T \begin{bmatrix} x^1_{1_1} - \mu_{11} & x^1_{1_2} - \mu_{12} & \dots & x^1_{1_n} \mu_{1n} \\ \dots & & & \\ x^1_{1_1} - \mu_{11} & x^1_{1_2} - \mu_{12} & \dots & x^1_{1_n} \mu_{1n} \end{bmatrix}$$

$$+ \sum_{x \in X_2} (x - \mu_2)^T (x - \mu_2)$$

- Спроецированные данные: $x'_i = \omega^T x_i^T$.
- Значение scatter matrix S_w для спроектированных данных:

$$S'_w = \sum_{i=1,2} \sum_{x' \in X_i} (x'_i - \mu'_i)^2 = \sum_{i=1,2} \sum_{x' \in X_i} (\omega^T x_i - \omega^T \mu_i)^2 =$$

$$= \sum_{i=1,2} \sum_{x_i \in X_i} (x_i - \mu_i)^T (x_i - \mu_i) \omega^T = \omega S_w \omega^T$$

- Межклассовые scatter matrix S_b для исходных и спроецированных данных:

$$S_b = (\mu_1 - \mu_2)^T (\mu_1 - \mu_2)$$

$$S'_b = (\mu'_1 - \mu'_2)^T (\mu'_1 - \mu'_2) = \omega S_b \omega$$

Подставим в выражение для w:

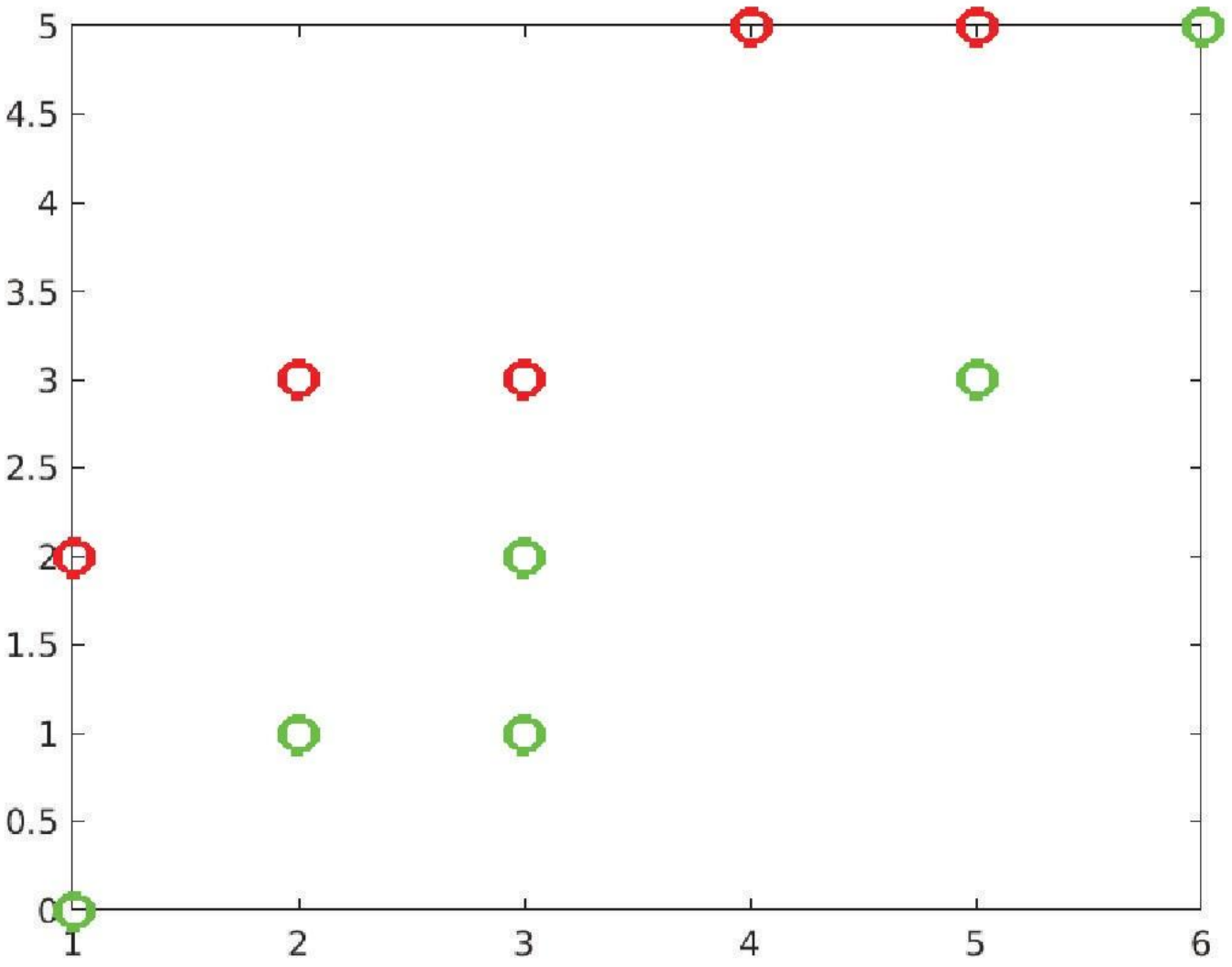
$$= \underset{\omega}{\operatorname{argmax}}\, I = \underset{\omega}{\operatorname{argmax}}\, J(\omega) = \underset{\omega}{\operatorname{argmax}} \frac{\omega S_b \omega^T}{\omega S_w \omega^T}$$

Продифференцировав по ω (подробнее см. здесь), приходим к уравнению:

$$S_w^{-1} S_b \omega = I \omega$$

Откуда, искомый вектор ω - собственный вектор матрицы $S_w^{-1} S_b$, соответстующей ее максимальному собственному значению.

Пример работы алгоритма



Пример работы алгоритма

- Данные: Первый класс и второй класс :

$$c_1 : \left[\ (1,2),(2,3),(3,3),(4,5),(5,5)\ \right]$$
$$c_2 : \left[\ (1,0),(2,1),(3,1),(3,2),(5,3),(6,5)\ \right]$$

- Запишем данные в матричном виде:

$$c_1 = \begin{bmatrix} 1 & 2 \\ \dots & \dots \\ 5 & 5 \end{bmatrix}, c_2 = \begin{bmatrix} 1 & 0 \\ \dots & \dots \\ 6 & 5 \end{bmatrix}$$

- Средние значения по каждому из классов и общее среднее:

$$\mu_1 = \begin{bmatrix} 3 & 3.6 \end{bmatrix}, \mu_2 = \begin{bmatrix} 3.3 & 2 \end{bmatrix}$$
$$\mu = \begin{bmatrix} 3.18 & 2.73 \end{bmatrix}$$

- Отклонение от среднего среди образцов каждого класса:

$$c_1 - \mu_1 = \begin{bmatrix} -2 & -1.6 \\ \dots & \dots \\ 2 & 1.4 \end{bmatrix}, c_2 - \mu_2 = \begin{bmatrix} -2.3 & -2 \\ \dots & \dots \\ 1.7 & 3 \end{bmatrix}$$

- Вычислим внутриклассовую(within) scatter матрицу S_w:

$$S_w = (c_1 - \mu_1)^T (c_1 - \mu_1) + (c_2 - \mu_2)^T (c_2 - \mu_2) =$$

$$\begin{bmatrix} -2 & -1.6 \\ \dots & \dots \\ 2 & 1.4 \end{bmatrix}^T \begin{bmatrix} -2 & -1.6 \\ \dots & \dots \\ 2 & 1.4 \end{bmatrix} + \begin{bmatrix} -2.3 & -2 \\ \dots & \dots \\ 1.7 & 3 \end{bmatrix}^T \begin{bmatrix} -2.3 & -2 \\ \dots & \dots \\ 1.7 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 27.3 & 24 \\ 24 & 23.2 \end{bmatrix}; \quad S_w^{-1} = \begin{bmatrix} 0.39 & -0.41 \\ -0.41 & 0.47 \end{bmatrix}$$

- Теперь межклассовую(between) scatter матрицу

$$S_b = (\mu_1 - \mu_2)^T (\mu_1 - \mu_2) = \begin{bmatrix} 0.3 & -1.45 \\ -1.45 & 6.98 \end{bmatrix}$$

- $$S_b S_w^{-1} = \begin{bmatrix} 0.3 & -1.45 \\ -1.45 & 6.98 \end{bmatrix} \begin{bmatrix} 0.39 & -0.41 \\ -0.41 & 0.47 \end{bmatrix} = \begin{bmatrix} 0.71 & -0.8 \\ -3.43 & 3.88 \end{bmatrix}$$

- Собственные значения и соответсвующее им собственные вектора:

$$\lambda_1 = 0 \ : \begin{bmatrix} -0.98 \\ -0.2 \end{bmatrix}, \quad \lambda_2 = 4.6 \ : \begin{bmatrix} 0.66 \\ -0.75 \end{bmatrix}$$

- Находим w - искомый вектор проекции, как вектор соответствующий макс собственному значению:

$$\omega = \begin{bmatrix} 0.66 & -0.75 \end{bmatrix}^T$$

- Проецируем дату на любую прямую, имеющую направление w:

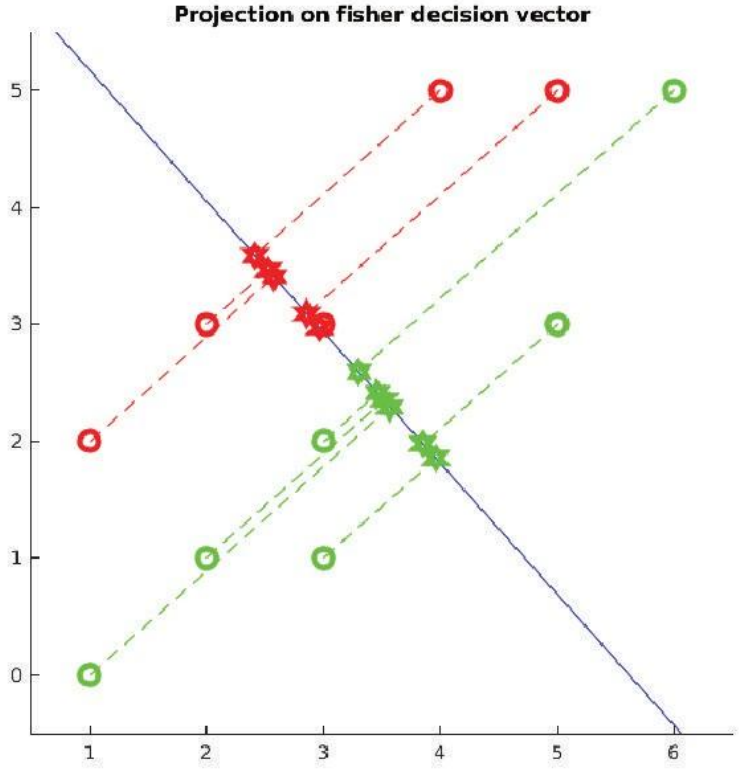


Рис. 2: Спроецированная дата легко разделяется на классы (например по расстоянию от среднего каждого класса)

Основной код для реализации FDA

```
def FisherVectorProject(X,y): #Возращает наилучший вектор для проекции, определямый
    dimension = X.shape[1];
    # label -массив индентификаторов классов
    labels = np.unique(y)
    Sw = np.zeros((dimension,dimension));
    Sb = np.zeros((dimension,dimension));
    mu_total = mean(X);
    for i in range (0,2):
        Xi = X[np.nonzero(y == labels[i])[0],:];
        n = Xi.shape[0];
        mu_i = mean(Xi);
        XMi = Xi - mu_i;
        Sw = Sw + np.dot(XMi.T, XMi );
        MiM =  np.array([mu_i - mu_total]);
        Sb = Sb + n*np.dot(MiM.T,MiM);
    #S = np.dot(np.linalg.inv(Sw),Sb);
    S = np.linalg.solve(Sw, Sb)
    #Собственный вектор, соответсвующий макс собств значению
    W_fda = MaxEg(S)
    return W_fda
def FisherDataProject(X, W, mu_total): #Проекция точек из X на вектор W
    X_proj = np.dot(X,W);
    X_proj = np.dot(X_proj,W.T);
    X_proj =  X_proj + mu_total;
    return X_proj
```

Более сложный пример

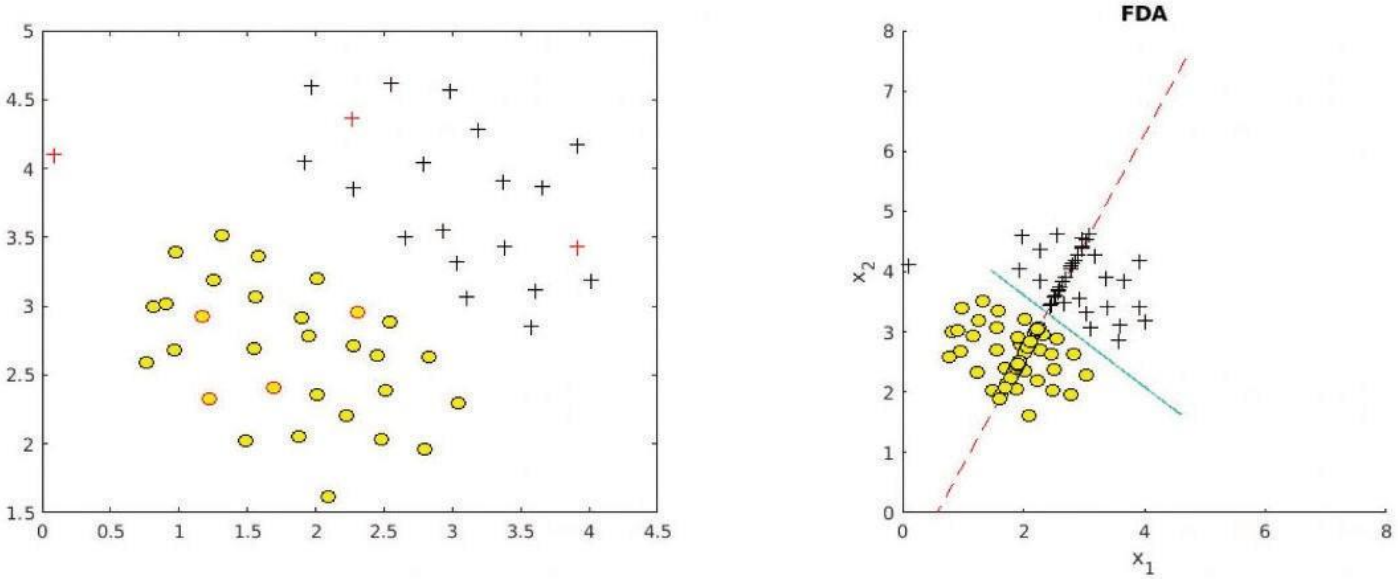


Рис. 3: Dataset1 и его классификация

Функция, используемая для классификации спроектированных данных:

```
def FDApred(X,W,mu0_red,mu1_red, labels):
    y = np.zeros((X.shape[0], 1))
    for i in range(0, X.shape[0]):
        y[i,0] = labels[0]
        Xi_red = FisherDataRed(np.array([X[i,:]]), W)
        if (abs(Xi_red-mu1_red) < abs(Xi_red - mu0_red)):
            y[i,0] = labels[1]
    return y
```

Дискриминантный анализ. Общие принципы и методы

- DA предполагает, что размерность пр-ва(кол-во predictors p) меньше p кол-во образцов n. Точность классификатора "сохраняется"при $n \geq 5p$.
- DA предсказывает вероятность попадения образцов, с определнным значением предикта, в конкретный класс отдельно для каждого класса.
- Используя Th. Байеса по этим вероятностям DA находит для определенного образца и класса *discriminant score*, определяющее вероятность его принадлежности к данному классу.
- DA относит образец к классу, для которого discriminant score макс.

Предположив, что образцы в каждом классе имеют ***нормальное распределение*** получим формулу, задающую discriminat score для k-го класса (подробнеее см. в статье):

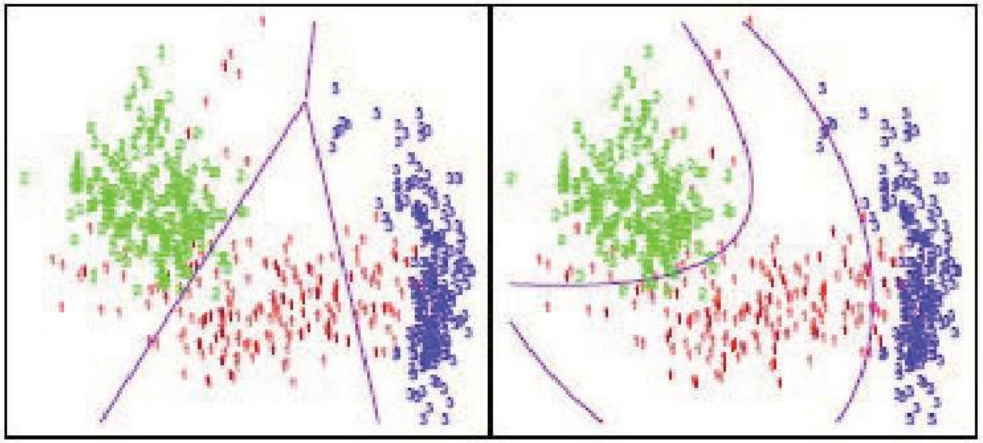
$\delta_k(x) = -\frac{l_k}{2} x^T S_k^{-1} x + l_k x^T S_k^{-1} \mu_k - \frac{l_k}{2} \mu_k^T S_k^{-1} \mu_k - \frac{1}{2} log|S_k| + \frac{1}{2} log l_k +$
$+ log(\pi_k)$ (1), π_k доля образцов класса k среди всех образцов.

Другие методы DA. LDA и QDA

Формула (1) задает квадратичное Decision Boundary ($\delta_1(x) = \delta_2(x)$) и лежит в основе метода **QDA**.
В **LDA** используется предположение, что распределение образцов в каждом классе (σ) одинаково, и формула(1) упрощается:

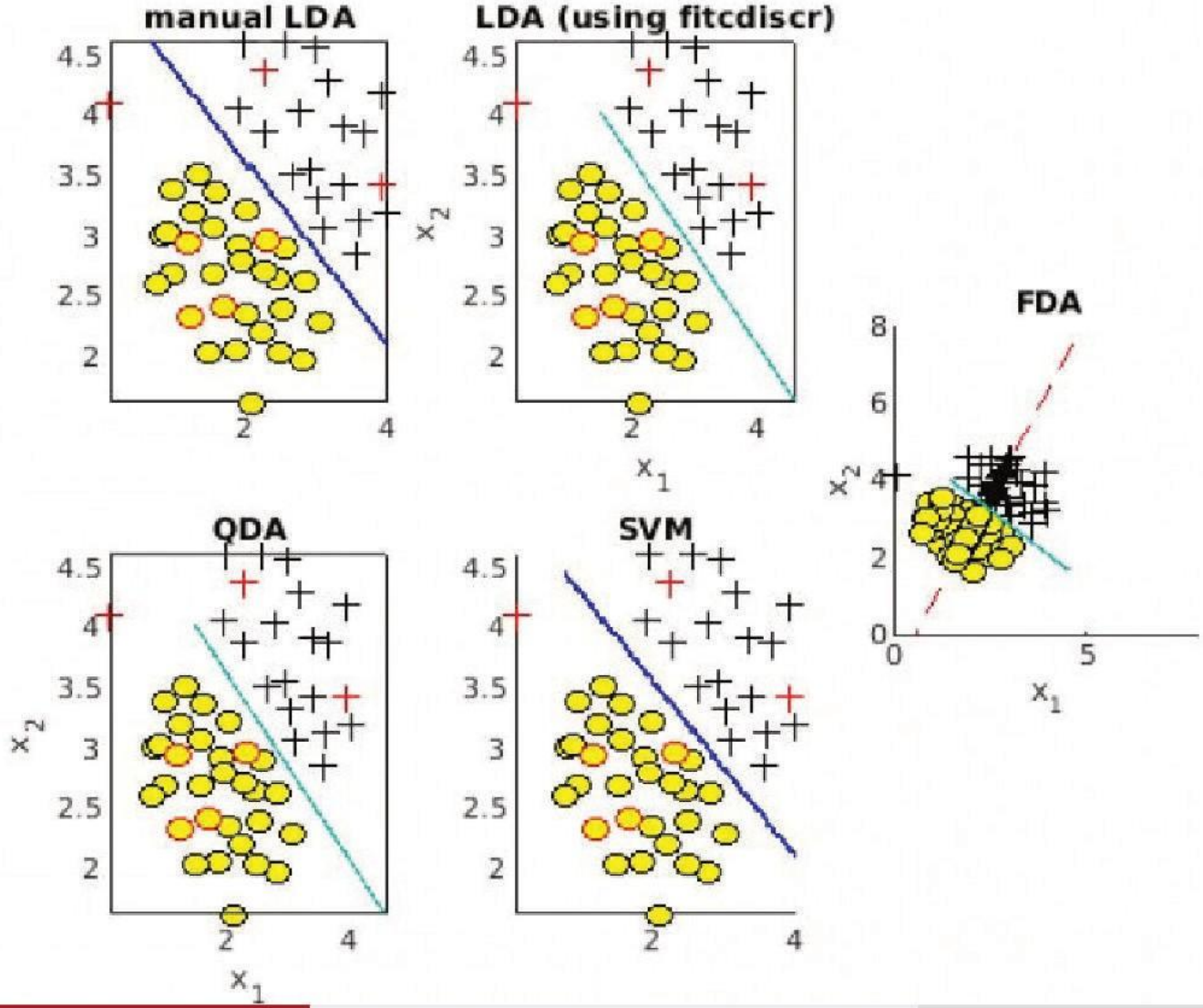
LDA

$\delta_k(x) = xl_k S_k^{-1}\mu_k - \frac{l_k}{2}\mu_k^T S_k^{-1} - \mu_k + log(\pi_k)$
l_k - количество образцов в классе k. Decision Boundary имеет линейную форму.



Распределние образцов внутри классов различно, и QDA работает лучше

Сравнение FDA с другими линейными классификаторами и QDA (на примере Dataset1)



FDA не справляется. Пример

Расмотрим данные, образующие две концентрические окружности. Как видно из рисунка, после применения FDA и проектирования на найденный вектор дата по-прежнему неразделима.

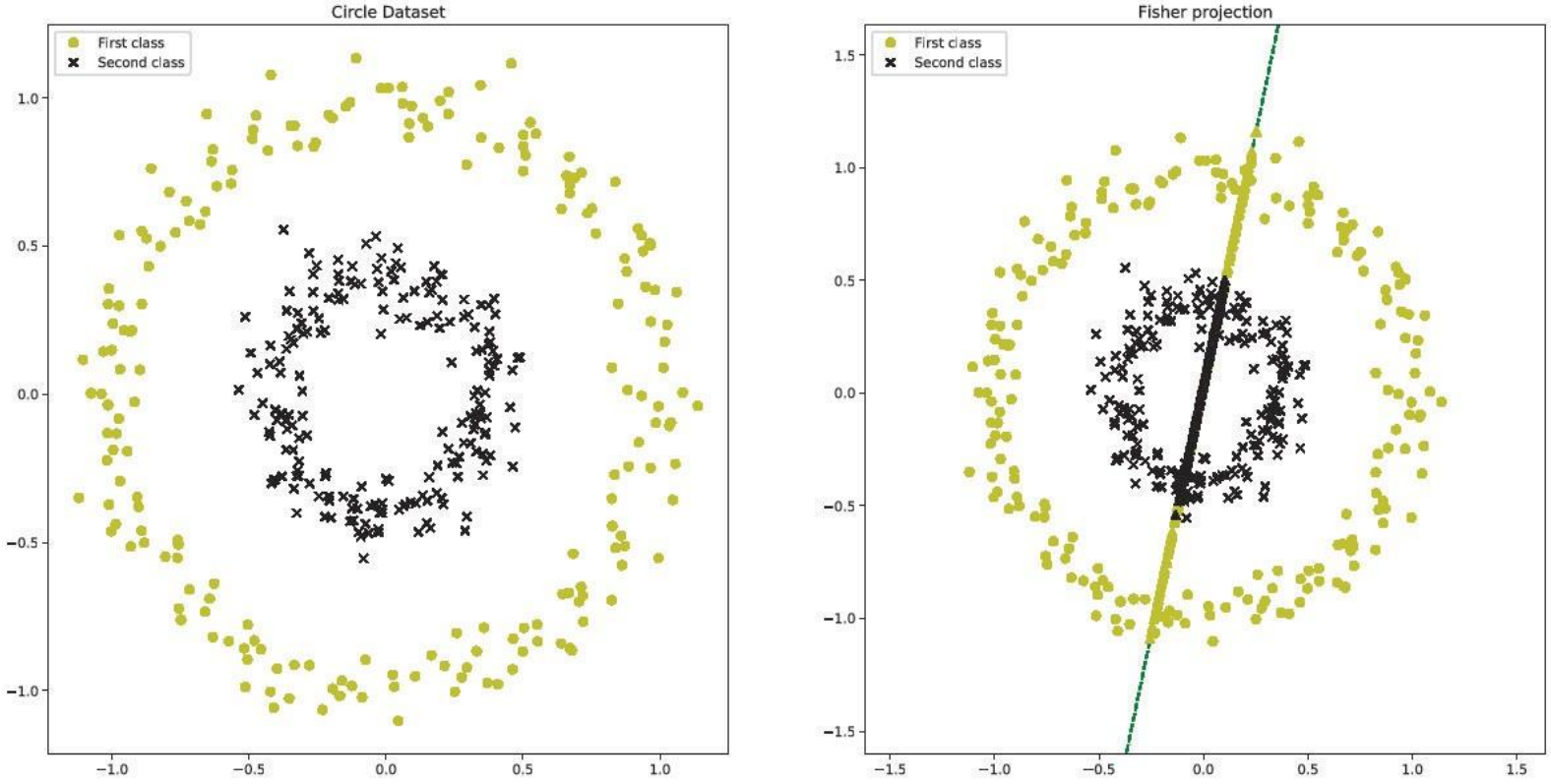


Рис. 4: Линейно неразделимые данные. CircleData

Как работать с такими данными?
Попробуем перейти в пространство большей размерности

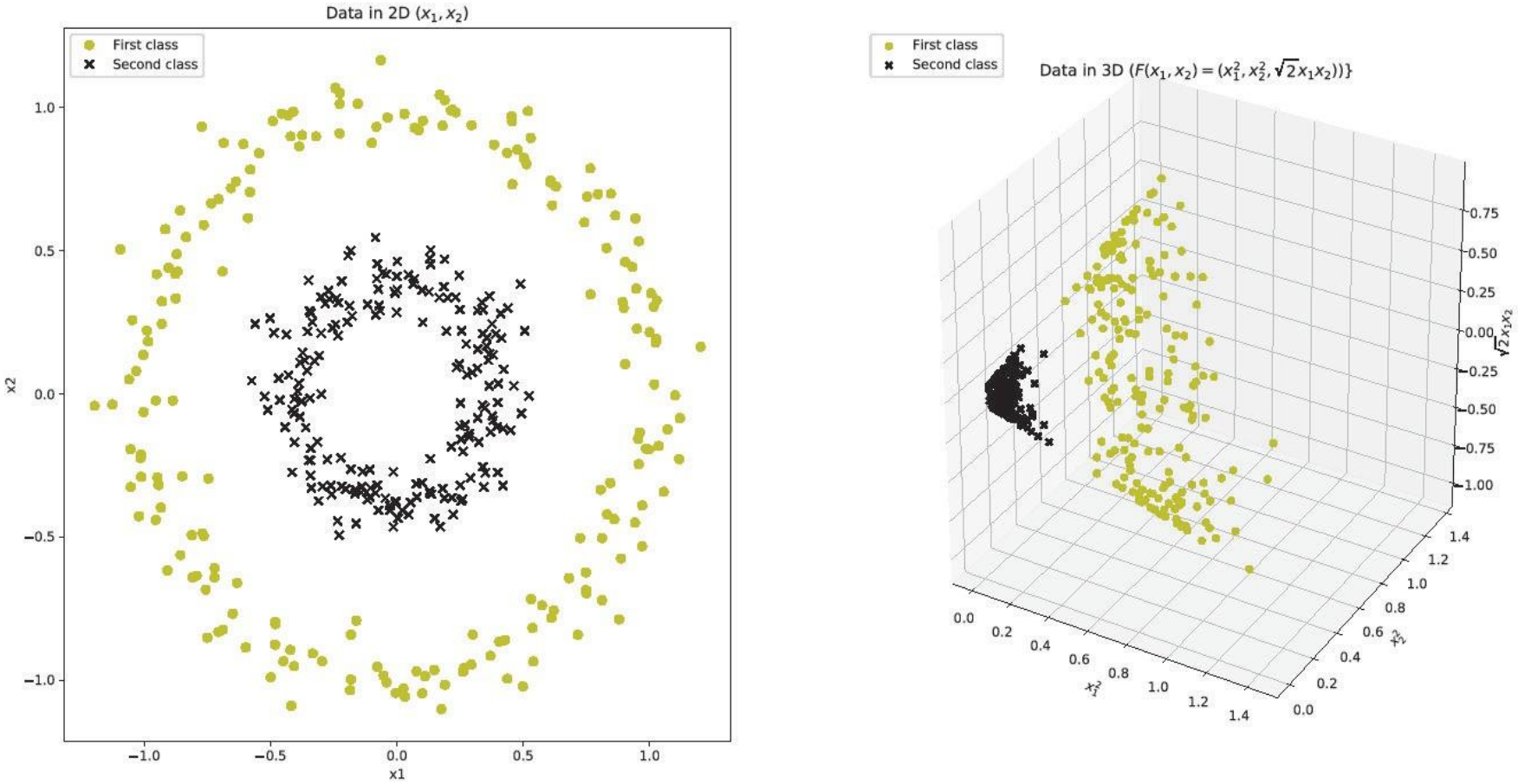
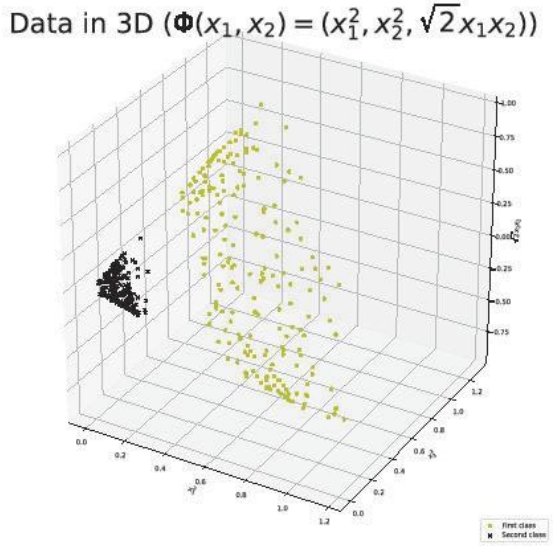
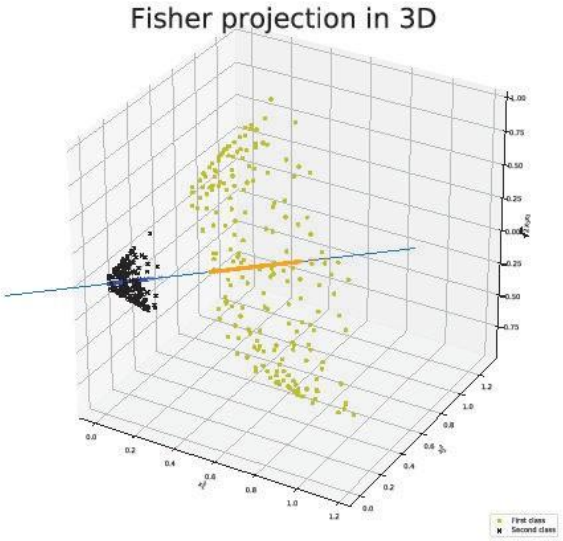


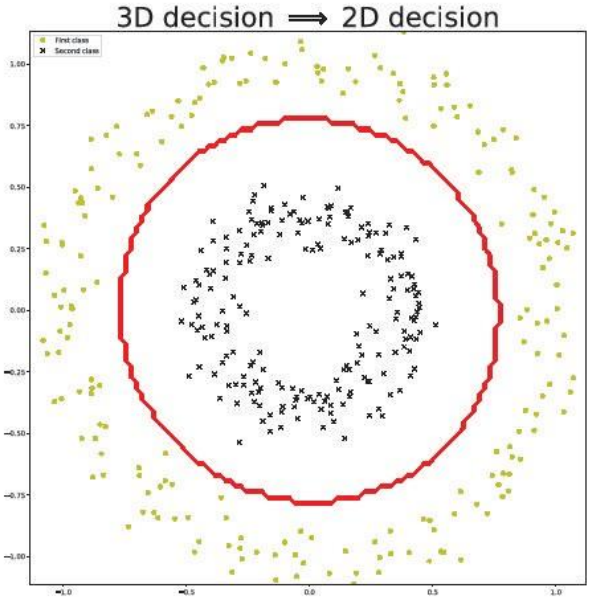
Рис. 5: $(x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1x_2)$

Как работать с такими данными? Попробуем перейти в пространство большей размерности

В пространстве большей размерности данные линейно разделимы. Классифицировав данные в 3D, можно спроектировать решение в 2D:







Что было сделано для решения задачи? (Способ №1)

- $\forall x$ вычислено $\Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$.
- Классификация даты $\Phi(x)$ методом FDA в более высоком пр-ве. Для этого необходимо вычислить скалярные произведения $(\Phi(x_i), \Phi(x_j)) = (\{x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}\}, \{x_{j1}^2, x_{j2}^2, \sqrt{2}x_{j1}x_{j2}\}) = (x_i, x_j)^2$.

Другой способ (№2):

- Для всевозожных пар образцов вычислим значение: $k(x_i, x_j) = (x_i, x_j)^2$.
- Т.к. при работе алгоритма FDA,только скалярные произведения (x_i, x_j), подставив вместо них $k(x_i, x_j)$, получим результат, совп со способом №1.

Идея, использованная в способе №2 - **Kernel trick**.

Kernel trick

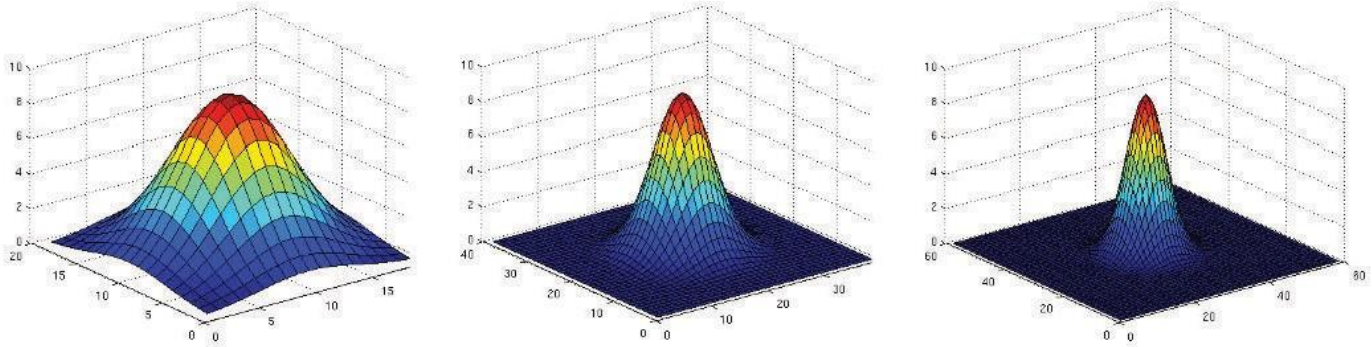
Метод, позволяющий использовать линейный классификатор для линейно неразделимых данных, если эти данные разделимы в более высоком пространстве m. При этом алгоритму классификатора на вход необходимо подать *только* kernel function, X и labels (метки классов объектов), где **kernel function**:

$$k(x_i, x_j) = (\Phi(x_i), \Phi(x_j)), \text{ где } \Phi : R^n \longrightarrow R^m.$$

Работа в бесконечномерном пространстве
Gaussian Kernel

Gausian Kernel (RBF)

$$K(x_i, x_j) = exp(\frac{-||x_i - x_j||^2}{2\sigma^2})$$



$$exp(\frac{-||x_i - x_j||^2}{2\sigma^2}) = exp(\frac{-(||x_i||^2 + ||x_j||^2)}{2\sigma^2} exp(\frac{(x_i, x_j)}{\sigma^2}) =$$

$$= C(1 - C_1(x_i, x_j) + C_2(x_i, x_j)^2 - C_3(x_i, x_j)^3 \cdots + (-1)^n C_n(x_i, x_j)^n \dots)$$

$\Rightarrow$ *Если дата разделима в ∞ пространстве, то применив RBF Kernel ее можно классифицировать в пространстве исходной размерности.*

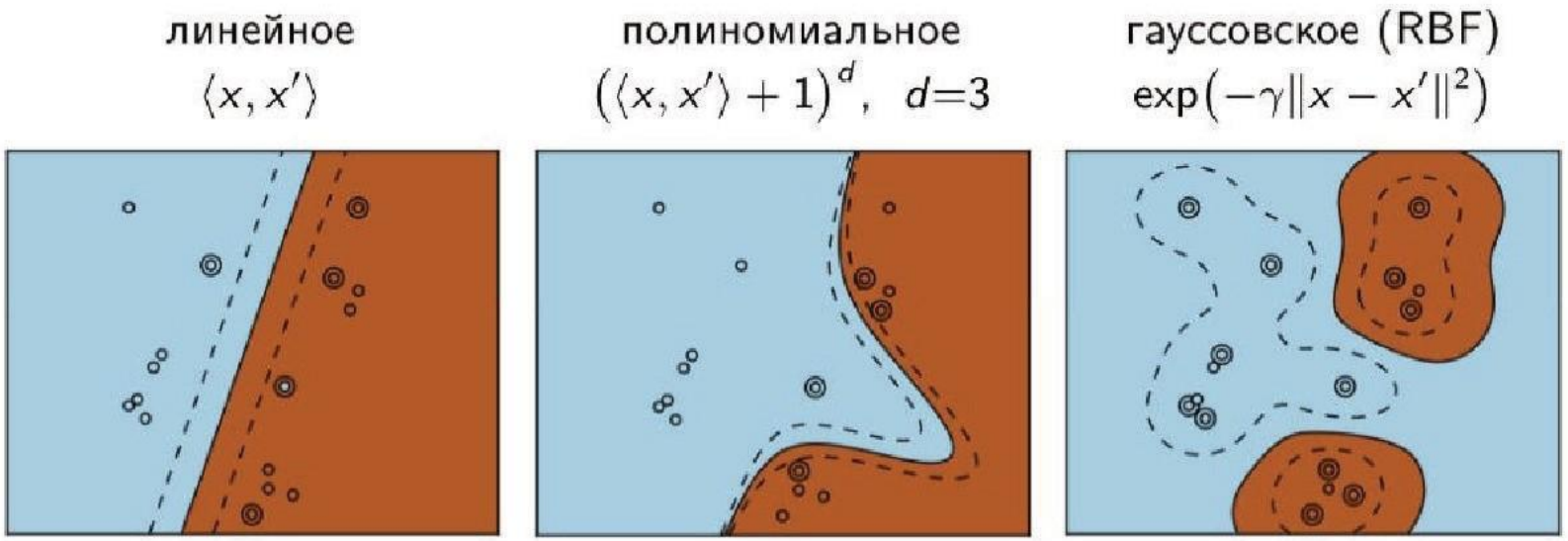
Типы ядер

Th. Мерсера

Функция K(x, z) является ядром тогда и только тогда, когда:

- Она симметрична: K(x, z) = K(z, x).
- Она неотрицательно определена, то есть для любой конечной выборки $(x_1 \dots x_l)$ матрица $K = (K(x_i, x_j))_{i,j=1}^{l}$ неотрицательно определена.

Наиболее распространенные ядра[3]:



[3] Рис., сгенерирован кодом из scikit-learn example.

Переход от FDA к KFDA

Основные равенства

$$k(x, y) = (\Phi(x) \cdot \Phi(y))$$

$$\omega = \underset{\omega}{\operatorname{argmax}} \frac{\omega S_b^{\Phi} \omega^T}{\$\omega S_w^{\Phi} \omega^T}$$

$$S_w^{\Phi} = S_1^{\Phi} + S_2^{\Phi} = \sum_{i=1,2} \sum_{\Phi(x) \in X_i} (\Phi(x_i) - \mu_i^{\Phi})^T (\Phi x_i - \mu_i^{\Phi})$$

$$S_b^{\Phi} = (\mu_1^{\Phi} - \mu_2^{\Phi})^T (\mu_1^{\Phi} - \mu_2^{\Phi})$$

$$\omega = \sum_{i=1}^{l} a_i \Phi(x_i)$$

Алгоритм KFDA

Kernel matrix Kj для класса j

Матрица размером $l \times l_j$, где $(\mathrm{K}_j)_{nm} = k(x_n, x_m^j)$.

$$\omega = \sum_{i=1}^{l} a_i \Phi(x_i)$$

$$(M_i)_j = \frac{1}{l_i} \sum_{k=1}^{l_i} k(x_j, x_k^i)$$

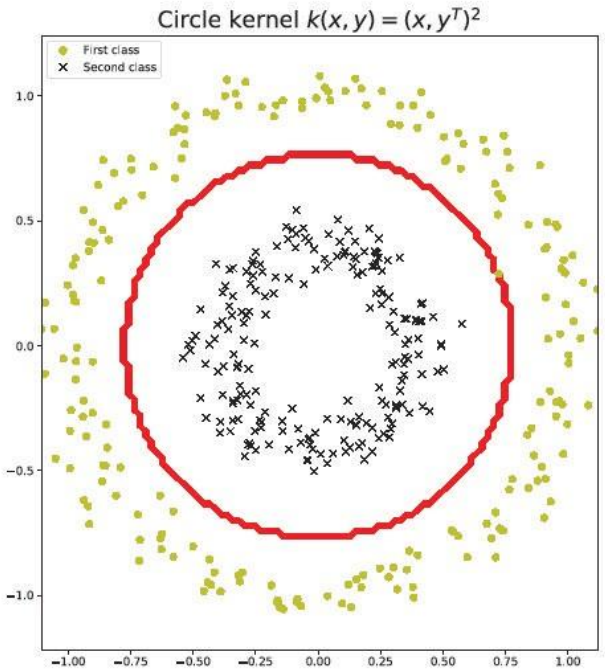
$$M = (M_1 - M_2)^T (M_1 - M_2)$$

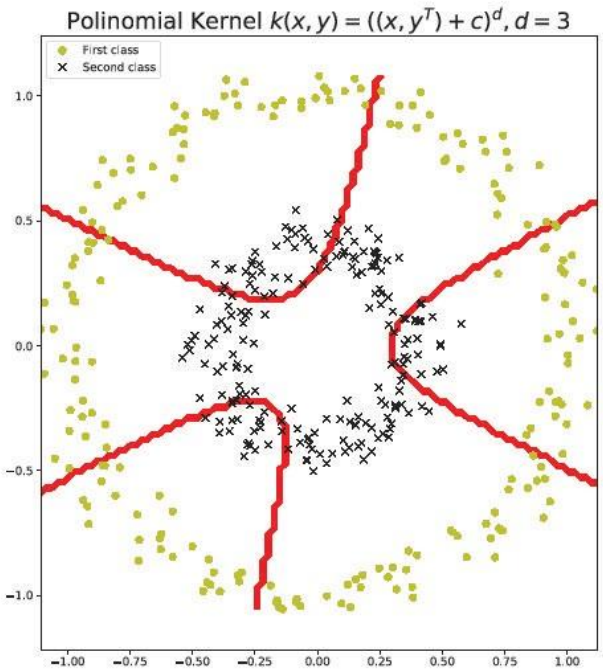
$N = \sum_{j=1,2} K_j (E - 1_{li}) K_j^T$, где 1_{li} -матрица с элементами $1/l_i$

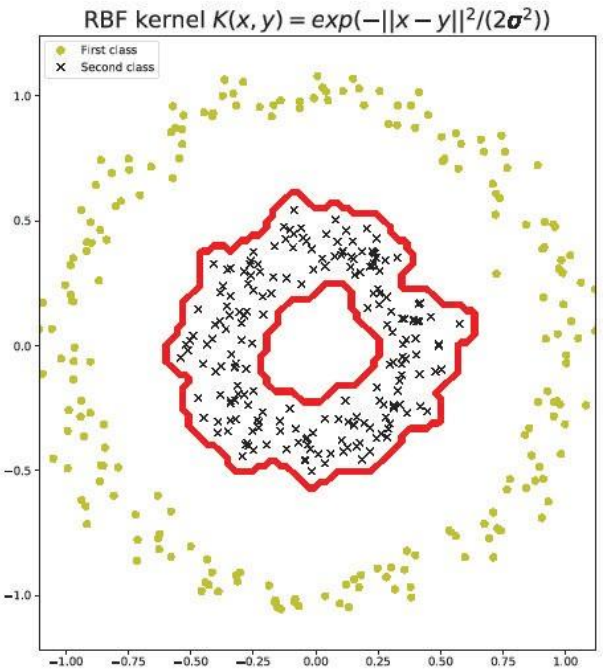
$J(a) = \frac{aMa^T}{aNa^T}, \Rightarrow a = MaxEg(MN^{-1})$

$$(w, \Phi(x)) = \sum_{i=1}^{l} a_i k(x_i, x) = K \cdot a$$

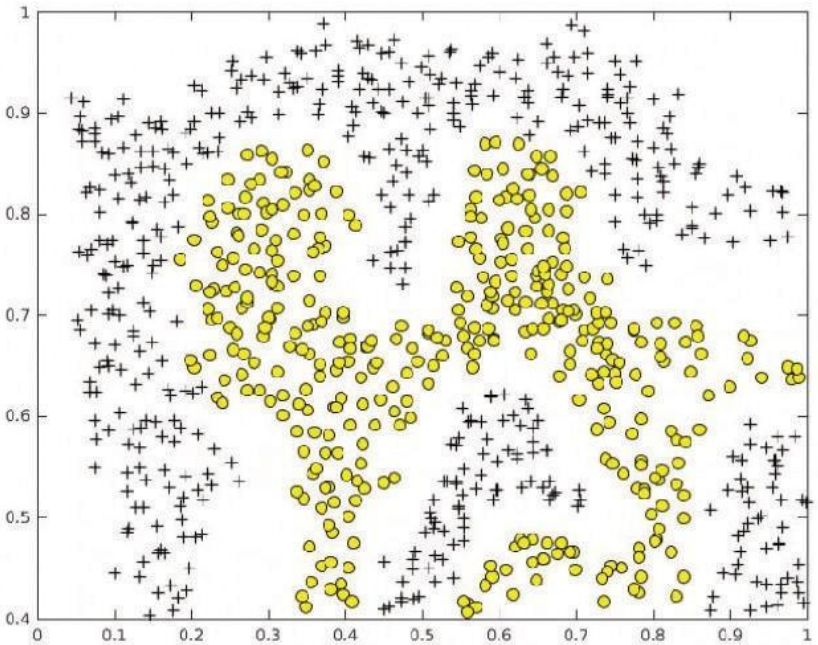
KFDA с различными ядрами на CircleData

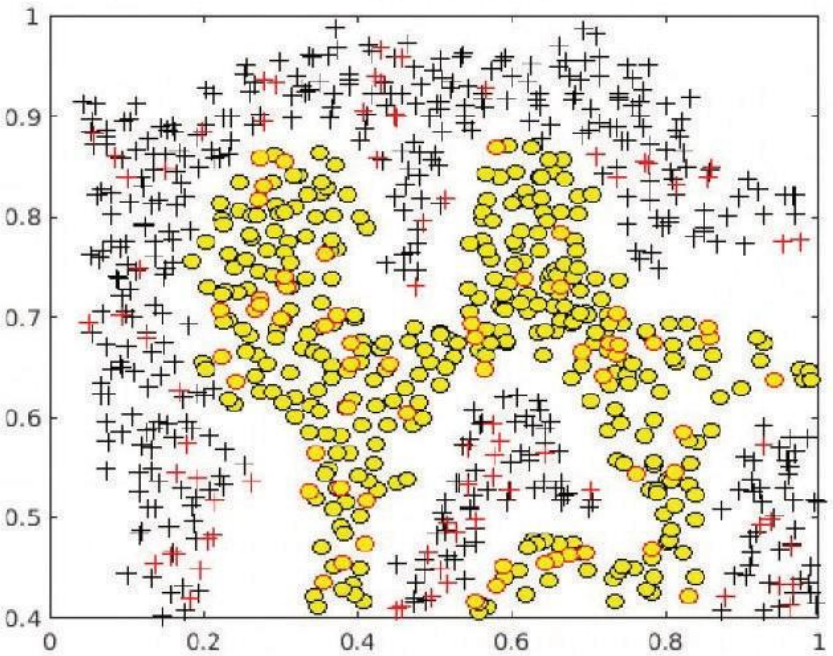






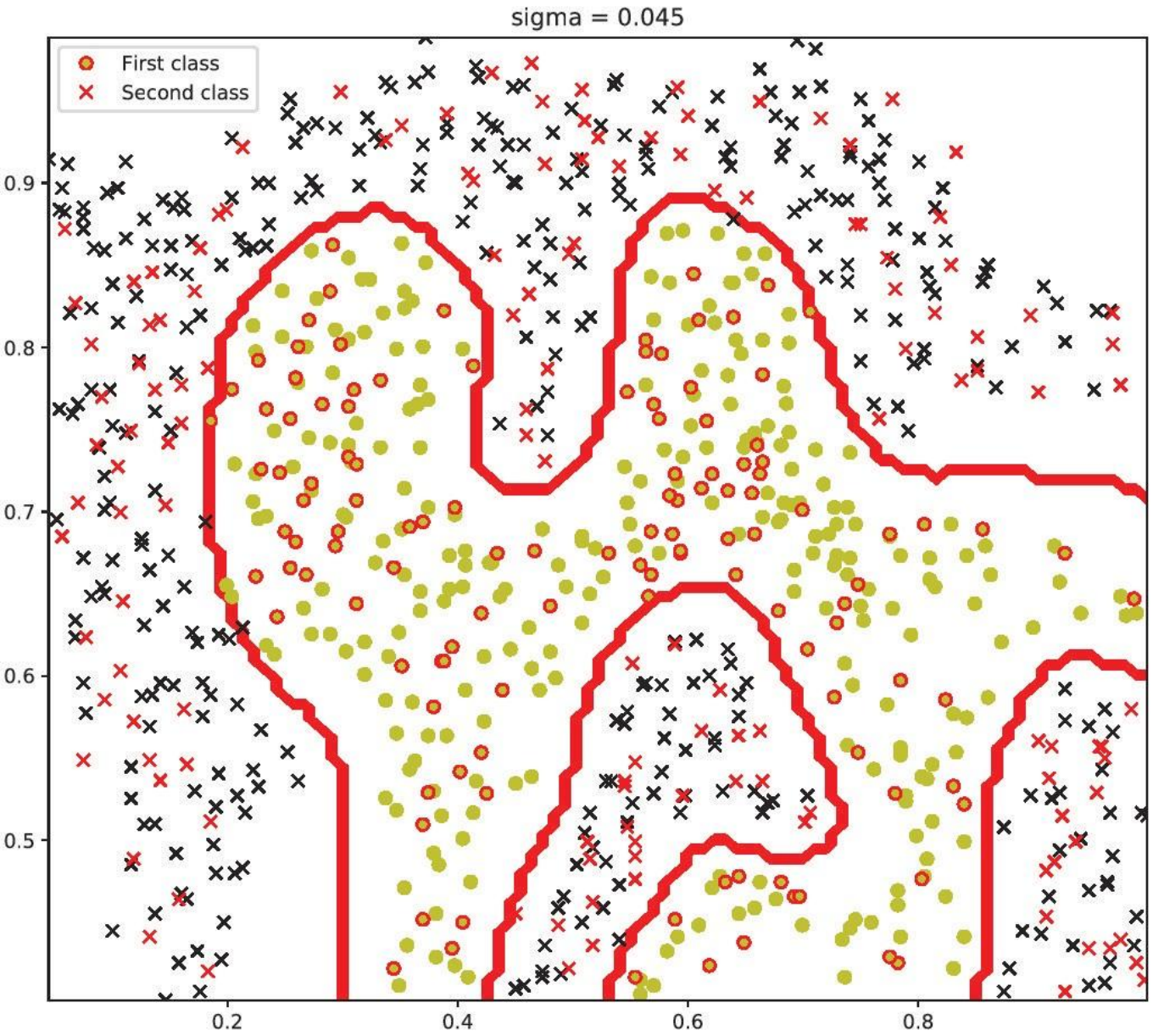
KFDA.Пример работы алгоритма Dataset2



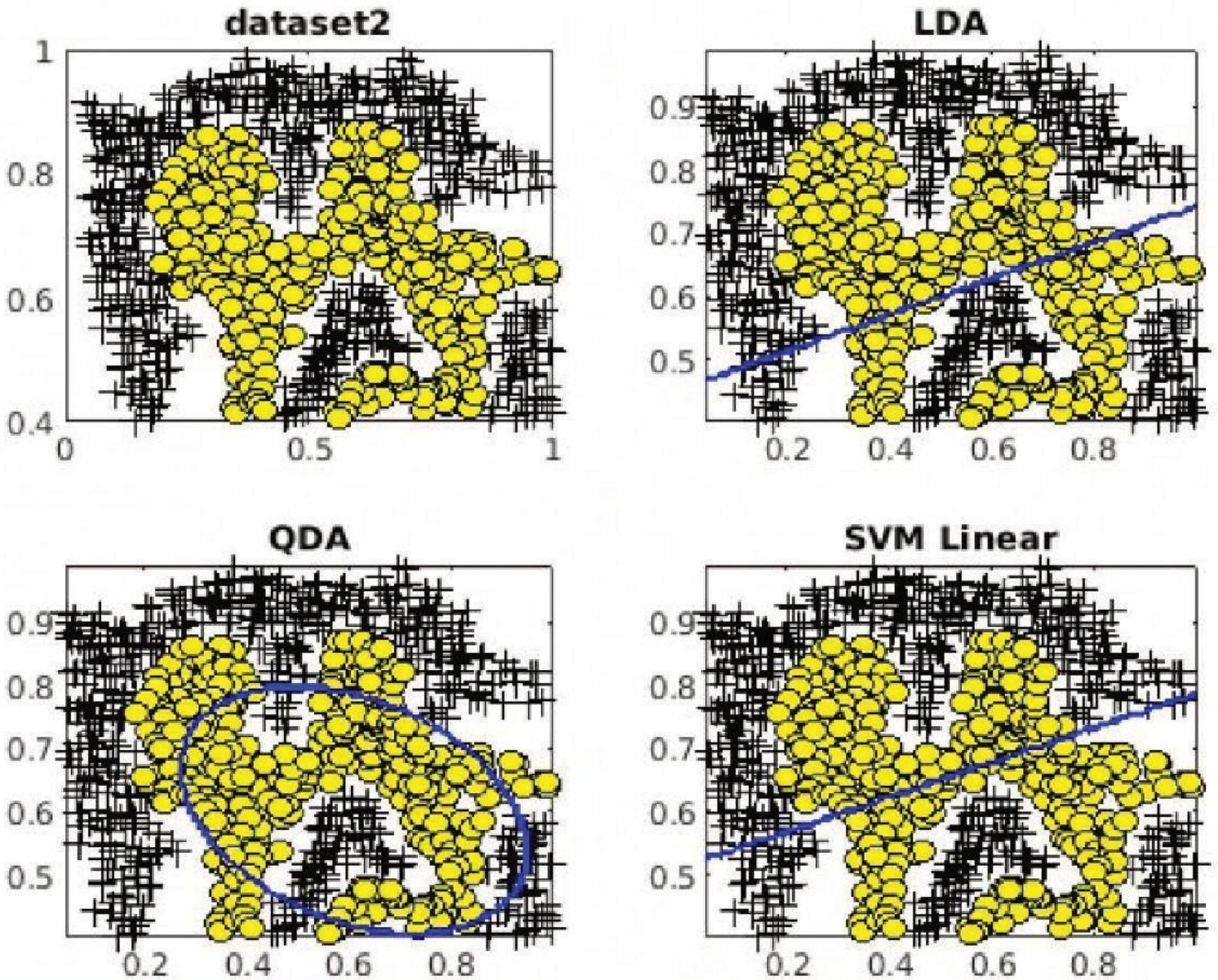


Данные Dataset2,разделенные на тренировочные и тестовые(обведены красным на рис.), на них будем проверять точность модели.

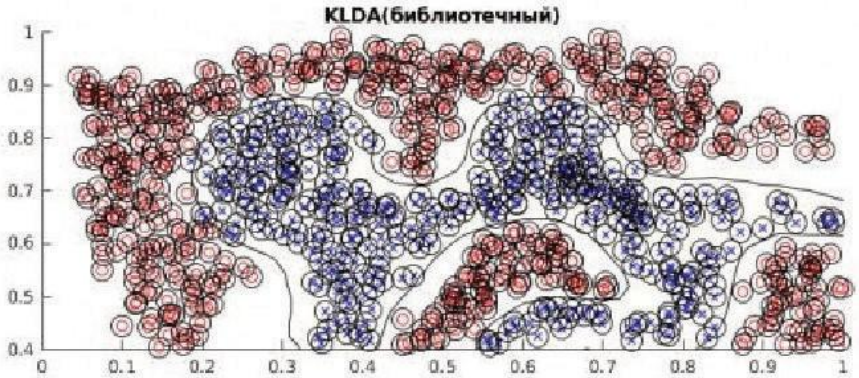
KFDA.Пример работы алгоритма

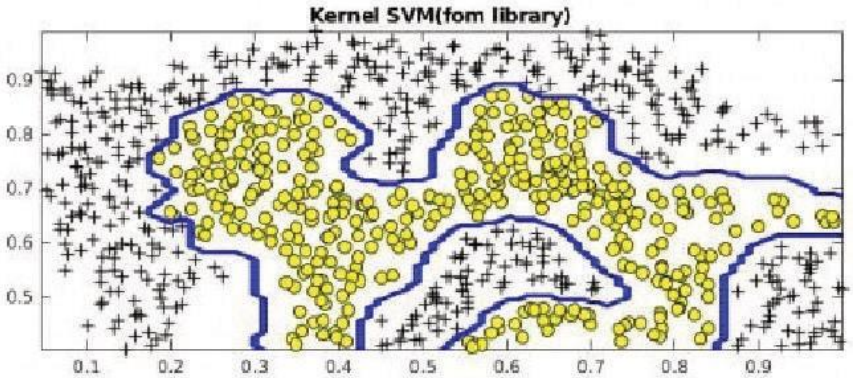


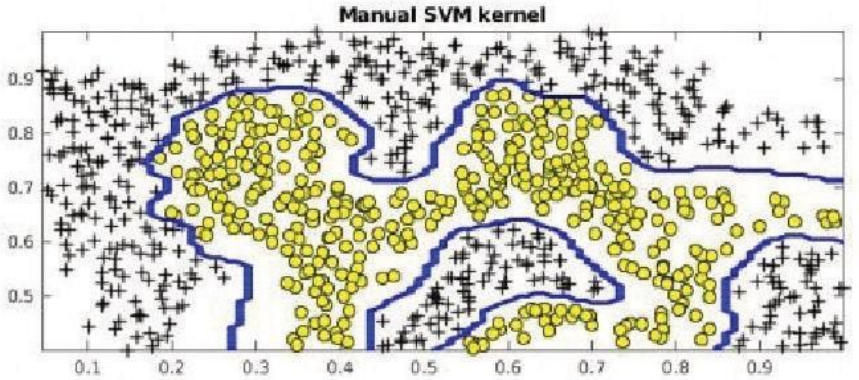
Линейные классификаторы и QDA на Dataset2



Другие реализации KFDA и KSVM на Dataset2







Результаты классификаторов не сильно различаются на Dataset2.

Параметры ядра

σ - параметр RBF Kernel: $K(x_i, x_j) = exp(\frac{-||x_i - x_j||^2}{2\sigma^2})$. Классификаторы очень чувствительны к этому параметру.

Таблица 1: Точность KFDA с RBF kernel на Dataset2 при различных σ

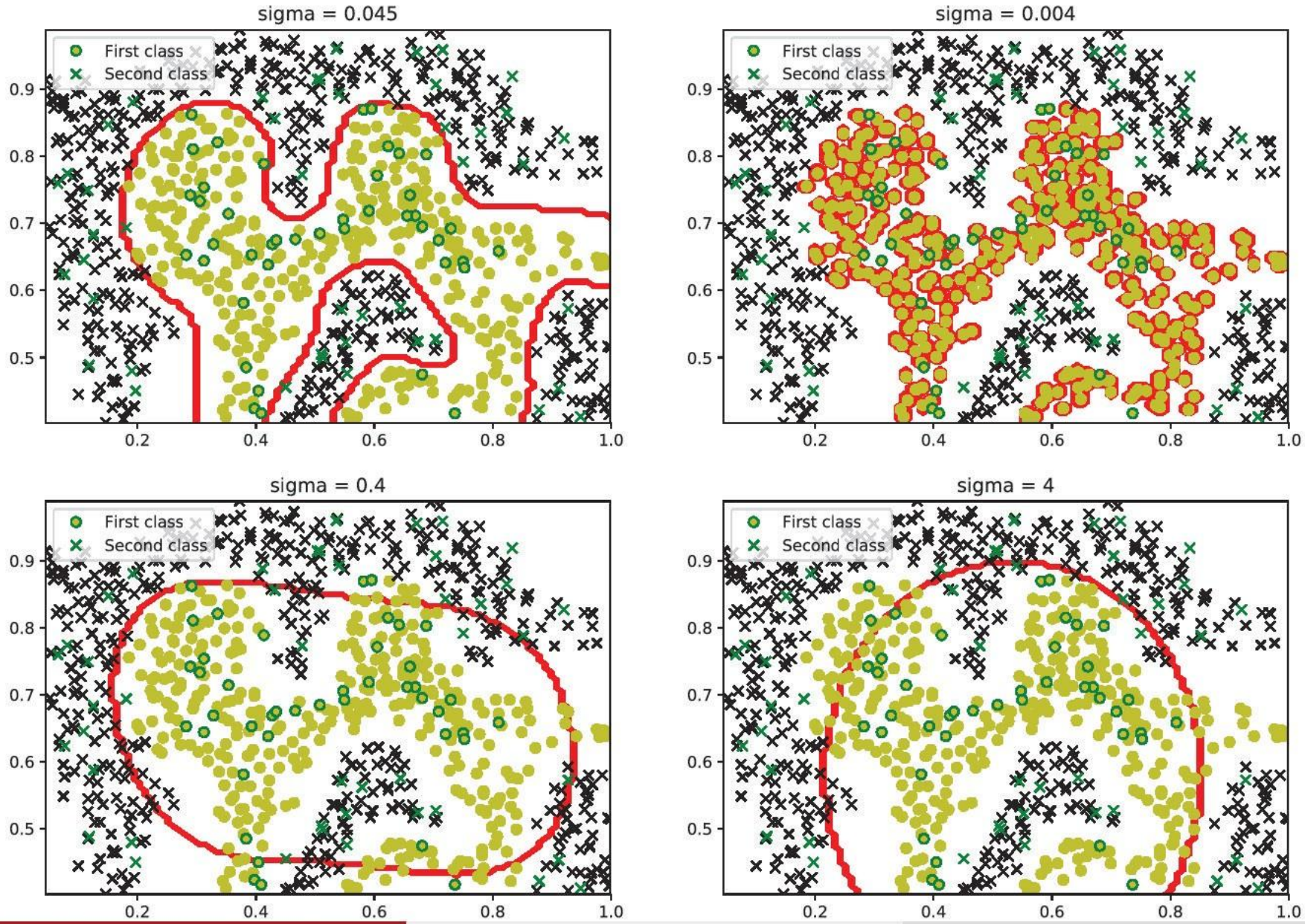
Параметр σ	Ошибка классификатора
0.045	0.0034
0.004	0.08
0.4	0.118
4	0.1793

В чем функция данного параметра?

σ определяет как далеко распространяется влияние каждого training образца:

- $\uparrow \sigma$: У ядра большая дисперсия. Область влияния вектора $\uparrow$.Модель приближается к линейной модели с набором гиперплоскостей.
- $\downarrow \sigma$: У ядра маленькая дисперсия. Область влияния вектора $\downarrow$. При классификации образуются "острова"вокруг образцов. Это приводит к overfitting.

Зависимость KFDA с RBF kernel от sigma (для Dataset2)



KFDA для многоклассовой задачи
Что изменится в алгоритме?

- В случае **c** классов данные можно спроецировать в $(c-1)$-мерное пространство.
- w - матрица проекции, состоящая из (c-1) вектора.
- Чтобы значение функции J(w) был скаляр, функция записывается как:

$$J(a) = \frac{\det(wS_b w^T)}{\det(wS_w w^T)}$$

Алгоритм

$$(M_i)_j = \frac{1}{l_i} \sum_{k=1}^{l_i} k(x_j, x_k^i)$$

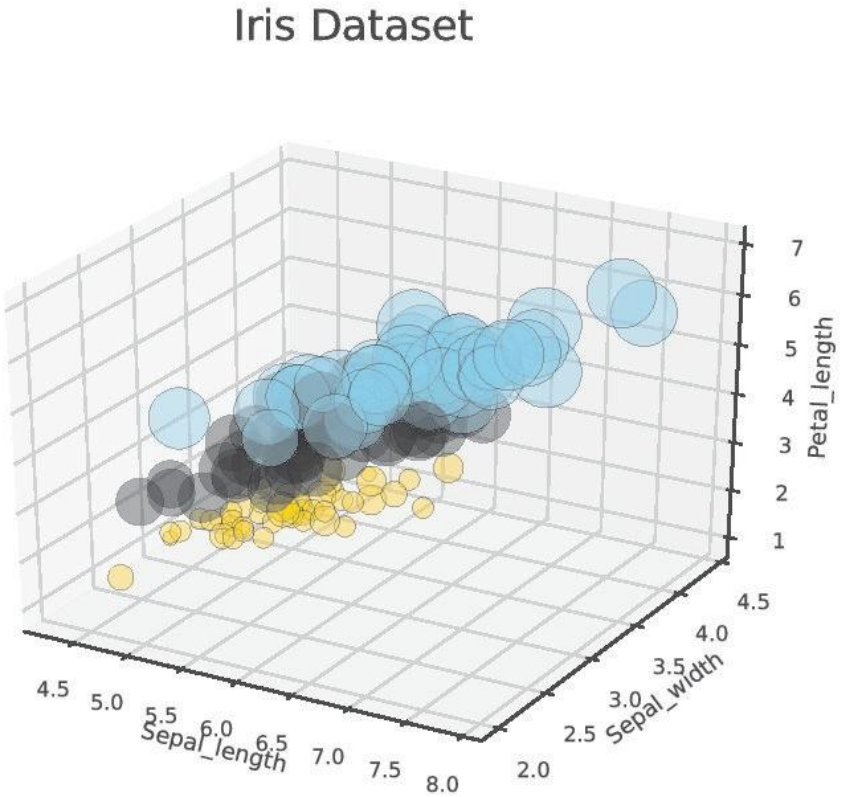
$$M = \sum_{i=1,2,3} (M_0 - M_i)^T (M_0 - M_i)$$

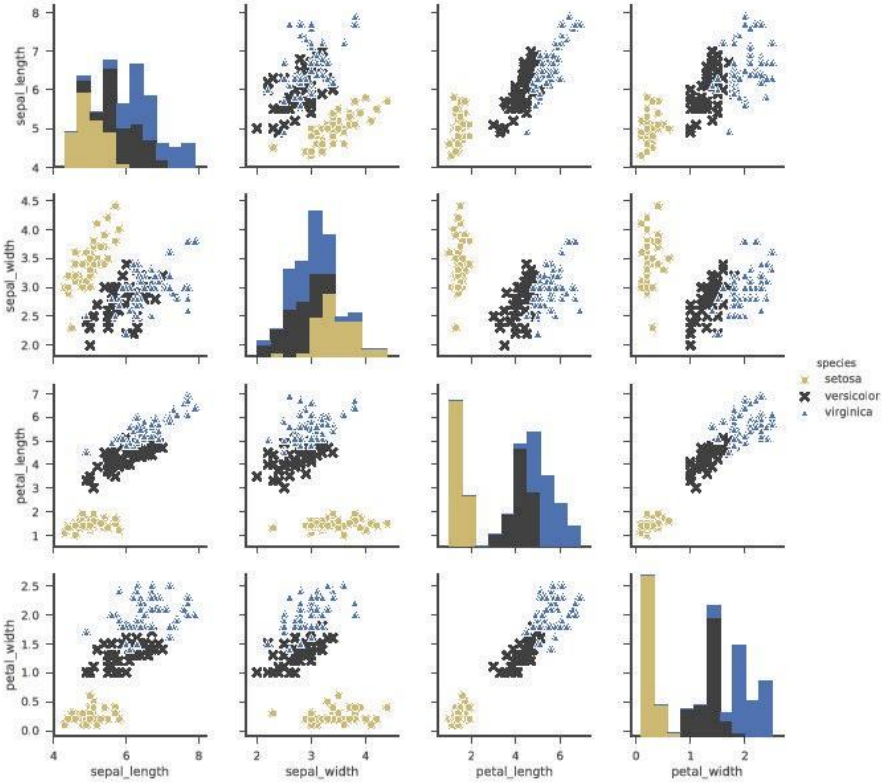
$$N = \sum_{j=1,2,3} K_j (Id - 1_{l_i}) K_j^T$$

a - собственный вектор $(M^{-1}N)$, соотв макс собственному значению.

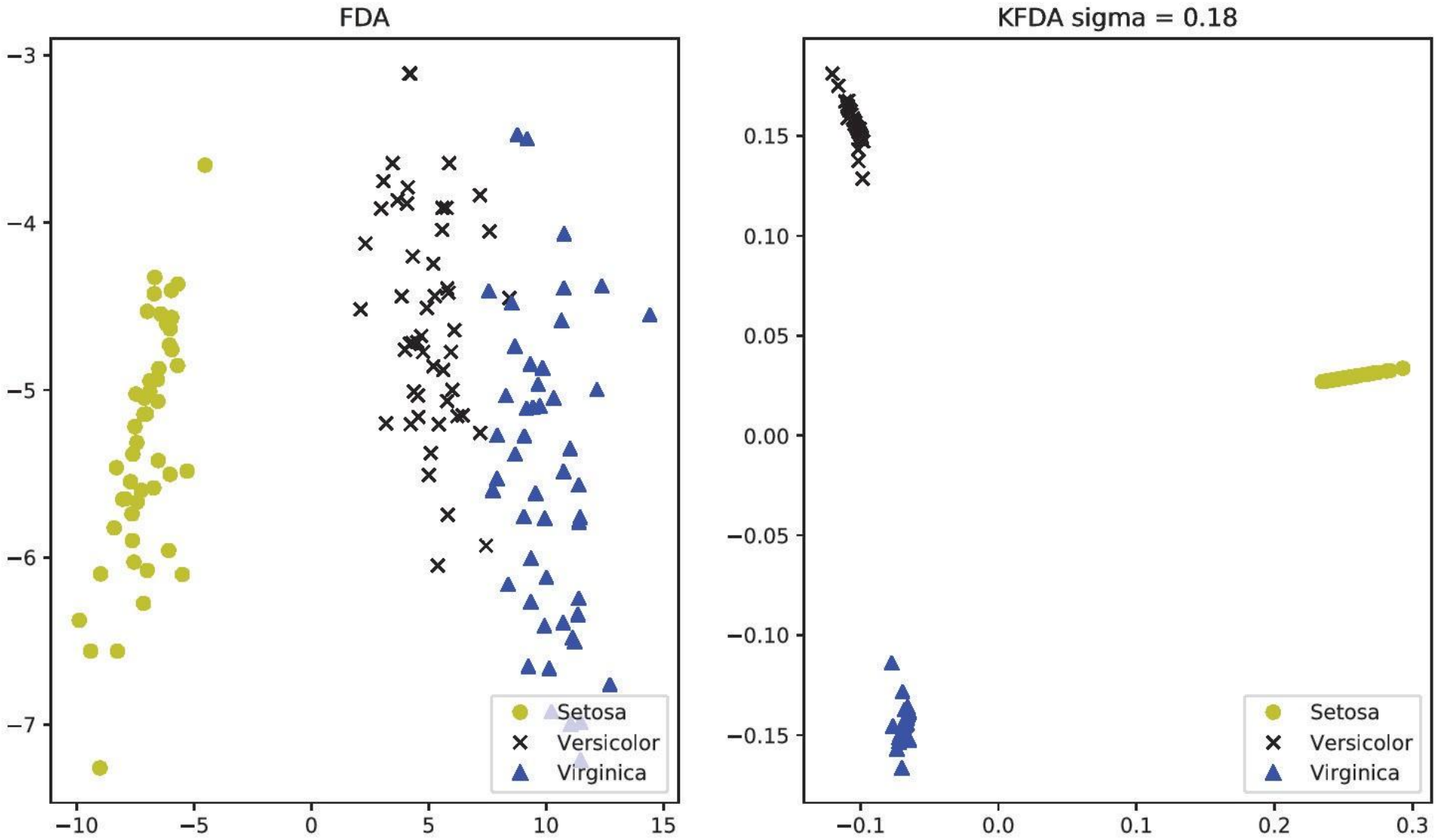
$$(w \cdot \Phi(x)) = K \cdot a$$

KFDA многоклассовый. Пример работы алгоритма Iris Dataset





KFDA многоклассовый. Пример работы алгоритма



Сравнительная таблица

Таблица 2: Dataset 1. Линейные классификаторы и QDA

Метод	Ошибка	Язык реализации
LDA	0.0227	MATLAB
SVM	0.0229	
QDA	0.02	
FDA	0.0209	Python 2.7

		Dataset2		***Iris***	
Метод	Реализцация	Пар-тр	**Ошибка**	**Ошибка**	Пар-тр
FDA	Python 2.7		0.3736	0.0137	
	MATLAB		0.41	0.0145	
QDA	MATLAB		0.091	0.0121	
KFDA (RBF)	Python 2.7	0.045	0.0034	0.0097	0.18
	MATLAB (SPRT lib)	0.045	0.0029	0.011	0.18
KSVM (RBF)	MATLAB	0.03	0.0054	0.0123	0.46

Применение

- **Распознавание лиц** FDA сокращает количество признаков. Создает тэмплэйты,состоящие из новых размерностей (линейных комбинаций значений пикселей, называемых *фишеровскими лицами*). Подробный алгоритм описан в статье.
- **Маркетинг** Разделение покупателей/товаров на группы на основании опросов или других форм сбора данных.
- **Медицина** Предсказание эффективности лечения, диагностика заболеваний:
  - Обработка и создание комплексов лабораторных тестов. Нахождение оптимальных комбинаций лабораторных тестов для диагностики заболевания.
  - Дифференциация между болезнями, схожими по проявлениям. Например, офтольмологическая проблема: отличие глаукомы от глазной гипертензии.
  - Предсказние доброкачественности/злокачественности опухоли по ее хар-кам, состоянию пациента.
  - Ранняя диагностика опухолей головного мозга. Выявление пораженных клеток по записям слуховых потенциалов ствола головного мозга.

Вывод FDA

Задача: Найти

$$\omega = \underset{\omega}{\operatorname{argmax}}\, I = \underset{\omega}{\operatorname{argmax}} \frac{\omega S_b \omega^T}{\omega S_w \omega^T}$$

Найдем точку экстремума функции: $\frac{d}{d\omega} J(\omega) =$
$= \frac{(\frac{d}{d\omega}\omega^T S_b \omega)\omega^T S_w \omega - (\frac{d}{d\omega}\omega^T S_w \omega)\omega^T S_b \omega}{(\omega^T S_w \omega)^2} = \frac{(2S_b\omega)\omega^T S_w \omega - (2S_w\omega)\omega^T S_b \omega}{(\omega^T S_w \omega)^2} = 0$
$\Rightarrow (S_b\omega)\omega^T S_w \omega - (S_w\omega)\omega^T S_b \omega = 0$
Разделим на $\omega^T S_w \omega$:
$S_b\omega - S_w\omega \cdot \frac{\omega^T S_b \omega}{\omega^T S_w \omega} = 0$
$\Rightarrow S_b\omega = I \cdot (S_w\omega)$
Если $\det(S_w) \neq 0$, то задача сводится к:

$$S_w^{-1} S_b \cdot \omega = I \cdot \omega$$

$\Rightarrow \omega = \operatorname{argmax}_\omega I$ - вектор, соответствующий максимальному собственному значению матрицы $(S_w^{-1} S_b)$ (т.к. значение I совпадает с собств знач.).

Переход от FDA к KFDA

$$k(x, y) = (\Phi(x) \cdot \Phi(y))$$

$$w = \underset{\omega}{\operatorname{argmax}} \frac{\omega S_b^{\Phi} \omega^T}{\omega S_w^{\Phi} \omega^T}$$

$$S_w = S_1 + S_2 = \sum_{i=1,2} \sum_{x \in X_i} (x_i - \mu_i)^T (x_i - \mu_i)$$

$$S_b = (\mu_1 - \mu_2)^T (\mu_1 - \mu_2)$$

Вектор w линейная оболочка векторов $\Phi(x_i)$:

$$\omega = \sum_{i=1}^{l} a_i \Phi(x_i)$$

$$m_i^{\Phi} \omega = \frac{1}{l_i} \sum_{j=1}^{l} \sum_{k=1}^{l_i} a_j k(x_j, x_k^i) = M_i a$$

$$\Rightarrow \omega S_b^{\Phi} \omega^T = aMa^T$$

Аналогично,

$$\omega S_w^{\Phi} \omega^T = aNa^T$$

где:

$$(M_i)_j = \frac{1}{l_i} \sum_{k=1}^{l_i} k(x_j, x_k^i)$$

$$M = (M_1 - M_2)^T (M_1 - M_2)$$

$$N = \sum_{j=1,2} K_j (Id - 1_{li}) K_j^T$$

Таким образом,

$$J(a) = \frac{aMa^T}{aNa^T}$$

$$(\omega, \Phi(x)) = \sum_{i=1}^{l} a_i k(x_i, x)$$

Список литературы

- Sebastian Mika "Fisher Discriminant analysis with Kernels."
- The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani (Раздел 4.3).
- Miller-Mika-Ratsch-Tsuda-Scholkopf "An Introduction to Kernel Based Learning Algorithms"
- Ядра и их применение в машинном обучении. Евгений Соколов.
- Statistical Patttern Recognition Toolbox for Matlab. User's guide.
- Wikipedia: KFDA и FDA.