

КЛАССИФИКАЦИЯ

Сергей Николенко



Академия больших данных MADE — VK

04 марта 2022 г.

Random facts:

- 04 марта 1733 г. был издан указ Анны Иоанновны «Об учреждении полиции в городах», согласно которому в 23 городах России были созданы полицмейстерские конторы
- 04 марта 1762 г., ровно через 29 лет после этого, император Пётр III подписал указ «О свободной для всех торговле»
- 04 марта 1803 г., ещё через 39 лет, император Александр I издал «Указ о вольных хлебопашцах», по которому землевладельцы получили право освобождать крестьян с обязательным наделением их землёй
- 04 марта 1837 г., ещё через 34 года и через 103 года после первого факта, Михаил Лермонтов был арестован за стихотворение «Смерть поэта»
- 04 марта 1990 г. на 5-летний срок был избран Съезд народных депутатов РСФСР, высший законодательный орган РСФСР; он был распущен указом Бориса Ельцина 21 сентября 1993 г. и разогнан вооружённой силой 4 октября 1993 г.
- 04 марта 2012 г. прошли выборы президента Российской Федерации; Владимир Путин был избран президентом России на третий срок

BIAS-VARIANCE-NOISE DECOMPOSITION

- Рассмотрим совместное распределение $p(y, \mathbf{x})$ и квадратичную функцию потерь $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$.
- Мы знаем, что тогда оптимальная оценка – это функция регрессии

$$\hat{f}(\mathbf{x}) = \mathbf{E}[y \mid \mathbf{x}] = \int y p(y \mid \mathbf{x}) dx.$$

- Давайте подсчитаем ожидаемую ошибку и перепишем её в другой форме:

$$\begin{aligned}\mathbf{E}[L] &= \mathbf{E}[(y - f(\mathbf{x}))^2] = \mathbf{E}[(y - \mathbf{E}[y | \mathbf{x}] + \mathbf{E}[y | \mathbf{x}] - f(\mathbf{x}))^2] = \\ &= \int (f(\mathbf{x}) - \mathbf{E}[y | \mathbf{x}])^2 p(\mathbf{x}) d\mathbf{x} + \int (\mathbf{E}[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy,\end{aligned}$$

потому что

$$\int (f(\mathbf{x}) - \mathbf{E}[y | \mathbf{x}]) (\mathbf{E}[y | \mathbf{x}] - y) p(\mathbf{x}, y) d\mathbf{x} dy = 0.$$

- Эта форма записи – разложение на bias-variance и noise:

$$\mathbf{E}[L] = \int (f(\mathbf{x}) - \mathbf{E}[y | \mathbf{x}])^2 p(\mathbf{x}) d\mathbf{x} + \int (\mathbf{E}[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy,$$

- Отсюда, кстати, тоже сразу видно, что от $f(\mathbf{x})$ зависит только первый член, и он минимизируется, когда

$$f(\mathbf{x}) = \hat{f}(\mathbf{x}) = \mathbf{E}[y | \mathbf{x}].$$

- А noise, $\int (\mathbf{E}[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy$, – это просто свойство данных, дисперсия шума.

- Если бы у нас был всемогущий компьютер и неограниченный датасет, мы бы, конечно, на этом и закончили, посчитали бы $\hat{f}(\mathbf{x}) = \mathbf{E}[y \mid \mathbf{x}]$, и всё.
- Однако жизнь – борьба, и у нас есть только ограниченный датасет из N точек. Предположим, что этот датасет берётся по распределению $p(\mathbf{x}, y)$ – т.е. фактически рассмотрим много-много экспериментов такого вида:
 - взяли датасет D из N точек по распределению $p(\mathbf{x}, y)$;
 - подсчитали нашу чудо-регрессию;
 - получили новую функцию предсказания $f(\mathbf{x}; D)$.
- Разные датасеты будут приводить к разным функциям предсказания...

- ...а потому давайте усредним теперь по датасетам.
- Наш первый член в ожидаемой ошибке выглядел как $(f(\mathbf{x}) - \hat{f}(\mathbf{x}))^2$, а теперь будет $(f(\mathbf{x}; D) - \hat{f}(\mathbf{x}))^2$, и его можно усреднить по D , применив такой же трюк:

$$\begin{aligned} & (f(\mathbf{x}; D) - \hat{f}(\mathbf{x}))^2 \\ &= (f(\mathbf{x}; D) - \mathbf{E}_D[f(\mathbf{x}; D)] + \mathbf{E}_D[f(\mathbf{x}; D)] - \hat{f}(\mathbf{x}))^2 \\ &= (f(\mathbf{x}; D) - \mathbf{E}_D[f(\mathbf{x}; D)])^2 + (\mathbf{E}_D[f(\mathbf{x}; D)] - \hat{f}(\mathbf{x}))^2 + 2(\dots)(\dots), \end{aligned}$$

и в ожидании получится...

- ...и в ожидании получится

$$\begin{aligned}\mathbf{E}_D \left[\left(f(\mathbf{x}; D) - \hat{f}(\mathbf{x}) \right)^2 \right] &= \\ &= \mathbf{E}_D \left[\left(f(\mathbf{x}; D) - \mathbf{E}_D [f(\mathbf{x}; D)] \right)^2 \right] + \left(\mathbf{E}_D [f(\mathbf{x}; D)] - \hat{f}(\mathbf{x}) \right)^2.\end{aligned}$$

- Разложили на дисперсию $\mathbf{E}_D \left[\left(f(\mathbf{x}; D) - \mathbf{E}_D [f(\mathbf{x}; D)] \right)^2 \right]$ и квадрат систематической ошибки $\left(\mathbf{E}_D [f(\mathbf{x}; D)] - \hat{f}(\mathbf{x}) \right)^2$; это и есть bias-variance decomposition.

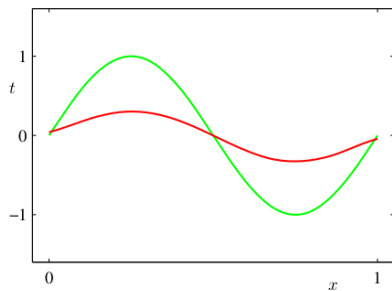
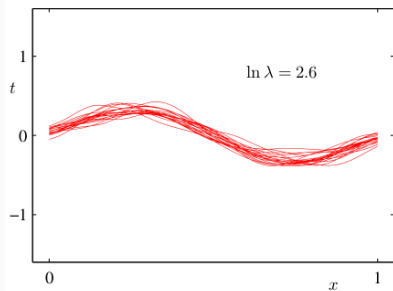
Expected loss = (bias)² + variance + noise,

где

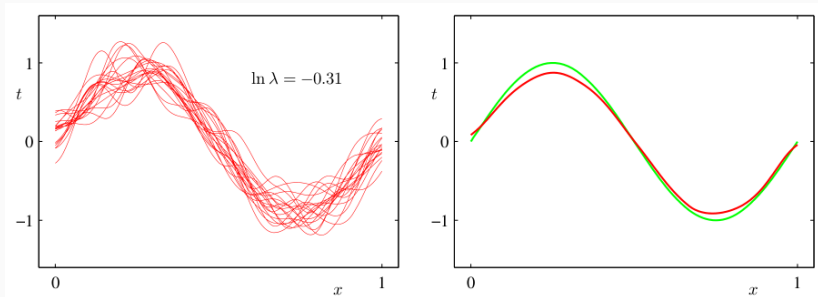
$$\begin{aligned}(\text{bias})^2 &= \left(\mathbf{E}_D [f(\mathbf{x}; D)] - \hat{f}(\mathbf{x}) \right)^2, \\ \text{variance} &= \mathbf{E}_D \left[(f(\mathbf{x}; D) - \mathbf{E}_D [f(\mathbf{x}; D)])^2 \right], \\ \text{noise} &= \int (\mathbf{E}[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy.\end{aligned}$$

- Теперь давайте посмотрим на пример: опять та же синусоида, опять приближаем её линейной регрессией с полиномиальными признаками (максимальным их числом).
- И мы регуляризуем эту регрессию с параметром α .
- Будем набрасывать много датасетов и смотреть, что меняется при этом.

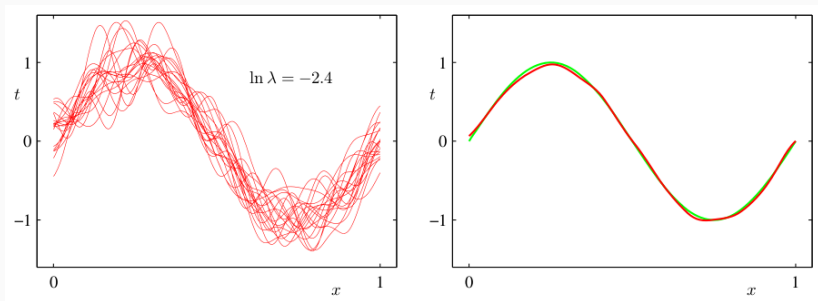
РЕГУЛЯРИЗАТОР И BIAS-VARIANCE

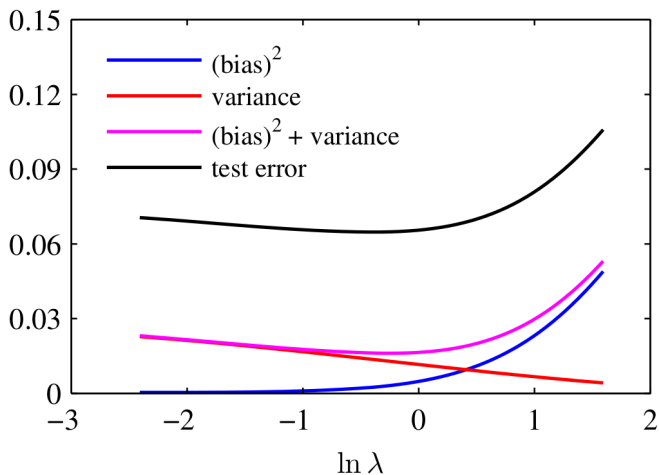


РЕГУЛЯРИЗАТОР И BIAS-VARIANCE



РЕГУЛЯРИЗАТОР И BIAS-VARIANCE





ВВЕДЕНИЕ В КЛАССИФИКАЦИЮ

- Теперь классификация: определить вектор $\mathbf{x}$ в один из K классов C_k .
- В итоге у нас так или иначе всё пространство разобьётся на эти классы.
- Т.е. на самом деле мы ищем *разделяющую поверхность* (decision surface, decision boundary).

- Как кодировать? Бинарная задача – очень естественно, переменная t , $t = 0$ соответствует C_1 , $t = 1$ соответствует C_2 .
- Оценку t можно интерпретировать как вероятность (по крайней мере, мы постараемся, чтобы было можно).
- Если несколько классов – удобно 1-of- K :

$$\mathbf{t} = (0, \dots, 0, 1, 0, \dots)^\top.$$

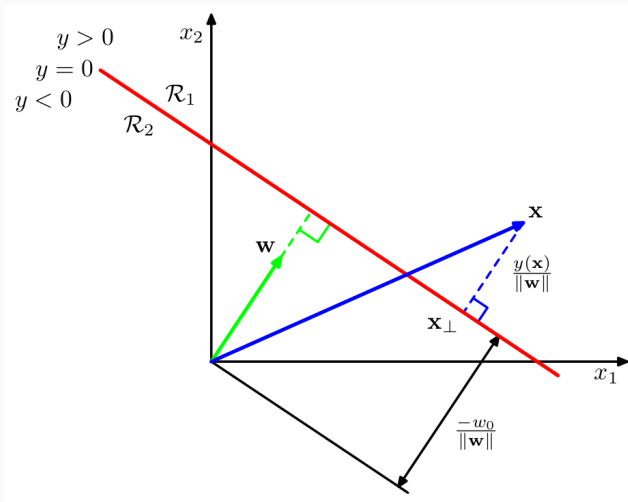
- Тоже можно интерпретировать как вероятности – или пропорционально им.

- Начнём с геометрии: рассмотрим линейную дискриминантную функцию

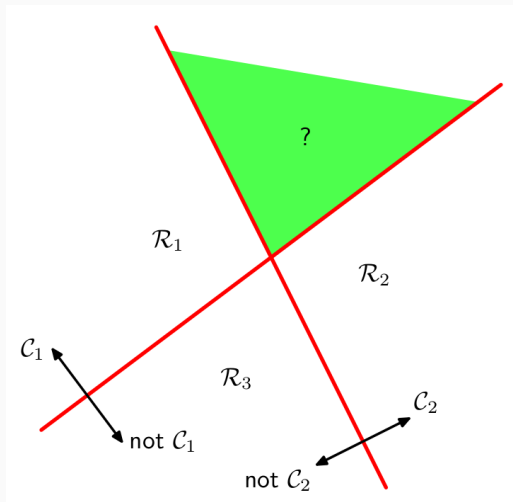
$$y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0.$$

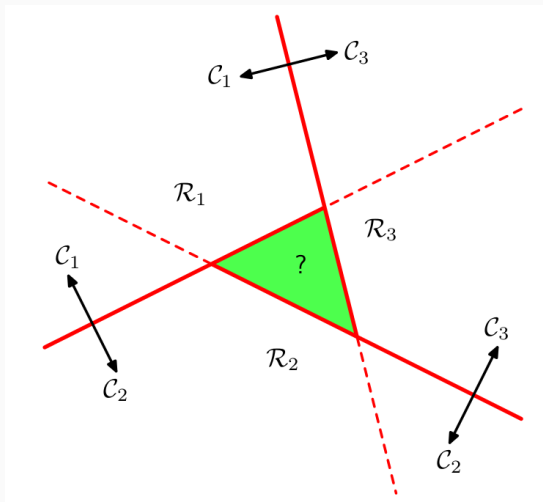
- Это гиперплоскость, и $\mathbf{w}$ – нормаль к ней.
- Расстояние от начала координат до гиперплоскости равно $\frac{-w_0}{\|\mathbf{w}\|}$.
- $y(\mathbf{x})$ связано с расстоянием до гиперплоскости: $d = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$.

РАЗДЕЛЯЮЩАЯ ГИПЕРПЛОСКОСТЬ



- С несколькими классами выходит задача.
- Можно рассмотреть K поверхностей вида «один против всех».
- Можно – $\binom{K}{2}$ поверхностей вида «каждый против каждого».
- Но всё это как-то нехорошо.



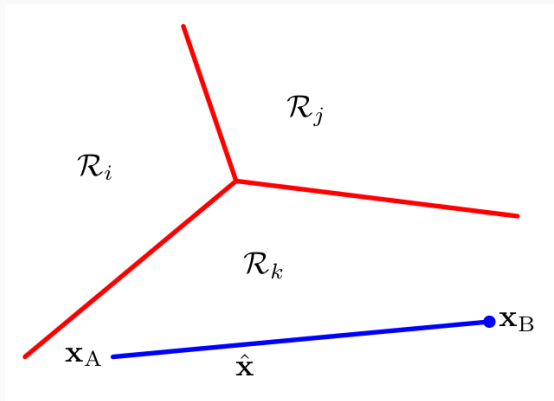


- Лучше рассмотреть единый дискриминант из K линейных функций:

$$y_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + w_{k0}.$$

- Классифицировать в C_k , если $y_k(\mathbf{x})$ – максимален.
- Тогда разделяющая поверхность между C_k и C_j будет гиперплоскостью вида $y_k(\mathbf{x}) = y_j(\mathbf{x})$:

$$(\mathbf{w}_k - \mathbf{w}_j)^\top \mathbf{x} + (w_{k0} - w_{j0}) = 0.$$



Упражнение. Докажите, что области, соответствующие классам, при таком подходе всегда односвязные и выпуклые.

- Мы снова можем воспользоваться методом наименьших квадратов: запишем $y_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + w_{k0}$ вместе (спрятав свободный член) как

$$\mathbf{y}(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}.$$

- Можно найти $\mathbf{W}$, оптимизируя сумму квадратов; функция ошибки:

$$E_D(\mathbf{W}) = \frac{1}{2} \text{Tr} [(\mathbf{XW} - \mathbf{T})^\top (\mathbf{XW} - \mathbf{T})].$$

- Берём производную, решаем...

- ...получается привычное

$$\mathbf{W} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{T} = \mathbf{X}^\dagger \mathbf{T},$$

где $\mathbf{X}^\dagger$ – псевдообратная Мура-Пенроуза.

- Теперь можно найти и дискриминантную функцию:

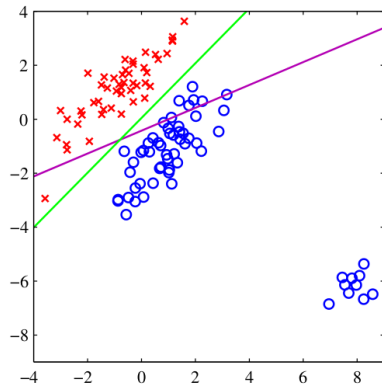
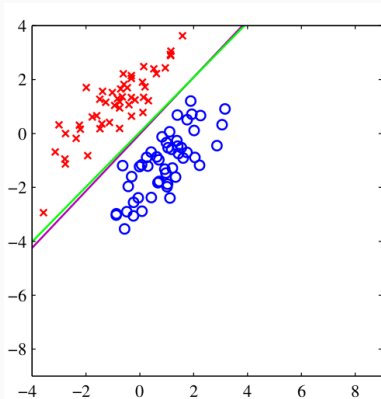
$$y(\mathbf{x}) = \mathbf{W}^\top \mathbf{x} = \mathbf{T}^\top (\mathbf{X}^\dagger)^\top \mathbf{x}.$$

- Это решение сохраняет линейность.

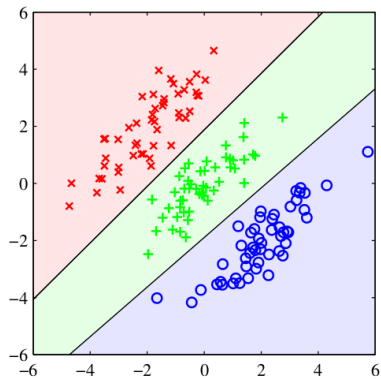
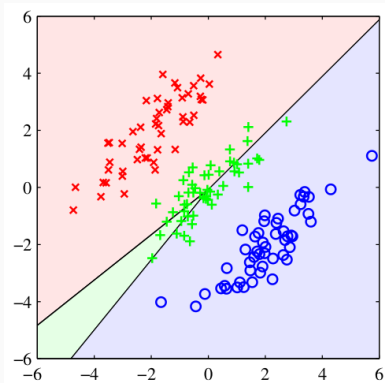
Упражнение. Докажите, что в схеме кодирования 1-of- K предсказания $y_k(\mathbf{x})$ для разных классов при любом $\mathbf{x}$ будут давать в сумме 1. Почему они всё-таки не будут разумными оценками вероятностей?

- Проблемы наименьших квадратов:
 - outliers плохо обрабатываются;
 - «слишком правильные» предсказания добавляют штраф.

ПРОБЛЕМЫ НАИМЕНЬШИХ КВАДРАТОВ



ПРОБЛЕМЫ НАИМЕНЬШИХ КВАДРАТОВ



ПРОБЛЕМЫ НАИМЕНЬШИХ КВАДРАТОВ

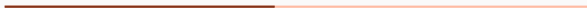
- Почему так? Почему наименьшие квадраты так плохо работают?

ПРОБЛЕМЫ НАИМЕНЬШИХ КВАДРАТОВ

- Почему так? Почему наименьшие квадраты так плохо работают?
- Они предполагают гауссовское распределение ошибки.
- Но, конечно, распределение у бинарных векторов далеко не гауссово.

ЛИНЕЙНЫЙ
ФИШЕРА

ДИСКРИМИНАНТ



- Другой взгляд на классификацию: в линейном случае мы хотим спроецировать точки в размерность 1 (на нормаль разделяющей гиперплоскости) так, чтобы в этой размерности 1 они хорошо разделялись.
- Т.е. классификация – это такой метод радикального сокращения размерности.
- Давайте посмотрим на классификацию с этих позиций и попробуем добиться оптимальности в каком-то смысле.

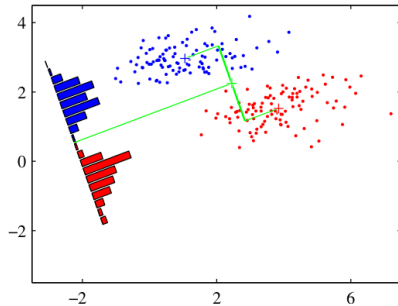
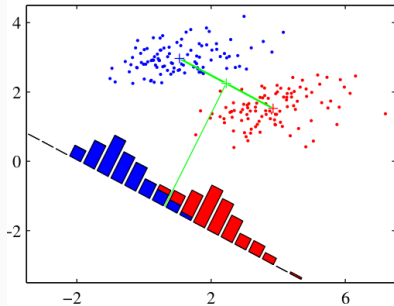
- Рассмотрим два класса C_1 и C_2 с N_1 и N_2 точками.
- Первая идея – надо найти серединный перпендикуляр между центрами кластеров

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{C_1} \mathbf{x}, \text{ и } \mathbf{m}_2 = \frac{1}{N_2} \sum_{C_2} \mathbf{x},$$

т.е. максимизировать $\mathbf{w}^\top (\mathbf{m}_2 - \mathbf{m}_1)$.

- Надо ещё добавить ограничение $\|\mathbf{w}\| = 1$, но всё равно не ахти как работает.

ЛИНЕЙНЫЙ ДИСКРИМИНАНТ ФИШЕРА



Чем левая картинка хуже правой?

- Слева больше дисперсия каждого кластера.
- Идея: минимизировать перекрытие классов, оптимизируя и проекцию расстояния, и дисперсию.
- Выборочные дисперсии в проекции: для $y_n = \mathbf{w}^\top \mathbf{x}_n$

$$s_1 = \sum_{n \in C_1} (y_n - m_1)^2 \text{ и } s_2 = \sum_{n \in C_2} (y_n - m_2)^2.$$

- Критерий Фишера:

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^\top \mathbf{S}_B \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_W \mathbf{w}}, \text{ где}$$

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\top,$$

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^\top + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^\top.$$

(between-class covariance и within-class covariance).

- Дифференцируя по $\mathbf{w}$...

- ...получим, что $J(\mathbf{w})$ максимален при

$$(\mathbf{w}^\top \mathbf{S}_B \mathbf{w}) \mathbf{S}_W \mathbf{w} = (\mathbf{w}^\top \mathbf{S}_W \mathbf{w}) \mathbf{S}_B \mathbf{w}.$$

- Т.к. $\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\top$, $\mathbf{S}_B \mathbf{w}$ всё равно будет в направлении $\mathbf{m}_2 - \mathbf{m}_1$, а длина $\mathbf{w}$ нас не интересует.
- Поэтому получается

$$\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1).$$

- В итоге мы выбрали направление проекции, и осталось только разделить данные на этой проекции.

- Любопытно, что дискриминант Фишера тоже можно получить из наименьших квадратов.
- Давайте для класса C_1 выберем целевое значение $\frac{N_1+N_2}{N_1}$, а для класса C_2 возьмём $-\frac{N_1+N_2}{N_2}$.

Упражнение. Докажите, что при таких целевых значениях наименьшие квадраты – это дискриминант Фишера.

- А что будет с несколькими классами? Рассмотрим $\mathbf{y} = \mathbf{W}^\top \mathbf{x}$, обобщим внутреннюю дисперсию как

$$\mathbf{S}_W = \sum_{k=1}^K \mathbf{S}_k = \sum_{k=1}^K \sum_{n \in C_k} (\mathbf{x}_n - \mathbf{m}_k) (\mathbf{x}_n - \mathbf{m}_k)^\top.$$

- Чтобы обобщить внешнюю (межклассовую) дисперсию, просто возьмём остаток полной дисперсии

$$\mathbf{S}_T = \sum_n (\mathbf{x}_n - \mathbf{m}) (\mathbf{x}_n - \mathbf{m})^\top,$$

$$\mathbf{S}_B = \mathbf{S}_T - \mathbf{S}_W.$$

- Обобщить критерий можно разными способами, например:

$$J(\mathbf{W}) = \text{Tr} [\mathbf{s}_W^{-1} \mathbf{s}_B],$$

где $\mathbf{s}$ – ковариации в пространстве проекций на $\mathbf{y}$:

$$\mathbf{s}_W = \sum_{k=1}^K \sum_{n \in C_k} (\mathbf{y}_n - \mu_k) (\mathbf{y}_n - \mu_k)^\top,$$

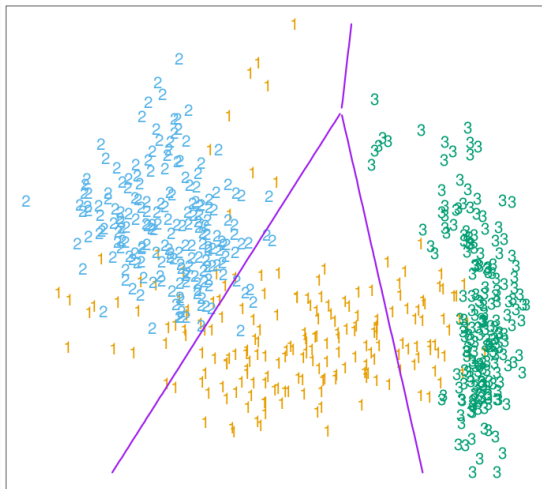
$$\mathbf{s}_B = \sum_{k=1}^K N_k (\mu_k - \mu) (\mu_k - \mu)^\top,$$

где $\mu_k = \frac{1}{N_k} \sum_{n \in C_k} \mathbf{y}_n$.

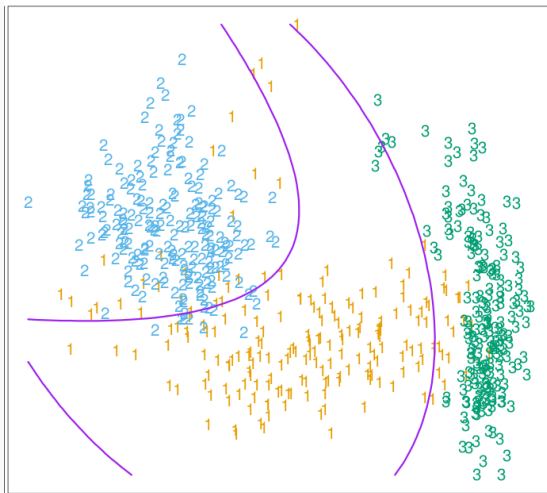
LDA и QDA

- Мы учились проводить разделяющие гиперплоскости.
- Но как же нелинейные поверхности?
- Можно делать нелинейные из линейных, увеличивая размерность.

НЕЛИНЕЙНЫЕ ПОВЕРХНОСТИ



НЕЛИНЕЙНЫЕ ПОВЕРХНОСТИ



- Теперь классификация через генеративные модели: давайте каждому классу сопоставим плотность $p(\mathbf{x} | C_k)$, найдём априорные распределения $p(C_k)$, будем искать $p(C_k | \mathbf{x})$ по теореме Байеса.
- Для двух классов:

$$p(C_1 | \mathbf{x}) = \frac{p(\mathbf{x} | C_1)p(C_1)}{p(\mathbf{x} | C_1)p(C_1) + p(\mathbf{x} | C_2)p(C_2)}.$$

- Перепишем:

$$p(C_1 | \mathbf{x}) = \frac{p(\mathbf{x} | C_1)p(C_1)}{p(\mathbf{x} | C_1)p(C_1) + p(\mathbf{x} | C_2)p(C_2)} = \frac{1}{1 + e^{-a}} = \sigma(a),$$

где

$$a = \ln \frac{p(\mathbf{x} | C_1)p(C_1)}{p(\mathbf{x} | C_2)p(C_2)}, \quad \sigma(a) = \frac{1}{1 + e^{-a}}.$$

- $\sigma(a)$ – логистический сигмоид:

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

- $\sigma(-a) = 1 - \sigma(a)$.
- $a = \ln\left(\frac{\sigma}{1-\sigma}\right)$ – логит-функция.

Упражнение. Докажите эти свойства.

- В случае нескольких классов получится

$$p(C_k | \mathbf{x}) = \frac{p(\mathbf{x} | C_k)p(C_k)}{\sum_j p(\mathbf{x} | C_j)p(C_j)} = \frac{e^{a_k}}{\sum_j e^{a_j}}.$$

- Здесь $a_k = \ln p(\mathbf{x} | C_k)p(C_k)$.
- $\frac{e^{a_k}}{\sum_j e^{a_j}}$ – нормализованная экспонента, или softmax-функция (сглаженный максимум).

- Давайте рассмотрим гауссовы распределения для классов:

$$p(\mathbf{x} \mid C_k) = N(\mathbf{x} \mid \mu_k, \Sigma).$$

- Сначала пусть Σ у всех одинаковые, а классов всего два.
- Посчитаем логистический сигмоид...

- ...получится

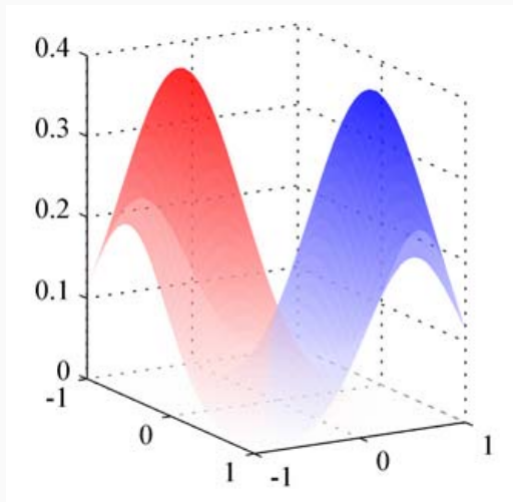
$$p(C_1 | \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + w_0), \text{ где}$$

$$\mathbf{w} = \Sigma^{-1} (\mu_1 - \mu_2),$$

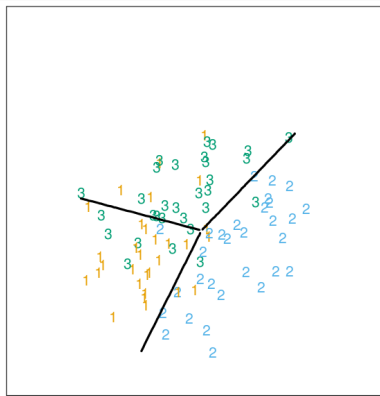
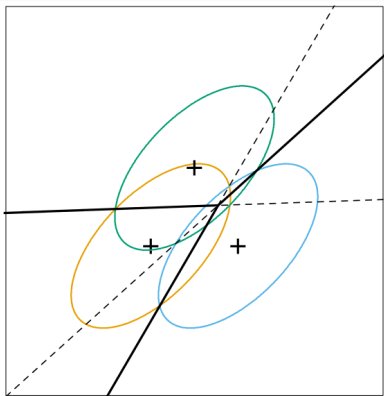
$$w_0 = -\frac{1}{2}\mu_1^\top \Sigma^{-1} \mu_1 + \frac{1}{2}\mu_2^\top \Sigma^{-1} \mu_2 + \ln \frac{p(C_1)}{p(C_2)}.$$

- Т.е. в аргументе сигмоида получается линейная функция от $\mathbf{x}$. Поверхности уровня – это когда $p(C_1 | \mathbf{x})$ постоянно, т.е. гиперплоскости в пространстве $\mathbf{x}$. Априорные вероятности $p(C_k)$ просто сдвигают эти гиперплоскости.

РАЗДЕЛЯЮЩАЯ ГИПЕРПЛОСКОСТЬ

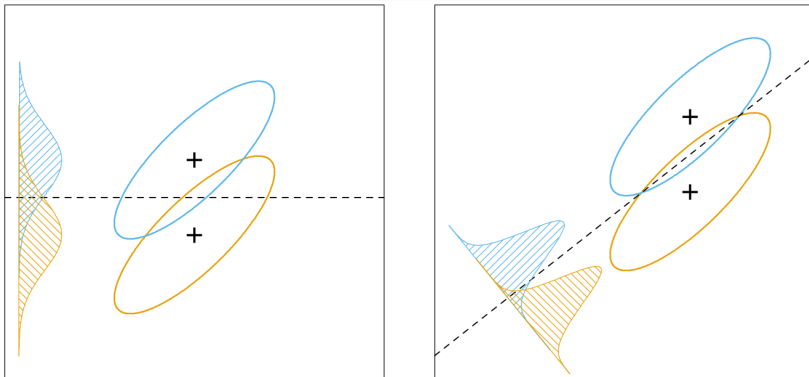


РАЗДЕЛЯЮЩАЯ ГИПЕРПЛОСКОСТЬ



ДИСКРИМИНАНТ ФИШЕРА

Кстати, с дискриминантом Фишера эта разделяющая поверхность отлично сходится.



- С несколькими классами получится тоже примерно так же:

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \ln \pi_k,$$

где $\pi_k = p(C_k)$.

- Получились линейные $\delta_k(\mathbf{x})$, и опять разделяющие поверхности линейные (тут разделяющие поверхности – когда две максимальных вероятности равны).
- Этот метод называется LDA – linear discriminant analysis.

- Как оценить распределения $p(\mathbf{x} \mid C_k)$, если даны только данные?
- Можно по методу максимального правдоподобия.
- Опять рассмотрим тот же пример: два класса, гауссианы с одинаковой матрицей ковариаций, и есть $D = \{\mathbf{x}_n, t_n\}_{n=1}^N$, где $t_n = 1$ значит C_1 , $t_n = 0$ значит C_2 .
- Обозначим $p(C_1) = \pi$, $p(C_2) = 1 - \pi$.

- Для одной точки в классе C_1 :

$$p(\mathbf{x}_n, C_1) = p(C_1)p(\mathbf{x}_n | C_1) = \pi N(\mathbf{x}_n | \mu_1, \Sigma).$$

- В классе C_2 :

$$p(\mathbf{x}_n, C_2) = p(C_2)p(\mathbf{x}_n | C_2) = (1 - \pi)N(\mathbf{x}_n | \mu_2, \Sigma).$$

- Функция правдоподобия:

$$\begin{aligned} p(\mathbf{t} | \pi, \mu_1, \mu_2, \Sigma) &= \\ &= \prod_{n=1}^N [\pi N(\mathbf{x}_n | \mu_1, \Sigma)]^{t_n} [(1 - \pi)N(\mathbf{x}_n | \mu_2, \Sigma)]^{1-t_n}. \end{aligned}$$

- Максимизируем логарифм правдоподобия. Сначала по π , там останется только

$$\sum_{n=1}^N [t_n \ln \pi + (1 - t_n) \ln(1 - \pi)],$$

и, взяв производную, получим, совершенно неожиданно,

$$\hat{\pi} = \frac{N_1}{N_1 + N_2}.$$

- Теперь по μ_1 ; всё, что зависит от μ_1 :

$$\sum_n t_n \ln N(\mathbf{x}_n \mid \mu_1, \Sigma) = -\frac{1}{2} \sum_n t_n (\mathbf{x}_n - \mu_1)^\top \Sigma^{-1} (\mathbf{x}_n - \mu_1) + C.$$

- Берём производную, и получается, опять внезапно,

$$\hat{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^N t_n \mathbf{x}_n.$$

- Аналогично,

$$\hat{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^N (1 - t_n) \mathbf{x}_n.$$

- Для матрицы ковариаций придётся постараться; в результате получится

$$\hat{\Sigma} = \frac{N_1}{N_1 + N_2} \mathbf{S}_1 + \frac{N_2}{N_1 + N_2} \mathbf{S}_2, \text{ где}$$
$$\mathbf{S}_1 = \frac{1}{N_1} \sum_{n \in C_1} (\mathbf{x}_n - \mu_1) (\mathbf{x}_n - \mu_1)^\top,$$
$$\mathbf{S}_2 = \frac{1}{N_2} \sum_{n \in C_2} (\mathbf{x}_n - \mu_2) (\mathbf{x}_n - \mu_2)^\top.$$

- Тоже совершенно неожиданно: взвешенное среднее оценок для двух матриц ковариаций.

- Это самым прямым образом обобщается на случай нескольких классов.

Упражнение. Сделайте это.

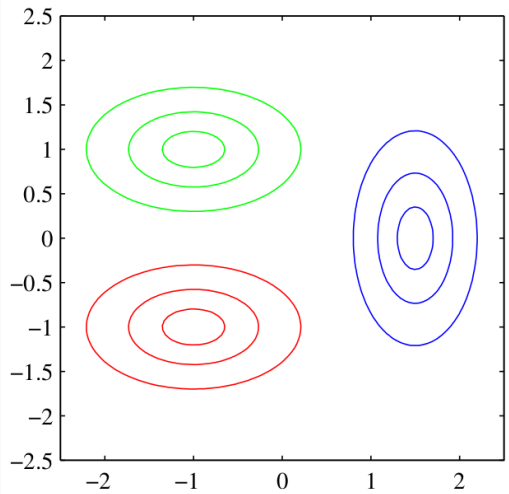
- А вот с разными матрицами ковариаций уже будет по-другому.
- Квадратичные члены не сократятся.
- Разделяющие поверхности станут квадратичными; QDA – quadratic discriminant analysis.

- В QDA получится

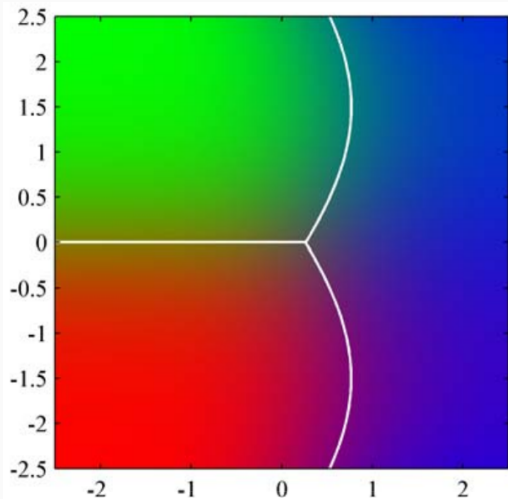
$$\delta_k(\mathbf{x}) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (\mathbf{x} - \mu_k)^{-1} \Sigma_k^{-1} (\mathbf{x} - \mu_k) + \log \pi_k.$$

- Разделяющая поверхность между C_i и C_j – это $\{\mathbf{x} \mid \delta_i(\mathbf{x}) = \delta_j(\mathbf{x})\}$.
- Оценки максимального правдоподобия такие же, только надо отдельно матрицы ковариаций оценивать.

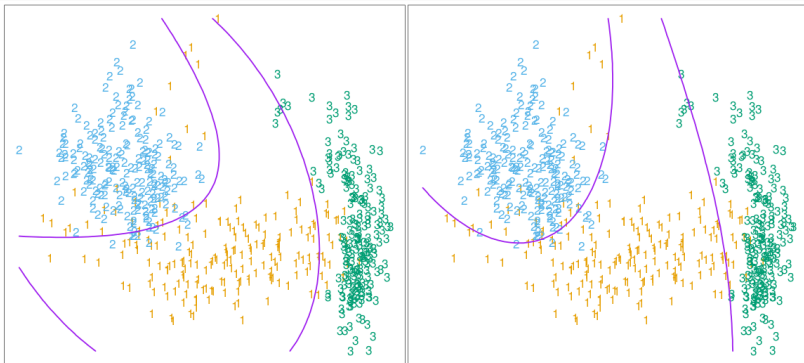
РАЗНЫЕ МАТРИЦЫ КОВАРИАЦИЙ



РАЗНЫЕ МАТРИЦЫ КОВАРИАЦИЙ



Разница между LDA с квадратичными членами и QDA обычно невелика.



- LDA и QDA неплохо работают на практике. Часто это первая идея в классификации.
- Число параметров:
 - у LDA $(K - 1)(d + 1)$ параметр: по $d + 1$ на каждую разницу вида $\delta_k(\mathbf{x}) - \delta_K(\mathbf{x})$;
 - у QDA $(K - 1)(d(d + 3)/2 + 1)$ параметр, но он выглядит гораздо лучше своих лет.

- Почему хорошо работают?
- Скорее всего, потому, что линейные и квадратичные оценки достаточно стабильны: даже если bias относительно большой (как будет, если данные всё-таки не гауссианами порождены), variance будет маленькой.

- Компромисс между LDA и QDA – регуляризованный дискриминантный анализ, RDA.
- Стянем ковариации каждого класса к общей матрице ковариаций:

$$\hat{\Sigma}_k(\alpha) = \alpha \hat{\Sigma}_k + (1 - \alpha) \hat{\Sigma},$$

где $\hat{\Sigma}_k$ – оценка из QDA, $\hat{\Sigma}$ – оценка из LDA.

- Или стянем к единичной матрице:

$$\hat{\Sigma}_k(\gamma) = \gamma \hat{\Sigma}_k + (1 - \gamma) \hat{\sigma}^2 I.$$

- Предположим, что размерность d больше, чем число классов K .
- Тогда центроиды классов $\hat{\mu}_k$ лежат в подпространстве размерности $\leq K - 1$.
- И когда мы определяем ближайший центроид, нам достаточно считать расстояния только в этом подпространстве.
- Таким образом, можно сократить ранг задачи.

- Куда именно проецировать? Не обязательно само подпространство, порождённое центроидами, будет оптимальным.
- Это мы уже проходили: для размерности 1 это линейный дискриминант Фишера.
- Это он и есть: оптимальное подпространство будет там, где межклассовая дисперсия максимальна по отношению к внутриклассовой.

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

- Итак, мы рассмотрели логистический сигмоид:

$$p(C_1 | \mathbf{x}) = \frac{p(\mathbf{x} | C_1)p(C_1)}{p(\mathbf{x} | C_1)p(C_1) + p(\mathbf{x} | C_2)p(C_2)} = \frac{1}{1 + e^{-a}} = \sigma(a),$$

$$\text{где } a = \ln \frac{p(\mathbf{x} | C_1)p(C_1)}{p(\mathbf{x} | C_2)p(C_2)}, \quad \sigma(a) = \frac{1}{1 + e^{-a}}.$$

- Вывели из него LDA и QDA, обучив их методом максимального правдоподобия.

- Два класса, и апостериорное распределение – логистический сигмоид на линейной функции:

$$p(C_1 | \phi) = y(\phi) = \sigma(\mathbf{w}^\top \phi), \quad p(C_2 | \phi) = 1 - p(C_1 | \phi).$$

- *Логистическая регрессия* – это когда мы напрямую оптимизируем $\mathbf{w}$.

- Для датасета $\{\phi_n, t_n\}$, $t_n \in \{0, 1\}$, $\phi_n = \phi(\mathbf{x}_n)$:

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^N y_n^{t_n} (1 - y_n)^{1-t_n}, \quad y_n = p(C_1 \mid \phi_n).$$

- Ищем параметры максимального правдоподобия, минимизируя $-\ln p(\mathbf{t} \mid \mathbf{w})$:

$$E(\mathbf{w}) = -\ln p(\mathbf{t} \mid \mathbf{w}) = -\sum_{n=1}^N [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)].$$

- Пользуясь тем, что $\sigma' = \sigma(1 - \sigma)$, берём градиент (похоже на перцептрон):

$$\nabla E(\mathbf{w}) = \sum_{n=1}^N (y_n - t_n) \phi_n.$$

- Если теперь сделать градиентный спуск, получим как раз разделяющую поверхность.
- Заметим, правда, что если данные действительно разделимы, то может получиться жуткий оверфиттинг: $\|\mathbf{w}\| \rightarrow \infty$, и сигмоид превращается в функцию Хевисайда. Надо регуляризовать.

- В логистической регрессии не получается замкнутого решения из-за сигмоида.
- Но функция $E(\mathbf{w})$ всё равно выпуклая, и можно воспользоваться методом Ньютона-Рапсона – на каждом шаге использовать локальную квадратичную аппроксимацию к функции ошибки:

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \mathbf{H}^{-1} \nabla E(\mathbf{w}),$$

где $\mathbf{H}$ (Hessian) – матрица вторых производных $E(\mathbf{w})$.

- Замечание: давайте применим Ньютона-Рапсона к обычной линейной регрессии с квадратической ошибкой:

$$\begin{aligned}\nabla E(\mathbf{w}) &= \sum_{n=1}^N (\mathbf{w}^\top \phi_n - t_n) \phi_n = \Phi^\top \Phi \mathbf{w} - \Phi^\top \mathbf{t}, \\ \nabla \nabla E(\mathbf{w}) &= \sum_{n=1}^N \phi_n \phi_n^\top = \Phi^\top \Phi,\end{aligned}$$

и шаг оптимизации будет

$$\begin{aligned}\mathbf{w}^{\text{new}} &= \mathbf{w}^{\text{old}} - (\Phi^\top \Phi)^{-1} [\Phi^\top \Phi \mathbf{w}^{\text{old}} - \Phi^\top \mathbf{t}] = \\ &= (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{t},\end{aligned}$$

т.е. мы за один шаг придём к решению.

- Для логистической регрессии:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^N (y_n - t_n) \phi_n = \Phi^\top (\mathbf{y} - \mathbf{t}),$$

$$\mathbf{H} = \nabla \nabla E(\mathbf{w}) = \sum_{n=1}^N y_n (1 - y_n) \phi_n \phi_n^\top = \Phi^\top R \Phi$$

для диагональной матрицы R с $R_{nn} = y_n(1 - y_n)$.

- Формула шага оптимизации:

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - (\Phi^{\top} R \Phi)^{-1} \Phi^{\top} (\mathbf{y} - \mathbf{t}) = (\Phi^{\top} R \Phi)^{-1} \Phi^{\top} R \mathbf{z},$$

где $\mathbf{z} = \Phi \mathbf{w}^{\text{old}} - R^{-1} (\mathbf{y} - \mathbf{t})$.

- Получилось как бы решение взвешенной задачи минимизации квадратического отклонения с матрицей весов R .
- Отсюда название: iterative reweighted least squares (IRLS).

СПАСИБО!

Спасибо за внимание!