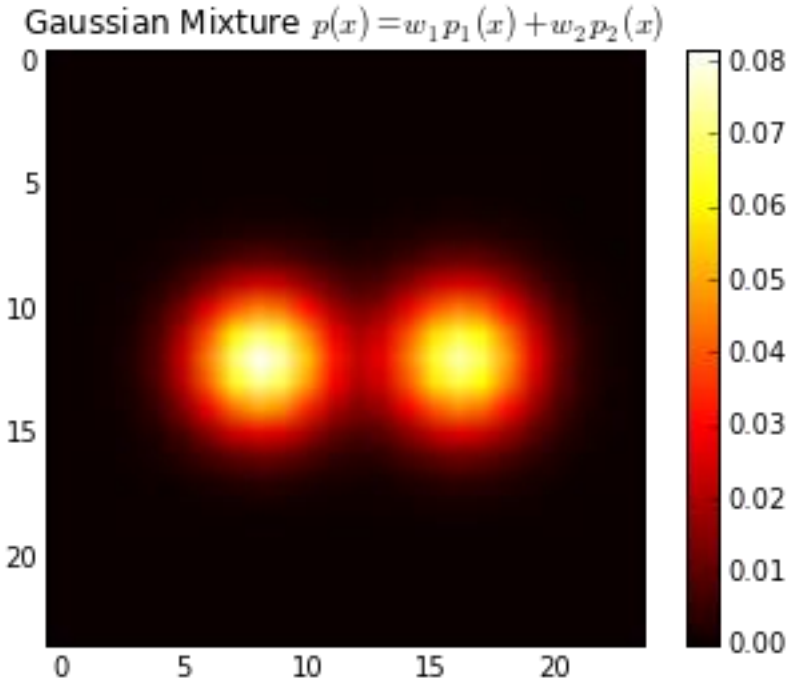
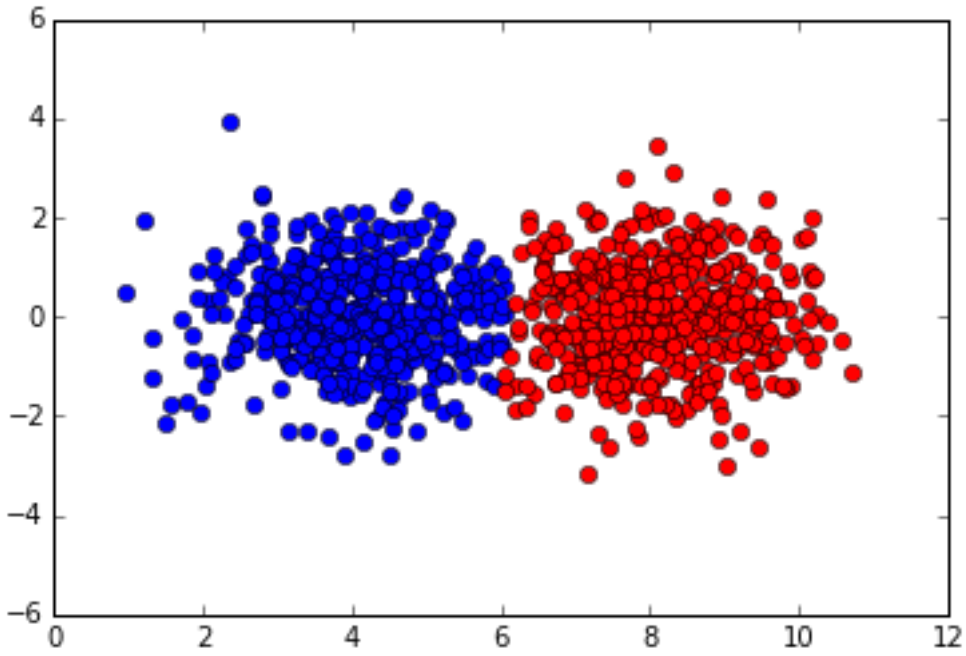




Анализ данных на практике

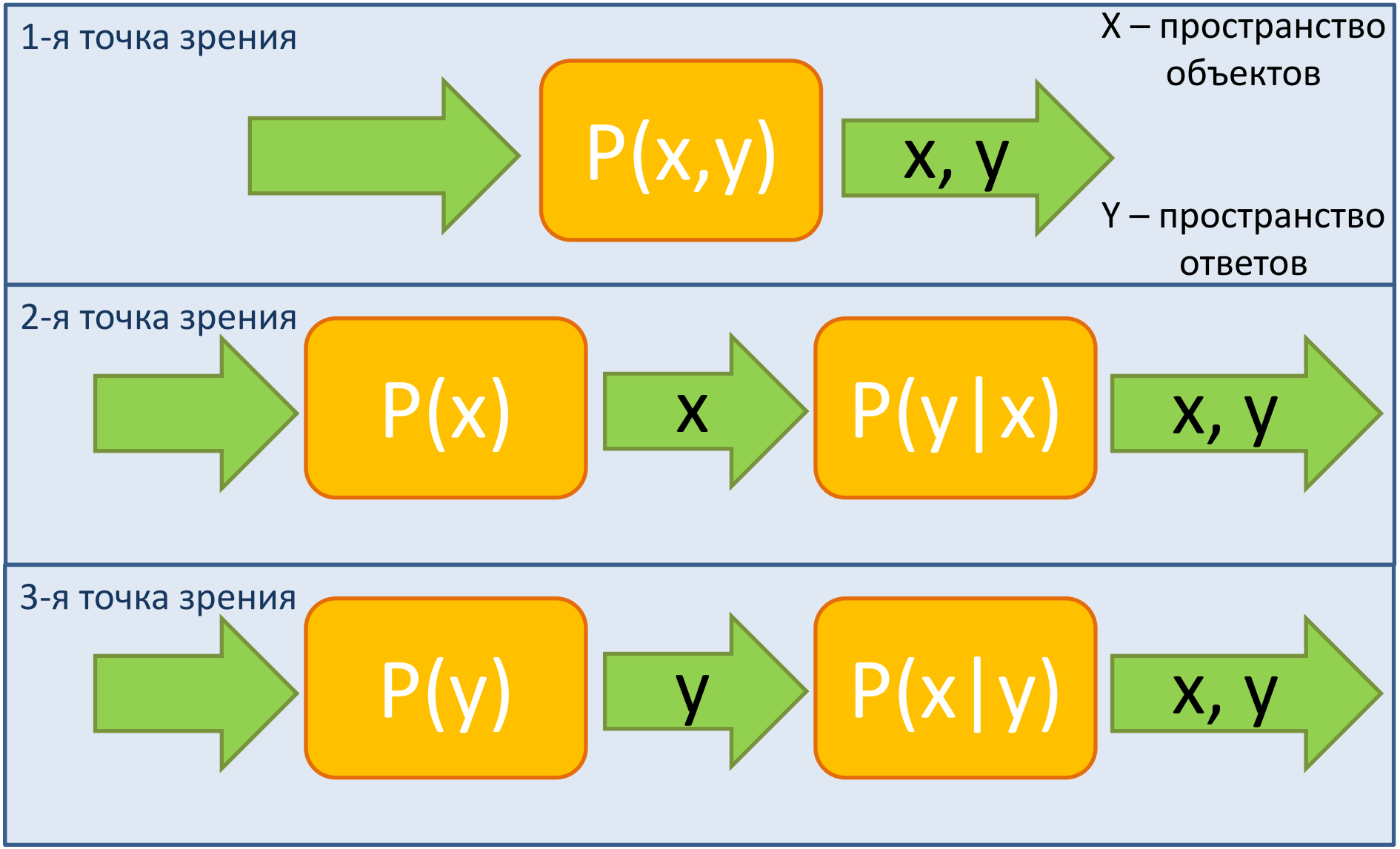
Байесовские методы классификации и регрессии

Виктор Кантор

Байесовские методы

1. Наивный Байес и его истоки: фильтр спама
2. Восстановление плотности распределения
3. Функции потерь и функционалы риска

Вероятностная модель данных



1. Наивный Байес

Простая идея фильтрации спама

Примеры спама:

- “Hi! :) **Purchase Exclusive** Tabs Online **http://**...”

- “We **Offer** Loan At A **Very Low Rate** Of 3%. If **Interested**, Kindly Contact Us,Reply by this email **….@hotmail.com**”

Простая идея фильтрации спама

1. Посчитать для слова **w** количество его вхождений $\mathbf{n_{ws}}$ в спам (spam) и $\mathbf{n_{wh}}$ в не спам (ham)

2. Вероятность появления слова **w**:

   $\mathbf{P(w|spam) = n_{ws} / \sum_{w'} n_{w's}}$

   $\mathbf{P(w|ham) = n_{wh} / \sum_{w'} n_{w'h}}$

3. «Наивное» предположение:

   $\mathbf{P(text|C) = P(w_1|C)\ P(w_2|C)\ \ldots\ P(w_N|C)}$

   C – “spam” или “ham”

4. $\mathbf{a(new\ text) = argmax_C\ P(new\ text|C)}$

Простая идея фильтрации спама

Но это же неправильно!

~~**a(new text) = argmax$_c$ P(new text|C)**~~

Мы уже знаем какие слова вошли в новый текст, значит нам нужна вероятность **P(C|new text)**

a(new text) = argmax$_C$ P(C|new text)

Однако по теореме Байеса:

argmax$_C$ P(C|new text) = argmax$_C$ P(new text|C)

если **P(spam) = P(ham)**

Наивный байесовский классификатор

1. Байесовский классификатор:

   $$a(x) = \operatorname{argmax}_C P(C|x) = \operatorname{argmax}_C P(x|C)\,P(C)$$

   (Теорема Байеса: $P(C|x) = P(x|C)\,P(C) / P(x)$)

2. С «наивной» гипотезой:

   $$P(x|C) = P(f_1(x)|C)\,P(f_2(x)|C)\,\ldots\,P(f_N(x)|C)$$

   $f_i(x)$ – i-ый признак объекта x

Сглаживание оценок вероятностей

Если $\mathbf{n_{ws} = n_{wh} = 0}$, $\mathbf{P(w|C) = 0}$?

Когда мы ничего не знаем про слово, можно предположить, что оно характеризует каждый класс с равными вероятностями:

$$\mathbf{P(w|spam) = (n_{ws} + 1)/ (n_s + 2)}$$

$$\mathbf{P(w|ham) = (n_{ws} + 1)/ (n_h + 2)}$$

Такая более устойчивая оценка называется сглаживанием Лапласа (*Laplas smoothing)*

Общая формула: $\mathbf{P(f|c) = (1 + n_{fc})/ (|C| + n_c)}$

2. Восстановление плотности

Случай дискретных признаков

Для бинарных признаков была интуитивная оценка:

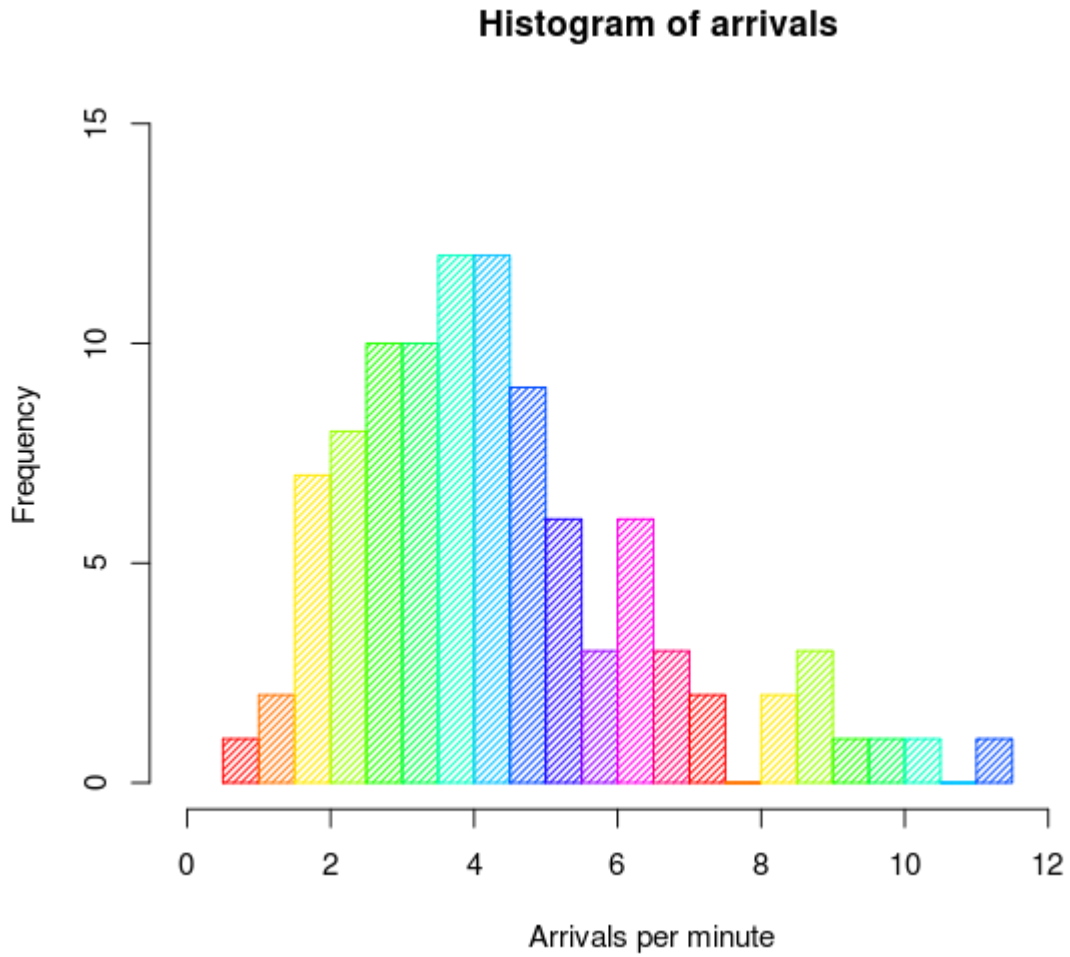
$$P(f|c) \approx n_{fc} / n_c$$

n_{fc} – точное число примеров в классе **c**, в которых признак **f = 1**

А если f – дискретный признак, принимающий значение из множества {f(1), f(2), …, f(k)} ?

Простая идея обобщения - гистограммы.

Случай дискретных признаков



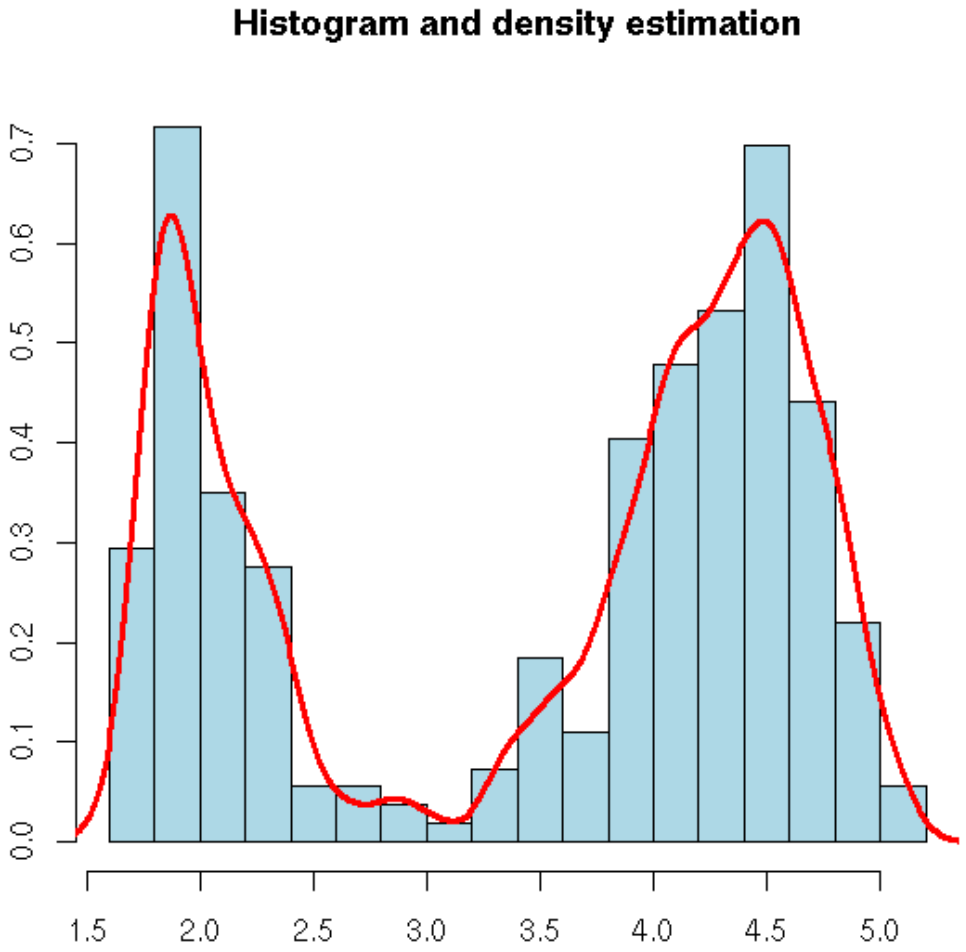
Дискретный признак сведется к бинарным – попаданиям в секции гистограммы.

Непрерывные (вещественные) признаки

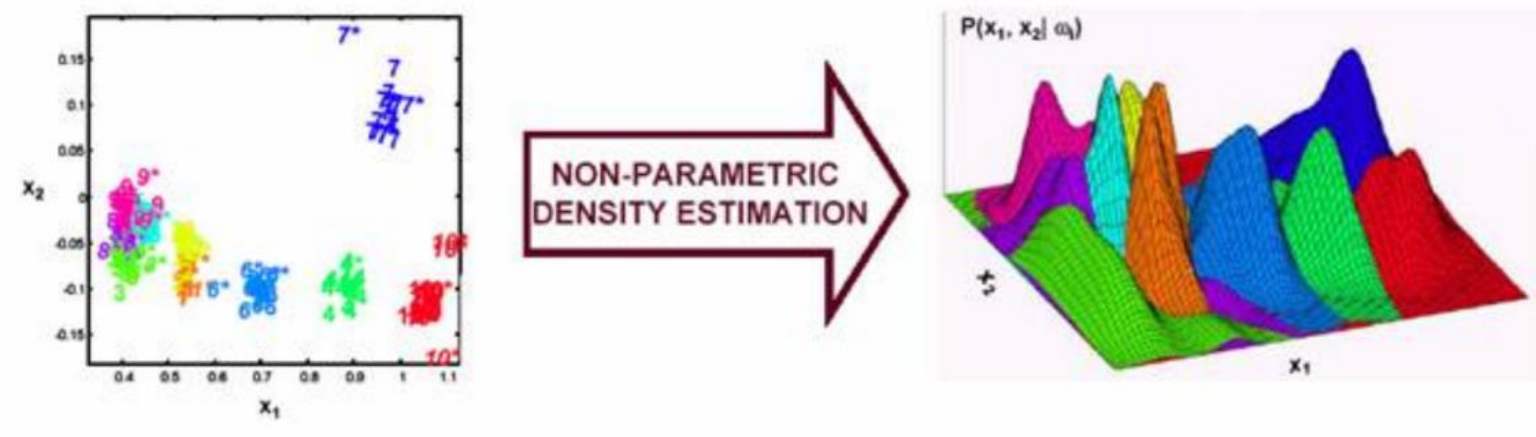
Что же делать с признаками, принимающими действительные значения? (0.2, 0.7, 0.333…)

- Сделать их дискретными ☺
- И сгладить

Непараметрическое восстановление плотности



Ядерные оценки плотности



$$\hat{p}_h(x) = \frac{1}{mV(h)} \sum_{i=1}^{m} K(\frac{\rho(x, x_i)}{h})$$

Параметрическое восстановление плотности

Другой способ – выбрать одно семейство распределений (например, нормальное распределение) и оценить параметры Θ_{ci} распределения каждого класса:

$$P(f_i(x) = t \mid c) = \varphi(\Theta_{ci}; t)$$

Для оценки Θ_c часто используется метод *максимального правдоподобия:*

$$\Theta_c = \operatorname{argmax}_\Theta \varphi(\Theta; f_i(x_{c1})) \varphi(\Theta; f_i(x_{c2})) \dots \varphi(\Theta; f_i(x_{cm}))$$

x_{ck} - **k**-ый пример из класса **c** в обучающей выборке

Рекомендации по выбору распределения

- Тексты/другие данные с разреженными дискретными признаками – мультиномиальное распределение
- Непрерывные признаки с маленьким разбросом – нормальное распределение
- Непрерывные признаки с выбросами в обучающей выборке – распределения с «тяжелыми хвостами»

Байесовский классификатор без наивной гипотезы

- Обобщение методов восстановления плотности – параметрическое и непараметрическое восстановление плотности в многомерном пространстве
- Параметрическое восстановление:
  - Вместо представления $\mathbf{P(f_i(x) = t \mid c) = \varphi(\Theta_{ci}; t)}$ используется $\mathbf{P(f(x) = t \mid c) = \varphi(\Theta_c; t)}$, где **f(x) – вектор** признаков
- Непараметрические оценки:
  - Обобщение сглаживания гистограмм

3. Функции потерь и функционалы риска

Обобщение: функции потерь

- Пусть алгоритм a дает прогноз $\hat{y} = a(x)$ в задаче регрессии или классификации, и $\hat{y}$ отличается от правильного ответа y. Допустим, нам стал известен y. Как измерить различие между y и $\hat{y}$?
- Базовые идеи: $(\hat{y} - y)^2$ или $|\hat{y} - y|$ для регрессии, $\lambda_{y\hat{y}}$ (costs matrix) или $\lambda_y[y \neq \hat{y}]$ или $[y \neq \hat{y}]$ для классификации
- В общем случае: $\mathrm{L}(\hat{y}, y)$ - функция потерь

Функционал риска

$R(a(x), x) = \boldsymbol{E}\big(L(a(x), y)\big|x\big)$ – матожидание штрафа

(нам известен алгоритм и x, мы хотим оценить ожидаемый штраф на объекте x для этого алгоритма)

В задаче классификации:

$\boldsymbol{E}\big(L(a(x), y)\big|x\big) = \sum_y L(a(x), y)\, P(y|x)$

В задаче регрессии:

$\boldsymbol{E}\big(L(a(x), y)\big|x\big) = \int_y L(a(x), y)\, dP(y|x)$

Байесовский классификатор с функцией потерь

$$a(x) = argmin_s\, R(s, x)$$

$$a(x) = argmin_s \sum_y L(s, y)\, P(y) P(x|y)$$

Если $L(s, y) = \lambda_y [y \neq \hat{y}]$:

$$a(x) = argmax_y\ \lambda_y P(y) P(x|y)$$

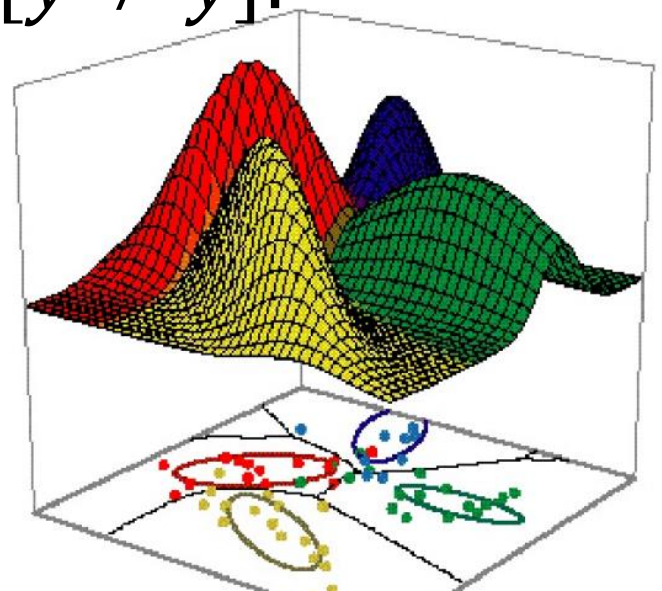
Байесовский классификатор с функцией потерь

$$a(x) = argmin_s\, R(s, x)$$

$$a(x) = argmin_s \sum_y L(s, y)\, P(y) P(x|y)$$

Если $P(y) = const,\ L(s, y) = [y \neq \hat{y}]$:

$$a(x) = argmax_y P(x|y)$$



Байесовская регрессия с функцией потерь

$$a(x) = argmin_s\, R(s, x)$$

$$a(x) = argmin_s \int_y L(a(x), y)\, dP(y|x)$$

Если $\mathrm{L(s, y)} = (\hat{y} - y)^2$:

$$a(x) = \int_y y\, dP(y|x)$$

Функционал среднего риска

$$R(a) = \boldsymbol{E}L(a(x), y) = \boldsymbol{E}R(a(x), x)$$

Теорема

$a(x) = argmin_s\, R(s, x)$ минимизирует R(a)

Доказательство:

Очевидно.