Машинное обучение

Лекция 8. Линейные модели: дополнительные темы

Подробнее об оптимизационных задачах, методах оптимизации и библиотеках

Виктор Кантор

План

I. Методы оптимизации в логистической регрессии и SVM
II. Библиотека Vowpal Wabbit
III. Semi-supervised модификации SVM и логистической регрессии
IV. Многоклассовая логистическая регрессия
V. Многоклассовый SVM
VI. Модуль linear_model в sklearn

I. Методы оптимизации в логистической регрессии и SVM

Напоминание: логистическая регрессия

$y_i \in \{0, 1\}$

$$Q = -\sum_{i=1}^{\ell} y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \to \min_w$$

$$p_i = \sigma(\langle w, x_i \rangle) = \frac{1}{1 + e^{-\langle w, x_i \rangle}}$$

Напоминание: логистическая регрессия

$y_i \in \{0, 1\}$

$$Q = -\sum_{i=1}^{\ell} y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \to \min_w$$

$$p_i = \sigma(\langle w, x_i \rangle) = \frac{1}{1 + e^{-\langle w, x_i \rangle}} = P(y = 1 | x)$$

Напоминание: логистическая регрессия

$$y_i \in \{0, 1\} \qquad Q = -\sum_{i=1}^{\ell} y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \to \min_w$$

$$p_i = \sigma(\langle w, x_i \rangle) = \frac{1}{1 + e^{-\langle w, x_i \rangle}} = P(y = 1|x)$$

Как правило, добавляется ℓ_1 или ℓ_2-регуляризация, а оптимизационная задача решается с помощью SGD или метода Ньютона-Рафсона

Эквивалентность оптимизационных задач

$$Q = -\sum_{i=1}^{\ell} y_i \ln \frac{1}{1 + e^{-\langle w, x_i \rangle}} + (1 - y_i) \ln \frac{1}{1 + e^{\langle w, x_i \rangle}} \to \min_w$$

$$y_i \in \{0, 1\}$$

$$Q = \sum_{i=1}^{\ell} \underbrace{\ln\left(1 + e^{-y_i \langle w, x_i \rangle}\right)}_{L(M) = \ln(1 + e^{-M_i})} \to \min_w \qquad y_i \in \{-1, 1\}$$

Методы оптимизации

- SGD
- Метод Ньютона-Рафсона
- IRLS

SGD в логистической регрессии

$$Q = \sum_{i=1}^{\ell} \ln\left(1 + e^{-y_i\langle w, x_i\rangle}\right) \to \min_w$$

$$\frac{\partial Q}{\partial w} = \sum_{i=1}^{\ell} \frac{-y_i x_i e^{-y_i\langle w, x_i\rangle}}{1 + e^{-y_i\langle w, x_i\rangle}}$$

$$w^{(t+1)} = w^{(t)} - \eta_t \frac{-y_i x_i e^{-y_i\langle w, x_i\rangle}}{1 + e^{-y_i\langle w, x_i\rangle}}$$

Метод Ньютона-Рафсона для логистической регрессии

$$Q(w) = \sum_{i=1}^{\ell} \ln\Bigl(1 + \exp(-w^{\intercal} x_i y_i)\Bigr) = -\sum_{i=1}^{\ell} \ln \sigma\bigl(w^{\intercal} x_i y_i\bigr) \to \min_{w}$$

Метод Ньютона-Рафсона для логистической регрессии

$$Q(w) = \sum_{i=1}^{\ell} \ln\Big(1 + \exp(-w^{\mathsf{T}} x_i y_i)\Big) = -\sum_{i=1}^{\ell} \ln \sigma\big(w^{\mathsf{T}} x_i y_i\big) \to \min_w$$

$$\sigma(z) = (1 + e^{-z})^{-1} \qquad \sigma'(z) = \sigma(z)(1 - \sigma(z))$$

Метод Ньютона-Рафсона для логистической регрессии

$$Q(w) = \sum_{i=1}^{\ell} \ln\Big(1 + \exp(-w^{\intercal} x_i y_i)\Big) = -\sum_{i=1}^{\ell} \ln \sigma\big(w^{\intercal} x_i y_i\big) \to \min_{w}$$

$$\sigma(z) = (1 + e^{-z})^{-1} \qquad \sigma'(z) = \sigma(z)(1 - \sigma(z))$$

$$w^{t+1} := w^{t} - h_t\big(Q''(w^t)\big)^{-1} Q'(w^t)$$

Метод Ньютона-Рафсона для логистической регрессии

$$w^{t+1} := w^t - h_t\big(Q''(w^t)\big)^{-1}Q'(w^t)$$

$$\sigma_i = \sigma\big(y_i w^\intercal x_i\big)$$

Метод Ньютона-Рафсона для логистической регрессии

$$w^{t+1} := w^t - h_t\big(Q''(w^t)\big)^{-1}Q'(w^t)$$

$$\sigma_i = \sigma\big(y_i w^{\mathsf{T}} x_i\big) \qquad \frac{\partial Q(w)}{\partial w_j} = -\sum_{i=1}^{\ell}(1-\sigma_i)y_i f_j(x_i)$$

Метод Ньютона-Рафсона для логистической регрессии

$$w^{t+1} := w^t - h_t \big(Q''(w^t)\big)^{-1} Q'(w^t)$$

$$\sigma_i = \sigma(y_i w^\intercal x_i) \qquad \frac{\partial Q(w)}{\partial w_j} = -\sum_{i=1}^{\ell} (1 - \sigma_i) y_i f_j(x_i)$$

$$\frac{\partial^2 Q(w)}{\partial w_j \partial w_k} = -\frac{\partial}{\partial w_k} \sum_{i=1}^{\ell} (1 - \sigma_i) y_i f_j(x_i) =$$

Метод Ньютона-Рафсона для логистической регрессии

$$w^{t+1} := w^t - h_t\big(Q''(w^t)\big)^{-1}Q'(w^t)$$

$$\sigma_i = \sigma\big(y_i w^\mathsf{T} x_i\big) \qquad \frac{\partial Q(w)}{\partial w_j} = -\sum_{i=1}^{\ell}(1-\sigma_i)y_i f_j(x_i)$$

$$\frac{\partial^2 Q(w)}{\partial w_j \partial w_k} = -\frac{\partial}{\partial w_k}\sum_{i=1}^{\ell}(1-\sigma_i)y_i f_j(x_i) =$$

$$= \sum_{i=1}^{\ell}(1-\sigma_i)\sigma_i f_j(x_i) f_k(x_i)$$

IRLS (Itreratively Reweighted Least Squares)

$$F_{\ell\times n} = \big(f_j(x_i)\big) \qquad \Gamma_{\ell\times\ell} = \mathrm{diag}\Big(\sqrt{(1-\sigma_i)\sigma_i}\,\Big)$$

$$\tilde{y}_i = y_i\sqrt{(1-\sigma_i)/\sigma_i} \qquad \tilde{F} = \Gamma F \qquad \tilde{y} = (\tilde{y}_i)_{i=1}^{\ell}$$

IRLS (Itreratively Reweighted Least Squares)

$$F_{\ell \times n} = \big(f_j(x_i)\big) \qquad \Gamma_{\ell \times \ell} = \mathrm{diag}\Big(\sqrt{(1-\sigma_i)\sigma_i}\,\Big)$$

$$\tilde{y}_i = y_i \sqrt{(1-\sigma_i)/\sigma_i} \qquad \tilde{F} = \Gamma F \qquad \tilde{y} = (\tilde{y}_i)_{i=1}^{\ell}$$

$$\big(Q''(w)\big)^{-1} Q'(w) = -(F^\intercal \Gamma^2 F)^{-1} F^\intercal \Gamma \tilde{y} = -(\tilde{F}^\intercal \tilde{F})^{-1} \tilde{F}^\intercal \tilde{y} = -\tilde{F}^{+} \tilde{y}$$

IRLS (Itreratively Reweighted Least Squares)

$$F_{\ell \times n} = \big(f_j(x_i)\big) \qquad \Gamma_{\ell \times \ell} = \operatorname{diag}\big(\sqrt{(1-\sigma_i)\sigma_i}\,\big)$$

$$\tilde{y}_i = y_i \sqrt{(1-\sigma_i)/\sigma_i} \qquad \tilde{F} = \Gamma F \qquad \tilde{y} = (\tilde{y}_i)_{i=1}^{\ell}$$

$$\big(Q''(w)\big)^{-1} Q'(w) = -(F^{\intercal}\Gamma^2 F)^{-1} F^{\intercal}\Gamma\tilde{y} = -(\tilde{F}^{\intercal}\tilde{F})^{-1}\tilde{F}^{\intercal}\tilde{y} = -\tilde{F}^{+}\tilde{y}$$

$$Q(w) = \|\tilde{F}w - \tilde{y}\|^2 = \sum_{i=1}^{\ell} \underbrace{(1-\sigma_i)\sigma_i}_{\gamma_i} \big(w^{\intercal}x - \underbrace{y_i\sqrt{(1-\sigma_i)/\sigma_i}}_{\tilde{y}_i}\,\big)^2 \to \min_w$$

IRLS (Itreratively Reweighted Least Squares)

$$F_{\ell\times n} = \big(f_j(x_i)\big) \qquad \Gamma_{\ell\times\ell} = \mathrm{diag}\big(\sqrt{(1-\sigma_i)\sigma_i}\,\big)$$

$$\tilde{y}_i = y_i\sqrt{(1-\sigma_i)/\sigma_i} \qquad \tilde{F} = \Gamma F \qquad \tilde{y} = (\tilde{y}_i)_{i=1}^{\ell}$$

$$\big(Q''(w)\big)^{-1}Q'(w) = -(F^{\intercal}\Gamma^2 F)^{-1}F^{\intercal}\Gamma\tilde{y} = -(\tilde{F}^{\intercal}\tilde{F})^{-1}\tilde{F}^{\intercal}\tilde{y} = -\tilde{F}^{+}\tilde{y}$$

$$\boxed{Q(w) = \|\tilde{F}w - \tilde{y}\|^2 = \sum_{i=1}^{\ell}\underbrace{(1-\sigma_i)\sigma_i}_{\gamma_i}\Big(w^{\intercal}x - \underbrace{y_i\sqrt{(1-\sigma_i)/\sigma_i}}_{\tilde{y}_i}\Big)^2 \to \min_w}$$

SGD в SVM

$$Q = \sum_{i=1}^{\ell} \big(1 - y_i(\langle w, x_i \rangle + w_0)\big)_+ + \frac{1}{2C} \|w\|^2 \to \min_w$$

SGD в SVM

$$Q = \sum_{i=1}^{\ell} \big(1 - y_i(\langle w, x_i \rangle + w_0)\big)_+ + \frac{1}{2C}\|w\|^2 \to \min_w$$

$$\frac{\partial Q}{\partial w} = \sum_{i=1}^{\ell} y_i x_i [y_i(\langle w, x_i \rangle + w_0) > 1] + \frac{1}{C} w$$

SGD в SVM

$$Q = \sum_{i=1}^{\ell} \big(1 - y_i(\langle w, x_i \rangle + w_0)\big)_+ + \frac{1}{2C} \|w\|^2 \to \min_w$$

$$\frac{\partial Q}{\partial w} = \sum_{i=1}^{\ell} y_i x_i [y_i(\langle w, x_i \rangle + w_0) > 1] + \frac{1}{C} w$$

$$w^{(t+1)} = w^{(t)} - \eta_t \left(y_i x_i [y_i(\langle w, x_i \rangle + w_0) > 1] + \frac{1}{C} w \right)$$

$$w_0^{(t+1)} = w_0^{(t)} - \eta_t y_i \cdot [y_i(\langle w, x_i \rangle + w_0) > 1]$$

SGD в SVM

$$Q = \sum_{i=1}^{\ell} \big(1 - y_i(\langle w, x_i \rangle + w_0)\big)_+ + \frac{1}{2C} \|w\|^2 \to \min_w$$

$$\frac{\partial Q}{\partial w} = \sum_{i=1}^{\ell} y_i x_i [y_i(\langle w, x_i \rangle + w_0) > 1] + \frac{1}{C} w$$

$$w^{(t+1)} = w^{(t)} - \eta_t \left(y_i x_i [y_i(\langle w, x_i \rangle + w_0) > 1] + \frac{1}{C} w \right)$$

$$w_0^{(t+1)} = w_0^{(t)} - \eta_t y_i \cdot [y_i(\langle w, x_i \rangle + w_0) > 1]$$

Другой способ – решать двойственную задачу как задачу квадратичного программирования

II. Библиотека Vowpal Wabbit

Что реализовано в VW

- Линейные модели классификации и регрессии с разными функциями потерь и регуляризаторами
- Некоторые другие онлайновые алгоритмы, например для тематического моделирования (Online LDA – Latent Dirichlet Allocation)

Что реализовано в VW

- Линейные модели классификации и регрессии с разными функциями потерь и регуляризаторами
- Некоторые другие онлайновые алгоритмы, например для тематического моделирования (Online LDA – Latent Dirichlet Allocation)

План был сделать библиотеку онлайновых алгоритмов машинного обучения, но по факту сейчас используют как библиотеку с онлайновыми линейными классификаторами

Формат входных данных

Одна строка — один объект:

`123 10 | 1:0.43 5:2.1 age:20 some raw text here`

- 123 — целевая переменная
- 10 — вес объекта (можно не указывать, по умолчанию 1)
- `name:value` — описание признака
  - если `name` — строка, то она хэшируется (см. Hashing Trick)
  - по умолчанию `value`=1
  - если признак не описан для данного объекта, то он считается равным нулю

Слайд взят из презентации Евгения Соколова с семинаров по машинному обучению на ВМК

Формат входных данных

Признаки можно разделять на группы:

```
123 10 |integer 1:0.43 5:2.1 age:20 |text some raw
text here age:120
```

- `integer` и `text` — два пространства признаков
- в обоих пространствах есть признак age, так можно

Слайд взят из презентации Евгения Соколова с семинаров по машинному обучению на ВМК

Как запускать VW: обучение модели

Пусть выборка записана в файле `train.txt`.

Обучение:

```
vw -d train.txt --passes 10 -c -f model.vw
```

- `-d filename` — имя входного файла
- `--passes n` — количество проходов по выборке
- `-c` — включает кэширование, позволяет ускорить все проходы после первого
- `-f filename` — имя файла с моделью

Слайд взят из презентации Евгения Соколова с семинаров по машинному обучению на ВМК

Как запускать VW: применение обученной модели

Как получить прогноз:

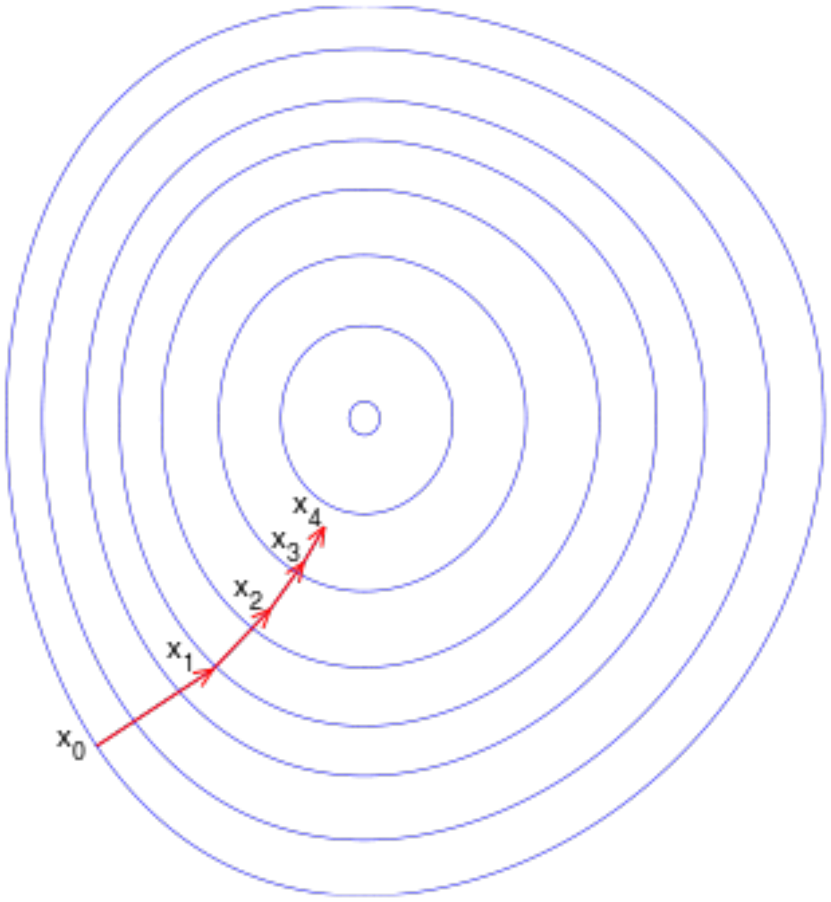
`vw -d test.txt -i model.vw -t -p predictions.txt`

- `-d filename` — имя входного файла
- `-i filename` — имя файла с моделью
- `-t` — режим применения существующей модели
- `-p filename` — имя файла с прогнозами

Слайд взят из презентации Евгения Соколова с семинаров по машинному обучению на ВМК

Напоминание: градиентный спуск

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \nabla F(\mathbf{x}_n),\ n \geq 0.$$



$$\nabla_w \tilde{Q} = \sum_{i=1}^{l} \nabla L(M_i)$$

$$\nabla \tilde{Q} = \sum_{i=1}^{l} L'(M_i) \frac{\partial M_i}{\partial w}$$

$$\frac{\partial M_i}{\partial w} = y_i x_i$$

$$\nabla \tilde{Q} = \sum_{i=1}^{l} y_i x_i L'(M_i)$$

$$\boxed{w_{n+1} = w_n - \gamma_n \sum_{i=1}^{l} y_i x_i L'(M_i)}$$

Напоминание: стохастический градиент

$$w_{n+1} = w_n - \gamma_n \cancel{\sum_{i=1}^{l}} y_i x_i L'(M_i)$$

$$\boxed{w_{n+1} = w_n - \gamma_n y_i x_i L'(M_i)}$$

x_i – случайный элемент обучающей выборки

Напоминание: стохастический градиент

$$w_{n+1} = w_n - \gamma_n \cancel{\sum_{i=1}^{l}} y_i x_i L'(M_i)$$

$$\boxed{w_{n+1} = w_n - \gamma_n y_i x_i L'(M_i)}$$

x_i – случайный элемент обучающей выборки

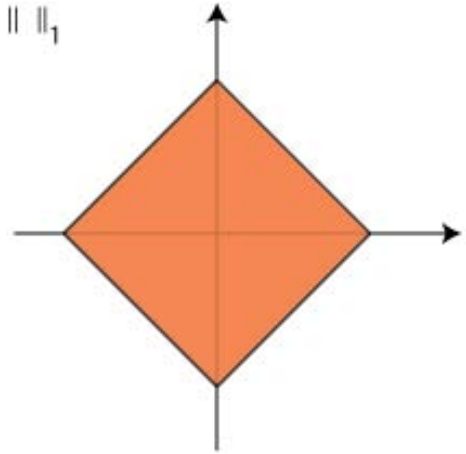
Бонус: возможность онлайнового обучения

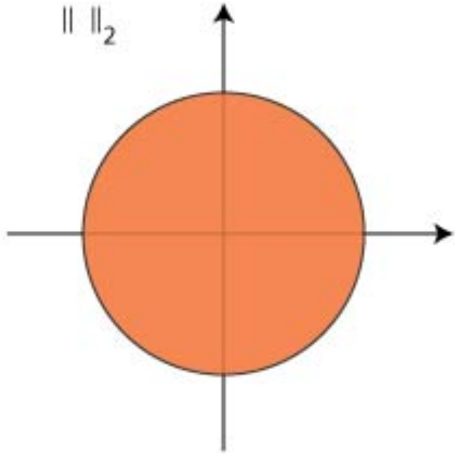
Напоминание: регуляризация

$$\sum_{i=1}^{l} L(M_i) + \gamma \sum_{k=1}^{m} |w_k| \to min \qquad \sum_{i=1}^{l} L(M_i) + \gamma \sum_{k=1}^{m} {w_k}^2 \to min$$

$l1$ – регуляризация

$l2$ – регуляризация





Напоминание: общий случай

$$Q = \sum_{i=1}^{\ell} L\big(y_i, f(x_i)\big) + \gamma V(w) \to \min_{w}$$

Функция потерь

Регуляризатор

Коэффициент регуляризации

Функции потерь в VW

Поддерживаемые функционалы (`--loss_function`):

- squared
$$\frac{1}{2}(y - a(x))^2$$
- classic — quadratic без перевзвешивания объектов
- quantile
$$\tau(a(x) - y)[y \leqslant a(x)] + (1 - \tau)(y - a(x))[y \geqslant a(x)]$$
- logistic
$$\log(1 + \exp(-ya(x)))$$
- hinge
$$\max(0, 1 - ya(x))$$

Слайд взят из презентации Евгения Соколова с семинаров по машинному обучению на ВМК

Регуляризаторы в VW

К функционалу можно добавить регуляризаторы:

- `--l1 coef`
- `--l2 coef`

Можно обучать SVM:

```
vw -d train.txt -f svm.vw --loss_function hinge --l2
0.1
```

Настройка весов в VW

Градиентный шаг:

$$w^{(t+1)} = w^{(t)} - \alpha_t \nabla Q(x_{i_t}).$$

Как выбирать α_t?

$$\alpha_t = s \left(\frac{i}{i + t} \right)^p,$$

где

- `-l s`
- `--initial_t i`
- `--power_t p`

Эти параметры сильно влияют на качество!

Слайд взят из презентации Евгения Соколова с семинаров по машинному обучению на ВМК

Другие параметры VW

- `-b n`: логарифм количества возможных значений хэш-функции для hashing trick
- `-q ab`: генерирует все парные признаки, где первый признак берется из пространств с именем «a*», второй — из «b*»
- `--cubic abc`: тройки признаков
- `--ngram an`: генерирует *n*-граммы для пространств «a*»
- `--skips ak`: разрешает делать пропуски длины *k* в *n*-граммах пространств «a*»

Слайд взят из презентации Евгения Соколова с семинаров по машинному обучению на ВМК

Напоминание: hashing trick

feature
a
b
c
b

$L=2^n$ столбцов

hash(a) % L = hash(c) % L = 1	hash(b) % L = 2
1	
	1
1	
	1

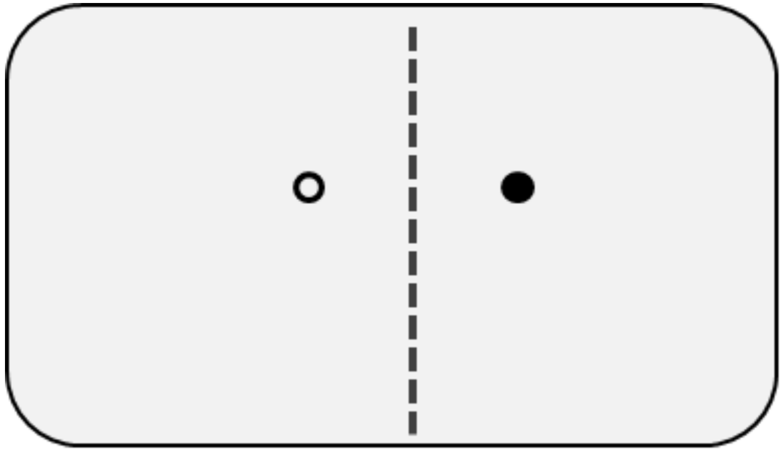
Что еще может VW

Читайте документацию!

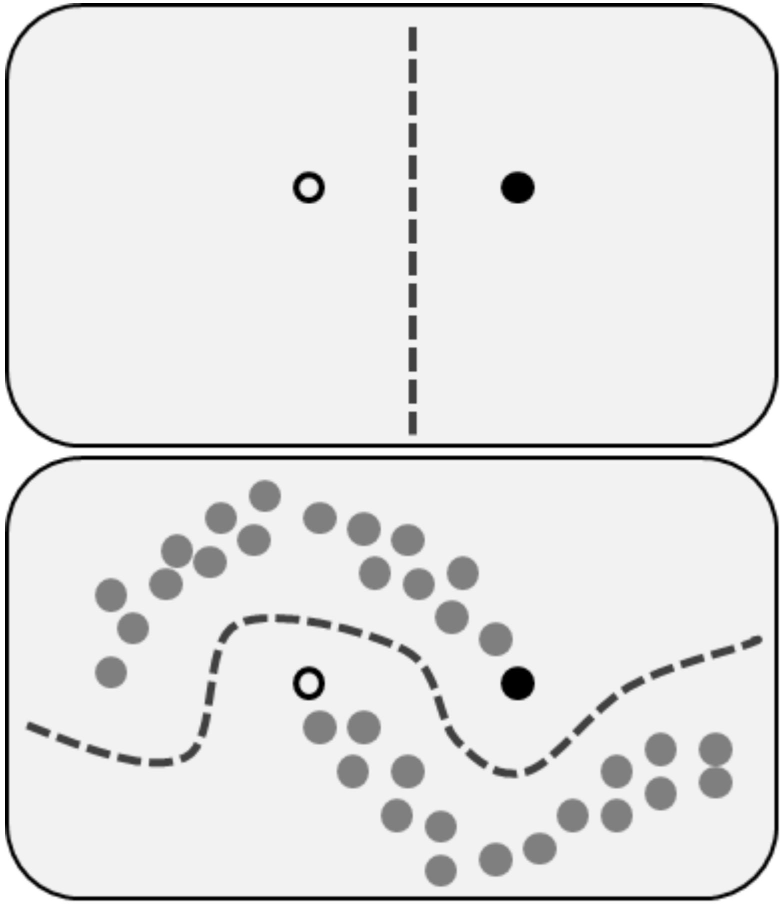
- `--holdout_after`: ранний останов, использует отложенную выборку
- `--bfgs`: квазиньютоновская оптимизация, должна работать лучше
- `--ksvm`: SVM с kernel trick

III. Semi-supervised линейные классификаторы

Semi-supervised обучение: мотивация



Semi-supervised обучение: мотивация



Semi-supervised SVM (S3VM)

SVM:

$$\sum_{i=1}^{l} \max\{0; 1 - y_i < w, x_i >\} \; + \; \alpha ||w||_{l_2}^2 \to \min_w$$

Semi-supervised SVM (S3VM)

SVM:

$$\sum_{i=1}^{l} \max\{0; 1 - y_i < w, x_i >\} \ + \ \alpha ||w||_{l_2}^2 \to \min_w$$

Идея:

$$y_i < w, x_i > \ \to \ a(x_i) < w, x_i >=$$
$$= sign\{< w, x_i >\} < w, x_i >= | < w, x_i > |$$

Semi-supervised SVM (S3VM)

SVM:

$$\sum_{i=1}^{l} \max\{0; 1 - y_i < w, x_i >\} + \alpha ||w||_{l_2}^2 \to \min_w$$

Идея:

$$y_i < w, x_i > \;\to\; a(x_i) < w, x_i >=$$
$$= sign\{< w, x_i >\} < w, x_i >= | < w, x_i > |$$

$$\boxed{\sum_{i=1}^{l} \max\{0; 1 - y_i < w, x_i >\} + \beta \sum_{i=l+1}^{l+u} \max\{0; 1 - | < w, x_i > |\} + \alpha ||w||_{l_2}^2}$$

Semi-supervised логистическая регрессия (энтропийная регуляризация)

$$\boxed{\sum_{i=1}^{l} \ln\left(1+e^{-M_i}\right) \;+\; \beta \sum_{i=l+1}^{l+u} H\left(P(+1|x_i); P(-1|x_i)\right) \;+\alpha V(w)}$$

$$H(p,q) = -p\ln p - q\ln q$$

IV. Многоклассовая логистическая регрессия

От логлосса к кросс-энтропии

$$Q = -\sum_{i=1}^{\ell} y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \to \min_w$$

$y_i \in \{0, 1\}$

$p_i = P(y_i = 1 | x_i, w)$

От логлосса к кросс-энтропии

$$Q = -\sum_{i=1}^{\ell} y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \to \min_w$$

$y_i \in \{0, 1\}$ $\qquad\qquad$ $p_i = P(y_i = 1 | x_i)$

$$Q = -\sum_{i=1}^{\ell} \sum_{k=0}^{K-1} y_i \ln P(y_i = k | x_i) \to \min_w$$

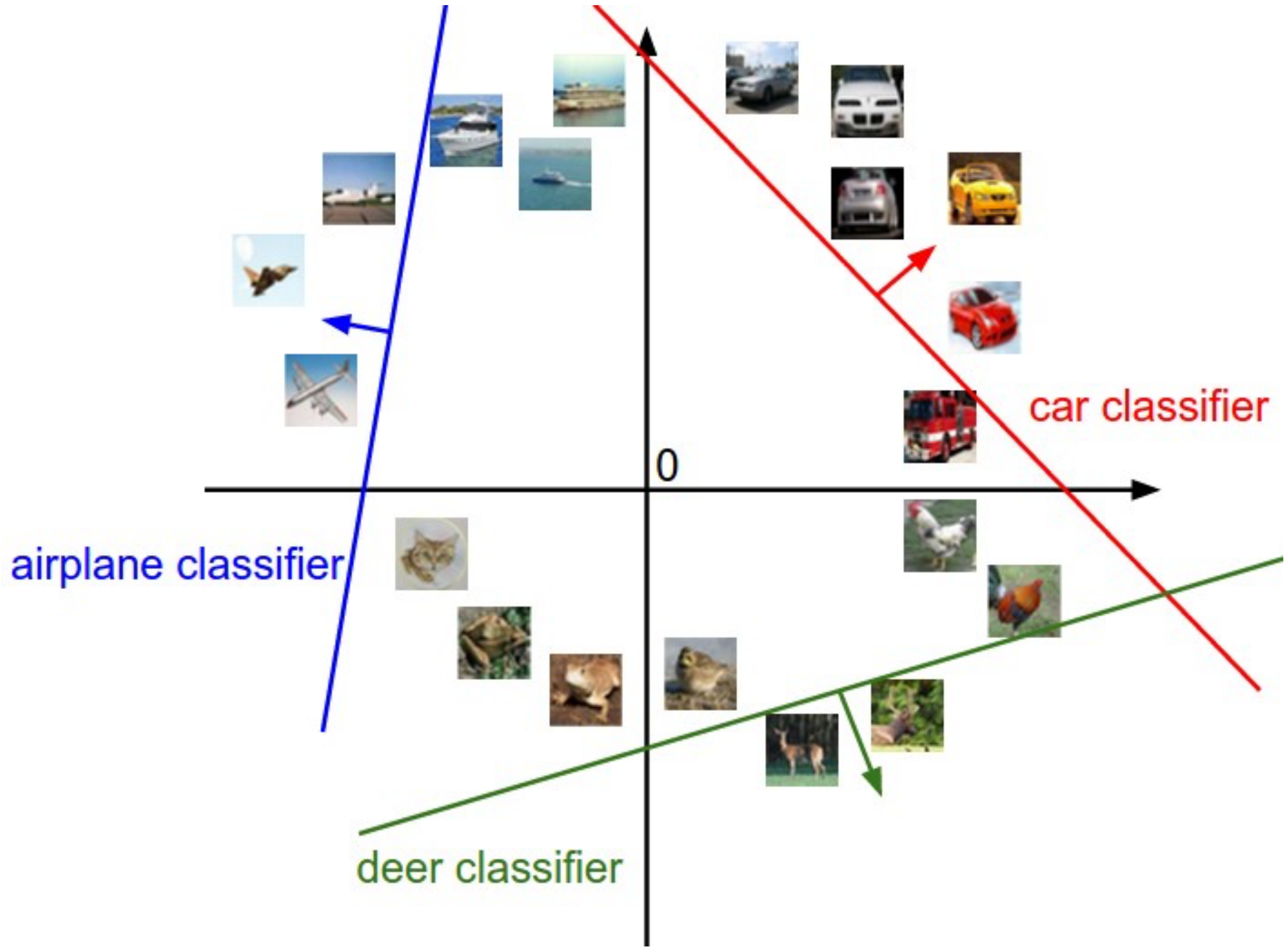
$y_i \in \{0, 1, \dots, K - 1\}$

От сигмоиды к softmax

$$p_i = \sigma(\langle w, x_i \rangle) = \frac{1}{1 + e^{-\langle w, x_i \rangle}} = P(y_i = 1 | x_i)$$

$$P(y_i = k | x_i) = \frac{e^{-\langle w_k, x_i \rangle}}{\sum_{r=0}^{K-1} e^{-\langle w_r, x_i \rangle}}$$

Векторы весов в многоклассовой задаче



Многоклассовая логистическая регрессия

$$Q = -\sum_{i=1}^{\ell} \sum_{k=0}^{K-1} y_i \ln P(y_i = k | x_i) \to \min_w$$

$$y_i \in \{0, 1, \dots, K-1\}$$

$$P(y_i = k | x_i) = \frac{e^{-\langle w_k, x_i \rangle}}{\sum_{r=0}^{K-1} e^{-\langle w_r, x_i \rangle}}$$

V. Многоклассовый SVM

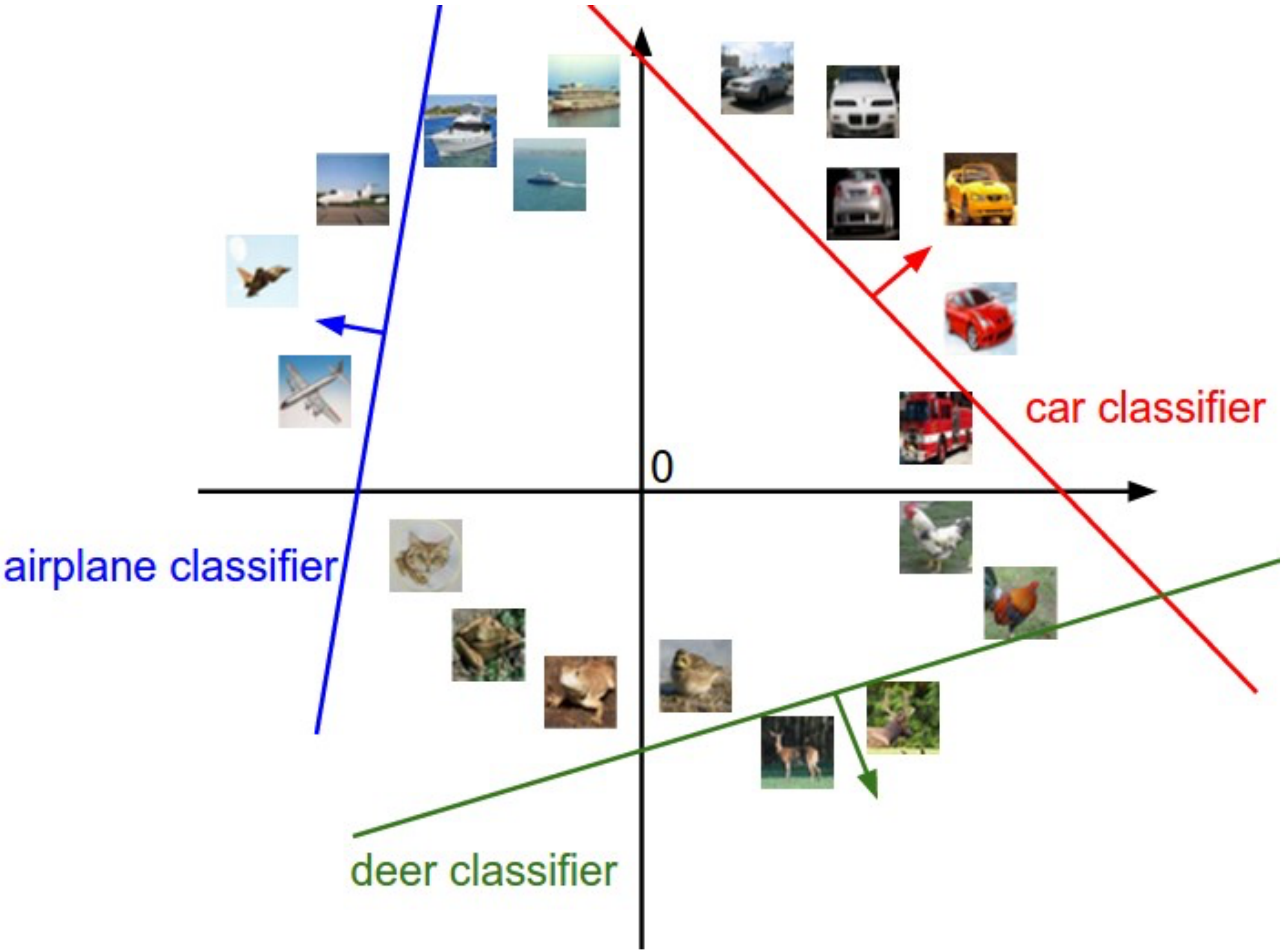
Безусловная оптимизационная задача в SVM

$$Q(w, w_0) = \sum_{i=1}^{\ell} \bigl(1 - M_i(w, w_0)\bigr)_+ + \frac{1}{2C}\|w\|^2 \to \min_{w, w_0}$$

Оптимизационная задача с ограничениями

$$
\begin{cases}
\frac{1}{2}\langle w, w\rangle + C\sum\limits_{i=1}^{\ell}\xi_i \to \min\limits_{w, w_0, \xi};\\
y_i\big(\langle w, x_i\rangle - w_0\big) \geqslant 1 - \xi_i, \quad i = 1, \ldots, \ell;\\
\xi_i \geqslant 0, \quad i = 1, \ldots, \ell.
\end{cases}
$$

Векторы весов в многоклассовой задаче



Оптимизационная задача и решающее правило в многоклассовом SVM

$$\min_{\boldsymbol{w}_m, \xi_i} \quad \frac{1}{2}\sum_{m=1}^{k} \boldsymbol{w}_m^T \boldsymbol{w}_m + C\sum_{i=1}^{l} \xi_i$$

$$\boldsymbol{w}_{y_i}^T \boldsymbol{x}_i - \boldsymbol{w}_m^T \boldsymbol{x}_i \geq e_i^m - \xi_i, \ \ i = 1, \ldots, l,$$

Оптимизационная задача и решающее правило в многоклассовом SVM

$$\min_{\boldsymbol{w}_m, \xi_i} \qquad \frac{1}{2} \sum_{m=1}^{k} \boldsymbol{w}_m^T \boldsymbol{w}_m + C \sum_{i=1}^{l} \xi_i$$

Оптимизационная задача и решающее правило в многоклассовом SVM

$$\min_{\boldsymbol{w}_m, \xi_i} \quad \frac{1}{2}\sum_{m=1}^{k} \boldsymbol{w}_m^T \boldsymbol{w}_m + C\sum_{i=1}^{l} \xi_i$$

$$\boldsymbol{w}_{y_i}^T \boldsymbol{x}_i - \boldsymbol{w}_m^T \boldsymbol{x}_i \geq e_i^m - \xi_i, \; i = 1, \dots, l,$$

$$e_i^m = \begin{cases} 0 & \text{if } y_i = m, \\ 1 & \text{if } y_i \neq m. \end{cases}$$

Оптимизационная задача и решающее правило в многоклассовом SVM

$$\min_{\boldsymbol{w}_m, \xi_i} \quad \frac{1}{2}\sum_{m=1}^{k} \boldsymbol{w}_m^T \boldsymbol{w}_m + C\sum_{i=1}^{l} \xi_i$$

$$\boldsymbol{w}_{y_i}^T \boldsymbol{x}_i - \boldsymbol{w}_m^T \boldsymbol{x}_i \geq e_i^m - \xi_i, \; i = 1, \ldots, l,$$

$$e_i^m = \begin{cases} 0 & \text{if } y_i = m, \\ 1 & \text{if } y_i \neq m. \end{cases}$$

$$\boxed{\arg \max_{m=1,\ldots,k} \boldsymbol{w}_m^T \boldsymbol{x}.}$$

VI. Sklearn.linear_model

SGD

SGD

- SGDClassifier
  - loss="hinge": (soft-margin) linear Support Vector Machine,
  - loss="modified_huber": smoothed hinge loss,
  - loss="log": logistic regression

SGD

- SGDClassifier
  - loss="hinge": (soft-margin) linear Support Vector Machine,
  - loss="modified_huber": smoothed hinge loss,
  - loss="log": logistic regression
- SGDRegressor
  - loss="squared_loss": Ordinary least squares,
  - loss="huber": Huber loss for robust regression,
  - loss="epsilon_insensitive": linear Support Vector Regression

SGD

- SGDClassifier
  - loss="hinge": (soft-margin) linear Support Vector Machine,
  - loss="modified_huber": smoothed hinge loss,
  - loss="log": logistic regression
- SGDRegressor
  - loss="squared_loss": Ordinary least squares,
  - loss="huber": Huber loss for robust regression,
  - loss="epsilon_insensitive": linear Support Vector Regression

- `penalty="l2"`: L2 norm penalty on `coef_`.
- `penalty="l1"`: L1 norm penalty on `coef_`.
- `penalty="elasticnet"`: Convex combination of L2 and L1; `(1 - l1_ratio) * L2 + l1_ratio * L1`

Библиотека liblinear

Из документации:

LIBLINEAR is a **linear** classifier for data with **millions** of instances and features. It supports
L2-regularized classifiers
L2-loss linear SVM, L1-loss linear SVM, and logistic regression (LR)
L1-regularized classifiers (after version 1.4)
L2-loss linear SVM and logistic regression (LR)
L2-regularized support vector regression (after version 1.9)
L2-loss linear SVR and L1-loss linear SVR.

Пример: LinearSVC

- **C** : float, optional (default=1.0)

Penalty parameter C of the error term.

- **loss** : string, ‘hinge’ or ‘squared_hinge’ (default=’squared_hinge’)

Specifies the loss function. ‘hinge’ is the standard SVM loss (used e.g. by the SVC class) while ‘squared_hinge’ is the square of the hinge loss.

- **penalty** : string, ‘l1’ or ‘l2’ (default=’l2’)

Specifies the norm used in the penalization. The ‘l2’ penalty is the standard used in SVC. The ‘l1’ leads to coef_ vectors that are sparse.

- **dual** : bool, (default=True)

Select the algorithm to either solve the dual or primal optimization problem. Prefer dual=False when n_samples > n_features.

Другие модели

- Linear Regression $\underset{w}{min}\, ||Xw - y||_2{}^2$
- Ridge $\underset{w}{min}\, ||Xw - y||_2{}^2 + \alpha ||w||_2{}^2$
- LASSO $\underset{w}{min}\, \frac{1}{2n_{samples}} ||Xw - y||_2^2 + \alpha ||w||_1$
- Multi-task LASSO $\underset{w}{min}\, \frac{1}{2n_{samples}} ||XW - Y||_{Fro}^2 + \alpha ||W||_{21}$

$$||A||_{Fro} = \sqrt{\sum_{ij} a_{ij}^2} \qquad ||A||_{21} = \sum_i \sqrt{\sum_j a_{ij}^2}$$

Другие модели

- Elastic Net $$\min_w \frac{1}{2n_{samples}} ||Xw - y||_2^2 + \alpha\rho||w||_1 + \frac{\alpha(1-\rho)}{2}||w||_2^2$$
- Multi-task Elastic Net $$\min_W \frac{1}{2n_{samples}} ||XW - Y||_{Fro}^2 + \alpha\rho||W||_{21} + \frac{\alpha(1-\rho)}{2}||W||_{Fro}^2$$
- OMP $$\arg\min ||y - X\gamma||_2^2 \text{ subject to } ||\gamma||_0 \leq n_{nonzero\_coefs}$$
- Logistic Regression $$\min_{w,c} ||w||_1 + C\sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

$$\min_{w,c} \frac{1}{2} w^T w + C\sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

Другие модели

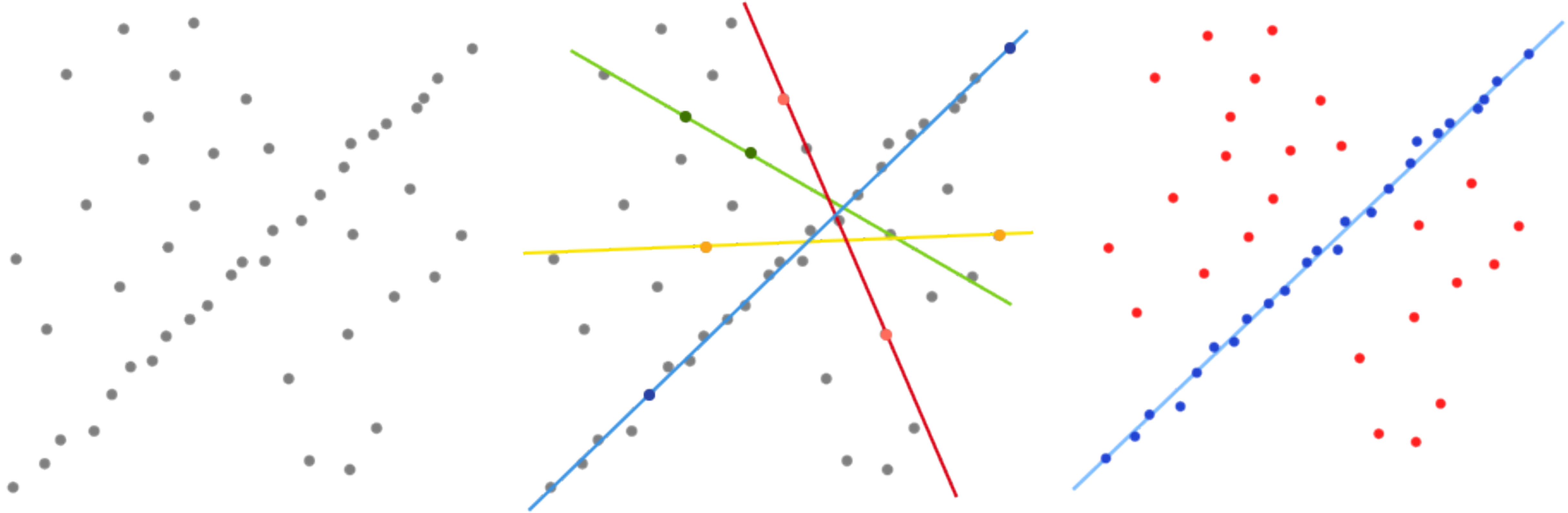
- LARS
- LARS LASSO
- Bayesian Regression
- Bayesian Ridge Regression
- Automatic Relevance Determination (ARD)

Робастные модели в linear_model

- RANSACRegressor
- HuberRegressor
- Theil-Sen Regressor

Бонус-трек: робастные модели

RANSACRegressor



HuberRegressor

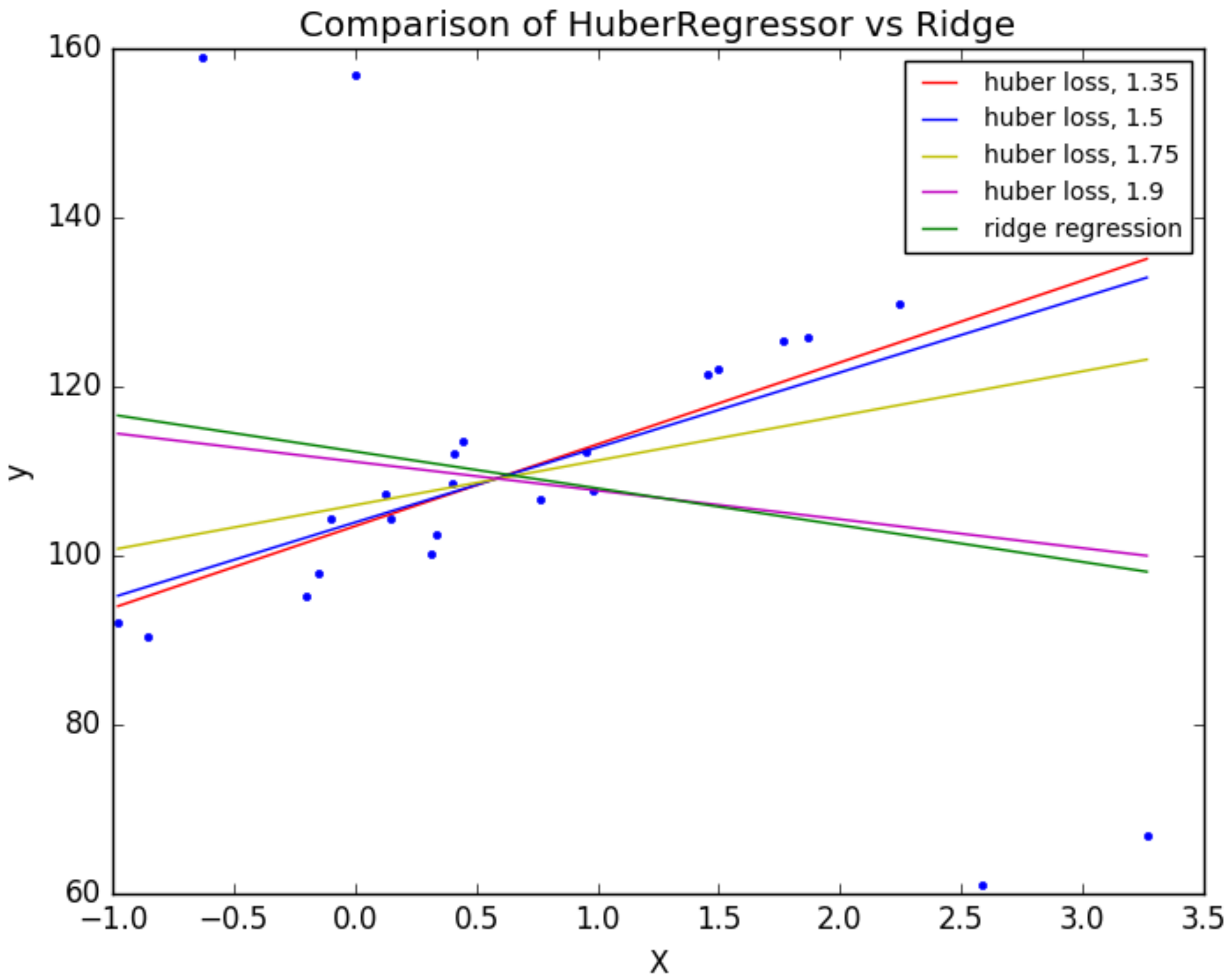
$$\min_{w,\sigma} \sum_{i=1}^{n} \left(\sigma + H_m\left(\frac{X_i w - y_i}{\sigma} \right) \sigma \right) + \alpha ||w||_2{}^2$$

$$H_m(z) = \begin{cases} z^2, & \text{if } |z| < \epsilon, \\ 2\epsilon|z| - \epsilon^2, & \text{otherwise} \end{cases}$$

HuberRegressor

$$\min_{w,\sigma} \sum_{i=1}^{n} \left(\sigma + H_m\left(\frac{X_i w - y_i}{\sigma} \right) \sigma \right) + \alpha ||w||_2{}^2$$

$$H_m(z) = \begin{cases} z^2, & \text{if } |z| < \epsilon, \\ 2\epsilon|z| - \epsilon^2, & \text{otherwise} \end{cases}$$

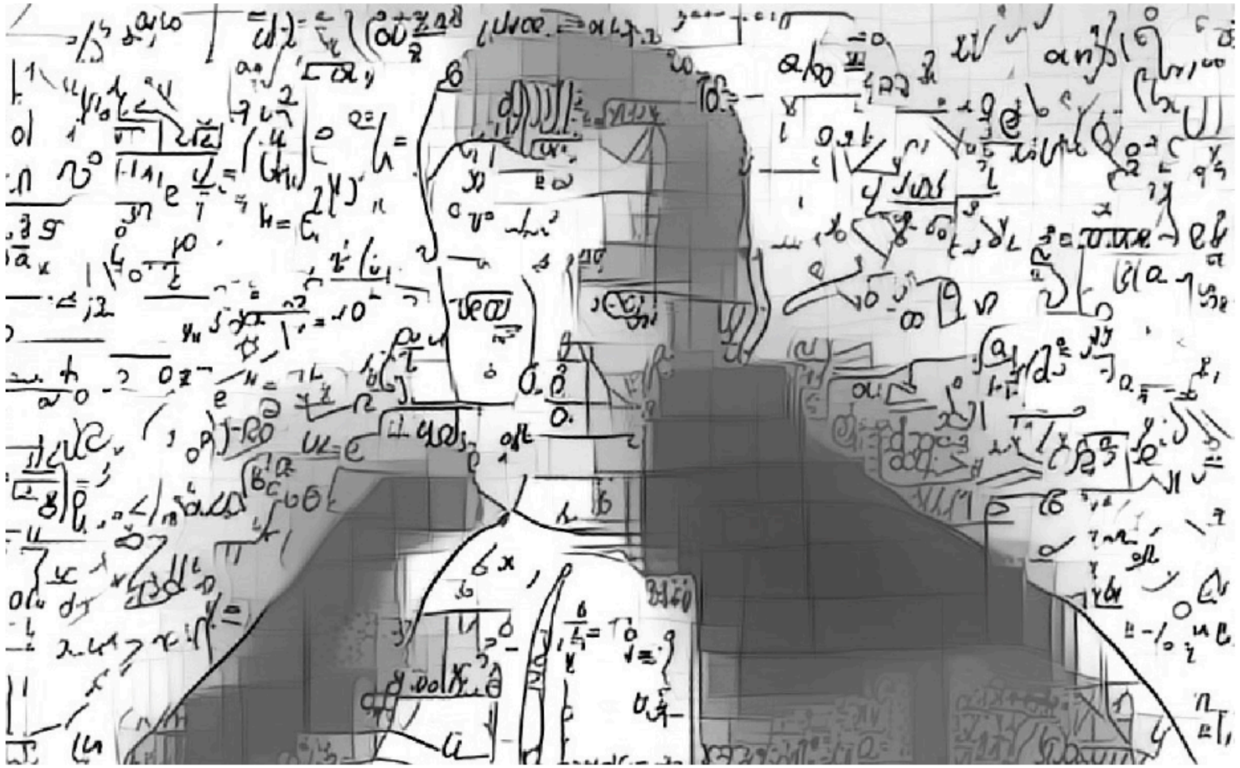


Отзывы

Отзывы о прошедших лекциях и семинарах можно и нужно оставлять здесь:

https://ml-mipt.github.io/2017part1/

Deep|Bayes Summer School



- Самые продвинутые нейросеточки
- Зачем там байесовские методы?
- Будут ли на школе кормить? Да :)

Deadline 31 Марта

Есть тестовое задание

26-30 Августа, deepbayes.ru