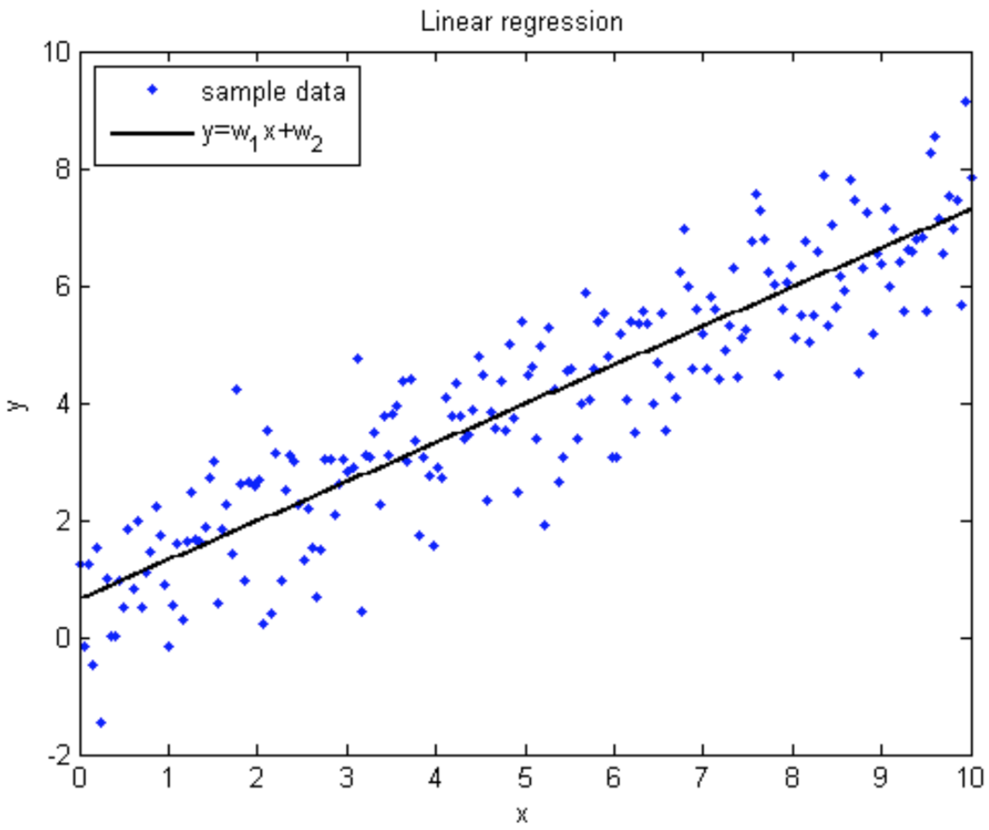
Линейные модели

18 февраля 2020 г.

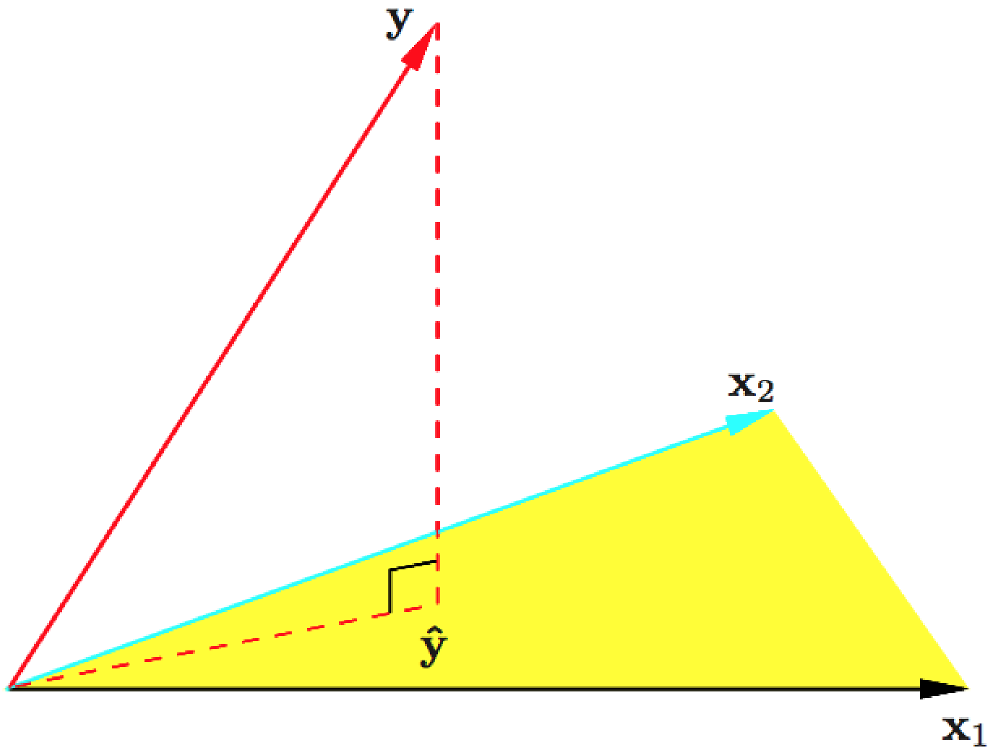
Линейная регрессия

- Данные: $X^l = (x_i, y_i)_1^l, x_i \in \mathbb{R}^n, y_i \in \mathbb{R}$
- Линейная модель: $a(x_i, w) = \langle x_i, w \rangle$
- Функция потерь: $\mathcal{L}_i(w) = (\langle x_i, w \rangle - y_i)^2$
- Обучение: $Q(w) = \sum_{i=1}^l \mathcal{L}_i(w) \to \min\limits_w$



Геометрическая интерпретация МНК

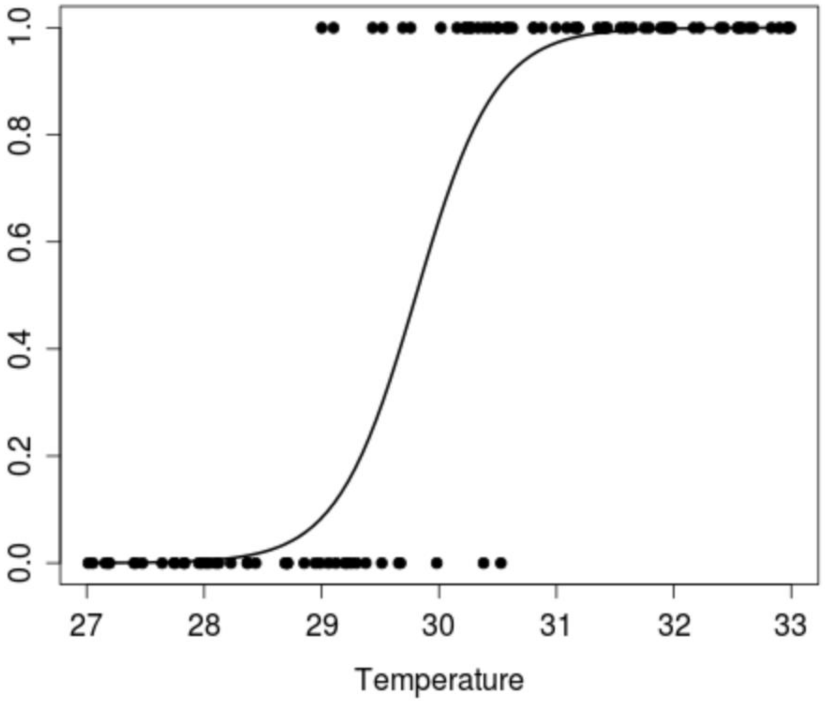
x_1, x_2 - столбцы матрицы объект-признак, y - вектор ответов,
$\hat{y}$ - наилучшая аппроксимация для y в подпространстве, натянутом на x_1, x_2



Минимизируем:
$$||y - Xw||^2 \to \min_w$$

Решение:
$$\hat{y} = Xw = X(X^T X)^{-1} X^T y$$

Логистическая регрессия



- Данные: $X^l = (x_i, y_i)_1^l, x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}$
- Линейная модель: $a(x_i, w) = \langle x_i, w \rangle$
- Отступ: $M_i = \langle x_i, w \rangle y_i$
- Предсказание вероятности класса:

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

- Функция потерь: $\mathcal{L}_i(w) = \log(1 + \exp(-\langle x_i, w \rangle y_i))$
- Обучение: $Q(w) = \sum_{i=1}^l \mathcal{L}_i(w) \to \min\limits_w$

Отступ

- Два класса:

$$\mathrm{M}(\mathrm{x_i}) = \langle x_i, w \rangle y_i$$

- Произвольное число классов:

$$M(x_i) = \langle x_i, w_{y_i} \rangle y_i - \max_{y \in \mathrm{Y}, y \neq y_i} \langle x_i, w_y \rangle$$

Функции потерь и метрики качества

Функции потерь

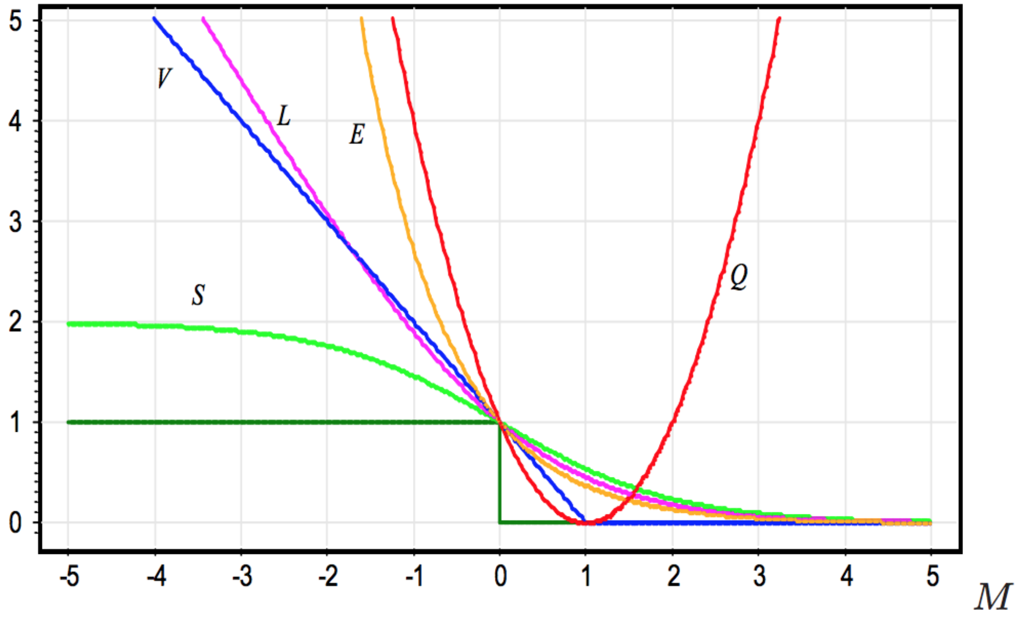


Рис. 9. Непрерывные аппроксимации пороговой функции потерь $[M < 0]$.

$Q(M) = (1 - M)^2$ — квадратичная;
$V(M) = (1 - M)_+$ — кусочно-линейная;
$S(M) = 2(1 + e^M)^{-1}$ — сигмоидная;
$L(M) = \log_2(1 + e^{-M})$ — логарифмическая;
$E(M) = e^{-M}$ — экспоненциальная.

Функции потерь

Фактически, выбор линейной модели для решения задачи осуществляется выбором функции потерь и регуляризации:

- Квадратичная функция потерь – линейная регрессия
- Логарифмическая функция потерь – логистическая регрессия
- Кусочно-линейная (Hinge loss) функция потерь – линейный SVM

Функция потерь и метрика качества

Стоит различать между собой функции потерь и метрики качества:

- Функция потерь – функция, которую удобно оптимизировать численно
- Метрика качества – функция, которая диктуется «заказчиком»

Santa Gift Matching challenge

Пример странной метрики оценки качества

Your goal is to maximize the

```
Average Normalized Happiness (ANH) = (AverageNormalizedChildHappiness (ANCH) ) ^ 3
+ (AverageNormalizedSantaHappiness (ANSH) ) ^ 3
```

where **NormalizedChildHappiness** is the happiness of each child, divided by the maximum possible happiness, and **NormalizedSantaHappiness** is the happiness of each gift, divided by the maximum possible happiness.

Note the cubic terms with ANCH and ANSH.

in the equation form:

$$ANCH = \frac{1}{n_c} \sum_{i=0}^{n_c-1} \frac{ChildHappiness}{MaxChildHappiness},$$

$$ANSH = \frac{1}{n_g} \sum_{i=0}^{n_g-1} \frac{GiftHappiness}{MaxGiftHappiness}.$$

n_c is the number of children. n_g is the number of gifts

Метрики качества

sklearn.metrics – все популярные метрики качества уже реализованы.

Можно легко реализовать свою.

В конкурсах для экзотических метрик качества обычно дается реализация.

Метрики качества

metrics.accuracy_score (y_true, y_pred[, …])
metrics.auc (x, y[, reorder])
metrics.average_precision_score (y_true, y_score)
metrics.brier_score_loss (y_true, y_prob[, …])
metrics.classification_report (y_true, y_pred)
metrics.cohen_kappa_score (y1, y2[, labels, …])
metrics.confusion_matrix (y_true, y_pred[, …])
metrics.f1_score (y_true, y_pred[, labels, …])
metrics.fbeta_score (y_true, y_pred, beta[, …])
metrics.hamming_loss (y_true, y_pred[, …])
metrics.hinge_loss (y_true, pred_decision[, …])
metrics.jaccard_similarity_score (y_true, y_pred)

metrics.log_loss (y_true, y_pred[, eps, …])
metrics.matthews_corrcoef (y_true, y_pred[, …])
metrics.precision_recall_curve (y_true, …)
metrics.precision_recall_fscore_support (…)
metrics.precision_score (y_true, y_pred[, …])
metrics.recall_score (y_true, y_pred[, …])

metrics.explained_variance_score (y_true, y_pred)
metrics.mean_absolute_error (y_true, y_pred)
metrics.mean_squared_error (y_true, y_pred[, …])
metrics.mean_squared_log_error (y_true, y_pred)
metrics.median_absolute_error (y_true, y_pred)
metrics.r2_score (y_true, y_pred[, …])

Функция потерь: Log-loss

- Классификация:

$$y_t \in \{0,1\}$$
$$y_p \in (0,1)$$
$$LogLoss(y_t, y_p) = -(y_t \log(y_p) + (1 - y_t) \log(1 - y_p))$$

- Мультиклассификация (Кросс-энтропия):

$$y_t = (y_{t1}, y_{t2}, \dots, y_{tk})$$
$$y_p = (y_{p1}, y_{p2}, \dots, y_{pk})$$
$$LogLoss(y_t, y_p) = \sum_{i=1}^{k} y_{tk} log(y_{pk})$$

Функция потерь Log-loss

- y_p «обрубается» до отрезка $[\epsilon, 1-\epsilon]$ из-за области определения логарифма
- В некоторых задачах y_t могут быть из отрезка $[0, 1]$
- Можно доказать, что минимум по-прежнему достигается при $y_p = y_t$

Функция потерь Log-loss

- Логистическая регрессия, $y_t \in \{0,1\}$

$$LogLoss\left(y_t, y_p\right) = \log(1 + \exp(-M(y_t, y_p)))$$

- Логистическая регрессия, $y_t \in \{0, \dots, k-1\}$

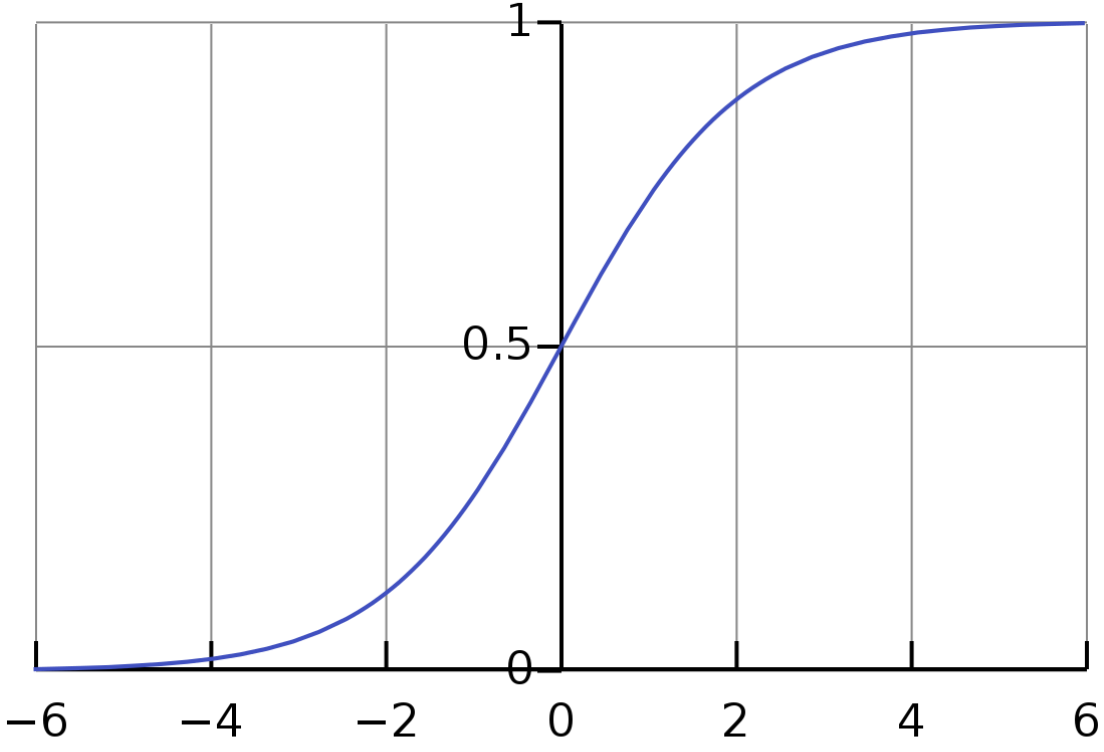
$$P(y|x,w) = \frac{\exp\langle w_y, \mathrm{x}\rangle}{\sum_{z \in \mathrm{Y}} \exp\langle w_z, \mathrm{x}\rangle} = SoftMax\,\langle w_y\,, \mathrm{x}\rangle$$

$$LogLoss\left(y_t, y_p\right) = -\log P(y_t|x,w)$$

Сигмоида

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

- Значения лежат в интервале (0, 1)
- $\sigma'(x) = \sigma(x) * (1 - \sigma(x))$ – полезный факт про сигмоиду



Логит

$$p = \frac{1}{1 + \exp(- < x, w >)}$$

$$logit(p) =< x, w >= -\log\left(\frac{1}{p} - 1\right) = log\left(\frac{p}{1-p}\right)$$

Если отношение $p : (1 - p)$ оказалось больше 1, то первый класс более вероятен.

Мультиклассовая логистическая регрессия

Матрица w размерности $M \times K$, где M - число факторов, K - число классов

Обучать можно двумя способами:

- Оптимизировать сразу по всей матрице
- Решать бинарные задачи, затем аккумулировать результаты

Логистическая регрессия для регрессии

- Пусть метки классов лежат в интервале $\mathrm{y_i} \in [0,1]$
- Хотим обучить логистическую регрессию
- Используем i-ый объект:
  - с весом $\mathrm{y_i}$ и значением метки 1
  - с весом $(1 - y_i)$ и значением метки 0
- Переходим к задаче классификации

Регуляризация

Регуляризация

Веса w линейной модели могут оказаться линейно зависимыми, тогда существует целое подпространство решений оптимизационной задачи

Если объектов меньше, чем признаков, то линейная зависимость точно присутствует

Большие значения весов делают решение неустойчивым

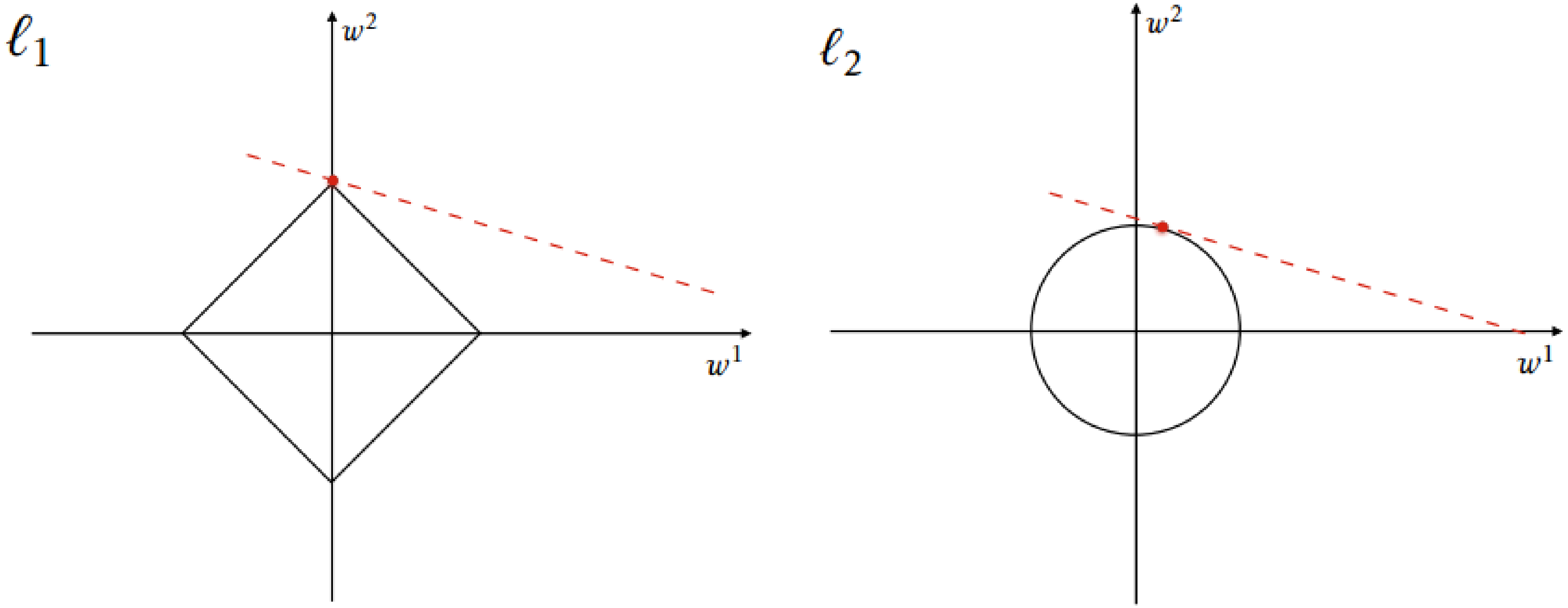
Регуляризация

Для борьбы с переобучением в оптимизационную задачу добавляется регуляризация:

$$Q(w) = \sum_{i=1}^{l} \mathcal{L}_i(w) + R(w) \to \min_w$$

$R(w)$ – штраф за сложность модели

Регуляризация



- L_1: $R(w) = ||w||_1$ - обычно используется как метод отбора признаков
- L_2: $R(w) = ||w||_2$ - улучшает качество модели

Разнообразие реализаций

`linear_model.ARDRegression` ([n_iter, tol, …])
`linear_model.BayesianRidge` ([n_iter, tol, …])
`linear_model.ElasticNet` ([alpha, l1_ratio, …])
`linear_model.ElasticNetCV` ([l1_ratio, eps, …])
`linear_model.HuberRegressor` ([epsilon, …])
`linear_model.Lars` ([fit_intercept, verbose, …])
`linear_model.LarsCV` ([fit_intercept, …])
`linear_model.Lasso` ([alpha, fit_intercept, …])
`linear_model.LassoCV` ([eps, n_alphas, …])
`linear_model.LassoLars` ([alpha, …])
`linear_model.LassoLarsCV` ([fit_intercept, …])
`linear_model.LassoLarsIC` ([criterion, …])
`linear_model.LinearRegression` ([…])
`linear_model.LogisticRegression` ([penalty, …])
`linear_model.LogisticRegressionCV` ([Cs, …])
`linear_model.MultiTaskLasso` ([alpha, …])
`linear_model.MultiTaskElasticNet` ([alpha, …])
`linear_model.MultiTaskLassoCV` ([eps, …])
`linear_model.MultiTaskElasticNetCV` ([…])
`linear_model.OrthogonalMatchingPursuit` ([…])

`linear_model.OrthogonalMatchingPursuit` ([…])
`linear_model.OrthogonalMatchingPursuitCV` ([…])
`linear_model.PassiveAggressiveClassifier` ([…])
`linear_model.PassiveAggressiveRegressor` ([C, …])
`linear_model.Perceptron` ([penalty, alpha, …])
`linear_model.RANSACRegressor` ([…])
`linear_model.Ridge` ([alpha, fit_intercept, …])
`linear_model.RidgeClassifier` ([alpha, …])
`linear_model.RidgeClassifierCV` ([alphas, …])
`linear_model.RidgeCV` ([alphas, …])
`linear_model.SGDClassifier` ([loss, penalty, …])
`linear_model.SGDRegressor` ([loss, penalty, …])
`linear_model.TheilSenRegressor` ([…])
`linear_model.enet_path` (X, y[, l1_ratio, …])
`linear_model.lars_path` (X, y[, Xy, Gram, …])
`linear_model.lasso_path` (X, y[, eps, …])
`linear_model.lasso_stability_path` (X, y[, …])
`linear_model.logistic_regression_path` (X, y)
`linear_model.orthogonal_mp` (X, y[, …])
`linear_model.orthogonal_mp_gram` (Gram, Xy[, …])

Разнообразие реализаций

Множество различных линейных регрессий:

- *ElasticNet* – L_1 и L_2 регуляризации;
- *RidgeRegression* – только L_2 регуляризация;
- *Lasso* – только L_1 регуляризация;
- *HuberRegression* – модель, более стойкая к выбросам;
- …

Категориальные признаки

Категориальные признаки

Некоторые категориальные признаки *могут* иметь непрерывную природу:

- Месяц в году
- День в неделе
- Яркость цвета (но не сам цвет: красный, желтый, синий)

Можно попробовать заменить категориальный признак на действительнозначный

Категориальные признаки

- Для остальных случаев используется one-hot-encoding

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

Категориальные признаки

One-hot-encoding порождает линейную зависимость в данных:

$$x_1 + x_2 + x_3 + \cdots + x_n = 1$$

Необходимо удалять любой из новых факторов

Решение оптимизационной задачи

Решение оптимизационной задачи

LIBLINEAR – пакет, включающий библиотеку и command-line tool для обучения. Есть обертка для sklearn.

- L_2-regularized L_1/L_2-loss Support Vector Machines
- L_2-regularized Logistic Regression
- https://www.csie.ntu.edu.tw/~cjlin/papers/logistic.pdf

Решение оптимизационной задачи

Итеративные методы оптимизации:
- Быстрая итерация, медленная сходимость
- Медленная итерация, быстрая сходимость

LIBLINEAR использует Trust Region Newton Method

Решение оптимизационной задачи

C V X O P T

PYTHON SOFTWARE FOR CONVEX OPTIMIZATION

CVXOPT – решение задач оптимизации на python

Особенности моделей

Шумные данные

Как объекты-«выбросы» могут повлиять на качество обучения:

- Логистической регрессии
- Линейной регрессии

Шумные данные

Как объекты-«выбросы» могут повлиять на качество обучения:

- Логистической регрессии – не очень сильно
- Линейной регрессии – очень сильно

Несбалансированные классы

Часто в задачах один класс сильно преобладает над другим

Можно исключить большую часть примеров доминирующего класса, а оставшимся объектам дать больший вес

Данные для линейных моделей

Признаки необходимо нормировать:

- Веса становятся интерпретируемыми
- Градиентный спуск лучше сходится

Линейные модели sklearn умееют могут нормировать данные из коробки

Веса линейной модели

Линейный модели хорошо интерпретируемы

Веса отражают зависимость предсказания от признака

Предсказания линейных моделей

Несколько советов для лучшего качества:

- predict_proba лучше, чем predict
- Если нужны бинарные предсказания, то стоит настроить пороговое значение
- Если нужна вероятность, то можно попробовать откалибровать предсказания: Isotonic regression, Platt scaling

Предсказания линейных моделей

Логистическая регрессия не дает истинных вероятностей:
- Модель не достаточно хорошо отражает действительность
- Регуляризация вносит смещение

Поэтому, настройка порогового значения обычно работает лучше, чем автоматическая бинаризация по порогу $p = 0.5$

Выводы

Плюсы и минусы

Плюсы:
- Понятные, интерпретируемые модели
- Быстро обучаются на больших объемах данных
- Сложно переобучиться
- Легко настраивать

Минусы:
- Слабое качество во многих задачах
- Необходим пре-процессинг данных:
  - Нормализация, создание «линейных факторов», обработка кат.факторов

Когда использовать линейные модели

В каких случаях целесообразно обучать линейные модели:

- Большие объемы данных
- Данные одной природы
- Существует требование интерпретируемости модели
- Целевая переменная линейно зависит от факторов
    - Хороший способ смешать несколько моделей

Спасибо за внимание!