Практический анализ данных и машинное обучение: искусственные нейронные сети

Ульянкин Филипп

13 июня 2019 г.

Обзор современных архитектур

Agenda

- Частичное обучение
- Как делать разметки
- Нейробайесовские методы (это никак не связано с частичным обучением, у нас два разных сюжета!)

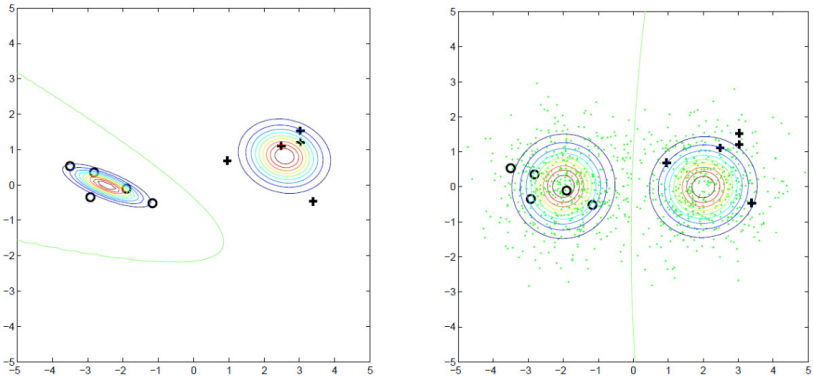
Частичное обучение

Задача частичного обучения

- У нас есть размеченная выборка X^l, для которой известны ответы y^l
- Кроме неё есть большой неразмеченный кусок X^k
- Обычно неразмеченный кусок гораздо больше размеченного, так как у нас мало модераторов, хочется как-то использовать его при обучении

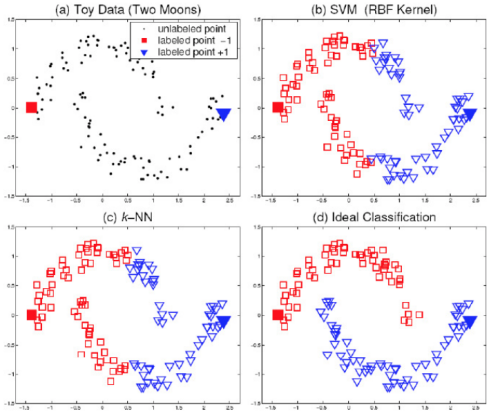
Задача частичного обучения не сводится к классификации

Пример 1. плотности классов, восстановленные:
по размеченным данным X^ℓ по полным данным $X^{\ell+k}$



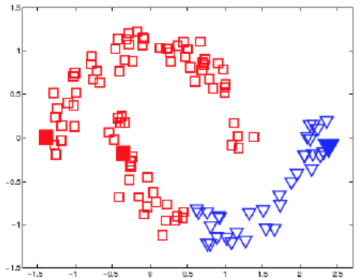
Задача частичного обучения не сводится к классификации

Пример 2. Методы классификации не учитывают кластерную структуру неразмеченных данных



Задача частичного обучения не сводится к кластеризации

Пример 3. Методы кластеризации не учитывают приоритетность разметки.



Простые обёртки над моделями

Self-training (1965)

- Пусть $a(x)$ - наш алгоритм классификации, он выдаёт на выход степень уверенности $a_i = a(x_i)$ (например, вероятности)

- Простейший вариант дополнить разметку следующий:
  1. Учим $a(x)$ на X^l и y^l
  2. Строим прогнозы для X^k
  3. Берём все объекты, где $a_i > M_0$ и добавляем их в обучающую выборку с меткой 1, отсекая с другой стороны получаем объекты с меткой 0
  4. Заново обучаем алгоритм $a(x)$, повторяем пока не надоест
  5. M_0 - гиперпараметр

Co-training (1998)

- Пусть $a_1(x)$ и $a_2(x)$ два очень разных алгоритма, которые используют
  - либо разные признаки;
  - либо разные парадигмы обучения
  - либо разные источники данных $X_1^{l_1}$, $X_2^{l_2}$

- Схема обучения:
  1. Обучаем первый алгоритм на своей подвыборке
  2. Строим прогнозы $a_1(x)$ для X^k
  3. Все наблюдения, где $a_1(x) > M_0$ закидываем в выборку для второго алгоритма
  4. Обучаем второй алгоритм на его подвыборке
  5. строим прогнозы $a_2(x)$ для X^k
  6. Все наблюдения, где $a_2(x) > M_0$ закидываем в выборку для первого алгоритма
  7. Повторяем, пока не надоест

Co-learning (1993)

- Это self-training для композиции алгоритмов
- Решение брать или не брать наблюдение в обучающую выборку принимается на основе голосования композицией алгоритмов

От кластеризации к частичному обучению

Кластеризация как задача дискретной оптимизации

- Пусть $\rho(x, x')$ - функция расстояния между объектами, а $w_{ij} = exp(-\beta\rho(x_i, x_j))$ - веса на парах объектов (близости), где β - параметр

- Задача кластеризации:

$$\sum_{i,j} w_{ij} \cdot [a_i \neq a_j] \to \min_{a_i \in Y}$$

- Задача частичного обучения:

$$\sum_{i,j} w_{ij} \cdot [a_i \neq a_j] + \lambda \cdot \sum_{i=1}^{l} [a_i \neq y_i] \to \min_{a_i \in Y}$$

- То есть мы решаем задачу кластеризации и накладываем штраф на неверную кластеризацию объектов с известными классами

Мораль

Надо просто взять алгоритм кластеризации и модернизировать его так, чтобы был штраф в объектах с известными классами!

Модернизация графовой кластеризации

Пусть мы хотим раздробить выборку на K кластеров, тогда алгоритм графовой кластеризации выглядел бы так:

1. Найти пару вершин (x_i, x_j) с наименьшим расстоянием между ними и соединить ребром
2. Пока выборке остаются изолированные точки, находим изолированную точку и соединяем её с ближайшей
3. В итоге у нас получается остовное дерево
4. Удалим из дерева $K - 1$ самых длинных ребер

Модернизация графовой кластеризации

Для перехода к задаче частичного обучения надо немного поменять последний шаг

1. Найти пару вершин (x_i, x_j) с наименьшим расстоянием между ними и соединить ребром
2. Пока выборке остаются изолированные точки, находим изолированную точку и соединяем её с ближайшей
3. В итоге у нас получается остовное дерево
4. ~~Удалим из дерева $K-1$ самых длинных ребер~~
5. Пока есть путь между двумя вершинами разных классов, будем удалять на этом пути самое длинное ребро

Модернизация иерархической кластеризации

1. Все классы 1 -элементные
2. Ищем пары кластеров с минимальным расстоянием между ними и сливаем, пока все объекты не сольются в единый кластер
3. Считать расстояния между кластерами можно по-разному

Модернизация иерархической кластеризации

1. Все классы 1 -элементные
2. Ищем пары кластеров с минимальным расстоянием между ними и сливаем, пока все объекты не сольются в единый кластер, следим за тем, чтобы при слиянии не было объектов с разными метками классов
3. Считать расстояния между кластерами можно по-разному

От классификации к частичному обучению

От SVM к частичному обучению

- В случае SWM мы пытаемся сделать разделяющую полосу как можно шире, для этого мы минимизируем функцию:

$$\sum_{i=1}^{l}(1 - M_i(w, w_0))_+ + \frac{1}{2c}||w||^2 \to \min_{w,w_0}$$

- Функция $L(M) = (1 - M)_+$ штрафует за уменьшение отступа

- Идея! Функция $L(M) = (1 - |M|)_+$ для объектов без метки будет штрафовать за попадание в зазор между классами

Transductive SVM

- Обучение весов можно провести по частично размеченной выборке:

$$\sum_{i=1}^{l}(1 - M_i(w, w_0))_+ + \frac{1}{2c}||w||^2 + \gamma \cdot \sum_{i=1}^{k}(1 - |M_i(w, w_0)|)_+ \to \min_{w, w_0}$$

- Гиперпараметр γ отвечает за то, насколько много внимания мы уделяем не размеченной части

- Если в выборке нет области разреженности, решение будет неустойчивым

От логрегрессии к частичному обучению

- Обучение логрегрессии происходит в результате максимизации правдоподобия:

$$\sum_{i=1}^{l} \ln P(y_i \mid x_i, w) - \frac{1}{2C} \sum_{y \in Y} ||w_y||^2 \to \max_w$$

- Функция выше для мультиклассовой задачи, если вы ещё не забыли, там у каждого класса свои веса, $P(y \mid x, w)$ - это softmax

- Нам нужно учесть в этом функционале неразмеченные данные

От логрегрессии к частичному обучению

- Пусть $b_j(x)$ - бинарные признаки, $j = 1, \dots, m$.
- Для неразмеченных объектов оценим $P(y \mid b_j(x = 1))$ двумя способами:
  1. Эмпирическая оценка по размеченным данным X^l

$$\hat{p}_j(y) = \frac{\sum_{i=1}^{l} b_j(x_i)[y_i = y]}{\sum_{i=1}^{l} b_j(x_i)}$$

  2. Оценка по неразмеченным данным X^k и линейной модели:

$$\hat{p}_j(y) = \frac{\sum_{i=1}^{k} b_j(x_i) \cdot P(y \mid x_i, w)}{\sum_{i=1}^{k} b_j(x_i)}$$

  3. Хочется, чтобы эти две вероятности были похожи

От логрегрессии к частичному обучению

- Расстояние между распределениями измеряет KL-дивергенция, будем её минимизировать

$$KL(\hat{p}_j(y)||p_j(y,w)) = \sum_y \hat{p}_j(y) \cdot \ln \frac{\hat{p}_j(y)}{p_j(y,w)} \to \min_w$$

- Итоговый функционал получается, если вычесть KL-дивергенцию, посчитанную по всем m признакам из правдоподобия с коэффициентом γ

$$\sum_{i=1}^{l} \ln P(y_i \mid x_i, w) - \frac{1}{2C} \sum_{y \in Y} ||w_y||^2 - \gamma \cdot \sum_{j=1}^{m} \sum_{y \in Y} \hat{p}_j(y) \cdot \ln \frac{\hat{p}_j(y)}{p_j(y,w)} \to \max_w$$

От логрегрессии к частичному обучению

- Оптимизация идёт методом стохастического градиентного спуска
- Метод слабо чувствителен к выбору C и γ
- Метод устойчив к погрешностям оценивания $\widehat{p}_j(y)$
- Не требует большого числа размеченных объектов, хорошо подходит для текстов, показывает неплохую точность
- Пример бинаризации $b_j(x)$ для текстов: [термин j входит в текст x]

Частичное обучение

- Задача занимает промежуточное положение между классификацией и кластеризацией, но не сводится к ним.
- Простые методы-обёртки требуют многократного обучения
- Методы кластеризации легко адаптировать к частичному обучению, введением ограничений (constrained clustering), но это обычно вычислительно сложно
- Методы классификации можно адаптировать чуть сложнее, но это приводит к более эффективному частичному обучению

Как делать разметки?

Как создать свой датасет с Киркоровым и Фейсом

https://habr.com/ru/company/ods/blog/358574/

Что я понял, работая в Data science

- Разметок мало, они плохие

Что я понял, работая в Data science

- Разметок мало, они плохие
- Люди - сволочи

Что я понял, работая в Data science

- Разметок мало, они плохие
- Люди - сволочи
- Толокеры не читают инструкцию, приходится делать на проектах обучение и экзамены

Что я понял, работая в Data science

- Разметок мало, они плохие
- Люди - сволочи
- Толокеры не читают инструкцию, приходится делать на проектах обучение и экзамены
- Люди - жадные сволочи

Что я понял, работая в Data science

- Разметок мало, они плохие
- Люди - сволочи
- Толокеры не читают инструкцию, приходится делать на проектах обучение и экзамены
- Люди - жадные сволочи
- Толокеры хотят побольше денег просто так, приходится следить за качеством разметки, делать ханипоты (примеры, ответы на которые мы знаем), выгонять толокеров, если они делают разметку недобросовестно

Что я понял, работая в Data science

- Разметок мало, они плохие
- Люди - сволочи
- Толокеры не читают инструкцию, приходится делать на проектах обучение и экзамены
- Люди - жадные сволочи
- Толокеры хотят побольше денег просто так, приходится следить за качеством разметки, делать ханипоты (примеры, ответы на которые мы знаем), выгонять толокеров, если они делают разметку недобросовестно

Что я понял, работая в Data science

- Люди - жадные сволочи, которые умеют учиться

Что я понял, работая в Data science

- Люди - жадные сволочи, которые умеют учиться
- Ханипоты приходится обновлять, для этого нужны отдельные процессы

Что я понял, работая в Data science

- Люди - жадные сволочи, которые умеют учиться
- Ханипоты приходится обновлять, для этого нужны отдельные процессы
- Люди - тупые жадные сволочи, которые умеют учиться

Что я понял, работая в Data science

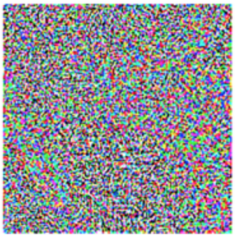
- Люди - жадные сволочи, которые умеют учиться
- Ханипоты приходится обновлять, для этого нужны отдельные процессы
- Люди - тупые жадные сволочи, которые умеют учиться
- Чем больше вопросов в разметке, тем хуже итоговые результаты

Нейробайесовские методы

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

https://www.youtube.com/watch?v=kFe5zSkro0E

Байесовский w2v

- ватерло - лондон - станция - поезд
- ватерло - наполеон - аустерлиц - битва
- ватерло - абба - мама-миа

Байесовский w2v

- S. Bartunov, D. Kondrashkin, A. Osokin, D. Vetrov. Breaking Sticks and Ambiguities with Adaptive Skip-gram. In AISTATS 2016 http://arxiv.org/abs/1502.07257
- Код и документация: https://github.com/sbos/AdaGram.jl
- Предобученная модель: https://yadi.sk/d/W4FtSjA5o3jUL


That's all Folks!