Категориальные признаки на больших данных

Илья Трофимов

24.11.2016

Машинное обучение и большие данные
ФИВТ, осень 2016

Признаки в машинном обучении

Задача обучения с учителем: по обучающей выборке

$$\{\mathbf{x}_i, y_i\}_{i=1}^{n}$$

где $\mathbf{x}_i \in \mathcal{X}$ - это признаки, $y \in \mathcal{Y}$ - ответы, построить функцию

$$f : \mathcal{X} \to \mathcal{Y}$$

так, чтобы минимизировать средний риск

$$\underset{f(\cdot)}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x},y}[L(f(\mathbf{x}), y)]$$

Виды признаков

$$\mathbf{x} = (x_1, \ldots, x_p)$$

Виды признаков

$$\mathbf{x} = (x_1, \ldots, x_p)$$

Виды признаков:

- $x_k \in \{0, 1\}$ - бинарный
- $x_k \in \mathbb{R}$ - количественный (вещественный)
- $x_k \in \mathcal{C}_k,\ |\mathcal{C}_k| < \infty$ - **категориальный** (номинальный)
- $x_k \in \mathcal{C}_k,\ |\mathcal{C}_k| < \infty,\ \mathcal{C}_k -$ упорядочено - порядковый

Виды признаков

Элементы $c \in \mathcal{C}_k$ называются значениями (уровнями, levels) категориальной переменной.

$$\mathcal{C}_k = \{c_{k1}, \ldots, c_{k|\mathcal{C}_k|}\}$$

Примеры категориальных признаков:

- идентификаторы: UserID, ItemID, ShopID, и т.п.
- категория объекта (товар, объявление, клиент), обычно с иерархией
- IP-адрес, регион, город, цвет, и т.п.

Виды признаков

Элементы $c \in \mathcal{C}_k$ называются значениями (уровнями, levels) категориальной переменной.

$$\mathcal{C}_k = \{c_{k1}, \ldots, c_{k|\mathcal{C}_k|}\}$$

Примеры категориальных признаков:

- идентификаторы: UserID, ItemID, ShopID, и т.п.
- категория объекта (товар, объявление, клиент), обычно с иерархией
- IP-адрес, регион, город, цвет, и т.п.

Обычно расширяют исходное признаковое пространство комбинациями (interactions) нескольких признаков:

комбинации 2-й степени: $\mathcal{C}_i \times \mathcal{C}_j$, для всех i $<$ j
комбинации 3-й степени: $\mathcal{C}_i \times \mathcal{C}_j \times \mathcal{C}_k$, для всех i $<$ j $<$ k
и т.д.

Методы работы с категориальными признаками

1. dummy variables, one-hot кодировка
2. Решающие деревья
3. Наивный баейсовский классификатор
4. Счетчики

Методы работы с категориальными признаками

dummy variables, one-hot кодировка.
Каждая переменная $x_k \in \mathcal{C}_k$ заменяется на $|\mathcal{C}_k|$ бинарных z_{km}, $m = 1 \ldots |\mathcal{C}_k|$.
Если $x_k = c_{km}$, то $\mathbf{z} = (0, 0, \ldots, 1, \ldots, 0)$, единица на $m-$й позиции.

Методы работы с категориальными признаками

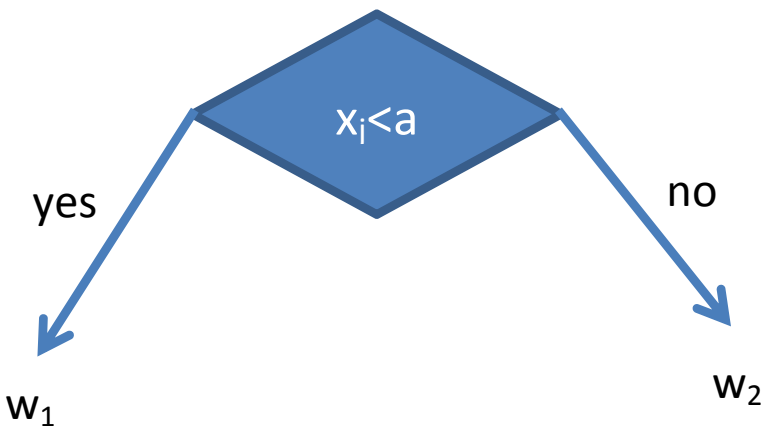
dummy variables, one-hot кодировка.
Каждая переменная $x_k \in \mathcal{C}_k$ заменяется на $|\mathcal{C}_k|$ бинарных $z_{km},\ m = 1 \dots |\mathcal{C}_k|$.
Если $x_k = c_{km}$, то $\mathbf{z} = (0, 0, \dots, 1, \dots, 0)$, единица на $m-$й позиции.

Размерность нового признакового пространства: $\sum_{k=1}^{p} |\mathcal{C}_k|$
После этого можно применять стандартные методы машинного обучения для количественных признаков.
Для больших выборок вычислительно эффективны будут **линейные модели (VW)**

Решающие деревья

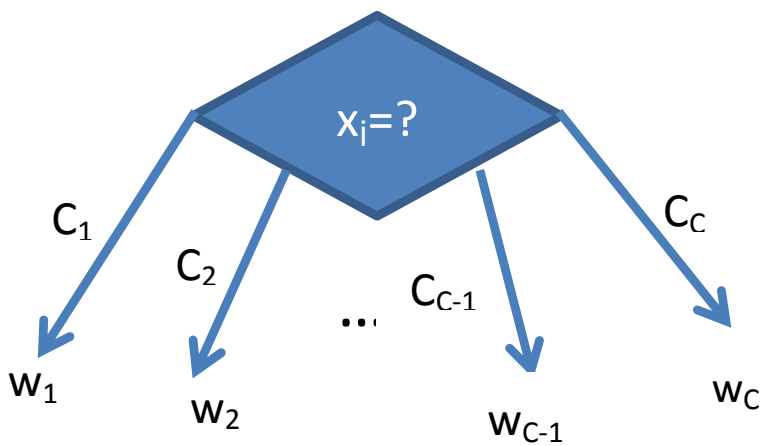
Пример: дерево регрессии, квадратичная ошибка

$$\min_{j,w_1,w_2} \sum_{i=1}^{n} \left((y_i - w_1)^2 [\mathbf{x}_{ij} < a] + (y_i - w_2)^2 [\mathbf{x}_{ij} \geq a]\right)$$



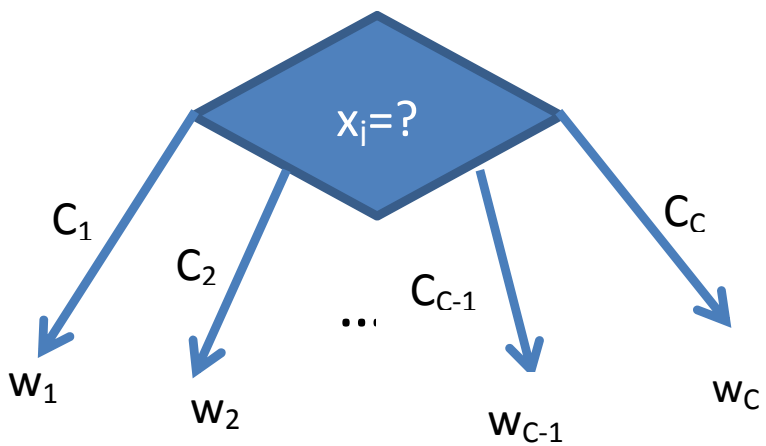
Решающие деревья

Ветвления в решающих деревьях по значениям категориальной переменной (CART, GBDT, random forest, ...).
При большом $|\mathcal{C}_k|$ ветвление будет выбираться слишком часто, это чревато переобучением.



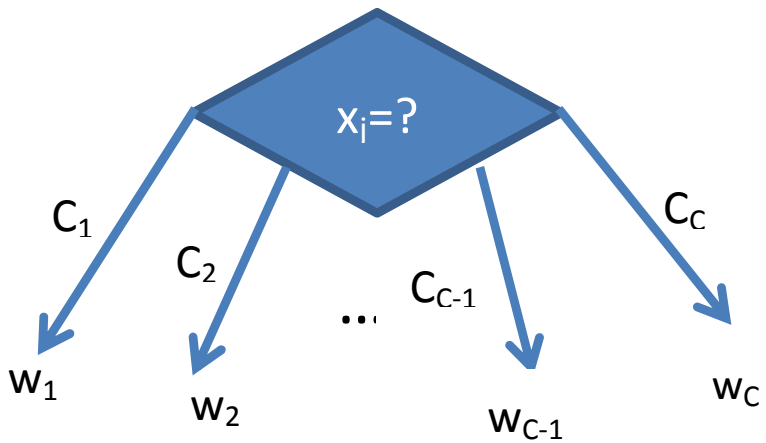
Решающие деревья

Ветвления в решающих деревьях по значениям категориальной переменной (CART, GBDT, random forest, ...).
При большом $|\mathcal{C}_k|$ ветвление будет выбираться слишком часто, это чревато переобучением.



Решающие деревья

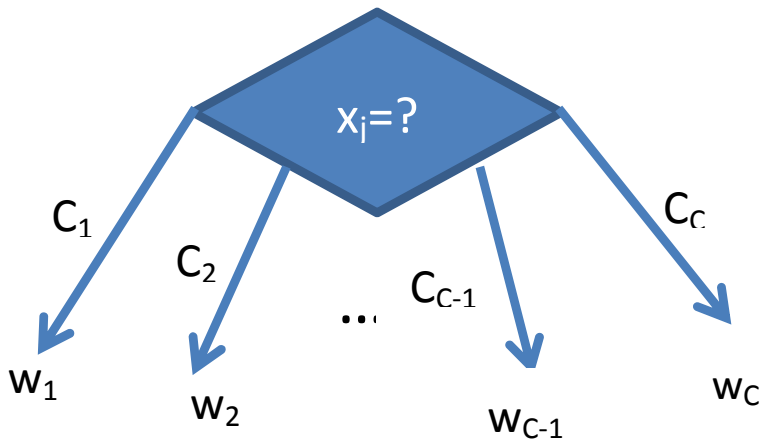
Ветвления в решающих деревьях по значениям категориальной переменной (CART, GBDT, random forest, ...).
При большом $|\mathcal{C}_k|$ ветвление будет выбираться слишком часто, это чревато переобучением.



$$\min_{j,w_1,\ldots,w_C} \sum_{i=1}^{n} \sum_{c=1}^{C} (y_i - w_c)^2 [\mathbf{x}_{ij} = C_c]$$

Решающие деревья

Ветвления в решающих деревьях по значениям категориальной переменной (CART, GBDT, random forest, . . .). При большом $|\mathcal{C}_k|$ ветвление будет выбираться слишком часто, это чревато переобучением.



$$\min_{j,w_1,\ldots,w_C} \sum_{i=1}^{n}\sum_{c=1}^{C}(y_i - w_c)^2[\mathbf{x}_{ij} = C_c] + \gamma C + \frac{1}{2}\lambda\sum_{c=1}^{C} w_c^2$$

см. XGBoost `http://xgboost.readthedocs.io/en/latest/model.html`

Наивный байесовский классификатор

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|y)P(y)}{\sum_t P(\mathbf{x}|t)P(t)}$$

$$y^* = \operatorname*{argmax}_y P(y|\mathbf{x}) = \operatorname*{argmax}_y P(\mathbf{x}|y)P(y)$$

Наивный байесовский классификатор

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|y)P(y)}{\sum_t P(\mathbf{x}|t)P(t)}$$

$$y^* = \operatorname*{argmax}_y P(y|\mathbf{x}) = \operatorname*{argmax}_y P(\mathbf{x}|y)P(y)$$

Почему Байес такой «наивный»? Предполагаем, что

$$P(\mathbf{x}|y) = \prod_k P_k(x_k|y)$$

Наивный байесовский классификатор

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|y)P(y)}{\sum_t P(\mathbf{x}|t)P(t)}$$

$$y^* = \operatorname*{argmax}_y P(y|\mathbf{x}) = \operatorname*{argmax}_y P(\mathbf{x}|y)P(y)$$

Почему Байес такой «наивный»? Предполагаем, что

$$P(\mathbf{x}|y) = \prod_k P_k(x_k|y)$$

$$y^* = \operatorname*{argmax}_y P(y|\mathbf{x}) = \operatorname*{argmax}_y P(\mathbf{x}|y)P(y)$$

$$= \operatorname*{argmax}_y \left(\prod_k P_k(x_k|y)P(y) \right)$$

Наивный байесовский классификатор

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|y)P(y)}{\sum_t P(\mathbf{x}|t)P(t)}$$

$$y^* = \underset{y}{\operatorname{argmax}}\, P(y|\mathbf{x}) = \underset{y}{\operatorname{argmax}}\, P(\mathbf{x}|y)P(y)$$

Почему Байес такой «наивный»? Предполагаем, что

$$P(\mathbf{x}|y) = \prod_k P_k(x_k|y)$$

$$y^* = \underset{y}{\operatorname{argmax}}\, P(y|\mathbf{x}) = \underset{y}{\operatorname{argmax}}\, P(\mathbf{x}|y)P(y)$$

$$= \underset{y}{\operatorname{argmax}} \left(\prod_k P_k(x_k|y)P(y) \right)$$

Если x_k - категориальная переменная, то

$$P_k(x_k|y) = \frac{|\{i|x_{ik} = x_k \;\&\; y_i = y\}|}{|\{i|y_i = y\}|}$$

Сравнение методов

Желаемые свойства методов:

- **Точность** - хорошее качество предсказания
- **Масштабируемость** - поддержка больших n, p при обучении (с учетом комбинаций!)
- **Адаптируемость** - умение подстраиваться к изменяющемуся распределению $P(y|\mathbf{x})$

Сравнение методов

Желаемые свойства методов:

- **Точность** - хорошее качество предсказания
- **Масштабируемость** - поддержка больших n, p при обучении (с учетом комбинаций!)
- **Адаптируемость** - умение подстраиваться к изменяющемуся распределению $P(y|\mathbf{x})$

Метод	**Точность**	**Масштаб.**	**Адапт.**
one-hot encoding, linear			
one-hot encoding, linear + hash			
множ. ветвления в деревьях			
наивный байес			

Сравнение методов

Желаемые свойства методов:

- **Точность** - хорошее качество предсказания
- **Масштабируемость** - поддержка больших n, p при обучении (с учетом комбинаций!)
- **Адаптируемость** - умение подстраиваться к изменяющемуся распределению $P(y|\mathbf{x})$

Метод	**Точность**	**Масштаб.**	**Адапт.**
one-hot encoding, linear	$+$	$-$	$\pm$
one-hot encoding, linear + hash			
множ. ветвления в деревьях			
наивный байес			

Сравнение методов

Желаемые свойства методов:

- **Точность** - хорошее качество предсказания
- **Масштабируемость** - поддержка больших n, p при обучении (с учетом комбинаций!)
- **Адаптируемость** - умение подстраиваться к изменяющемуся распределению $P(y|\mathbf{x})$

Метод	**Точность**	**Масштаб.**	**Адапт.**
one-hot encoding, linear	+	−	±
one-hot encoding, linear + hash	+	+	±
множ. ветвления в деревьях			
наивный байес			

Сравнение методов

Желаемые свойства методов:

- **Точность** - хорошее качество предсказания
- **Масштабируемость** - поддержка больших n, p при обучении (с учетом комбинаций!)
- **Адаптируемость** - умение подстраиваться к изменяющемуся распределению $P(y|\mathbf{x})$

Метод	**Точность**	**Масштаб.**	**Адапт.**
one-hot encoding, linear	+	−	±
one-hot encoding, linear + hash	+	+	±
множ. ветвления в деревьях	+	−	−
наивный байес			

Сравнение методов

Желаемые свойства методов:

- **Точность** - хорошее качество предсказания
- **Масштабируемость** - поддержка больших n, p при обучении (с учетом комбинаций!)
- **Адаптируемость** - умение подстраиваться к изменяющемуся распределению $P(y|\mathbf{x})$

Метод	**Точность**	**Масштаб.**	**Адапт.**
one-hot encoding, linear	+	−	±
one-hot encoding, linear + hash	+	+	±
множ. ветвления в деревьях	+	−	−
наивный байес	−	+	+

Наивный байесовский классификатор

$$P(y|\mathbf{x}) = \frac{\prod_k P_k(x_k|y)P(y)}{P(\mathbf{x})}$$

Наивный байесовский классификатор

$$P(y|\mathbf{x}) = \frac{\prod_k P_k(x_k|y)P(y)}{P(\mathbf{x})}$$

$$P_k(x_k|y) = \frac{P_k(y|x_k)P_k(x_k)}{P(y)}$$

Наивный байесовский классификатор

$$P(y|\mathbf{x}) = \frac{\prod_k P_k(x_k|y)P(y)}{P(\mathbf{x})}$$

$$P_k(x_k|y) = \frac{P_k(y|x_k)P_k(x_k)}{P(y)}$$

$$P(y|\mathbf{x}) = \frac{\prod_k \frac{P_k(y|x_k)P_k(x_k)}{P(y)} P(y)}{P(\mathbf{x})} =$$

$$= \left(\prod_k P_k(y|x_k)\right) \left(\prod_k \frac{P_k(x_k)}{P(y)}\right) \frac{P(y)}{P(\mathbf{x})}$$

Наивный байесовский классификатор

$$P(y|\mathbf{x}) = \frac{\prod_k P_k(x_k|y)P(y)}{P(\mathbf{x})}$$

$$P_k(x_k|y) = \frac{P_k(y|x_k)P_k(x_k)}{P(y)}$$

$$P(y|\mathbf{x}) = \frac{\prod_k \frac{P_k(y|x_k)P_k(x_k)}{P(y)} P(y)}{P(\mathbf{x})} =$$

$$= \left(\prod_k P_k(y|x_k)\right) \left(\prod_k \frac{P_k(x_k)}{P(y)}\right) \frac{P(y)}{P(\mathbf{x})}$$

$$y^* = \operatorname*{argmax}_y \left(\prod_k P_k(y|x_k)\right) P(y)^{1-p}$$

Счетчики

Обозначим $F_k(c) = \{j \mid x_{jk} = c\}$.
Вычислим «эмпирические средние» по каждому значению категориальных переменных

$$\hat{y}(k, c) = \frac{\sum_{j \in F_k(c)} y_j}{|F_k(c)|}$$

Счетчики

Обозначим $F_k(c) = \{j \mid x_{jk} = c\}$.
Вычислим «эмпирические средние» по каждому значению категориальных переменных

$$\hat{y}(k, c) = \frac{\sum_{j \in F_k(c)} y_j}{|F_k(c)|} \approx E[y | x_k = c]$$

Счетчики

Обозначим $F_k(c) = \{j \mid x_{jk} = c\}$.
Вычислим «эмпирические средние» по каждому значению категориальных переменных

$$\hat{y}(k, c) = \frac{\sum_{j \in F_k(c)} y_j}{|F_k(c)|} \approx E[y | x_k = c]$$

и заменим вектора признаков

$$(x_{i1}, x_{i2}, \ldots, x_{ip}) \to (\hat{y}(1, x_{ik}), \hat{y}(2, x_{2k}), \ldots, \hat{y}(p, x_{pk}))$$

Счетчики

Обозначим $F_k(c) = \{j \mid x_{jk} = c\}$.
Вычислим «эмпирические средние» по каждому значению категориальных переменных

$$\hat{y}(k, c) = \frac{\sum_{j \in F_k(c)} y_j}{|F_k(c)|} \approx E[y | x_k = c]$$

и заменим вектора признаков

$$(x_{i1}, x_{i2}, \ldots, x_{ip}) \to (\hat{y}(1, x_{ik}), \hat{y}(2, x_{2k}), \ldots, \hat{y}(p, x_{pk}))$$

$$(UserID : 12398234, Region : Moscow, IP : 127.123.10.20) \to (0.1, 0.33, 0.001)$$

После такой замены мы получаем обучающую выборку с количественными признаками, которую можно подавать на вход алгоритму машинного обучения. Рекомендуется использовать хороший нелинейный метод, например **бустинг решающих деревьев**

Эмпирическое среднее

Обозначим $F_k(c) = \{j \mid x_{jk} = c\}$.
Вычислим «эмпирические средние» по каждому значению категориальных переменных

$$\hat{y}(k, c) = \frac{\sum_{j \in F_k(c)} y_j}{|F_k(c)|}$$

Эмпирическое среднее

Обозначим $F_k(c) = \{j \mid x_{jk} = c\}$.
Вычислим «эмпирические средние» по каждому значению категориальных переменных

$$\hat{y}(k, c) = \frac{\sum_{j \in F_k(c)} y_j}{|F_k(c)|}$$

Вариант, сглаживание средних:

$$\hat{y}(k, c) = \frac{\sum_{j \in F_k(c)} y_j + \bar{y} n_0}{|F_k(c)| + n_0}$$

Параметры $\bar{y}, n_0$ подбираются минимизацией ошибки на валидационном множестве

Эмпирическое среднее

Обозначим $F_k(c) = \{j \mid x_{jk} = c\}$.
Вычислим «эмпирические средние» по каждому значению категориальных переменных

$$\hat{y}(k, c) = \frac{\sum_{j \in F_k(c)} y_j}{|F_k(c)|}$$

Вариант, сглаживание средних:

$$\hat{y}(k, c) = \frac{\sum_{j \in F_k(c)} y_j + \bar{y} n_0}{|F_k(c)| + n_0}$$

Параметры $\bar{y}, n_0$ подбираются минимизацией ошибки на валидационном множестве
Вариант: использовать в качестве признаков несколько $\hat{y}(k, c)$ с разными $\bar{y}, n_0$

Счетчики

Можно сделать еще лучше. Каждая переменная x_{ik} заменяется на 3:

1. $\hat{y}(k, x_{ik})$
2. $\sum_{j \in F_k(x_{ik})} y_j$
3. $|F_k(x_{ik})|$

Счетчики

Можно сделать еще лучше. Каждая переменная x_{ik} заменяется на 3:

1. $\hat{y}(k, x_{ik})$, эмпирическое среднее
2. $\sum_{j \in F_k(x_{ik})} y_j$, ?
3. $|F_k(x_{ik})|$, мера уверенности

Как разбивать выборку

Вариант 1 (неправильный).

Часть 1 (большая)	**Часть 3**
Вычисление счетчиков, обучение регрессии	Тест

Как разбивать выборку

Вариант 2.

Часть 1 (большая)	**Часть 2 (маленькая)**	**Часть 3**
Вычисление счетчиков	Обучение регрессии	Тест

Как разбивать обучающую выборку

Вариант 2. Если в данных есть временные метки:

Период 1 (большой)	**Период 2 (маленький)**	**Период 3**
Вычисление счетчиков	Обучение регрессии	Тест

Ось времени

---------->

Как разбивать обучающую выборку

Вариант 3. Онлайн-счетчики, если в данных есть временные метки:

Период 1 (большой)	**Период 3**
Онлайн вычисление счетчиков, обучение регрессии	Тест

Ось времени

---------->

Вычисление большого числа счетчиков

Подход 1. Оффлайн. MapReduce

Вычисление большого числа счетчиков

Подход 1. Оффлайн. MapReduce

(key = "i", value = "$y_i \;\; 1 : x_{i1} \;\; 2 : x_{i2} \;\; \ldots \;\; p : x_{ip}$")

Вычисление большого числа счетчиков

Подход 1. Оффлайн. MapReduce

(key = "i", value = "y_i 1 : x_{i1} 2 : x_{i2} ... p : x_{ip}")

map:

(key = "1 x_{i1}", value = "y_i"), ..., (key = "p x_{ip}", value = "y_i")

Вычисление большого числа счетчиков

Подход 1. Оффлайн. MapReduce

(key = "i", value = "y_i $1 : x_{i1}$ $2 : x_{i2}$ $\ldots$ $p : x_{ip}$")

map:

(key = "1 x_{i1}", value = "y_i"), $\ldots$, (key = "p x_{ip}", value = "y_i")

reduce:

(key = "1 c_{11}", value = "$\sum_{i \in F_1(c_{11})} y_i$")

(key = "1 c_{12}", value = "$\sum_{i \in F_1(c_{12})} y_i$")

$\ldots$

(key = "p $c_{p|C_p|}$", value = "$\sum_{i \in F_p(c_{p|C_p|})} y_i$")

Вычисление большого числа счетчиков

Подход 1. Оффлайн. MapReduce

$(\text{key} = "i", \text{value} = "y_i \;\; 1 : x_{i1} \;\; 2 : x_{i2} \;\; \ldots \;\; p : x_{ip}")$

map:

$(\text{key} = "1\ x_{i1}", \text{value} = "y_i"), \ldots, (\text{key} = "p\ x_{ip}", \text{value} = "y_i")$

reduce:

$(\text{key} = "1\ c_{11}", \text{value} = "\sum_{i \in F_1(c_{11})} y_i")$

$(\text{key} = "1\ c_{12}", \text{value} = "\sum_{i \in F_1(c_{12})} y_i")$

$\ldots$

$(\text{key} = "p\ c_{p|C_p|}", \text{value} = "\sum_{i \in F_p(c_{p|C_p|})} y_i")$

Алгоритм WordCount

Count-Min Sketch

Подход 2. Онлайн

Проблема: нужно считать онлайн n счетчиков

Count-Min Sketch

Подход 2. Онлайн

Проблема: нужно считать онлайн n счетчиков

Обновление: (i_t, c_t) : $\quad a_{i_t} \leftarrow a_{i_t} + c_t$

Count-Min Sketch

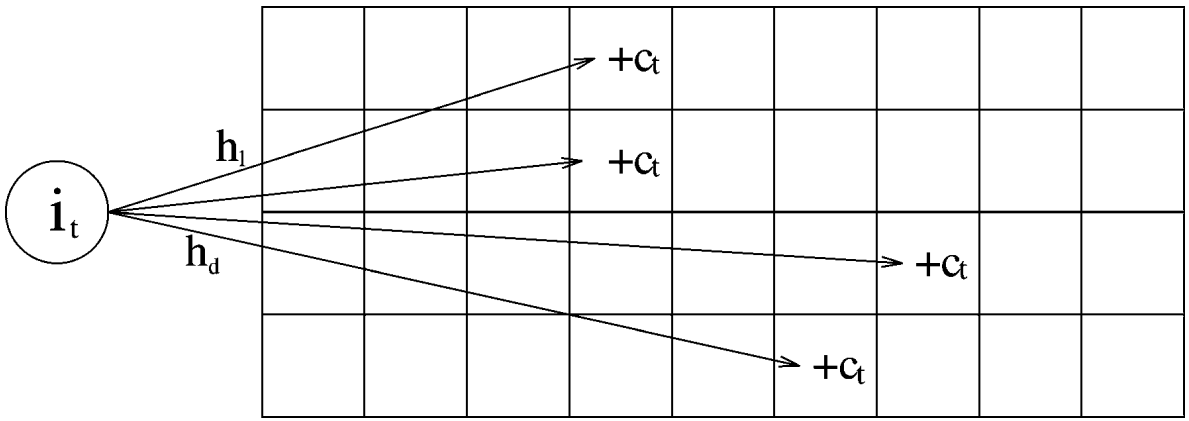
Подход 2. Онлайн

Проблема: нужно считать онлайн n счетчиков

Обновление: (i_t, c_t) : $\quad a_{i_t} \leftarrow a_{i_t} + c_t$

Count-Min Sketch с параметрами (ϵ, δ) это

- таблица размера $w \times d$, где $w = \lceil \frac{e}{\epsilon} \rceil$, $d = \lceil \ln \frac{1}{\delta} \rceil$
- d хеш-функций $h_1 \dots h_d : \{1 \dots n\} \to \{1 \dots w\}$



Обновление:

$$\forall\, 1 \le j \le d : \quad count[j, h_j(i)] \leftarrow count[j, h_j(i)] + c_t$$

Count-Min Sketch

Приближенное значение a_i:

$$\hat{a}_i = \min_j count[j, h_j(i)]$$

Count-Min Sketch

Приближенное значение a_i:

$$\hat{a}_i = \min_j count[j, h_j(i)]$$

Теорема. С вероятностью как минимум $1 - \delta$ будет выполняться

$$\hat{a}_i \leq a_i + \epsilon \sum_{k=1}^{n} |a_k|$$

Сравнение методов

Желаемые свойства методов:

- **Точность** - хорошее качество предсказания
- **Масштабируемость** - поддержка больших n, p при обучении
- **Адаптируемость** - подстраиваться к изменяющемуся распределению $P(y|\mathbf{x})$

Метод	**Точность**	**Масштаб.**	**Адапт.**
one-hot encoding, linear	+	−	±
one-hot encoding, linear + hash	+	+	±
множ. ветвления	+	−	−
наивный байес	−	+	+
счетчики	+	+	+

Декрементальное обучение

Забывание обучающих примеров

Декрементальное обучение

Каждая переменная x_{ik} заменяется на 3:

1. $\hat{y}(k, x_{ik})$
2. $\sum_{j \in F_k(x_{ik})} y_j$
3. $|F_k(x_{ik})|$

Декрементальное обучение

Каждая переменная x_{ik} заменяется на 3:

1. $\hat{y}(k, x_{ik})$
2. $\sum_{j \in F_k(x_{ik})} y_j$
3. $|F_k(x_{ik})|$

Если нужно «забыть» обучающие примеры G, то нужно просто уменьшить счетчики:

$$\sum_{j \in F_k(x_{ik})} y_j \to \sum_{j \in F_k(x_{ik})} y_j - \sum_{j \in G} y_j$$

$$|F_k(x_{ik})| \to |F_k(x_{ik})| - |G|$$

$$\hat{y}(k, x_{ik}) = \frac{\sum_{j \in F_k(x_{ik})} y_j - \sum_{j \in G} y_j}{|F_k(x_{ik})| - |G|}$$

Список литературы

Bilenko `https://yandexdataschool.com/conference/2015/program/bilenko`

CountMin Cormode, G., Muthukrishnan, S. An improved data stream summary: The count-min sketch and its applications.