Энтропия

Терверессы: Лиза, Олеся, Даша, Марина, Яна

БЭК 171

Откуда появилась?

- 1948 год - Клод Шеннон создал первую, истинно математическую, теорию энтропии
- Его идеи послужили основой разработки двух основных направлений: теории информации и теории кодирования

Для дискретных случайных величин

- Энтропия – наименьшее среднее число бит, необходимое для кодирования некоторой информации.
$$H = -\sum_{i=1}^{n} p_i \log p_i,$$
где p_i — вероятность i-го исхода
- Условная энтропия — количество бит, необходимое для того, чтобы узнать значение случайной величины Y при условии, что случайная величина X известна.
$$H(Y|X) = -\sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)}$$
- Совместная энтропия — степень неопределенности, связанная со множеством случайных величин.
$$H(X,Y) = -\sum_x \sum_y p(x,y) \log p(x,y)$$
- Взаимная информация $I(X;Y)$ — мера взаимной зависимости двух случайных величин.
$$I(X;Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

Свойства

- $H \geqslant 0$
- $H(Y|X) = H(X,Y) - H(X)$
- $H(Y|X) \leqslant H(Y)$
- $H(X,Y) = H(X|Y) + H(Y|X) + I(X;Y) = H(X) + H(Y) - I(X;Y)$, где $I(X;Y)$ – взаимная информация о случайных величинах X и Y
- $I(X;Y) \leqslant H(X)$, $I(X;Y) = H(X) - H(X|Y)$

- Кросс энтропия — минимальное среднее количество бит, необходимое для того, чтобы закодировать информацию, если схема кодирования базируется на некотором распределении q, а не истинном, p.
$$CE(P||Q) = -\sum_{i=1}^{n} p_i \log q_i$$
- Дивергенция Кульбака – Лейблера — степень отдаленности одного вероятностного распределения от другого.
$$D_{KL}(P\,||\,Q) = -\sum_{i=1}^{n} p_i \log q_i \; - (-\sum_{i=1}^{n} p_i \log p_i)$$

Пример. Текст задачи

Для непрерывных случайных величин

- Самая главная и простая энтропийка:
$$H(X) = -\int_{-\infty}^{+\infty} f(x) \log f(x) dx$$
- Условная энтропия:
$$H(Y|X) = -\int_{-\infty}^{+\infty} f(x,y) \log f_{Y|X}(y) dy$$
- Совместная энтропия:
$$H(X,Y) = -\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y) \log f(x,y) dx dy$$
- Взаимная информация:
$$I(X;Y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y) \log \frac{f(x,y)}{f(x)f(y)} dx dy$$
- Кросс-энтропия:
$$CH(p,q) = -\int_{-\infty}^{+\infty} p(x) \log q(x) dx$$
- Дивергенция Кульбака – Лейблера:
$$D_{KL}(P\,||\,Q) = \int_{-\infty}^{+\infty} p(x) \log p(x) dx - \int_{-\infty}^{+\infty} p(x) \log q(x) dx$$

Пример. Текст задачи

Применение

Энтропийное кодирование

Энтропия показывает наименьшее среднее число бит, необходимое для кодирования некоторой информации. С целью минимизации энтропии и оптимизации кода элементы с большой вероятностью появления кодируются меньшим числом символов. Это позволяет передавать большее количество информации, затрачивая меньший объем памяти.

Построение решающих деревьев

Каждое ветвление дерева представляет собой разделение выборки на две части по порогу некоторого признака. Расчет энтропии помогает определить оптимальный порог для каждого узла — при котором взвешенная сумма энтропий получившихся выборок минимальна среди возможных разбиений.

Например, у нас есть выборка объектов с одним признаком, длина: обыкновенный удав (22 попугая), анаконда (46 попугаев), анаконда (40 попугаев), обыкновенный удав (31 попугай).

Попробуем разделить выборку по 38 попугаям:

Длина $<$ 38 попугаев

Да — Нет

Обыкновенный удав, обыкновенный удав — Анаконда, анаконда

При расчете энтропии $0 \cdot \log_2 0$ считается равным 0, несмотря на $\log_2 0$. За вероятность принимается вероятность встретить данный класс в новой выборке.

Энтропия левой части: $-(1 \cdot \log_2 1 + 0 \cdot \log_2 0) = 0$.

Энтропия правой части: $-(1 \cdot \log_2 1 + 0 \cdot \log_2 0) = 0$.

Суммарная энтропия получилась: $\frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 0 = 0$, $\frac{1}{2}$ — доля каждой выборки в исходной.

Так как 0 — минимально возможное значение энтропии, критерий «длина $<$ 38 попугаев» дает оптимальный результат.

Применение в алгоритме UMAP

В анализе данных алгоритмы снижения размерности используют кросс-энтропию как показатель эффективности перенесения свойств объектов. Чем меньше кросс-энтропия, тем ближе к истинному оказалось подобранное отображение.

Приведем пример работы алгоритма UMAP. Мы возьмем набор данных об одежде, который включает в себя 70000 черно-белых изображений различной одежды по 10 классам: футболки, брюки, свитеры, платья, кроссовки и т.д. Каждая картинка имеет размер 28x28 пикселей или 784 пикселя.

Результатом преобразования будет следующее отображение:

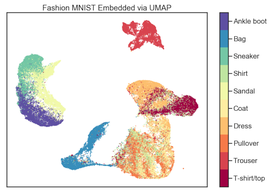

Рис. 1:Алгоритм UMAP

UMAP строит ориентированный взвешенный граф: ребрами соединяются каждый объект с наиболее похожими на него из выборки. Вес ребра можно интерпретировать как вероятность его существования. Тогда ребро e является случайной величиной: $e \sim B(w(e))$. Множество ребер построенного графа — множество E из случайных величин Бернулли.

Чтобы перенести граф в низкоразмерное пространство, UMAP подбирает для множества E_h похожее на него множество E_l с функцией $w_l(e)$, соответствующие низкоразмерному пространству

Для этого UMAP минимизирует сумму дивергенций Кульбака-Лейблера для каждой случайной величины из множеств:

$$S(E_h||E_l) = \sum_{e \in E} w_h(e) \log \frac{w_h(e)}{w_l(e)} + (1 - w_h(e)) \log \left(\frac{1 - w_h(e)}{1 - w_l(e)}\right) \to \min_{w_l}$$

Результатом является граф в низкоразмерном пространстве с подобранной функцией весов w_l.

Contact Information

- Web: http://www.YourWebsite.com/lab
- Email: youremail@emailprovider.com