Методология Importance Sampling на основе LSH-бакетов для обучения моделей на больших данных

1 Введение

Ниже представлено описание метода комбинирования Locality-Sensitive Hashing (LSH) и Importance Sampling для эффективного обучения моделей на больших данных. Метод позволяет сократить вычислительные затраты при сохранении статистических свойств исходного распределения данных. Приводятся теоретические обоснования, практическая реализация и рекомендации по применению метода.

Locality-Sensitive Hashing (LSH) — вероятностный метод приближенного поиска ближайших соседей в многомерных пространствах. Алгоритм проектирует данные в пространство меньшей размерности с сохранением свойств локальности: семантически близкие объекты с высокой вероятностью попадают в одинаковые бакеты.

Importance Sampling — статистический метод оценки свойств распределения по взвешенной выборке. В контексте машинного обучения позволяет проводить обучение на репрезентативной подвыборке данных с компенсацией смещения через весовые коэффициенты.

Комбинация этих методов представляет практический интерес для обработки крупномасштабных данных, где полное обучение на всех данных computationally expensive.

2 Теоретическое обоснование

2.1 Стратификация данных посредством LSH

LS-хэширование естественным образом осуществляет стратификацию данных — разделение на однородные подгруппы (страты) по признаку семантической близости. Формально, для семейства хэш-функций $\mathcal{H}$ и метрики d выполняется:

$$Pr[h(x) = h(y)] = sim(x, y)$$

где $sim(x, y)$ — функция сходства, монотонно убывающая относительно $d(x, y)$.

Бакеты LSH можно рассматривать как страты с высокой внутригрупповой когерентностью. Выборка из таких страт сохраняет свойства исходного распределения при условии репрезентативного покрытия пространства данных.

2.2 Теория Importance Sampling

Для несмещенной оценки математического ожидания функции потерь $\mathcal{L}$ на распределении $p(x)$ по выборке из распределения $q(x)$ вводится весовой коэффициент:

$$w(x) = \frac{p(x)}{q(x)}$$

Оценка приобретает вид:

$$\mathbb{E}_p[\mathcal{L}] \approx \frac{1}{n} \sum_{i=1}^{n} w(x_i)\mathcal{L}(x_i)$$

В контексте LHS-бакетов, $q(x)$ соответствует вероятности выборки из конкретного бакета.

2.3 Статистическое обоснование репрезентативности выборки

Согласно теоретическим результатам, представленным в работе *Why locality sensitive hashing works: A practical perspective*, вероятность коллизии $P_{\mathbf{h}_k^\ell}(r)$ для точек на расстоянии r может быть выражена как:

$$P_{\mathbf{h}_k^\ell}(r) = 1 - (1 - p_{\mathbf{h}}(r)^k)^\ell$$

где $p_{\mathbf{h}}(r)$ — вероятность коллизии для отдельной хэш-функции, k и ℓ — параметры LSH-схемы.

Важное наблюдение: для большинства реальных датасетов фактический recall rate $P_{h_k^\ell}(\mathcal{D}_r)$ очень близок к теоретическому ожиданию $P_{\mathbf{h}_k^\ell}(r)$. Это означает, что дисперсия recall rate крайне мала при правильном выборе параметров k и ℓ и достаточно большом объеме данных.

Данное свойство обеспечивает статистические гарантии репрезентативности выборки, полученной из LSH-бакетов. Конкретно, величина ошибки аппроксимации может быть ограничена с помощью неравенства Чебышева:

$$Pr\left[|P_{h_k^\ell}(\mathcal{D}_r) - P_{\mathbf{h}_k^\ell}(r)| \geq \varepsilon\right] \leq \frac{\mathbb{D}[P_{h_k^\ell}(\mathcal{D}_r)]}{\varepsilon^2}$$

где дисперсия $\mathbb{D}[P_{h_k^\ell}(\mathcal{D}_r)]$ уменьшается с ростом количества хэш-функций и размера бакетов.

3 Практическая реализация

3.1 Построение LSH-индекса

Критичные параметры настройки:

- k — количество хэш-функций (определяет точность группировки)
- w — ширина бакета (влияет на granularity стратификации)
- L — количество хэш-таблиц (увеличивает recall)

3.2 Стратегии выборки

Алгоритм построения:

1. Для каждой из L хэш-таблиц сгенерировать k независимых хэш-функций
2. Для каждого объекта $x \in \mathcal{D}$ вычислить L супер-хэшей:
$$H_j(x) = (h_{j,1}(x), \ldots, h_{j,k}(x)), \quad j = 1..L$$
3. Поместить идентификатор объекта x в бакет с ключом $H_j(x)$ для каждой из L таблиц

Уточнение:

Поскольку объект может находиться в нескольких бакетах (из-за использования L таблиц), при формировании выборки необходимо устранять дублирование. Для объекта, попавшего в выборку из нескольких бакетов, учитывается его вес, усредненный по всем бакетам, в которые он попал.

Рассматриваемые подходы:

- **Пропорциональная выборка:**
$$n_i = \frac{|B_i|}{N} \cdot S$$
- **Сбалансированная выборка:**
$$n_i = \min(c, |B_i|)$$
- **Оптимизированная выборка (на основе дисперсии):**
$$n_i \propto |B_i| \cdot \sigma_{B_i}$$
где σ_{B_i} — оценка стандартного отклонения целевой переменной или функции потерь внутри бакета B_i, S — общий размер выборки, c — константа ограничения.

3.3 Расчет весовых коэффициентов

Для точки $x \in B_i$ вероятность её включения в выборку $\pi(x)$ равна вероятности того, что бакет B_i будет выбран для семплинга И точка x будет выбрана внутри бакета.

Пусть $P_{\text{select}}(B_i)$ — вероятность выбора бакета B_i для семплинга. Тогда:

$$\pi(x) = P_{\text{select}}(B_i) \cdot \frac{n_i}{|B_i|}$$

Следовательно, вес для точки x вычисляется по классической формуле Importance Sampling:

$$w(x) = \frac{1}{\pi(x) \cdot N} = \frac{1}{N} \cdot \frac{1}{P_{\text{select}}(B_i)} \cdot \frac{|B_i|}{n_i}$$

Для пропорциональной выборки, где $P_{\text{select}}(B_i) = \frac{|B_i|}{N}$, формула сводится к:

$$w(x) = \frac{1}{N} \cdot \frac{N}{|B_i|} \cdot \frac{|B_i|}{n_i} = \frac{1}{n_i}$$

3.4 Обучение с взвешиванием

Модификация функции потерь:

$$\mathcal{L} = \frac{1}{\sum_{i=1}^{S} w(x_i)} \sum_{i=1}^{S} w(x_i) \cdot \ell(f(x_i), y_i)$$

Реализация требует поддержки взвешенного обучения в используемых алгоритмах.

Таблица 1: Сравнение стратегий семплинга на основе LSH

Стратегия	**Преимущества**	**Недостатки**	**Подходящие сценарии**
Пропорциональный семплинг	Простота реализации, несмещённость	Может быть неэффективен для редких классов	Данные с равномерным распределением
Сбалансированный семплинг	Лучшее покрытие редких классов	Требует настройки максимального размера бакета	Данные с длинным хвостом распределения
Multi-probe LSH	Улучшенное покрытие соседних областей	Увеличивает вычислительную стоимость	Высокоточные приложения
Query-aware LSH	Высокая релевантность для конкретных запросов	Сложность реализации	Рекомендательные системы, поиск

4 Ограничения метода

1. Сильная зависимость качества от параметров LSH
2. Чувствительность к аномалиям и выбросам в данных
3. Вычислительная сложность построения индексной структуры
4. Необходимость эмпирического подбора размера выборки