A-012

クラスタを用いた学習型インデックスにおけるデータ更新処理の改善
Improvement of data update process in learned index using clusters

田 佳良† 山崎 憲一†
Tian JiaLiang Kenichi Yamazaki

1. はじめに

学習型インデックスは，キーに対応するデータの保存・検索（クエリ）を行う機構である．データ分布を学習して，key の値からデータの場所を予測するにより，クエリの高速化とメモリ使用量の削減を両立する．従来の B-tree インデックスを置き換えるデータ構造として期待されているが，データの追加・削除時にインデックスの更新に手間が掛かるという問題がある．この問題を解決するため，データセットをクラスタに分割し，データ挿入時には該当クラスタだけを再学習することでデータ挿入の速度を向上させる研究がある．本研究では，クラスタの大きさの偏りを防ぐ方法を提案する．

2. 関連研究

Kraska ら[1]は B-tree をキーからレコードの位置へマッピングするモデルと見なし，これを機械学習モデル（ニューラルネットワークや線形回帰など）に置き換えた RMI を提案した．RMI の基本的なアイデアは，学習モデルを使用して，順序配列キーの累積分布関数 CDF（Cumulative Distribution Function）を近似することである．RMI の構造は図 1 に示され，主にルートモデル，中間層モデル，およびリーフモデルから構成されている．ルートモデルはデータセット全体で訓練されるため，データ分布が複雑な場合があり，そのためにニューラルネットワークを使用する．一方，下層のモデル（中間層モデルまたはリーフモデル）はより小さなサブデータセットで訓練され，スペースと構造コストを節約するために，単純な線形回帰モデルを使用する．また，リーフモデルがデータ分布を効果的にフィットできない場合は，B-tree でリーフモデルを置き換えることが可能である．Kraska らによって提案された学習型インデックスモデルは，B-tree と比較して探索の性能とメモリフットプリントが大きく改善された．一方，データが挿入・削除されたことによりインデックスが更新された場合，RMI のリーフモデルおよびそれに接続されている上層モデルは再学習が必要となり，学習コストが発生するという問題がある．

†芝浦工業大学大学院理工学研究科 Graduate School of Engineering and Science, Shibaura Institute of Technology

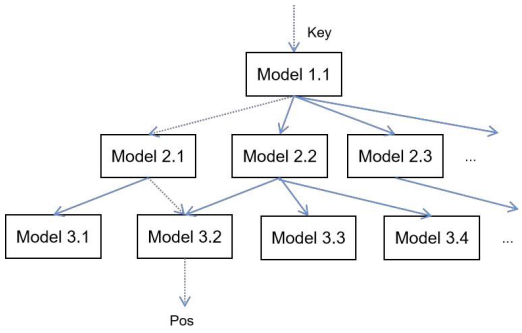

図 1 RMI

インデックス更新問題については，いくつかの研究がある．Gao ら[2]は，スケーラブルな学習型インデックスモデル Dabble を提案した．クラスタリングアルゴリズムを用いて，データセットを K 個の部分に分割した．これにより，より良い学習が可能となった．

Dabble の問題点の一つは，クラスタ数 K を決める方法が提案されていない点である．もう一つの問題は，新しいデータが挿入された際のクラスタサイズの偏りである．Dabble ではモデル選択器を使って，新しいデータが格納されるべきクラスタを決定する．この際，クラスタの大きさの偏りを考慮していないため，特定のクラスタだけサイズが大きくなる可能性がある．また，データの削除についても同様に，特定のクラスタのサイズだけが小さくなる可能性がある．クラスタのサイズが偏ると，探索効率の低下や不均衡なリソース利用が起こり，全体のパフォーマンスに悪影響を及ぼす．

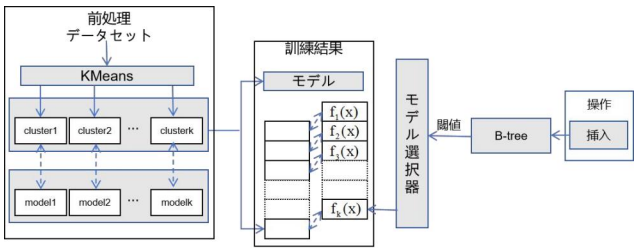

図 2 Dabble モデル

Copyright © 2024 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

3. 提案

以下では，クラスタの数 K を決める方法，各クラスタの大きさの偏りを防ぐ方法を提案する．

3.1 クラスタ数 K の決定法

クラスタ数 K は従来から知られているシルエット分析を用いて決定する．以降の提案に関係するので，詳細に述べる．

シルエット係数 $S_{(i)}$ は，与えられた標本点 $x_{(i)}$ に対して次のように定義される．

$$S^{(i)} = \frac{b^{(i)} - a^{(i)}}{\max(a^{(i)}, b^{(i)})}$$

ここで $a^{(i)}$ はクラスタの凝集度であり， $x^{(i)}$ から $x^{(i)}$ が属するクラスタ内他の点までの平均距離である．$b^{(i)}$ は別クラスタからの乖離度であり，$x^{(i)}$ に最も近い別クラスタに属する点までの平均距離である．すべての標本点のシルエット係数を求め，それらを平均する．平均シルエット係数は[-1,1]の範囲をとり，クラスタ内の標本点の距離が近いほど，またクラスタ間の標本点の距離が遠いほど，平均シルエット係数は大きくなる．異なるクラスタ数で KMeans を実行し，それぞれの平均シルエット係数を計算する．平均シルエット係数が最大となるクラスタ数が最適なクラスタ数 K である．

3.2 データの挿入

提案方式では，関連研究[2]と同じモデル選択器で挿入先のクラスタ C を決めるが，そのサイズが一定の閾値よりも大きかった場合は，クラスタ C を分割する．ここで，3.1 節で述べたシルエット係数を利用して，実データ領域の分割点を決める．クラスタ C の中で，シルエット係数が一番高いデータを選んで分割点とする．シルエット係数が最大のデータを分割点として選択する根拠は，その点の周囲のデータが非常に似ており，クラスタの密集点であることを意味するからである．また，他のクラスタとの距離も遠いため，分割点として適している．図 3 に示すように 2 つに分割されたクラスタのうち，データ数が少ないクラスタ C_1 とし，もう一方をクラスタ C_2 とする．新しいデータを C_1 に加え再学習する．C_2 は元のクラスタ C をそのまま利用する．

以上により，クラスタ数は K から K+1 に増加する．このように動的にクラスタ数を調整する方法は，データの構造を最適化し，データ処理の柔軟性と効率を向上させる．

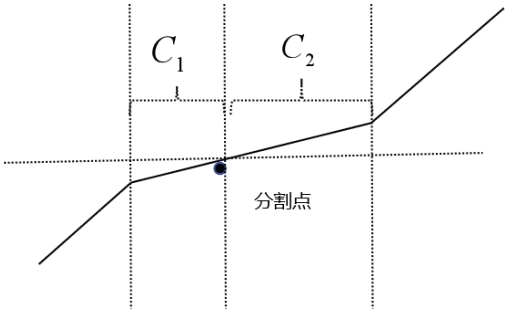

図 3　クラスタの分割方法

3.3 データの削除

データがあるクラスタから削除され，一定の閾値以下になる場合，最も近い別クラスタとの統合するのが可能である．このプロセスにより，二つのクラスタが一つに統合され，クラスタが過小になるのを防げる．統合後には，クラスタの特性が変わるため，適切に反映させるために再学習を行う．再学習のコストと過小なクラスタが残るコストはトレードオフ関係にあり，前述の閾値を慎重に決定する必要がある．

4. 実装

提案システムは現在 python を用いて実装中である．また本稿では述べなかったが，実際のデータ挿入はバッファにデータを溜めておき，複数個の挿入を一度に行う．同じクラスタに複数のデータが追加された際の最適化なども実装する予定である．

5. おわりに

本稿では既存研究である学習型インデックス Dabble に対して，データを挿入・削除した際に生じるクラスタのサイズの偏りを防ぐ方法を提案した．今後は実装を完了させ，データの挿入と削除の性能を評価する．

参考文献

[1] Kraska T, Beutel A, Chi EH, Dean J, Polyzotis N. “The case for learned index structures.” In: Proc. of the Conf. on Management of Data (SIGMOD). ACM, 2018. 489−504.
[2] Gao YN, Ye JB, Yang NZ, Gao XF, Chen GH. Middle layer based scalable learned index scheme. Ruan Jian Xue Bao/Journal of Software, 2020,31(3):620−633 (in Chinese).

Copyright © 2024 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.