構造化 P2P ネットワークにおける類似度検索のための次元圧縮手法の検討

本山洸 [1,a)]　　菅谷至寛 [1]　大町真一郎 [1]

1. まえがき

ユーザーによって生成される膨大な量のデータをリアルタイムで扱うことに適しているシステムとして，peer-to-peer (P2P) システムが挙げられる．生成される文書や画像などのコンテンツは一般的に非常に高次元な特徴量で表されることが多い．このようなデータに対して，分散環境で類似したデータの検索（類似度検索）を行うとき，すべてのデータにアクセスしてしまうと通信コストが膨大になるため現実的ではない．あらかじめ類似性の高いものを近くに集めて配置することによって探索すべきノードの範囲を限定し，通信コストを抑える事ができると考えられる．先行研究である ZNet[2)]では低次元のデータを対象としているので，高次元データを扱うと通信コストが大きくなってしまう．そこで，高次元コンテンツの局所性を保ちつつ次元を削減することができれば，類似度検索を行うときに通信コストを抑制することができると考えられる．本稿では，データセットの大局的情報を用いることなく，分散環境で類似度検索を可能にする次元圧縮の手法を提案する．

2. 分散環境での類似度検索

分散環境で類似度検索を行う手法の一つとして，構造化 P2P ネットワークを用いた ZNet[2)]が挙げられる．ZNet は多次元空間を空間充填曲線により 1 次元値にすることによって範囲検索を行う．データが比較的低次元の場合は 1 次元値においても局所性が保たれるが，データが高次元になるほど，1 次元値にした時に遠くに離れてしまうため局所性を保つことができなくなる．そのため通信コストが膨大になり扱うことがとても難しくなる．よって，高次元のデータの次元を削減した後 ZNet を用いることで類似度検索を実現する．

3. Vivaldi

Vivaldi[1)]は，分散ネットワークにおいてノード間の距離を反映した仮想的な座標を推定するための手法である．ネットワーク全体をばねモデルとみなし，ノード間はばねによって接続されていると仮定する．ネットワーク全体のばねエネルギーを最小化することによって，一部のノード間の距離を用いて m 次元空間における相対位置を推定する．あるノードが他のノードと通信を行うとき，その 2 つのノードはばねでつながっているとみなすことで，座標を更新する．この動作を逐次的に行うことでネットワーク全体がばねでつながっているとみなすことができる．

本稿では Vivaldi においてデータをノードとみなすことによって高次元データの類似性に基づいた圧縮に応用した．

$$E = \sum_i \sum_j (L_{ij} - \|x_i - x_j\|)^2 \qquad (1)$$

L_{ij}はデータi, jの実測距離，$\|x_i - x_j\|$はデータi, jの座標のユークリッド距離である．

4. Counting Filter

Counting Filter[3)]はハッシュ関数を用いてデータを格納するデータ構造である．c ビットのカウンタからなる長さ m の配列と l 個のハッシュ関数 h_i から構成されている．データを x とするとハッシュ関数は $h_i(x)$=[1, m]の整数値を返す．データを格納するときは，まずデータを l 個のハッシュ関数全てにかけて，それぞれが返した数値の場所のカウンタをインクリメントする．このカウンタ列を圧縮後のデータとみなすことによって，m 次元に圧縮することができる．

5. 実験

高次元データを Vivaldi または Counting Filter によって低次元化した後，ZNet 上で範囲検索を行った時の検索精度と Vivaldi による座標推定についてシミュレーションした．

5.1 次元削減による検索精度

Vivaldi または Counting Filter を用いて次元を圧縮し，ZNet 上に配置した時の検索精度について検証した．高次元データとして Hondana.org の 4,741 人の本棚リストを用いた．一つのデータ（本棚）は 143,462 タイトルの本の有無を表すベクトル列であり，すなわち，143,462 次元の 2 値ベクトル列とみなせる．このデータセットについて Vivaldi または Counting Filter を用いて 16 次元に圧縮した後，ZNet を用いて類似度の高いものから順に k 個検索する，Top-k 検索を行った．精度は以下の式によって定義する

$$Accuracy = \frac{\sum_{i=1}^{k} Retrieved_i}{\sum_{i=1}^{k} Relevant_i} \qquad (2)$$

$Relevant_i$ はデータセット全体から Top-k 検索を行った時の

1 東北大学大学院工学研究科
a) mtym@iic.ecei.tohoku.ac.jp

i 番目のデータの類似度の値であり，$Retrieved_i$ は ZNet 上で Top-k 検索を行った時の i 番目のデータの類似度の値である．本実験では k=10 とした．類似度は Dice 係数を使用した．以下に式を示す．

$$\text{Dice 係数} = \frac{2|I \cap I'|}{|I| + |I'|} \quad (3)$$

$|I|$はデータのキーワード数を表す．

結果を図 1 に示す．クエリ範囲は圧縮後の 16 次元空間におけるクエリ点からの半径であり，通信量と比例するものである．クエリの範囲を広げることによってより多くのピアにクエリが到達する．クエリの到達したピアが多くなるほど，検索精度は高い結果となったが，クエリ範囲が大きくなるにつれて Counting Filter による圧縮のほうが良い精度になった．

Counting Filter を用いるならばデータの座標は一意に定まり，類似したデータ（Hondana.org では同じ本を含む本棚に相当）は比較的近傍に配置されることが期待できる．しかし，データ間の距離（類似度）を直接利用しているのではなくハッシュ関数を用いているため，局所性が保たれなくなる場合もあると考えられる．一方，Vivaldi による次元圧縮では，データ間の距離を直接用いてデータの配置が決定される．したがって，Vivaldi によって類似度に基づいて座標を定めることができれば ZNet 上における範囲検索時の精度の低下を防ぐことができると期待されたが，その予想に反した結果が得られた．

5.2 データセットによる座標の収束性

前実験から Vivaldi ではデータセットによって推定される座標の誤差が大きくなってしまった．その原因として推定誤差が収束していないと考え，Vivaldi の収束性に関して調査を行った．一様分布の 10 次元[0,s]範囲のデータ 50 個を Vivaldi によって 2 次元空間で座標推定した時の座標の収束性について s=10,20,30,40 について検証をした．結果を図 2 に示す．s が 30 より大きくなると，発散する座標が多くなり，平均誤差が多くなった．よってデータが疎らになると Vivaldi では発散してしまうことがわかった．

5.3 Vivaldi の推定誤差

Vivaldi は自分の座標と相手の座標を用いて座標を推定する手法である．ある点が座標を更新するときに使うもう一点の座標について，ランダムに選んだ場合と，埋め込もうとしている 2 次元空間において最も近い点(元の空間の最近傍点とは限らない)を選んだ場合についての Vivaldi の推定誤差について検証をした．データセットは 10 次元の[0,20]の範囲で一様分布に生成した 50 個のデータを使用した．図 3 に平均誤差を示す．

収束までの更新回数はどちらも 100 回程度であった．また，最も近い座標によって更新することによって平均誤差が小さくなった．よって推定誤差を小さくするためには更新相手の選択が非常に重要であるという結果が得られた．

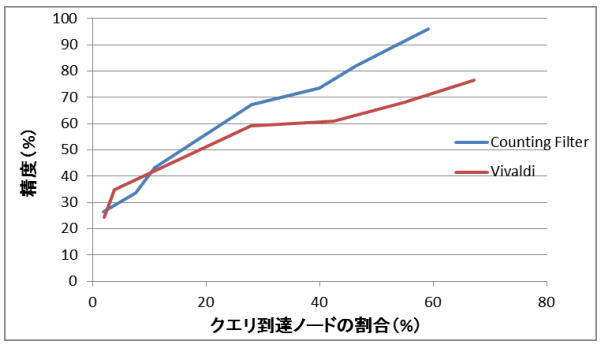

図 1.圧縮後の検索精度

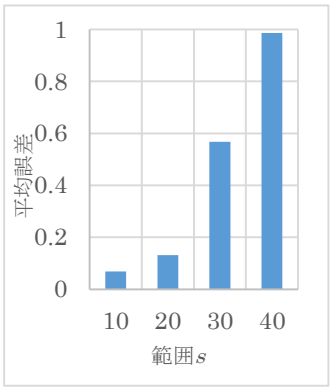

図 2.データセットの密度による収束性

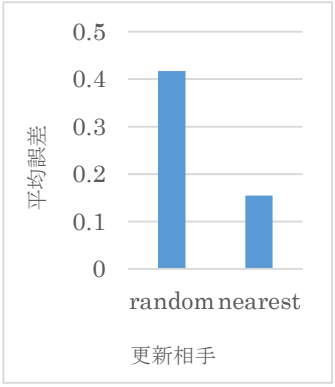

図 3.座標更新に使用する相手による平均誤差

6. まとめ

分散環境では類似度検索を行うことが難しい高次元データについて，次元圧縮をすることによって通信コストを減らす手法について検証をした．データセットによっては Vivaldi による座標推定が発散してしまうため，次元圧縮が難しい場合が見られた．よって，発散してしまう場合は Counting Filter と組み合わせることにより収束させる方法を考えている．また，Vivaldi による次元圧縮では，座標更新時に近い座標のデータを用いることによって推定誤差が小さくなる．同一ノードの中にはある程度類似したデータが徐々に集まってくると期待されるため，ノード内で Vivaldi による座標推定を重点的に行うことで通信量を抑えることができると考えられる．

謝辞

本研究の一部は JSPS 科研費 24500072 の助成を受けたものである．

参考文献

1) F. Dabek, et al., “Vivaldi: A Decentralized Network Coordinate System.” SIGCOMM ’04, pp. 15-26, 2004

2) Y. Shu, et al., “Supporting multi-dimensional range queries in peer-to-peer systems”, IEEE Intl Conf. on Peer-to-Peer Computing, pp.173-180, 2005

3) L. Fan, et al., “Summery cache: A Scalable Wide-Area Web Cache Sharing Protocol,” SIGCOMM ’98, pp. 254-265, 1998