未知スコア関数で評価される多次元データを対象とする計量学習を用いたTop-k検索アルゴリズム

野沢　充彦† 常　　穹† 宮崎　純†

† 東京工業大学情報理工学院情報工学系　〒 152–8550 東京都目黒区大岡山 2-12-1
E-mail: †nozawa@lsc.c.titech.ac.jp, ††{q.chang,miyazaki}@c.titech.ac.jp

あらまし　本研究では，検索対象のデータを事前にインデックスすることができず，検索に用いられるスコア関数の性質が未知かつ多次元データに対しても適用可能な Top-k 検索アルゴリズムを提案する．提案手法では，タスクにとって望ましいデータ間の類似関係を表現することができる距離関数を獲得するための機械学習手法である計量学習を用いることで，データ間の距離とスコア関数の出力値の大小関係が維持されないという問題の解決を試み，検索精度の向上を図る．評価実験では，比較手法に対し提案手法が同程度の Top-k 検索精度を達成するために抽出するデータ数を大幅に減らし，実行時間を削減することに成功した．

キーワード　情報検索，Top-k，計量学習

1 はじめに

近年，インターネットの普及に伴い，膨大な量の情報が電子化されたデータとして私たちの身の回りに溢れるようになった．そのような状況下において情報要求が生じた際，一つ一つの情報を人力で調べ，要求を満たすかどうか判別するというのは途方もない労力を要する．この問題を解決するため，コンピュータを用いて，データ集合中から情報要求を満たすデータを取り出すことを可能にする，情報検索という技術が存在する．また点在する検索対象データをあらかじめ集めて組織化して保持し，その対象データに対して情報検索を行う情報検索システム（検索システム）や，ストレージシステム内に格納されたデータをキーワードで検索するファイルサーチやデスクトップサーチシステムが存在する．

適合度の高い上位 k 件のデータ抽出するための方法を Top-k 検索という．検索システムが保持する膨大なデータ量に対し，検索システムの利用者が期待する検索結果データの件数は非常に小さいことが多いので，検索が行われる多くのケースを Top-k 検索に当てはめることができる．Top-k 検索を実現する単純な方法は，検索システムが保持する全てのデータに対して適合度を計算し，適合度の順にソートされたデータに対して上位 k 件のデータを選び，返却することである．しかしこの方法は，検索システムが莫大な量のデータを保持しており，適合度の計算処理に時間を要する場合，全データに対しての適合度の算出に非常に時間がかかってしまい不適である．そのため，全データのスコアリングを行わずに高速に動作する Top-k 検索手法が求められている．

これまで [1] をはじめとし，様々な Top-k 検索手法が提案されてきた．しかしその多くは，スコア関数や検索対象のデータに対して何らかの制約を設けており，Top-k 検索を行うことができる場面が限られている．例えば, [1] は適合度を算出するためのスコア関数が単調性を持っている場合のみ有効な手法である．また, [4], [5] は任意のスコア関数に対して適用可能だが，高次元データに対しての適用は計算量的に困難であるという問題を抱えている．また検索対象のデータが事前にインデックス付けされていることを仮定している既存手法も多い．データのインデックス付けは，索引後にデータの更新が行われない場合は効果的だが，ユーザプロファイルや時系列情報など，日々変化するデータをスコア値の計算に取り入れることが困難になる．

これに対し池田らは，Top-k 検索を近傍点探索問題に置き換え，近似最近傍探索手法の一つである Locality Sensitive Hashing(LSH) [6] を用いることで，高次元のデータにおいても適用可能な Top-k 検索手法を提案した．この手法は，LSH によって同じバケットに格納されたデータ，すなわちデータ間の距離が確率的に近いと判断されたデータはスコア関数の値も近いということを想定しているが，スコア関数が多次元かつ複雑な形状を有している場合においては，データ間の距離とスコア関数の出力値の近さには関係性がないことも多い．

それを踏まえ本研究では，データ間の類似関係を考慮した特徴量変換器を獲得することができる計量学習と, [6] で提案されている LSH を用いた Top-k 検索アルゴリズムをを組み合わせた Top-k 検索手法を提案する．実験により，LSH に格納する前のデータに対して計量学習によって獲得された特徴量変換器を適用することで，LSH の同じバケット内にスコア値の近いデータが格納されやすくなり，Top-k 検索の性能が向上することを示す．

2 準　　備

2.1 LSH

Locality Sensitive Hashing(LSH) は，距離が近いデータを高い確率で同じ値にハッシュ化する技術である．S を入力空間の領域，U をハッシュ関数の出力が取りうる領域として，ハッ

シュ関数族 $\mathcal{H} = \{h : S \to U\}$ が以下の式 (1) の条件を満たすとき，$\mathcal{H}$ は $(r_1, r_2, p1, p2)$ - sensitive であるという．

$$\begin{aligned} d(\boldsymbol{v}, \boldsymbol{q}) < r_1 \Rightarrow Pr[h(\boldsymbol{v}) = h(\boldsymbol{q})] \geqq p_1 \\ d(\boldsymbol{v}, \boldsymbol{q}) > r_2 \Rightarrow Pr[h(\boldsymbol{v}) = h(\boldsymbol{q})] \leqq p_2 \end{aligned} \tag{1}$$

このとき，$p_1 > p2$，$r_1 < r2$ である．

$(r_1, r_2, p1, p2)$ - sensitive なハッシュ関数族 $\mathcal{H}$ を得る方法として，安定分布を用いたものがある [7]．$\boldsymbol{a} \in \mathbb{R}^D$ を安定分布からのサンプル，b を $[0, W]$ の一様分布からのサンプルとしたとき，ハッシュ関数を式 (2) のように設計する．

$$h_{\boldsymbol{a},b}(\boldsymbol{v}) = \left\lfloor \frac{\boldsymbol{a} \cdot \boldsymbol{v} + b}{W} \right\rfloor \tag{2}$$

ハッシュ関数族 $\mathcal{H}$ から関数を取り出すことは，$\boldsymbol{a}$ と b のサンプリングを行うことと同義である．

[8] では，LSH を用いた近似最近傍探索アルゴリズムが提案されている．$(r_1, r_2, p1, p2)$ - sensitive であるハッシュ関数族 $\mathcal{H}$ から取り出されたハッシュ関数 $h(\boldsymbol{v})$ を j 個連結することでできる関数 $g(\boldsymbol{v}) = (h_1(\boldsymbol{v}), ..., h_j(\boldsymbol{v}))$ の族を $\mathcal{G} = \{g : S \to U^j\}$ とする．$g(\boldsymbol{v})$ の出力値をバケット，$g(\boldsymbol{v}) = g(\boldsymbol{q})$ となることを，$\boldsymbol{v}$ と $\boldsymbol{q}$ は同じバケットに格納されると表現する．$\mathcal{G}$ から L 個の関数 $g_1, ..., g_L$ を独立かつ無作為に取り出す．このときの g_l をハッシュテーブル，L をテーブル数と表現する．

探索を行う前に，探索対象のデータ $x \in \mathbb{R}^D$ に対して L 個のハッシュテーブルで $g_l(\boldsymbol{x})$ を計算し，LSH のバケットに格納しておく．探索時には，入力クエリ $\boldsymbol{q}$ に対して L 個のテーブルで $g_l(\boldsymbol{q})$ を求め，同じバケットに格納されているデータを抽出することで探索結果を得る．

2.2 計量学習

計量学習とは，あるタスクにとって望ましいデータ間の類似関係を表現することができる距離関数を獲得するための機械学習手法である．データ間の類似度を測定するために，ユークリッド距離やコサイン類似度が用いられることが多いが，何を持って類似と見なすかはそれぞれのタスクによって異なり，上記のような単純な類似度測定方法のみでデータ間の類似関係を表すのが困難なこともある．そのような場合において，あらかじめ用意されたデータセットを用いて，単純な類似度測定方法でもデータ間の複雑な類似関係を表現できる特徴量を得るための変換器を獲得する，というのが計量学習の考え方である．

計量学習の学習方法としては，教師あり学習と弱教師あり学習に分かれる．教師あり学習では，データが持つラベルなどの教師データを用いて距離関数の獲得を行う．弱教師あり学習では，データが教師データを持っていないとき，タスクが要求する類似性を満たすようなデータの組や三つ組などを作成し，それを元に距離関数の獲得を行う．

2.2.1 マハラノビス距離学習

データ $x_1, x_2 \in \mathbb{R}^D$ に対して，ユークリッド距離は式 (3) で表される．

$$d(x_1, x_2) = ||x_1 - x_2||_2^2 \tag{3}$$

またマハラノビス距離は式 (4) で表せる．

$$d_M(x_1, x_2) = \sqrt{(x_1 - x_2)^\top \Sigma^{-1} (x_1 - x_2)} \tag{4}$$

Σ は共分散行列である．ここで，Σ^{-1} が半正定値行列であるとき，$\Sigma = L^\top L$ と分解できるので，式 (4) は式 (5) に変形できる．

$$\begin{aligned} d_M(x_1, x_2) &= \sqrt{(x_1 - x_2)^\top \Sigma^{-1} (x_1 - x_2)} \\ &= \sqrt{(x_1 - x_2)^\top L^\top L (x_1 - x_2)} \\ &= ||Lx_1 - Lx_2||_2^2 \\ &= d(Lx_1, Lx_2) \end{aligned} \tag{5}$$

これは，x_1 と x_2 に対して，線形変換 L を行った上で算出されるユークリッド距離に等しい．そのような線形変換 L を，機械学習によって得るというのがマハラノビス距離学習である．

代表的な手法に，Jacob らによって提案された Neighbourhood Components Analysis(NCA) [9]，Kilian らによって提案された Large Margin Nearest Neighbor(LMNN) [10] などがある．

2.2.2 深層距離学習

求めたい距離関数が，ある変換 $f : \mathbb{R}^D \to \mathbb{R}^{D'}$ に対して，$d_f(x_1, x_2) = d(f(x_1), f(x_2))$ と表されるとする．このとき d_f は距離の公理を満たしている．この変換 f の部分にニューラルネットワークを用いた，深層計量学習が広く研究されている．ニューラルネットワークによって，画像やテキストデータなど複雑な形状を持つデータを扱いやすくなるだけでなく，特徴量変換器が非線形性を持ち，より複雑な類似関係を表現できる．代表的な手法として，Elad らによって提案された Triplet Network [11]，Jiankang らによって提案された ArcFace [12] などがある．

Triplet Network は，アンカーと呼ばれる代表となるデータ x と，アンカーに対して近いとみなされるデータ x^+，遠いとみなされるデータ x^- の三つ組 (x, x^+, x^-) を入力として作成し，弱教師あり学習を行う．特徴量変換器を f とすると，$f(x)$ と $f(x^+)$ との距離 d^+ が，$f(x)$ と $f(x^-)$ との距離 d^- より小さくなるように損失関数を設計し，学習を行う. [11] 中では，式 (6) の損失関数を用いることで，三つ組に対する損失の計算を行う．

$$\begin{aligned} L(d_+, d_i) &= ||(d_+, d_- - 1)||_2^2 \\ d_+ &= \frac{e^{||f(x) - f(x^+)||_2^2}}{e^{||f(x) - f(x^+)||_2^2} + e^{||f(x) - f(x^-)||_2^2}} \\ d_- &= \frac{e^{||f(x) - f(x^-)||_2^2}}{e^{||f(x) - f(x^+)||_2^2} + e^{||f(x) - f(x^-)||_2^2}} \end{aligned} \tag{6}$$

ArcFace は，ニューラルネットワークにおいて分類問題を解く際に使用される，交差エントロピー損失を最適化するように教師あり学習を行う方法である．ソフトマックス関数を含む交差エントロピー損失を式 (7) に表す．

$$L = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{W_{y_i}^\top x_i + b_{y_i}}}{\sum_{j=1}^{C} e^{W_j^\top x_i + b_j}} \tag{7}$$

N はデータ数，x_i, y_i は入力データに対するニューラルネットの出力ベクトルと入力データに対応するクラス，W_{y_i}, b_{y_i} はクラス y_i に対応する重み行列の i 行目と i 番目のバイアス項を表す．

ArcFace では，ニューラルネットの出力ベクトルと，重み行列 $W \in \mathbb{R}^{C \times D'}$ の間でクラスごとにコサイン類似度を計算し，それをロジットとみなすことで交差エントロピー損失が計算される．C はクラス数，D' は出力ベクトルの次元数である．ArcFace の損失関数を式 (8) に表す．

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_j}^{C}e^{s\cos\theta_j}} \quad (8)$$
$$cos\theta_j = \frac{W_j^\top}{|W_j^\top|}\cdot\frac{x_i}{|x_i|}$$

s はロジットの値の大きさを制御するパラメータ，m はクラス間距離を遠くするために用いられる正則化パラメータである．

3 関 連 研 究

本節では，既存の Top-k 検索アルゴリズムと，提案手法で用いる計量学習についての関連研究を紹介する．特に断りのない限り，スコア値の大きいデータを適合度が高いデータとする．

3.1 Threshold Algorithm(TA)

TA [1] は，Fagin らによって提案された，スコア関数が単調性を持つ場合のみ使用可能な Top-k 検索手法である．検索対象のデータが持つ属性それぞれのソート済みのリストを必要とする．検索時には，ソート済みのリストそれぞれに対して並行にアクセスし，データをスコアリングしていく．その際にアクセスしたデータから，未アクセスのデータが取り得るスコア値の上限を算出する．スコアリングを行ったデータのうち，算出したスコア値の上限を超えるデータが要求数以上になった時，TA は停止し，Top-k 検索結果を返却する．

TA は非単調なスコア関数に対しては適用できない．また，検索対象のデータが持つそれぞれの属性のソート済みのリストを必要とするため，検索対象のデータの更新が頻繁に起きる場合，リストの作成に高いコストを要する．

3.2 メッシュ分割法

メッシュ分割法 [4] は，佐々木らによって提案された，スコア関数が非単調な場合においても適用可能な Top-k 検索手法である．入力空間を事前にメッシュと呼ばれる小領域に分割し，各メッシュが取り得るスコア値の最大値と最小値を調べておく．検索時には，検索対象のデータをメッシュに配置した後，メッシュが取り得るスコアの最小値の降順にアクセスすることで，適合度が k 件のデータを抽出するための閾値を求める．その後，取り得るスコアの最大値が閾値以上のメッシュからデータを取り出し，スコアリングすることで Top-k 検索結果を返却する．

[4] で提案されている等分割法と呼ばれるメッシュの分割方法は，スコア関数の形状やデータの分布を考慮せずにメッシュの作成を行う．そのため，多峰性などの複雑な関数ではアクセスするメッシュの数が増加し，偏った分布のデータではあるメッシュに多くのデータが集中して格納されてまうといった問題が起こる．そこで池田らは，ピークと呼ばれる，スコア関数の擬似的な極大点を利用してメッシュの分割を行うことで，メッシュへのアクセスを改善した [5]．

メッシュ分割法は，適用対象のデータの次元数の増加に伴い，分割後のメッシュの数が指数関数的に増加してしまうため，多次元データに対しての適用は困難であるという問題点がある．

3.3 LSH を用いた Top-k 検索 (LSH 法)

LSH 法 [6] は，池田らによって提案された，非単調なスコア関数かつ多次元データに対して適用可能な Top-k 検索手法である．オフライン時，事前に用意した訓練データセットを用いて，スコア関数のピークとなるデータを求め，LSH のバケットに格納しておく．得られたピークには，スコア値の大小に基づいて優先順位をつけておく．検索が行われるオンライン時は，検索対象のデータを LSH に格納し，優先順位の高いピークが属するバケットから順にデータを取り出し，スコアリングしていく．LSH 法による Top-k 検索の流れを図 1 に示す．

LSH 法で用いられる LSH よって同じバケットに格納されるデータは，データ間の距離が確率的に近いと判定されたデータである．しかし複雑な形状のスコア関数においては，データ間の距離が近くなる程，そのデータの持つスコア値も近くなるとは限らず，真の Top-k 検索結果には含まれないデータを多くスコアリングしてしまいかねない．

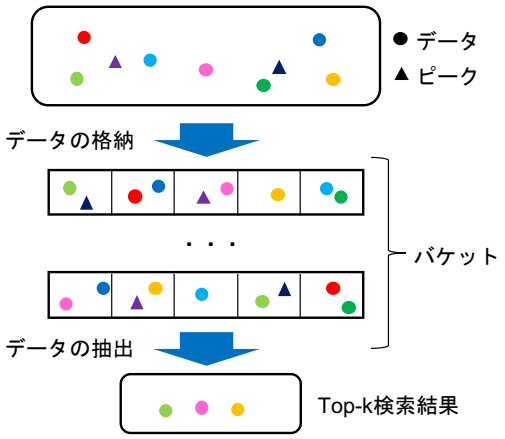


図 1: LSH による Top-k 検索の流れ

4 提 案 手 法

本節では，提案手法である計量学習を用いた Top-k 検索手法について述べる．LSH 法では，距離の近いデータ同士であってもスコア値が近いとは限らず，データの抽出時に真の Top-k 検索結果には含まれないデータを多く抽出してしまいかねないことが課題であった．そのため本手法では，計量学習を導入することにより，距離の近いデータ同士はスコア値も近くなるような状況を実現することができる特徴量変換器を獲得すること

を考える．具体的には，特徴量変換器による変換後の空間において，LSH 法で用いられるピークとなるデータの周辺にスコア値の高いデータが集まる状態を実現することを目指す．提案手法では，検索が行われないオフライン時に，事前に用意されたデータセットを用いて計量学習モデルの訓練を行い，特徴量変換器の獲得を行う．LSH 法で用いられるピークとなるデータと，オンライン時の検索対象データの LSH への格納は，実際のデータではなく，特徴量変換器を適用することによって得られる特徴量を格納することによって行われる．提案手法による Top-k 検索の流れを図 2 に示す．

4.1 学習用データの作成

本項では，教師あり学習によって計量学習モデルの訓練を行うためのデータセットの作成方法を述べる．データセット $\mathcal{X} \in \mathbb{R}^{N \times D}$ とスコア関数 f が事前に与えられているとする．N はデータセットのサイズ，D は次元数である．スコア関数 f に $\mathcal{X}$ を入力として与えることで，データセットに対するスコア値 $\mathcal{Y} \in \mathbb{R}^N$ を得る．このスコア値 $\mathcal{Y}$ に基づいて，ArcFace で学習を行う際に必要となるラベル $\mathcal{L} \in \mathbb{R}^N$ を生成することを考える．スコア値の近いデータ間の距離が小さくなるようにするため，同じラベルに属するデータはスコア値が近いことが望まれる．提案手法では，スコア値 $\mathcal{Y}$ に対しビン数 s の v-optimal ヒストグラムを適用することで，各ビンに属するデータの数が等しくなるようにデータセット $\mathcal{X}$ を s 個のビンに分割する．この時，各ビンに対し 0 から $L-1$ まで数字を割り当てたものを，データセット $\mathcal{X}$ に対するラベル $\mathcal{L}$ をとし，訓練データ $\mathcal{D}_{train} = (\mathcal{X}, \mathcal{L})$ を作成する．検証用データ，テスト用データに対しても同じ手順で作成を行うが，データセットのビンへの割り当ては，訓練データ生成時に v-optimal ヒストグラムによって得られるビン間の閾値に基づいて行う．

4.2 特徴量変換器の設計

4.1 で述べた手順によって生成された訓練データを用いて，計量学習モデルの訓練を行い，特徴量変換器の獲得を行う．特徴量変換器には，ニューラルネットワークを用いる．隠れ層の層数，各層のユニット数は自由に決めることができるが，大きすぎる値を設定すると特徴量への変換にかかる時間が増え，Top-k 検索の実行時間が増大してしまう．また，変換後の特徴量は，元のデータとは異なるデータ分布をしていることが考えられる．そのため LSH 法において良い性能であったパラメータを提案手法においてそのまま使用しても同じように良い性能を出すとは限らず，LSH の各パラメータを適切に決め直す必要がある．

5 評価実験

本節では，提案手法である計量学習を用いた Top-k 検索手法についての評価実験を行う．評価実験では，提案手法の精度と実行時間を計測する．評価実験は，データの次元数 D が 100, 300 のときに対してそれぞれ行った．実験で使用するパラメータの一部について表 1 に示す．

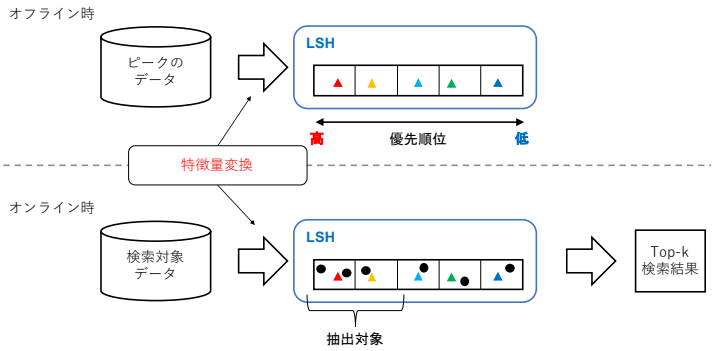


図 2: 計量学習を用いた Top-k 検索の流れ

表 1: Top-k 検索に用いられるパラメータ

表記	説明	値
N	Top-k 検索ときの検索対象のデータ数	1.0×10^5
D	データの次元数	100, 300
$dist$	データの分布	uni, normal
f_{score}	スコア関数	Rastrigin, Rosenbrock
s_p	v-optimal ヒストグラムのビン数	10
k	Top-k 検索で抽出するデータ数	100
j, L, W	LSH のパラメータ	実験ごとに設定

5.1 比較手法

評価実験で用いる，それぞれの比較手法について説明する．

全探索法 検索対象のデータ全てをスコアリングした後ソートを行い，スコアが上位のデータを返却する手法である．

LSH 法 [6] で提案されている，Top-k 検索に LSH を用いた手法である．

提案手法 (計量学習+LSH 法) 計量学習によって得られた特徴量変換器の出力となる特徴量を Top-k 検索手法に利用する方法である．評価実験では計量学習手法に ArcFace を使用する．

5.2 実験の詳細

まず，データセットの作成について説明する．様々な状況を想定して，データの次元数 D，データの分布 $dist$，スコア関数 f_{score} を変えて複数の種類のデータセットを作成し，それぞれのデータセットについて実験を行った．データの分布 $dist$ は，uni と corr の 2 つを使用した．uni は区間 [0, 1] の一様分布を表し，corr は平均ベクトル $(0.5, ..., 0.5)^\top$ と式 (9) の共分散行列 Σ を持つ多変量正規分布を表す．このとき，$\rho = 0.8$ とした．

$$\Sigma = \begin{pmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \vdots & \cdots & 1 \end{pmatrix} \tag{9}$$

$dist\_type$ が corr のときは，データの取りうる値の範囲が [0, 1] になるように正規化を行う．

スコア関数には，Rastrigin 関数と Rosenbrock 関数を使用した．Rastirigin 関数を式 (10) に，Rosebrock 関数を式 (11) に示す．

$$f(x_1, ..., x_d) = 10d + \sum_{i=1}^{d}((x_i^2) - 10\cos(2\pi x_i)) \quad (10)$$

$$f(x_1, ..., x_d) = \sum_{i=1}^{d-1}(100(x_{i+1} - x_i)^2 + (x_i - 1)^2) \quad (11)$$

Ratrigin 関数は多峰性の関数であり，非常に多くの極値を持つ関数として知られている．Rosenbrock 関数は，隣り合う変数間に交互作用が存在する関数である．なお本稿では [6] にならい，スコア関数に機械学習などを用いて獲得された計算時間を要する関数を想定し，スコアの算出に Rastrigin 関数の計算を 200 回行うこととした．

実験に用いるデータには，計量学習モデルの学習用データ，LSH 法におけるピーク探索用データ，Top-k 検索における検索対象となるデータが存在し，それぞれのデータは前述の方法を用いることで，$(D, dist, f_{score})$ の組に対して作成される．計量学習モデルの学習用データのラベルは，4.1 で述べた手順に沿って作成される．

次に計量学習手法 ArcFace の学習方法について説明する．ArcFace の学習に使用するパラメータについて表 2 に示す．

表 2: 計量学習モデルの訓練に用いられるパラメータ

表記	説明	値
N_{train}	訓練用データのサイズ	5.0×10^4
N_{valid}	検証用データのサイズ	1.0×10^4
N_{test}	テスト用データのサイズ	1.0×10^5
s	ラベル数	10
$scale$	ArcFace のハイパーパラメータ	64
$margin$	ArcFace のハイパーパラメータ	0.5
$epochs$	エポック数	500
lr	学習率	5.0×10^{-3}
$batch\_size$	バッチサイズ	512
α	重み減衰パラメータ	1.0×10^4

損失関数には交差エントロピー損失，最適化手法には Adam を用いた．各エポックの学習終了時，検証用データセットに対して評価指標の算出を行い，この指標を元に計量学習モデルの学習が進んでいるかどうかを判断する．評価指標には正解率 (Accuracy) と平均絶対誤差 (Mean Absolute Error; MAE) を用いる．評価指標の算出は，検証用データのラベル L_{valid} と，一つ抜き交差検証によって得られる予測ラベル L'_{valid} の間で行われる．Accuracy, MAE の計算式をそれぞれ式 (12)，式 (13) に示す．

$$Accuracy = \frac{1}{N_{valid}} \sum_{i=1}^{N_{valid}} \delta_{l_i, l'_i} \quad (12)$$

$$MAE = \frac{1}{N_{valid}} \sum_{i=1}^{N_{valid}} |l_i - l'_i| \quad (13)$$

ここで l_i は検証用データ x_i のラベル，l'_i は x_i の予測ラベル，δ_{l_i, l'_i} は l_i と l'_i の間で計算されるクロネッカーのデルタである．評価指標が 100 エポックの間改善しなかった場合，学習を早期に打ち切る．また特徴量変換器に用いるニューラルネットワークの構造は，予備実験の結果から隠れ層の大きさを 2，ユニット数を入力データの次元数 D の 5 倍とした．

次に Top-k 検索で用いる LSH 法の詳細について説明する．LSH 法で使用するピークとなるデータの探索とピークの優先順位付けの方法は, [6] で述べられている方法と同じ手順で行なった．またピークの優先順位付けで用いる v-optimal ヒストグラムのビン数は，計量学習モデルの学習用データセットの作成で用いられる v-optimal ヒストグラムのビン数 s と同じ値を使用した．ピークの探索に用いるデータセットは，計量学習モデルの学習用データセットを用いても問題ないと判断し，同じものを使用した．LSH パラメータセットは $D = 100$ の時と $D = 300$ の両方でそれぞれ実験ごとに設定した．

実験では，精度の評価のために Top-k 件の Recall@k を計測する．Recall@k の計算式を式 (14) に示す．

$$Recall@k = \frac{|Top_{true}@k \cap Top_{pred}@k|}{k} \quad (14)$$

ここで，$Top_{true}@k$ は真の Top-k 検索結果の集合であり，$Top_{pred}@k$ は Top-k 検索手法によって返却された検索結果の集合を表す．

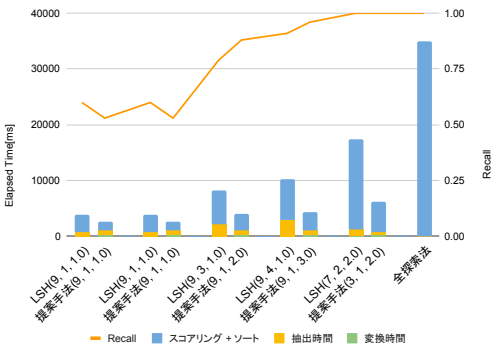
5.3 実験結果

5.3.1 $D = 100$ のとき

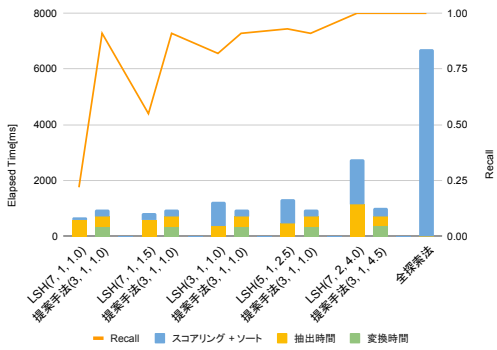
LSH 法と提案手法のパラメータセットについて，表 3 に示す．図 3 は次元数 $D = 100$ の時に各データセットに対して提案手法と比較手法を適用した時の Top-k 検索結果の $Recall@k$ と実行時間を，各 LSH のパラメータに対して図示したものである．LSH 法と提案手法では，左から順に，Recall が 0.1 以上，0.5 以上，0.75 以上，0.9 以上，1.0，となる性能を出すことができた LSH のパラメータの中で，最も実行時間が小さかったものを，LSH 法，提案手法の順に列挙した．また，手法の中には要求された Recall を満たすことができなかったものもあり，その場合は Recall が最良となるパラメータの組を列挙し，接尾辞に*をつけた．図中の $LSH(j, L, W)$ と 提案手法 (j, L, W) は LSH のパラメータ j，L，W における LSH 法と提案手法を表す．以降の項で触れる図 4 においても，同じ方法で図の理解をすることができる．また表 4 に，各手法において選択されたパラメータ設定の時に抽出されたデータ数をまとめた．

図 3 を見ると，提案手法では，全てのデータセットにおいて，Recall が 1.0 となるパラメータの組が存在しており，かつ全探索法と LSH 法よりも少ない実行時間で Top-k 検索を行うことができている．その時の，検索対象の全データに対するデータの抽出数の割合は，表 4 からおよそ 5%から 15%ほどに抑えることができている．

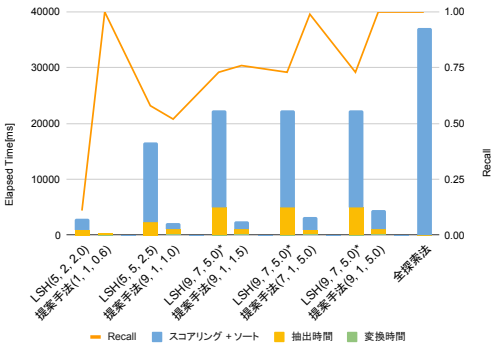
f_{score} = Rosenbrock においては，提案手法は全体の Top-k 検索の実行時間に対し，特徴量変換にかかる時間が支配的になるため，Recall の要求水準によっては LSH 法のほうが優れた実行時間になることがあった．加えて $dist$ = uni のとき，提案手法は Recall の要求水準が 0.9 以上から 1 に引き上げられると，実行時間が倍ほどかかってしまっている．



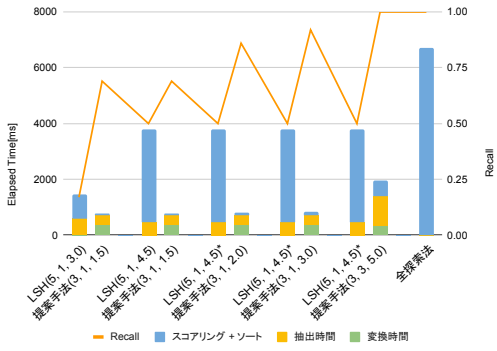
(a) $dist = \text{corr}$, $f_{score} = \text{Rastrigin}$



(b) $dist = \text{corr}$, $f_{score} = \text{Rosenbrock}$



(c) $dist = \text{uni}$, $f_{score} = \text{Rastrigin}$



(d) $dist = \text{uni}$, $f_{score} = \text{Rosenbrock}$

図 3: $D = 100$ の時の各データセットに対する Top-k 検索の実験結果

表 3: $D = 100$ のときの LSH のパラメータ

表記	値
j	3, 5, 7, 9
L	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
W	1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0

5.3.2 $D = 300$ のとき

LSH 法と提案手法のパラメータセットについて，表 5 に示す．図 4 は次元数 $D = 10$ の時に各データセットに対して提案手法と比較手法を適用した時の Top-k 検索結果の $Recall@k$ と実行時間を，各 LSH のパラメータに対して図示したものである．また表 6 に，各手法において選択されたパラメータ設定の時に抽出されたデータ数をまとめた．

図 4 を見ると，$dist = \text{corr}$ のとき，LSH 法と提案手法の両方が全探索法と比べ良い性能を出せている．特に $f_{score} = \text{Rosenbrock}$ においては，提案手法は $D = 100$ の時と同様，特徴量の変換にかかる時間が Top-k 検索の実行時間に対してより支配的になり，どの Recall の要求随順においても LSH 法の方が優れた実行時間を記録している．

一方で，$dist = \text{uni}$ のときは，LSH 法，提案手法の両方でどのパラメータの組みにおいても，Recall が 1 となる要求水準を満たすことはできなかった．$f_{score} = \text{Rastrigin}$ のとき LSH 法では，全てのパラメータの組において Recall の最大値は 0.7 未満であった．データの抽出数も，LSH 法と提案手法の両方で，Recall の要求水準の増加に伴って増えてしまっている．これは，提案手法では，計量学習モデルの訓練自体がうまくいっていないことに起因すると思われる．

表 5: $D = 300$ のときの LSH のパラメータ

表記	値
j	3, 5, 7, 9
L	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
W	1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0

6 おわりに

本稿では，検索対象のデータのインデックスを必要とせず，スコア関数の性質が未知かつ多次元のデータに対して適用可能な Top-k 検索手法を提案した．提案手法では，計量学習を用いることでスコア値の近いデータ間の距離を近くすることができるような特徴量変換器の獲得を行い，これにより獲得された変換器の出力となる特徴量を用いて Top-k 検索を行うことで，Top-k 検索の性能の向上を図った．

評価実験では，$D = 100, 300$ のときにおいて，複数のデータの分布とスコア関数の組みに対して，提案手法と比較手法の性能と実行時間を評価した．$D = 100$ のとき，提案手法は特徴量の変換にかかる時間を抑えつつ，比較手法に対して同程度の Recall の要求水準を達成するために LSH によって抽出されるデータの数を大幅に削減した．これにより，抽出されたデータのスコアリングにかかる時間を削減することに成功した．一方

手法	データ数
$LSH(9,1,1.0)$	9175
$LSH(9,3,1.0)$	17404
$LSH(9,4,1.0)$	21021
$LSH(7,2,2.0)$	44826
提案手法 $(9,1,1.0)$	4411
提案手法 $(9,1,2.0)$	8074
提案手法 $(9,1,3.0)$	9274
提案手法 $(3,1,2.0)$	15738

(a) $dist$ = corr, f_{score} = Rastrigin

手法	データ数
$LSH(7,1,1.0)$	1776
$LSH(7,1,1.5)$	4701
$LSH(3,1,1.0)$	10568
$LSH(5,1,2.5)$	14621
$LSH(7,2,4.0)$	27330
提案手法 $(3,1,1.0)$	4649
提案手法 $(3,1,4.5)$	5353

(b) $dist$ = corr, f_{score} = Rosenbrock

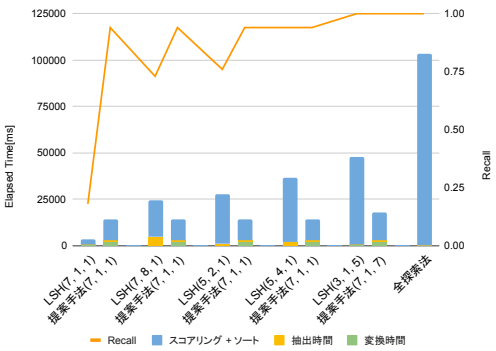
手法	データ数
$LSH(5,2,2.0)$	5748
$LSH(5,5,2.5)$	39113
$LSH(9,7,5.0)$	46270
提案手法 $(1,1,0.6)$	302
提案手法 $(9,1,1.0)$	3038
提案手法 $(9,1,1.5)$	3617
提案手法 $(7,1,5.0)$	6502
提案手法 $(9,1,5.0)$	9324

(c) $dist$ = uni, f_{score} = Rastrigin

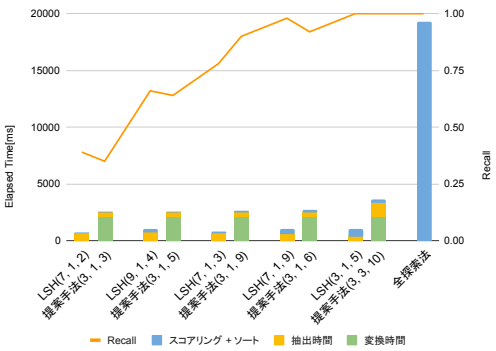
手法	データ数
$LSH(5,1,3.0)$	14051
$LSH(5,1,4.5)$	47931
提案手法 $(3,1,1.5)$	1896
提案手法 $(3,1,2.0)$	2314
提案手法 $(3,1,3.0)$	2529
提案手法 $(3,3,5.0)$	9343

(d) $dist$ = uni, f_{score} = Rosenbrock

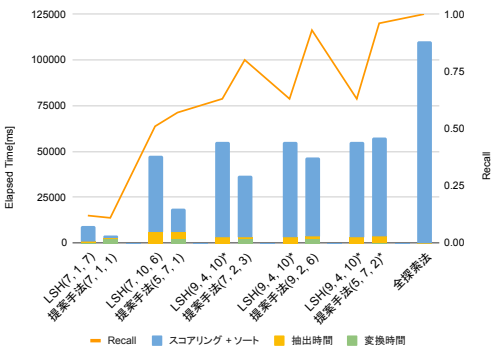
表 4: $D = 100$ の時の各データセットに Top-k 検索を適応した時の抽出されたデータ数



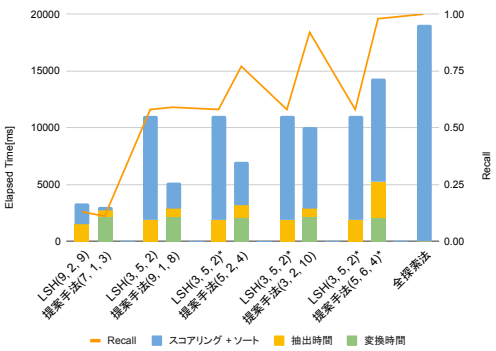
(a) $dist$ = corr, f_{score} = Rastrigin



(b) $dist$ = corr, f_{score} = Rosenbrock



(c) $dist$ = uni, f_{score} = Rastrigin



(d) $dist$ = uni, f_{score} = Rosenbrock

図 4: $D = 300$ の時の各データセットに対する Top-k 検索の実験結果

で $D = 300$ のときは，データの分布とスコア関数の性質によっては，Top-k 検索にかかる実行時間に対する特徴量変換器の適用時間の割合が支配的になることがあり，うまくいかない場合も見られた．また，一様分布のデータに対しては計量学習モデルの訓練自体がうまくいかないこともあり，Top-k 検索時に特徴量変換器の適用を行ってもデータの抽出数を抑えることができない場合もあった．

今後の課題として，計量学習を行う際の学習方法の改善が挙

手法	データ数
$LSH(7,1,1)$	2698
$LSH(7,8,1)$	18349
$LSH(5,2,1)$	25047
$LSH(5,4,1)$	32696
$LSH(3,1,5)$	45129
提案手法 $(7,1,1)$	10632
提案手法 $(7,1,7)$	13826

(a) $dist = \text{corr}$, $f_{score} = \text{Rastrigin}$

手法	データ数
$LSH(7,1,2)$	717
$LSH(9,1,4)$	2074
$LSH(7,1,3)$	1482
$LSH(7,1,9)$	2460
$LSH(3,1,5)$	4101
提案手法 $(3,1,3)$	357
提案手法 $(3,1,5)$	666
提案手法 $(3,1,9)$	1085
提案手法 $(3,1,6)$	1352
提案手法 $(3,3,10)$	1845

(b) $dist = \text{corr}$, $f_{score} = \text{Rosenbrock}$

手法	データ数
$LSH(7,1,7)$	8258
$LSH(7,10,6)$	37149
$LSH(9,4,10)$	47183
提案手法 $(7,1,1)$	1421
提案手法 $(5,7,1)$	11745
提案手法 $(7,2,3)$	30138
提案手法 $(9,2,6)$	38542
提案手法 $(5,7,2)$	48595

(c) $dist = \text{uni}$, $f_{score} = \text{Rastrigin}$

手法	データ数
$LSH(9,2,9)$	10677
$LSH(3,5,2)$	47420
提案手法 $(7,1,3)$	1768
提案手法 $(9,1,8)$	10690
提案手法 $(5,2,4)$	19714
提案手法 $(3,2,10)$	36836
提案手法 $(5,6,4)$	47351

(d) $dist = \text{uni}$, $f_{score} = \text{Rosenbrock}$

表 6: $D = 300$ の時の各データセットに Top-k 検索を適応した時の抽出されたデータ数

げられる．提案手法では，連続的な情報を持つスコア値をラベルに割り当てて計量学習を行ったが，この方法ではラベルの境界付近のスコア値の近いデータが異なるラベルを持つことになってしまい，学習の困難さを招く可能性がある．そのため，ラベル間の近さを考慮した学習方法を導入する必要性がある．

謝　　辞

本研究は，JST CREST JPMJCR22M2 の支援を受けたものである．

文　　献

[1] Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. In Journal of Computer and System Sciences, Vol. 66, pp. 614–656, 2003.

[2] Zhen Zhang, Seung-won Hwang, Kevin Chen-Chuan Chang, Min Wang, Christian A Lang, and Yuan-chi Chang. Boolean+ ranking: querying a database by k-constrained optimization. In Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pp. 359–370, 2006.

[3] Dong Xin, Jiawei Han, and Kevin C Chang. Progressive and selective merge: computing top-k with ad-hoc ranking functions. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pp. 103–114, 2007.

[4] 佐々木 夢, 欅　惇志, 宮崎 純, 検索対象データの事前インデックスを必要としない Top-k 検索アルゴリズムの提案と評価, DEIM Forum, C6-2, 2017.

[5] 池田 達樹, 宮崎 純, 未知スコア関数に対する Top-k 検索アルゴリズムの提案, IEICE-DE2020-3, 2020-06-20.

[6] 池田 達樹, 常　穹, 宮崎 純, 未知スコア関数で評価される多次元データを対象とする LSH を用いた Top-k 検索アルゴリズム, DEIM Forum, J21-2, 2022.

[7] M. Datar, P. Indyk, N. Immorlica and V. Mirrokni, Locality-Sensitive Hashing Scheme Based on p-Stable Distributions, In Proceedings of the Symposium on Computational Geometry, 2004.

[8] Indyk, Motwani. Approximate nearest neighbor: towards removing the curse of dimensionality. STOC ' 98, section 4.2.

[9] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbour- hood components analysis. Advances in neural information processing systems, Vol. 17, 2004.

[10] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. Advances in neural information processing systems, Vol. 18, , 2005.

[11] Hoffer, Elad, and Nir Ailon. Deep metric learning using triplet network. International workshop on similarity-based pattern recognition. Springer, Cham, 2015.

[12] Deng, Jiankang, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[13] Ihab Ilyas, George Beskales, and Mohamed Soliman. A survey of top-k query processing techniques in relational database systems. ACM Computing Surveys, Vol. 40, No.4, pp.1–58, 2008.

[14] Kulis, Brian. Metric learning: A survey. Foundations and Trends® in Machine Learning 5.4 (2013): 287-364.