ユーザの嗜好を考慮した情報推薦のための Skyline 演算の拡張

Skyline Operator for Recomendation Considering User Preference

吉武 亮 † 宮崎 純 † 藤澤 誠 ‡ 天野 敏之 †† 加藤 博一 †
Ryo YOSHITAKE Jun MIYAZAKI Makoto FUJISAWA Toshiyuki AMANO Hirokazu KATO

1 はじめに

計算機の発展によって情報推薦システムは様々なコンテンツを扱うことができるようになった. しかし, 近年ではデータベースが巨大化し, システムが扱うべきデータ量が増大した. さらに時間帯や位置情報, 個人の趣味や嗜好などのユーザに状態に応じて推薦結果を臨機応変に対応させる必要があり, システムに要求される推薦の質の向上が求められている.

Stephan Börzsönyi らによって提案されたスカイライン演算 [1] により, 大規模なデータベースから高速に優秀なデータを抽出することができるようになった. 例えばホテルのデータベースに宿泊費と駅からの距離を表すデータが格納されているとし, その値が小さければ小さいほど優れているという状況でスカイライン演算を行う. データを二次元平面にプロットし, スカイライン演算を行ったときの様子を図 1 を示す.

この図で赤く記されいる線がスカイラインであり, スカイライン上にある点がスカイライン点である. すべての次元で他のデータよりも優れていればそのデータは他のデータを支配すると定義すれば, 他のいかなるデータにも支配されていないデータがスカイライン点である. スカイライン演算を用いた推薦ではこのスカイライン点の集合, すなわちスカイライン集合がユーザに推薦される. スカイライン演算で求められるスカイライン集合はデータ間の支配関係にのみに基づいて決定されるため一意に定まる. そのため限られた情報しかユーザに提示することが出来ず, ユーザの状況や嗜好を反映することが出来なかった.

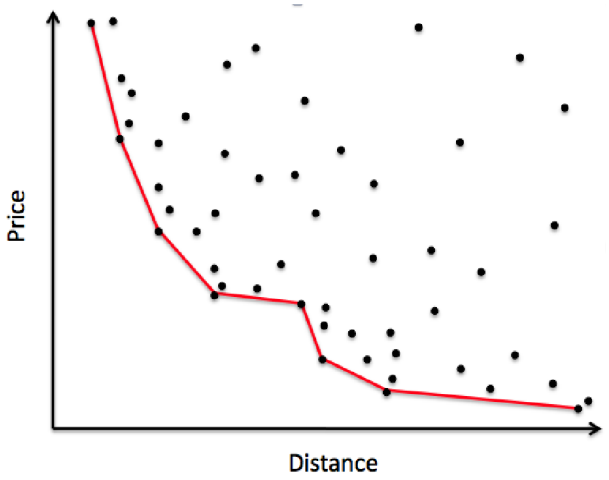

図 1: 二次元におけるスカイライン.

そこで本稿ではスカイラインの周辺に優れたデータがあることに注目し, スカイライン演算にユーザの嗜好を反映させるためにスカイライン付近のデータを提示するスカイライン演算の拡張について述べる.

2 関連研究

これまでにデータ間の距離や支配関係に注目してスカイライン演算にユーザの嗜好を反映させる研究が行われてきた. データ間の距離に着目した Wen Jin らの Thick Skyline[2] ではスカイライン集合付近のデータを付加して抽出する手法を提案した.Thick Skyline ではスカイライン集合付近に優秀なデータが存在することを利用し, スカイライン集合に含まれるデータの周囲に円形の近傍領域を図 2 のように設定し, その領域に含まれるデータを探索, 推薦する手法を提案した.

しかし, この手法ではスカイライン演算と近傍探索が同時に行われ, ユーザの嗜好が変化したときはもう一度計算する必要がある. また, 優秀なデータはスカイライン点の周辺だけでなく, スカイライン周辺にも存在しうるのでスカイライン点周辺のみを探索することでユーザにとって有益なデータが抽出されるとは限らない.

支配関係に着目した Dimitris Papadias らの k-Skyband[3] では支配条件の緩和によりスカイライン周辺のデータを抽出する手法を提案した. 従来のスカイライン演算ではすべての次元において他のデータよりも優れていなければスカイライン集合になることができないが,k-Skyband ではある k 個の次元で劣っていてもスカイライン集合になることができる. この支配条件の緩和により, スカ

† 奈良先端科学技術大学院大学　情報科学研究科
‡ 筑波大学大学院　図書館情報メディア研究科
†† 山形大学　工学部システム創成工学科

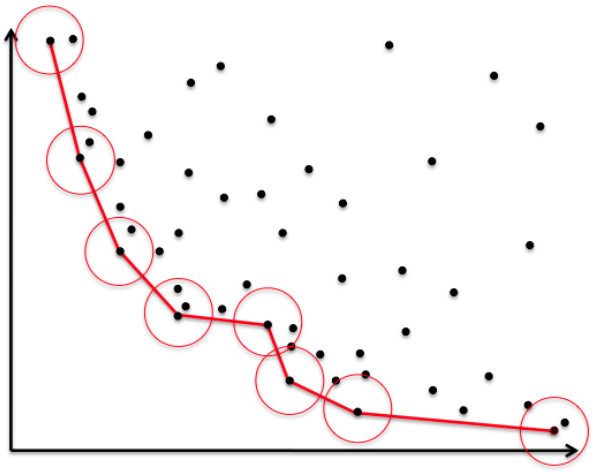

図 2: Thick Skyline.

イライン点付近だけでなくスカイライン付近のデータも抽出することができるようになった. この手法によりスカイラインは帯状に拡張され, スカイライン周辺のデータも推薦できるようになった. しかし,k-Skyband ではデータの優劣を定量的に扱うことが出来ないため推薦には適さない. また, スカイラインの優秀性が失われ, パラメータによりスカイライン集合が劇的に変化するため, ユーザの僅かな嗜好の変化に対応することができない.

本稿では, スカイライン演算でユーザの嗜好を表現するために球形の近傍領域をスカイライン点周囲に設定し, さらにスカイライン周辺のデータも抽出するために隣接する球形の近傍領域の接線からできる帯状の近傍領域を設定する. また, ユーザの嗜好に柔軟に対応するために近傍探索とスカイライン演算は分割して行う点が従来の手法と大きく異なる. ユーザの嗜好が変化したときは近傍探索のみを行い, データ追加, 更新などのデータベースに対する操作が行われたときのみ, スカイライン演算を行い, 対話的な速度での推薦結果の更新を目指す.

3 提案手法

本稿で述べる提案手法は二段階処理を行う. 第一段階にスカイライン演算を行い, 第二段階に近傍領域探索を行う. ユーザの嗜好が変化したときは第二段階の近傍探索のみを行う. ここでは説明を簡単にするため, 各データが持つ値が小さければ優れているという仮定でスカイライン演算を行う. しかし, 一般の場合も計算方法は同様である.

3.1 スカイライン演算

近傍領域はスカイライン点に基づいて決定されるため, まずスカイライン演算を行ないスカイライン点を求める. スカイライン演算には様々な手法が提案されており, 高速にスカイライン点を求める手法 [4, 5] があるが, 例えば,Stephan Börzsönyi らが提案した Block-Nested - Loop を用いた演算 [1] 等がある.

3.2 近傍領域

本提案手法の近傍領域の概念図を図 3 に示す. この近傍領域はスカイライン上を球が大きさを変えながら移動した軌跡の上半分に相当する. この近傍領域により従来の手法よりユーザが興味を持つ点を多く含み, そうでない点を効率よく省くことができる. 例えばユーザが駅からより近いホテルを探しているとすると距離軸が小さくなればなるほど球が大きくなり, 宿泊費が安いホテルを探しているとすると宿泊費軸が小さくなるほど球が大きくなる. 図 3 では距離軸が小さくなるに連れて近傍領域が広がっているため, 駅からの距離を重視するユーザにとって有益なスカイライン演算となる. ユーザにはこの近傍領域に含まれるデータを推薦する.

近傍領域は二種類に分類され, スカイライン点を中心に定義される球領域とその接線により定義される帯領域に分類される.

近傍領域の探索は例えば R 木の範囲問い合わせを用いて行うことができる. そのため, スカイライン演算に用いるデータは R 木構造に格納する必要がある. 範囲問い合わせは矩形領域の問い合わせのみに対応しているため, 近傍領域を包含する最小の範囲問い合わせを行ないデータを抽出する, その後, 抽出されたデータに対しスカイライン点, 及びスカイラインとの距離を計算し,

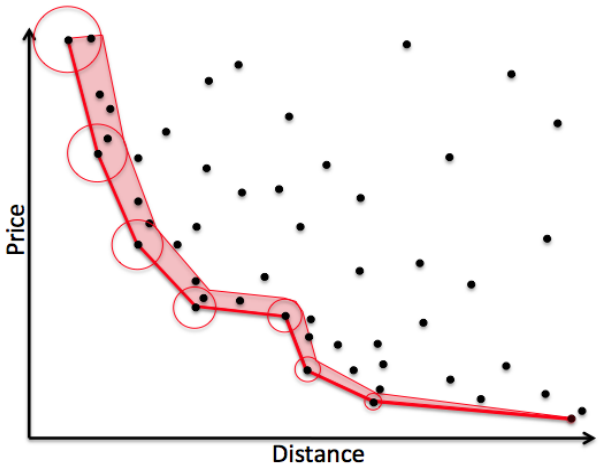

図 3: 提案するスカイライン.

近傍領域に含まれないデータを取り除く. 各データが持つ値は小さいほど優れているという前提ではスカイライン点より原点側にはデータは存在しない. なぜならばもし抽出されたスカイライン点よりも小さな値を持つデータがあればそのデータはスカイライン点にならなければならないためである.

3.2.1 球領域の探索

球領域はスカイライン点を中心に半径 r で与えられる球によって定義される. 図 4 では二次元上のスカイライン点から球領域を問い合わせている. 図 4 中の赤く囲まれた領域にのみ他のデータが存在しうるので (x_1, x_2) から $(x_1 + r, x_2 + r)$ の範囲に存在するデータを範囲問い合わせ, その後, スカイライン点との距離 d を計算し $d > r$ のデータを取り除く. 高次元においても同様に $(x_1, x_2, x_3, ...x_n)$ から $(x_1 + r, x_2 + r, x_3 + r, ...x_n + r)$ のように行ない, スカイライン点との距離 d を計算し $d > r$ のデータを取り除く.

球領域はユーザが重視する次元に依存して大きさが変化する. 現段階では球領域の増加率, 減少率は線形であると仮定しているが, 必ずしもその増加率, 減少率は線形である必要はない.

3.2.2 二次元における帯領域の探索

帯領域は隣接する球領域の共通外接線によって定義され, 図 5 で示される, スカイライン点 $S_1(x_1, y_1), S_2(x_2, y_2)$ の周囲にそれぞれ半径 $r_1, r_2(r_1 > r_2)$ の球領域を定義すると球 R_1 と円 R_2 の共通外接線と点 S_1 から各軸に水平に伸ばした中心線との交点 H により範囲問い合わせの領域が定まる. この帯領域において範囲問い合わせを行う範囲は赤く記されている矩形で決定されるため点 A と点 H を与える必要がある.

まずは共通外接線である直線 AB を求める. 点 S_2 から線分 AS_1 に垂線を $S_v(x_v, y_v)$ におろす. $\angle AS_1S_2 = \angle\theta$ とし, 点 S_1 から点 S_2 の距離を v とする. この時, $\triangle S_vS_1S_2$ に注目すると

$$\cos\theta = \frac{r_1 - r_2}{|v|} \tag{1}$$

となり, $\angle\theta$ は

$$\theta = \arccos\frac{r_1 - r_2}{|v|} \tag{2}$$

となる. 次に $\angle HS_1S_2$ を $\angle\phi$ とすると

$$\tan\phi = \frac{y_2 - y_1}{x_2 - x_1} \tag{3}$$

となり, $\angle\phi$ は

$$\phi = \arctan\frac{y_2 - y_1}{x_2 - x_1} \tag{4}$$

となる. $\angle\theta$, $\angle\phi$ により, 点 $A(x_a, y_a)$, 点 $B(x_b, y_b)$ は

$$x_a = x_1 + r_1\cos(\theta + \phi)$$
$$y_a = y_1 + r_1\sin(\theta + \phi) \tag{5}$$
$$x_b = x_2 + r_2\cos(\theta + \phi)$$
$$y_b = y_2 + r_2\sin(\theta + \phi) \tag{6}$$

で与えられる. 点 A は必ず点 S_1 よりも大きな値を, 点 B は必ず点 S_2 よりも大きな値を持つので点 A, 点 B は一意に求まる. 直線上の 2 点が確定したため直線 AB の方程式は

$$y = \frac{y_b - y_a}{x_b - x_a}(x - x_a) + y_a \tag{7}$$

で与えられる. 式 (7) の y に y_1 を代入すると点 H が決定され, 点 A と点 H で与えられる矩形で範囲問い合わせで近傍のデータを問い合わせる.

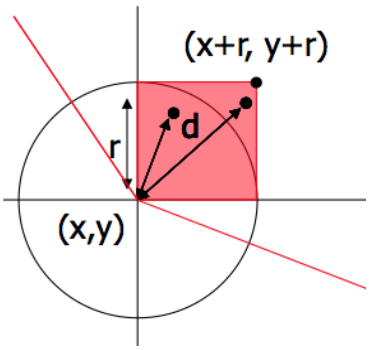

図 4: 二次元の球領域.

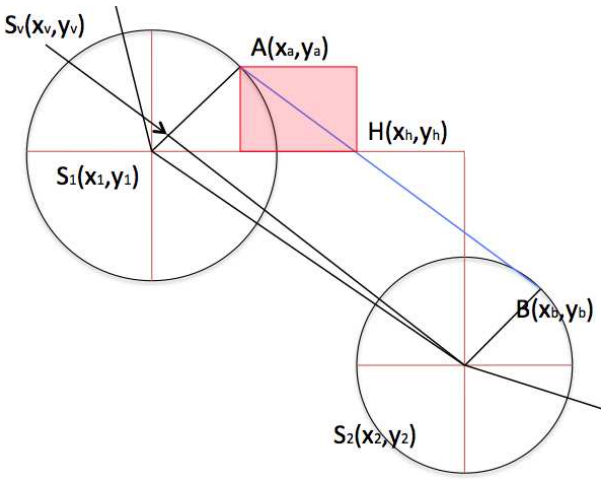

図 5: 二次元の帯領域.

次に問い合わされたデータ P が帯領域に含まれるかどうかを点 S_1 と点 S_2 を結ぶ直線 S からの距離を計算し確かめる. 最初に点 A と点 H から直線 S までの距離 d_a, d_h を求める. 直線 S の方程式を式 (7) と同様に求めると

$$y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1 \tag{8}$$

となる. これを変形すると

$$(y_2 - y_1)x + (x_1 - x_2)y + x_2y_1 - x_1y_2 = 0 \tag{9}$$

よって点 $A(x_a, y_a)$ と点 $H(x_h, y_h)$ から直線 S までの距離 d_a と d_h は

$$d_a = \frac{(y_2 - y_1)x_a + (x_1 - x_2)y_a + x_2y_1 - x_1y_2}{\sqrt{(y_2 - y_1)^2 + (x_1 - x_2)^2}} \tag{10}$$

$$d_h = \frac{(y_2 - y_1)x_h + (x_1 - x_2)y_h + x_2y_1 - x_1y_2}{\sqrt{(y_2 - y_1)^2 + (x_1 - x_2)^2}} \tag{11}$$

範囲問い合わせによって抽出されたデータ点 P から点 S_1 と S_2 を結ぶ直線 S までの距離 d_p は式 (10)(11) と同様に与えられる. この距離 d_p がスカイラインから共通外接線までの距離より小さければデータ点 P は近傍領域に含まれ, 大きければ取り除かれる. 点 H から直線 S への垂線と直線 S の交点を H' とし, 点 S_1 から点 H' までの距離を T とする. 点 P からも同様に直線 S への垂線と直線 S の交点を P' とし, 点 S_1 から点 P' までの距離を t とする. 点 H から S_1 と A を結ぶ直線に垂線を引き, その交点を B とすると点 A から点 B までの距離は $d_a - d_h$ となる. この時, 点 P を通る共通外接線とスカイラインの距離 d_ps は

$$d_ps = d_h + \frac{t}{T}(d_a - d_h) \tag{12}$$

で与えられる.$d_p <= d_ps$ ならば帯領域に含まれ,$d_p > d_ps$ ならば帯領域に含まれないと判定する.

3.2.3 三次元以上における帯領域探索

三次元以上では球領域と帯領域が立体になるため任意の一軸に垂直な平面に対し射影して低次元化して近傍探索を行う. 図 7 では三次元空間での帯領域の状態を表している. 球体は三次元における球領域で, 明るい灰色で表されている領域が三次元空間における帯領域である. 三次元以上における帯領域も前後のスカイライン点の球領域同士の共通外接線によって定まる.

スカイライン点 S_1, S_2 の周囲に半径 $r_1, r_2 (r_1 > r_2)$ の球領域を定義し,Z 軸に垂直な平面に対して射影すると,xy 平面上で二つのスカイライン点 S_1, S_2 の半径それぞれ R_1, R_2 の球領域の共通外接線が与えられる (図

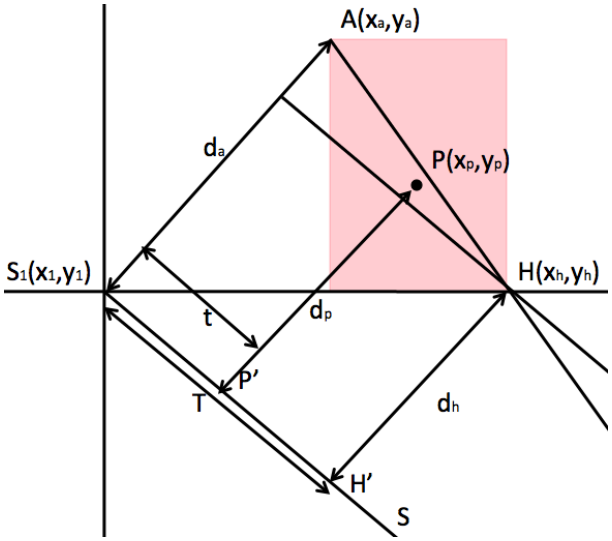

図 6: 帯領域における距離判定.

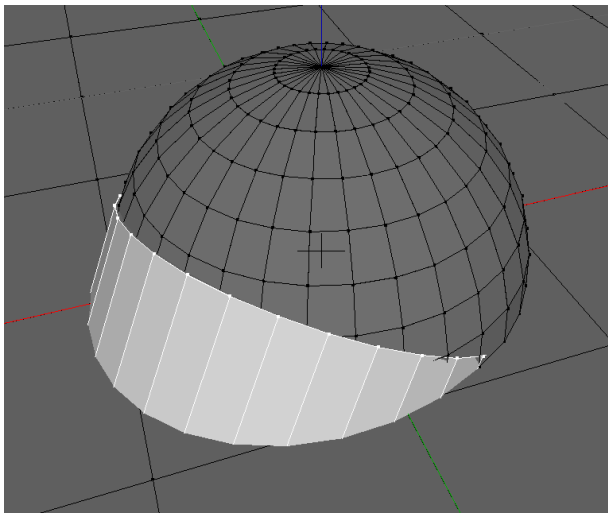

図 7: 三次元の帯領域概念図.

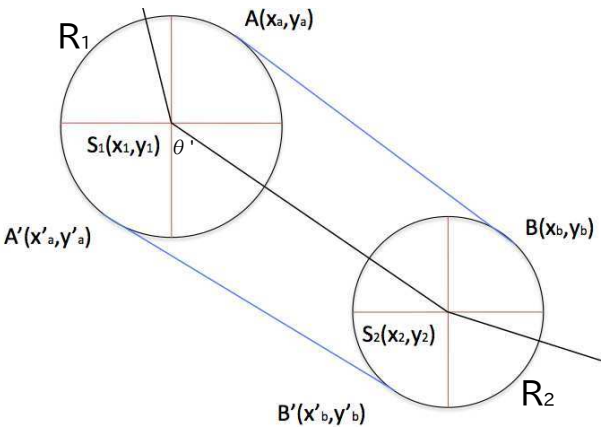

図 8: Z 軸に対する写像.

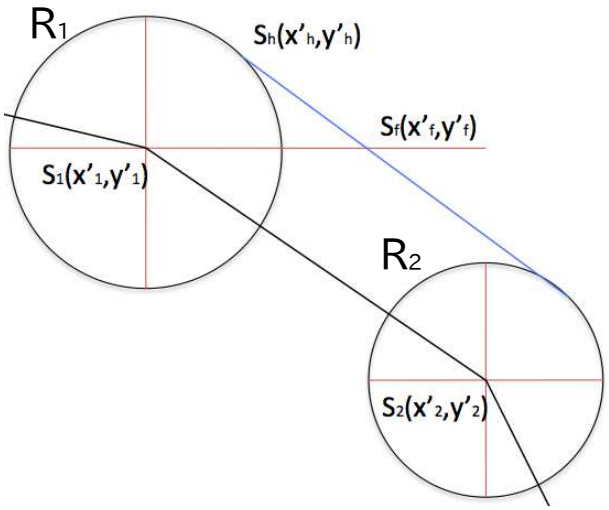

図 9: 横から見たスカイライン.

8). この時スカイラインと y 軸がなす角を θ' とする. 点 A, B と点 A', B' に関しては式 (5)(6) を利用して同じように求めることができる.

次にスカイライン点 S_1 から xy 平面上で最も離れた点 S_f を求める. 点 S_f は Z 軸に対して射影したときスカイラインと重なった位置に現れる特性を利用して点 S_f を求める. 点 S_f を求めるためにスカイラインを真横から見た時の様子を図 9 に示す.

点 S_f と点 S_h は節 3.3.2 の式 (1) から (6) と同じくして求めることができる. 点 $S_f(x'_f, y'_f)$ と点 $S_h(x'_h, y'_h)$ は元の座標系より θ 度回転しているので

$$x' = x\cos\theta' - y\sin\theta' \quad y' = x\sin\theta' + y\cos\theta' \quad (13)$$

により座標系を変換する.

以上より, 帯領域を包含する範囲問い合わせは共通外接線によって与えられる点 $A(x_a, y_a, z_a)$ 点 $A'(x'_a, y'_a, z'_a)$ 点とスカイラインによって与えられる $S_f(x_f, y_f, z_f)$ と $S_h(x_h, y_h, z_h)$ の 4 点により決定される. 四次元以上の場合も同様に任意の一軸から射影することで低次元に落とし込み計算する.

4 まとめ

本稿ではスカイライン点付近に球領域と帯領域による近傍領域を設定し, その領域に含まれるデータを R 木の範囲問い合わせを用いて探索しスカイライン集合に付加して推薦することでスカイライン演算でユーザの嗜好や状況を表現する手法を提案した. スカイラインはその特性上次元数が多くなればなるほど他のデータを支配することが困難になってくるのでスカイライン集合が従来のスカイライン集合よりも多くなってしまうことが予想されるため, ユーザに提示する手法とユーザの嗜好や状況をどのようなインタフェースで取り込むかの工夫が必要となる.

今後の課題として本手法を用いたスカイライン演算による推薦システムの開発を行ない, 近傍領域計算が対話的な時間内で行えるか実験を行う必要がある. そのため, 近傍領域の探索を計算コストのより低い近似を利用して, データ更新での逐次計算の検討を行う予定である. また, 各次元の正規化のため, 円形の球領域でなく, 楕円形領域への拡張も行う予定である.

5 謝辞

本研究の一部は，科研費補助金基盤研究 (C)(課題番号:23500121) の支援による．ここに記して謝意を表す.

参考文献

[1] Stephan Börzsönyi, Donald Kossmann, Konrad Stocker ”The Skyline Operator” In Proceeding of the 17th International Conference on Data Engineering 2001

[2] Wen Jin, Jiawei Han, Martin Ester ”Mining Thick Skylines over Large Databases” In PKDD2004 2004

[3] Dimitris Papadias, Greg Fu, Jp Morgan Chase, Bernhard Seeger ”Progressive Skyline Computation in Database Systems” In Proceeding of ACM Trans.Database Syst 2005

[4] Donald Kossamann, Frank Ramsak, Steffen Rost ”Shooting stars in the sky : an online algorithm for skyline queries” In Proceeding of the 28th international conference on Very Large Data Bases 2002

[5] Jan Chomicki, Parke Godfrey Jarek Gryz Dongming Liang ”Skyline with Presorting” In Proceeding of Data Engineering 19th international Conference 2003