データマイニング手法を用いたモバイルエージェント分散データ検索システム

何 斌達　相田 仁
東京大学 新領域創成科学研究科 基盤情報学

ネットワーク上にある分散したデータを検索したいときに、もし検索が非常に複雑、検索対象のデータ量が大きい、かつ検索手法多変な場合、検索コードの再利用性が低い。したがって、検索したいときに、その場で計算コードをエージェントのように検索ターゲットに配って、ターゲットの場所で実行したほうが効率的である。つまり、モバイルエージェントを利用した検索がより効率のいいものができる。本稿では、データマイニングの手法を用いて、モバイルエージェントのより賢い移動プランを作り出し、より指向性の高い検索を行うためのアプローチを提案し、それの実証と応用について論じる。

Mobile Agent Distributed Data Search System Using Data Mining Approach

He Binda　Aida Hitoshi
Department of Frontier Informatics, Graduated School of Frontier Sciences, University of Tokyo

When search on the distributed data on network, if the search process is very complicated, data quantity of search target is very large, and search approaches are variable, search code cannot be reused frequently. Therefore, when search, to distribute search code like an agent to the search target is more efficient. In this paper, a new approach to use data mining on mobile agent, which creates a more intelligent move plan and a more directional search, will be introduced.

1 研究背景

現在、あらゆるコンピュータデバイスが世の中に存在し、人間の補助ツールとして発展してきた。一般のコンピュータから、ノートコンピュータ、PDA、携帯電話までが人のために情報サービスを提供している。同時に、このあらゆる情報機器はパーソナルな情報を収集している。将来、ユビキタスコンピュータ社会になるとともに、冷蔵庫や電子レンジなどの日常生活用品まで情報機能もつけられるだろうと思われている。そこで、コンピュータデバイスはパーソナルなデータだけではなく、情報端末のある場所のあらゆる情報まで提供することができる。この分散されている情報の中から、自分のほしいものを検索するために、分散データ検索システムを構築することが必要とされている。

このようなシステムで賢い検索方法を見いだすのが、本研究の目的である。

2 モバイルエージェントを用いた検索の考察

このような検索モデルを考えよう。

- 検索対象が分散されている。
- 分散されている場所に存在するあらゆるものが検索計算用の資源になる。
- さらに、検索の種類も変化自由自在で、数多く存在するものである。
- その上、検索のリアルタイム性が非常に重要である。

一般的に、このようなオープンな検索を行いたい場合、よく使われるのがバケツリレー方式で、図1のような形で、ネットワーク上にある

検索ターゲット候補にリクエストをマルチキャストし、さらにそれぞれの行き先にある候補にリレーするというようなやり方で検索を行っている。実際にはGnutella[gnutella]はこれを採用している。

しかし、このような方法を使うと、ネットワークにも負担かけるし、盲目な検索であり、指向性の低い検索であり、検索自体が難しく、インテリジェントなものではなく、効率が低い。

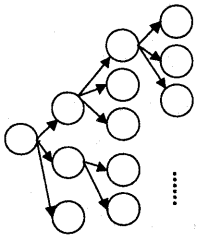

図1:relay search model

そこで研究対象とするモバイルエージェントを用いた分散データ検索システムの特徴を分析する。

モバイルエージェントシステムを利用する場合、複数のエージェントがシステム上で生きている。したがって、他のエージェントから情報提供を求め、自分の利用価値のある情報を吸収してから、出発したほうが、賢いやり方である。もちろん、エージェントは様々存在するが、自分の要求と似ているエージェントの情報を吸収し、似ていないエージェントの情報を逆利用すればいい。

したがって、複数のサービス（エージェント種類）が存在する場合、異なったサービスの間にも、相互関連があり、その相互関係を見いだすと、エージェントの移動プランにプラスの成分を加えることができる。

同種のサービス同士だけだと、単純にエージェントの数を計算するような統計的手法[Minar99]だけで、有効な情報が得られるが、複数の場合、それほど簡単ではない。そこで、データマイニングの手法によるプランニングを考案する。

3 データマイニングの手法による相関分析

3.1 相関関係分析（Association）

複数のアイテムがある場合、その項目間の相関関係を導き出す手法の一つは「相関関係分析」、いわゆる「バスケット分析」。簡単に言うと、スーパーマーケットでどの商品とどの商品が同時に買われる可能性が高いかを分析する手法である。

それは個々のアイテムの集合を一つ一つのトランザクションとして、全トランザクションの集計を行う。

たとえば、スーパーマーケットで、商品の品目をアイテムとして、顧客のショッピングカート内のものを個々の集合、つまりトランザクションとすると、

{ビール、冷凍食品、たばこ}
{ビール、たばこ}
{冷凍食品、アイスクリーム}

のようなトランザクションの集合ができる。そこで、各アイテム集合Aの支持度は

sup(A) = Aを含むトランザクション数 / 全トランザクション数

次に確信度を計算するが、あるトランザクションTが発見された場合、その集合の中にAも含まれる確信度は

conf(A,T) = sup(T+A) / sup(T)

このような計算でより高い確信度を見いだすことがバスケット分析と言われ、スーパーマーケットの商品陳列分析やオンライン書店のお勧め本などでよく使われている。

ここで、各トランザクションのビール、冷凍食品などのアイテムを数々のエージェントの種類に置き換えると、違う種類のエージェント間の関係を見いだすことができる。

3.2 Apriori アルゴリズム

Apriori アルゴリズムはバスケット分析を行う高速アルゴリズムである。それは下で示されている。

```
F1 = { frequent 1 - item sets };
K = 2;
```

```
While ( F(k-1) is not empty ) {
        C(k) = Apriori_generate( F(k-1) );
        For all transactions t in T {
                Subset( C(k), t );
        }
        F(k) = { c in C(k) s.t. c.count>=minimum_support };
}
Answer = union of sets F(k);
```

このアルゴリズムでは、ある値以上のサポート度を持つ集合を取り出すことができ、その最小サポート度に満たさない集合を切り落とす。

このアルゴリズムは高速にバスケット分析を行うことが可能であるが、トランザクションの数多いことや、トランザクションの平均サイズが大きいことや、最小サポート度の制限が低いことなどは計算速度に影響する。その関係は図2、3、4で表示されている。

同時に、サポート度が小さい集合、つまり全体トランザクション群の中の少数派である集合は無視されることで、確信度の信頼性が高まる。

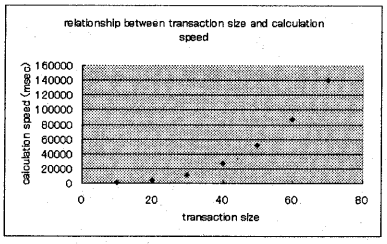

図2 トランザクションのサイズと計算速度の関係

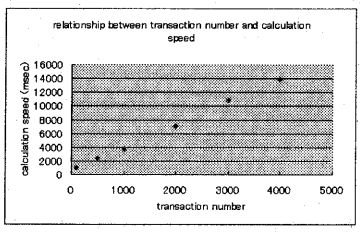

図3 トランザクションの数と計算速度の関係

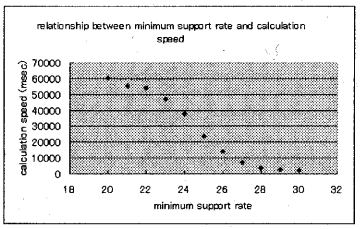

図4 最小指示度と計算速度の関係

4 モバイルエージェントのプランニングにおける利用法の提案

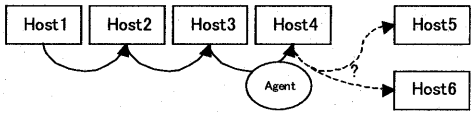

図7: moving of agent

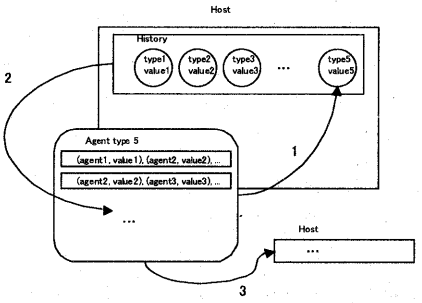

図8: planning of agent using data mining

モバイルエージェントを用いた分散検索システムの場合、エージェントの種類は多様であり、これらをトランザクションの要素として考え、バスケット分析を行うと、どのようなエージェントが行っているホストに行ったほうがいいのか、あるいはどのようなエージェントが行かないホストに行ったほうがいいのかを求めだすことができる。これをプランニングの判断素材として扱うと、効率のいい検索ができる。

したがって、以下のような移動プランを提案する。

(a) エージェントの種類

エージェントは様々な種類があり、各々の種類にユニークな認識記号（GUID）がある。以下はtype　nで記述する。

(b) 評価の定義

エージェントが様々なホストを歩き回り、そのホストで計算を行うが、その計算結果に満足と不満足というホストに対する評価を出し、それを以下で value n で記述する。そこで、value はホストの持っているデータに対する計算で得られる結果で、value n = function (host x, GUID n)になる。そこで、value の値の範囲は無限であるなら、分析の計算量が非常に多い。かつ、これはホストに対する満足度であるため、絶対値で表しても、種類の違うエージェント同士の間でも意味の通じない値になる。したがって、すべてのエージェントで理解のできる相対値にしなければいけない。ここで、満足を 1 で、不満足を 0 で計算する。

(c) ホストの履歴

ホストには数々のエージェントが訪れるが、それらのエージェントがホストに評価した値をホストのログとして、保存する。すると、以下のような形になる。

{(type1, value1), (type2, value2)...}

(d) エージェントの履歴

エージェントがホストを通過したときに、ホストの履歴を取り、自分の分析対象として格納する。格納の場所は、エージェントのモバイル性を考慮し、サービスを提供しているホストに転送することも考えられる。履歴は以下のような形になる。

{(type1, value1), (type2, value2)...},
{(type2, value2), (type3, value3)...}
...

(e) 支持度計算

履歴を分析の対象トランザクションとし、ある指示度の範囲でバスケット分析を行い、下のような結果を求める。指示度は support n で記述する。

{(type1, value1)} = support1
{(type1, value1), (type2, value2)} = support2
{(type2, value2), (type3, value3)} = support3
...

(f) 候補履歴の取得

エージェントの次の移動先を計算するときに、候補となるホストのリストをあげ、それらのホストに対して c で説明した履歴を問い合わせする。すると、以下のようなリストになる。

host1: {(type1, value1), (type2, value2)...}
host2: {(type2, value2), (type3, value3)...}
...

(g) 確信度の計算

f で得られたホスト履歴リストと e で得られた指示度リストで各々ホストの確信度を計算する。具体的には、ホスト履歴リスト内の各々の履歴に対して、支持度リスト内で一番マッチするトランザクションを見つけ出し、そのサポート度を、エージェント自分を加えたサポート度で割ると、確信度が得られる。

(h) 候補選出

g で得られた数々の確信度の内、一番高い値を持つホストを捜し、次の移動先として選出する。

(i) 移動

移動後は e に戻り、移動、分析の操作を繰り返して実行する。満足できる数の結果が得られたなら、検索作業が終了。

(j) 計算の場所

バスケット分析は大きな計算量が前提で正確結果を出すことが可能であるため、必ずしも、常に、ユーザのコンピュータで実行することはなく、場合によって、ある計算サーバがその分析を担うこともありうる。

(k) 計算の頻度

バスケット分析の計算量は大きいので、ログが入ってきたときに、すぐ計算し直すことはなく、ある量のログがたまってきたら、計算を始めるというような頻度で行ったほうが効率的である。

5 評価実験

シミュレーションしアルゴリズムを評価するために、ある相関関係を持っているエージェントモデルを作る必要がある。モデルは下のようになる。

図9のように、n×nのエージェントプラットフォームが存在し、それぞれのプラットフォームが一つの整数を持ち、つまり、0～n×nの整数が存在する。

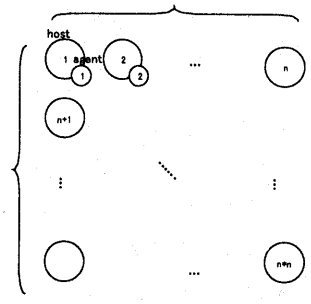

図9: Simulation Model

初期状態では、一つのプラットフォームに一つのエージェントが存在する。エージェントにもランダムに数字のGUIDが振られている。

エージェントにプラットフォームに対する視野があり、それは周りのn個とし、次の移動プランで使用する候補となる。周りのn個以外のプラットフォームは見えなく、移動プランで使用しない。

エージェントが探す目標は、下の二つの条件に満たすプラットフォームである。

1、 プラットフォームのIDは自分のGUIDの倍数である。

2、 自分のGUIDがプラットフォームのIDの倍数である。

たとえば、21のGUIDを持っているエージェントが正しい結果出しているならば、3のGUIDを持っているエージェントにも正しい結果出す可能性が高い。

このようなモデルで、n=10の環境で実行した結果、図10，11，12，13のようになる。

比較するために、同じモデルを採用し、移動プランをランダムに決めることにする。

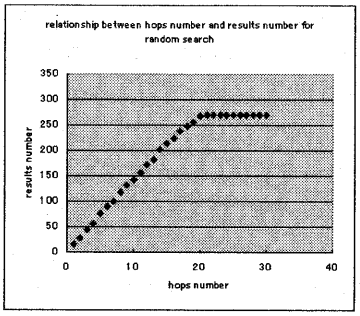

図10 ランダム検索におけるホップ数と結果数の関係

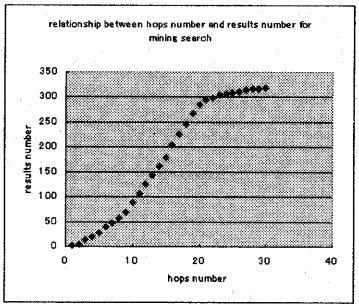

図11 マイニング検索におけるホップ数と結果数の関係

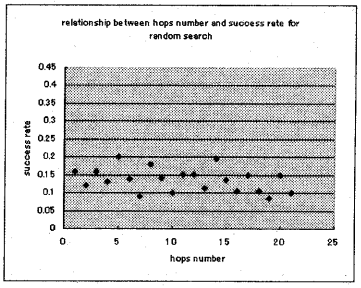

図12 ランダム検索におけるホップ数と成功率の関係

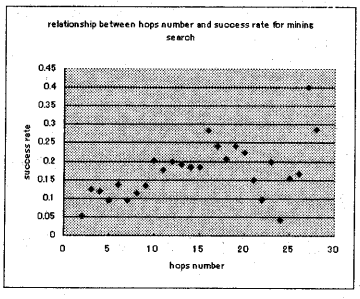

図13 マイニング検索におけるホップ数と成功率の関係

図 10 と図 11 から見ると、ランダムに検索をする場合、エージェントは一部の答えを漏れてしまうことがあるに対し、バスケット分析を行った場合、より完全な解を求めることが可能。図 12 と図 13 から見ると、ランダムに検索をする場合、プランの成功率は常に一定の水準に保つことに対し、バスケット分析を行った場合、学習能力を持ち、ホップ数が増えるとともに、成功率も高くなる。

図 13 の場合、20 歩数後の成功率が下がっているように見えるが、それはモデルとした問題では、すでに正解が少なくなり、見つかりにくくなっているからだ。図 10 と図 11 の比較で分かるように、両方とも 20 歩数の後で結果数の上昇が収まるが、図 11 での結果数はまだ粘り強く上昇していた。

6 結論

データマイニングの手法はモバイルエージェントのルーティングに情報を提供し、あらかじめ行き先の結果を予測する手法であり、あらゆる分散検索あるいは計算システムで利用する可能性がある。その手法はターゲットのデータそのものを扱うのではなく、モバイルエージェントの特徴であるエージェントの移動履歴を用いてエージェント間の関係を求め出し、間接的に予測する方法である。

データマイニングの手法は集積する履歴の数が多ければ多いほど、より正確な予測が可能になるが、その計算量も増えていく。バスケット分析による計算では、履歴の数は予測の正確度を影響し、サンプルとする履歴の数が多ければ、より正しい計算ができる。

各履歴のサイズは予測の結果とは無関係でありながら、計算速度を落とす悪影響を与える。Apriori で使用する最小支持度は大きく設定すると、正しい結果も落とされるので、履歴の数が多くなると、適切な支持度を設定することが必要である。

データマイニングを用いたモバイルエージェント移動プランニングは同種のエージェント同士のデータ交換ではなく、異種エージェント同士でも特定の関係を見出す方法である。それはすでに出来上がっているエージェントプラットフォームに新しく参加する新種エージェントに対して、より高速な環境適応能力を与えることが可能である。

7 課題

今後の課題として、シミュレーションで、的中率だけではなく、ネットワーク上行き渡る能力を考察し、より大きいプラットフォーム群でのプランニングをすることが必要である。そのアルゴリズムも考案していきたい。なお、時系列パターンも意味もつシステムに対するアルゴリズム修正も必要とされている。

参考文献

[Minar99] Nelson Minar and Kwindla Hultman Kramer and Pattie Maes, Cooperating Mobile Agents for Mapping Networks, Proceedings of the First Hungarian National Conference on Agent Based Computation, 1999, http://www.media.mit.edu/~nelson/research/routes-coopagents/

[猪口 00] 猪口 明博, 鷲尾 隆, 元田 浩, 熊澤 公平, 荒井 尚英, 多頻度グラフパターンの完全な構想区マイニング手法, 人工知能学会誌 Vol. 15 No. 6 (Nov 2000)

[鹿山 00] 鹿山 俊洋, 堀内 匡, 元田 浩, 鷲尾 隆, 遂次ペア拡張による機構像データからの分類規則学習, 人工知能学会誌 Vol. 15 No. 3 (May 2000)

[gnutella] http://gnutella.wego.com/

[Agrawal 94] Agrawal, R. and Srikant, R.: Gast Algorithm for Mining Association Rules in Large Databases. Proc. Of the 20th Very Large Data Bases Conference, pp. 487-499, 1994