4E-4

P2P ネットワークにおける保有コンテンツの類似性を考慮した検索効率向上手法

遠藤 慶一† 最田 健一‡ 川原 稔§ 高橋 豊†
†京都大学大学院 情報学研究科 ‡有限会社 シー・オー・コンヴ
§愛媛大学 総合情報メディアセンター

1 はじめに

インターネットの普及により，日常生活の様々な局面でデジタル情報の共有・活用が計られている．しかし，個々のサーバに蓄積されたコンテンツを利用するには，多くの場合コンテンツを保有しているサーバとは別の情報源の助けを借りて目的とするコンテンツにアクセスする必要がある．したがって，良質なコンテンツが公開されていたとしても，そのような情報源に取り込まれていないコンテンツに対しては，利用者はその存在を知る機会がない．本研究の目的は，特定の情報源に依存することなし（ブローカレス）にコンテンツの流通を可能とする情報流通ネットワーク構築の基盤技術となる，自己組織化アルゴリズムの開発である．通信処理の基盤技術としては，Peer-to-Peer (P2P) ネットワーク技術を用いる．P2P ネットワークにおいて検索を実現するには，検索クエリの隣接ノードへのリレーを繰り返すことにより多数のノードに検索クエリを行き渡らせればよい．しかし，ネットワークの規模が大きくなると，限られた帯域や許容遅延のもとでは，限られた範囲のノード（一定ホップ数以内にあるノード）にしか検索クエリを届けることができない．したがって，何らかの方法で検索効率を高め，通信量を低く抑えつつ検索成功率（ヒット率）を上げることが必要となる．

文献 [2] では，各ノードの保有コンテンツ内容などをもとに検索クエリのルーティングを行う手法が提案されている．本研究では，その手法を発展させ，各ノードが保有するコンテンツの類似性に基づいてネットワークの論理的な接続トポロジを変更することにより，検索効率の向上を図る．保有コンテンツの種類が似通っているノード同士が少ないホップ数でつながるようにすることにより，検索クエリが届く範囲のノードに必要な情報が存在する確率を高める．P2P ネットワークにおいては，ネットワーク全体を管理するコンピュータが存在しないため，前述のようなトポロジ変更を実現するには，各ノードが自律的に他ノードとの接続，切断を行う必要がある．この作業のことを本稿では自己組織化と呼ぶ．

2 自己組織化アルゴリズム

本章では，本稿で提案する自己組織化アルゴリズムについて説明する．

2.1 カテゴリ空間

各ノードの保有コンテンツの類似性を判定するために，カテゴリ空間というものを導入する．カテゴリ空間は N_c 次元のベクトル空間であり（N_c をカテゴリ数と呼ぶ），どのような種類のコンテンツをどのような割合で保有しているかによって，カテゴリ空間におけるノード j の座標 $\boldsymbol{x}_j$ が決定される．カテゴリ空間において近くに存在するノード同士は，保有コンテンツの種類が似通っていることとなる．実際に保有コンテンツからノードの座標を決定するには，コンテンツの種類（カテゴリ）を判別する方法が必要となるが，それにはコンテンツの種類を示すタグをあらかじめ人手で付加しておく方法の他，Latent Semantic Indexing[1] などの手法を用いて，コンテンツの内容から種類を判別する方法も考えられる．本稿ではノードの座標を決定する具体的な方法に関しては述べず，何らかの方法により各ノードがカテゴリ空間にマッピングされたものとして以後の説明を行う．

2.2 ノードの満足度

本稿で提案する自己組織化アルゴリズムでは，各ノードが以下で定義する満足度という評価値を大きくすることを目標として，接続ノード変更などの動作を行う．ノードの満足度は，カテゴリ空間における接続ノードとの位置関係から算出する．本研究では，カテゴリ空間において近くのノードと接続しているほど，また，様々な方向のノードと接続しているほど，ノードの満足度は高くなることとする．具体的には，接続しているノードの集合を $\mathcal{A}$ としたときのノード j の満足度 $f_j(\mathcal{A})$ を以下のように定義する：

$$f_j(\mathcal{A}) = \sum_{k\in\mathcal{A}} \frac{1}{\beta_\mathrm{d}|\boldsymbol{x}_k - \boldsymbol{x}_j| + 1} \cdot \frac{\theta_{\min}(j,k,\mathcal{A})}{\pi},$$

$$\theta_{\min}(j,k,\mathcal{A}) = \min_{l\in\mathcal{A}\setminus\{k\}} \arccos \frac{(\boldsymbol{x}_k - \boldsymbol{x}_j)\cdot(\boldsymbol{x}_l - \boldsymbol{x}_j)}{|\boldsymbol{x}_k - \boldsymbol{x}_j||\boldsymbol{x}_l - \boldsymbol{x}_j|}.$$

ここで，距離影響係数 β_d は，カテゴリ空間におけるノード間の距離がどの程度満足度に寄与するかを定める非負定数である．

2.3 自己組織化の手順

自己組織化は以下の手順で行われる．

1. **広告パケットのフラッディング**
   満足度を高めるために各ノードは定期的に Time To Live (TTL) を τ_a とした広告パケットをフラッディングする．

2. **広告パケット送信元の評価**
   広告パケットを受け取ったノード r は，送信元ノード s（広告パケットをフラッディングしたノード）と接続することにより自身の満足度を上昇させることができるならば，接続許可パケットをノード s に返信する．各ノードが接続できるノード数の上限を定数 N_a とし，現在ノード r が接続しているノードの集合を $A(r)$ とすると，ノード r がノード s と接続することにより得られる満足度上昇の最大値 $d_r(s)$ は以下の式で計算できる：

$$d_r(s) = \begin{cases} f_r(A(r)\cup\{s\}) - f_r(A(r)), & [|A(r)| < N_\mathrm{a}], \\ \max\limits_{e\in A(r)} f_r((A(r)\setminus\{e\})\cup\{s\}) - f_r(A(r)), & \\ & [|A(r)| = N_\mathrm{a}]. \end{cases}$$

   $d_r(s) > 0$ ならばノード s に接続許可パケットを返信する．

3. **接続許可パケット返信ノードとの接続**
   広告パケットの送信元ノード s は，接続許可パケットを返したノードのうち，自身の満足度上昇が最大とな

Improving Search Efficiency in Consideration of Content Similarity on P2P Networks
Keiichi Endo† Kenichi Saita‡
Minoru Kawahara§ Yutaka Takahashi†
†Graduate School of Informatics, Kyoto University
‡CO-CONV, Inc.
§Center for Information Technology, Ehime University

るようなノードと接続する（満足度上昇が正の場合に限る）．

3 シミュレーション

本章では，自己組織化の効果を確認するために行ったシミュレーションに関して述べる．

3.1 コンテンツベクトルと興味ベクトル

本研究のシミュレーションでは，(内容が異なる) N_f 個のファイルを N_n 個のノードに N_p 個ずつ重複を許して配置しておき，初期ネットワークでのファイル検索ヒット率と，自己組織化実行後のヒット率を比較する．ファイルには様々な内容のものがあり，ノードによってどのような種類のファイルをよく検索するか／多く保有しているかが異なるという状況を想定する．そのために，N_c 次元のコンテンツベクトル (向きはファイルの大まかな内容を表す) $\boldsymbol{f}_i$ と人気度 p_i を各ファイル i ($i = 1,\ \ldots,\ N_\mathrm{f}$) に，ノードの嗜好を表す興味ベクトル $\boldsymbol{n}_j$ を各ノード j ($j = 1,\ \ldots,\ N_\mathrm{n}$) にそれぞれ設定する．そして e_r を定数として，ノード j がファイル i を検索する頻度が $p_i(\boldsymbol{f}_i \cdot \boldsymbol{n}_j)^{e_\mathrm{r}}$ に比例すると仮定する．

シミュレーションにおいては，Zipf 係数 α_c, α_p を用いて，以下の式によってコンテンツベクトル，人気度，および興味ベクトルを設定している．

$$\boldsymbol{f}_i^m = \frac{(\hat{\boldsymbol{f}}_i^m)^{-\alpha_\mathrm{c}}}{\sqrt{\sum_{k=1}^{N_\mathrm{c}} k^{-2\alpha_\mathrm{c}}}},\ p_i = (\boldsymbol{r}^i)^{-\alpha_\mathrm{p}},\ \boldsymbol{n}_j^m = \frac{(\hat{\boldsymbol{n}}_j^m)^{-\alpha_\mathrm{c}}}{\sqrt{\sum_{k=1}^{N_\mathrm{c}} k^{-2\alpha_\mathrm{c}}}}.$$

ここで，$\boldsymbol{X}^n$ はベクトル $\boldsymbol{X}$ の第 n 成分を表す．また，$\hat{\boldsymbol{f}}_i$ ($i = 1,\ \ldots,\ N_\mathrm{f}$) および $\hat{\boldsymbol{n}}_j$ ($j = 1,\ \ldots,\ N_\mathrm{n}$) はそれぞれ $[1 \quad 2 \quad \cdots \quad N_\mathrm{c}]^T$ の要素をランダムに並べ替えて生成したベクトル，$\boldsymbol{r}$ は $[1 \quad 2 \quad \cdots \quad N_\mathrm{n}]^T$ の要素をランダムに並べ替えて生成したベクトルである．

3.2 ノードの座標

ノード j ($= 1,\ 2,\ \ldots,\ N_\mathrm{n}$) のカテゴリ空間における座標 $\boldsymbol{x}_j$ は以下の式により計算する：

$$\boldsymbol{x}_j^m = \sum_{k \in F(j)} \frac{\boldsymbol{f}_k^m}{|F(j)|}. \qquad (m = 1,\ 2,\ \ldots,\ N_\mathrm{c})$$

ただし，$F(j)$ はノード j が持っているファイルの集合である．

3.3 ヒット率の計算方法

本稿では，以下の手順により自己組織化前後での検索ヒット率を計算し，自己組織化の有効性を検証する．

1. **保有ファイルの設定**
   まず，前述の方法によりコンテンツベクトルを各ファイルに，興味ベクトルを各ノードにそれぞれ設定する．その後，ファイルを N_p 個ずつ各ノードに配置する．配置するファイルは前述の検索頻度 $p_i(\boldsymbol{f}_i \cdot \boldsymbol{n}_j)^{e_\mathrm{r}}$ に従って決定する．このとき，同じノードが同じファイルを選択した場合は選び直すが，異なるノードが同じファイルを保有することは許すものとする．
2. **初期ネットワークの構成**
   N_n 個のノードから 2 ノードをランダムに選択して，双方が接続数の上限 N_a に達していなければ接続するという処理を，接続可能なノードのペアが存在しない状態になるまで繰り返す．このような方法で構成したランダムネットワークを初期のネットワークとする．

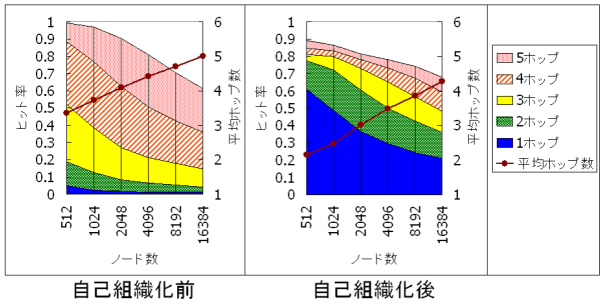

図 1: 自己組織化前後でのヒット率および平均ホップ数の比較

3. **初期ネットワークにおけるヒット率の計算**
   (i, j) の組を，$p_i(\boldsymbol{f}_i \cdot \boldsymbol{n}_j)^{e_\mathrm{r}}$ の値に比例した確率で選ぶことにより，検索を行うノード j と，そのノードが検索するファイル i を決定する．そして，ノード j がファイル i の検索を目的とした検索クエリを TTL τ_s でフラッディングし，その検索クエリが届いたノードのうちいずれかがファイル i を保有していれば，検索成功とする．この検索を規定回数 (N_s) 行い，検索成功回数を全検索回数 N_s で割ったものをヒット率とする．
4. **自己組織化の実行**
   2 章で述べた手順により，自己組織化を規定回数（各ノード N_r 回ずつ）実行する．なお，ノードの参加や離脱は起こらないものとする．
5. **自己組織化後ヒット率の計算**
   再度ヒット率を計算し，自己組織化前と比較する．

3.4 パラメータ設定

シミュレーションに用いたパラメータは以下の通りである：
$N_\mathrm{f} = 128000$, $N_\mathrm{n} = 6400$, $N_\mathrm{c} = 64$, $\alpha_\mathrm{c} = 1$, $\alpha_\mathrm{p} = 1$, $e_\mathrm{r} = 16$, $\beta_\mathrm{d} = 1000$, $\tau_\mathrm{s} = 3$, $\tau_\mathrm{a} = 5$, $N_\mathrm{p} = 100$, $N_\mathrm{a} = 5$, $N_\mathrm{r} = 20$, $N_\mathrm{s} = 1000$.

3.5 シミュレーションの結果

シミュレーションにより自己組織化前後の検索ヒット率および平均ホップ数を計算した結果を図 1 に示す（8 回シミュレーションを行った平均）．ここで平均ホップ数とは，TTL を設定せずに検索クエリをフラッディングしたと仮定した場合に，ファイルが見つかるまでに要するホップ数の平均のことである．この図から，自己組織化によって，少ないホップ数でファイルを発見できる可能性が高くなることが読みとれる．

4 まとめ

本稿では，P2P 型の情報流通ネットワークにおいて，各ノードが保有するコンテンツの類似性に基づいてネットワークの論理的な接続トポロジを変更することにより，検索効率を向上させる手法を提案した．

今後の課題としては，満足度の計算に用いている距離影響係数などの値を各ノードが自動的に調整する手法を検討したいと考えている．

参考文献

[1] Berry, M. W., Dumais, S. T. and Letsche, T. A.: Computational Methods for Intelligent Information Access, *Proceedings of ACM/IEEE Supercomputing '95* (1995).

[2] Nakatsuji, M., Kawahara, M. and Kawano, H.: The Architecture and Performance of Topic-Driven P2P Resource Discovery System, *IEICE Transactions on Communications*, Vol. J87-D1, No. 2, pp. 126–136 (2004).