社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－DSM－40 (7)
2006／3／29

PeerSommelier: 分散ハッシュテーブルを用いたスケーラブルな情報推薦フレームワーク

片山博順† 村岡洋一†

近年ユーザが接するコンテンツ数は膨大になり，その中からユーザが自分の嗜好に応じたコンテンツを効率的に選出することが困難になっている．この状況を解決するアプローチとして，協調フィルタリングなどを利用した情報推薦システムが提案されている．しかし従来手法では対象とするユーザ数とコンテンツ数の増加に伴い推薦サーバに負荷が集中する問題と，データが疎であることによって精度が低下する問題が存在した．本提案手法 PeerSommelier では P2P モデルを利用することで効率的に処理を分散させ，スケーラビリティの問題を解決した．また，ユーザプロファイルが分散した状況で高い推薦精度を実現するために，コンテンツごとの推薦処理を分散ハッシュテーブルのルートノードに割り当て，推薦処理に十分な評価情報を集積させた．本提案手法により，高い推薦精度（MAE 値 0.82～0.85）を保ちながらも優れたスケーラビリティを持つ情報推薦システムを実現した．

PeerSommelier:A Scalable Item-based Recommendation System with DHT

Hiroyuki Katayama† and Yoichi Muraoka†

In this paper, we propose a decentralized Item-based Collaborative Filtering algorithm, called PeerSommelier, based on distributed hash table method(DHT). The traditional CF argorithm, based on C/S model, is one of the most successful techniques in recommendation system. However, traditional CF based on C/S model has scalability problem. We achieved to realize a scalable Item-based CF algorithm using *Peer-to-Peer* model. And we also achieved to improve the prediction accuracy, using effectively *RootNode*.

1. はじめに

近年の広帯域インターネットの普及により，インターネットを経由した映画や音楽などのコンテンツ配信サービスが行われている．ユーザがコンテンツを取得する形態は，クライアントサーバモデルによるストリーミング配信や，Peer-to-Peer ファイル共有型のアプリケーションによるコンテンツの取得，あるいは Apple 社の iTunes のような専用アプリケーションでの動画・音楽購入まで様々である．コンテンツ配信技術の発展および，コンテンツ配信サービスの多様化に伴い，ユーザが取得可能なコンテンツは膨大な量になっている．今や，これらの膨大なコンテンツの中からユーザの嗜好に応じてコンテンツを効率的に選出する技術が必要である．この状況を解決する一つのアプローチとして，いくつかの情報推薦システム (*RecommendationSystem*) が提案されている．情報推薦システムの具体的なサービス例には Amazon.com や CDnow.com などがある．これらのシステムでは，ユーザの商品購入履歴からユーザの嗜好性を抽出し，関連する商品を推薦することで効率的な商品選択の提供と，販売機会の拡大をもたらしている．前述の情報推薦システムの基本となっている技術が協調フィルタリング (*collaborativefiltering*)[4] である．協調フィルタリングはビジネス上も学術的研究上も大きな成功を生んでいると言えるが，対象とするユーザ数とコンテンツ数の増加に伴い，スループットが低下するスケーラビリティ (Scalability) の問題と，スパーシティ (Sparsity) による精度の低下の問題が指摘されている[5]．本研究は次の 2 点の実現を目的とする．

- **目的 1**
  コンテンツとユーザの増加に耐性のある，スケーラビリティの高い情報推薦システムを実現する
- **目的 2**
  推薦精度の高い情報推薦を可能にする

とりわけ既存研究で解決できていない前者のスケーラビリティの問題解決に重点を置く．また，従来の協調

† 早稲田大学大学院理工学研究科
Waseda University

フィルタリングの長所である優れた推薦精度を保持し，のスケーラブルでありながら既存の情報推薦システムと同等あるいはそれ以上の推薦精度を実現することが目的である．

2. 既存の情報推薦手法と問題点

2.1 協調フィルタリングの特徴

協調フィルタリングは各ユーザのアイテムへの評価履歴をプロファイルとして利用し，類似した評価履歴を持つ他のユーザを選抜し，情報推薦を受けるユーザの未評価アイテムを推薦する情報推薦手法である．協調フィルタリング技術はオンラインショッピングでの商品推薦などにも用いられ，実用面でも広く成功をおさめている．

2.2 協調フィルタリングの課題

しかし，近年のユーザ数と対象とする情報（アイテム数）の増加により以下のような問題を抱えていることが指摘されている

- **スパーシティ(Sparsity)**
  Sparsity とはアイテム総数に対してユーザが既に評価を行ったアイテム数がしめる割合が非常に低い希薄な状態を指す．Amazon. com や CDnow. com などでは扱う書籍や音楽アルバムなどの膨大なアイテム全体に対して，わずか 1%以下しか購入されない．従って Nearest Neighbor アルゴリズムでは特定のユーザに対して適切なアイテムが推薦できないなどの精度の低下を及ぼしている．
- **スケーラビリティ(Scalability)**
  ユーザの増加と対象とするアイテムの増加の両方に応じて，その計算量が増加する．100 万以上のユーザやアイテムを扱うような Web ベースの情報推薦システムは深刻なスケーラビリティの問題を抱えていると指摘されている．

2.3 DHT を活用したフィルタリング：PipeCF

PipeCF ではこうした Scalability の問題を Peer-to-Peer のコンテンツ発見技術である分散ハッシュテーブル (以下，DHT) を用いて解決を試みている．PipeCF では Nearest neigbor の演算を既存の中央集権ではなく，を分散環境で行う．推薦情報を受けるアクティブユーザは，既に評価したアイテムの ID と評価値のタプル $\langle ITEM_ID, VOTE \rangle$ をローカルに持っている．$\langle ITEM_ID, VOTE \rangle$ をキーに DHT にかけ，同じ $\langle ITEM_ID, VOTE \rangle$ の組を持つユーザを発見し，それらのユーザから評価履歴情報を取得し，フィルタリングをかけている．これにより，あるアイテムに対し同じ評価をおこなったユーザのプロファイルから推薦情報を得ることができる．

2.4 PipeCF のプロファイル収集手法

図 1 に PipeCF のプロファイル収集手法を示す．図 1 では DHT の ID 空間を Chord や Pastry のように ring 状で示している．他の DHT の ID 空間には ring 状以外のものもあるが，ここでは解説を容易にするために ring 状の ID 空間を用いる．

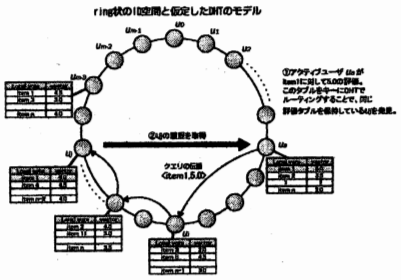


図 1 PipeCF におけるプロファイル収集法

ring 状の ID 空間（サイズは m とする）に配置されている $\{u_0, u_1, \cdots u_i, \cdots u_{m-2}, u_{m-1}\}$ は各ユーザのノードを表している．各ノードにひもづいて表現されているテーブルは各ユーザの評価履歴である．ここではアクティブユーザ u_a が，自身の評価履歴にある $item_1$ に対して評価値 5.0 を付けている neighbor ノードを ID 空間から探す場合を考える．図 1 にあるように u_a は，タプル $< item_1, 5.0 >$ を key として DHT にかけて，一致するタプルを保持しているノードを発見する．DHT の発見アルゴリズムに従い，同じハッシュ値を持つノードが発見される．発見したノードからアクティブユーザは評価履歴を受け取り，推薦情報計算に用いる．

2.5 PipeCF の課題

ここでは PipeCF が $\langle itemID, vote \rangle$ の評価タプルをハッシュにかけて利用することによる生じる解決すべき課題について述べる．一般にハッシュ機能は key に対して一意に value を割り当てる機能である．DHT は文字通りハッシュ機能であるので，key が完全に一致する場合でないと異なるハッシュ値を返す．図 2 のような評価履歴の場合について考えてみたい．評価値は 5 段階評価（1，2，3，4，5）であると仮定する．$user_1$ と $user_i$ はともに $item_1$ に対して比較的低い評価値（1，2）を与え，$item_3$，$item_5$ には比較的高い評価値（4，5），$item_4$ に対しては中程度の評価値（3，4）を与えている．前述の PipeCF のように評価値をハッシュにかけてプロファイル取得をする場合には，

このように類似した評価履歴を持つユーザが存在しても，評価値が異なるためプロファイル取得の対象とならない．これは予測評価値を求める場合に情報が十分に集まらず，精度が上がらないことにつながる． 各

近い評価傾向があるが評価値が一致しない場合DHTではヒットしない

	Item1	Item2	Item3		Item n-1	Item n
user 1	1.5		4.5		2.0	4.5
user 2		3.0	5.0			4.0
user 3	1.0	2.0	4.0		2.0	3.0
↑						
user i	2.0	2.5	5.0		3.5	5.0
↓						
user m-1	4.0	4.0			4.0	5.0
user m	2.0	2.5	3.0		3.0	

図 2 アイテム ID と評価値が完全に一致しない場合

ユーザがアイテムを評価する際には，ユーザごとに評価値を与えるスケールが異なる（比較的高い評価に偏るユーザ，低い評価に偏るユーザ，中程度に集中するユーザ）がいるため，それぞれの評価値そのものだけでなくユーザの評価スケールも考慮する必要がある．本研究では DHT をコンテンツの評価値に対してではなく，コンテンツ ID に対して用いる．また，PipeCF では user-based 協調フィルタリングを P2P で実現しているが，本研究では item-based 協調フィルタリングを P2P で実現することで，これらの情報の欠損を防ぐ．次に本研究の提案手法である PeerSommelier（ピア・ソムリエ）の概要について述べる．

3. PeerSommelier による情報推薦

本節では PeerSommelier の概要について述べる．PeerSommelier という名は，ユーザの嗜好性に応じて魅力的なアイテムを推薦する Sommelier（ソムリエ）のような P2P システムという意味を持つ．PeerSommelier では DHT を用いることでスケーラビリティ問題の解決を試みながら，精度の高い情報推薦を実現する．ユーザプロファイルの解析には，アイテム間の嗜好的類似度を求める Item-based 情報推薦を採用する．Chord や Pastry などの DHT ではあるコンテンツ ID に対応するノードが一意に定まる特徴がある．このノードをルートノード（Root Node）と呼ぶ．本手法では，アイテムの評価を行なうユーザの評価履歴が，その評価対象のアイテムのルートノードへ集めることで効率的なプロファイル収集を実現している．これにより，ルートノードには担当するアイテムに関する評価情報および，その周辺コンテンツに関する評価情報が得られるため，アイテム間の類似度計算および，アクティブユーザに対する予測評価値計算に十分な情報を取得することが可能になる．アクティブユーザは情報推薦のクエリをこのルートノードに送信することで推薦アイテムを取得する．次節では，より詳細にユーザプロファイル収集手法の流れを述べる．

3.1 PeerSommelier のプロファイル収集

本節では具体例を用いてアイテムのルートノードにユーザプロファイルが集まる仕組みを述べる．図 3，図 4 はルートノードにユーザプロファイルが集まる一連の流れを示している．図 3 のように，あるアイテム k に評価値を与えるユーザ u_i は，アイテム k を key として DHT にかけて，得られたルートノード u_r に評価値および自身の評価履歴を送信する．ルートノード u_r はこれを受け取り，保存領域に u_i の評価値および評価履歴保存する．

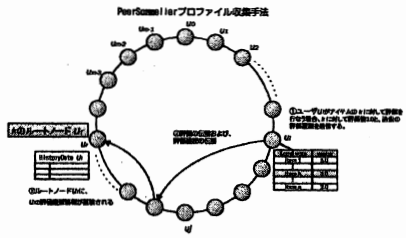


図 3 PeerSommelier におけるプロファイル収集法 その 1

次いで，図 4 のようにユーザ u_i とは異なるユーザ u_j がアイテム k を評価した場合も，同様にアイテム k のルートノードに評価履歴を伝播させ，ルートノードはこれを保存する． 以上の流れを繰り返すことで，

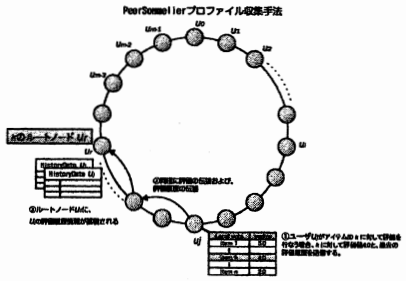


図 4 PeerSommelier におけるプロファイル収集法 その 2

アイテム k のルートノードにはアイテム k を評価したユーザの評価履歴が蓄積される．これまでの流れをまとめると以下のようになる．

(1) u_a は評価値を与えるコンテンツ ID をハッシュにかけてハッシュ値を得る

(2) 得られたハッシュ値をあて先として，コンテンツに対する評価を含んだ u_a の評価履歴を転送する

(3) DHTのルーティングアルゴリズムによって，評価履歴はコンテンツのルートノード u_r に向かって転送される

(4) u_r は u_a の評価履歴を受信し，ローカルに保存する

(5) 1から4を複数のユーザが繰り返すことで，u_r には評価履歴が蓄積する

3.2 ルートノードに集まる情報の特徴

図5は前節の例で用いたアイテム k のルートノードに集積する評価履歴の特徴を示している．アイテム k に評価値を与える際に評価者の評価履歴を受信するため，評価履歴中には必ずアイテム k の評価値が含まれている．ルートノードではアイテム k と類似度の

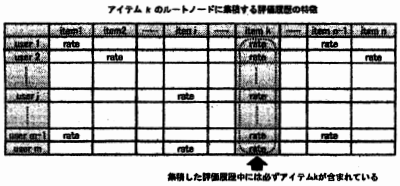


図5 ルートノードに集積する情報の特徴

高いアイテムを類似計算によって求め，求められたアイテム群からアクティブユーザに情報推薦する．アイテム i とアイテム j の類似度 $sim(i,j)$ の計算にはアイテム i とアイテム j の両方に対して評価を行なっている評価履歴が必要となる．よって，アイテム k との類似度を計算する際に，必ずアイテム k の評価が含まれている事は，類似度計算の高速化に大きな意味を持つ．前節で述べたプロファイル収集方法を用いると，次第にルートノードへ蓄積する評価履歴が増加して各ルートノードが保持する評価履歴テーブルサイズが巨大になる．そこで，本提案手法ではルートノードが保持する評価履歴テーブルのアイテム空間およびノード空間に制限を加える．

3.3 類似度計算

アイテム空間のサイズを m とすると，ユーザの評価履歴情報は m 次元ベクトルで表現できる．アイテム i に対する各ユーザの評価を m 次元のベクトル $\vec{i}$ と，アイテム j に対するベクトルを $\vec{j}$ としたとき，コサイン法を用いた2つのアイテムの類似度は次式で与えられる．

$$sim(i,j) = cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2} \qquad (1)$$

これは2つのベクトルの内積値をそれぞれのベクトルのスカラー値の積で割ることで，2つのベクトル間のコサイン値を求めるものである．最もシンプルで計算コストの低い類似度計算手法である．本研究ではコサイン法の他に相関係数法を用いた．

3.4 予測評価値計算

前節で導いたアイテム間の類似度 sim とユーザの評価履歴を用いて，未評価アイテムの評価値を予測する．未評価アイテム i への予測計算は次式で求める．

$$P_{u,i} = \frac{\sum_{allsimilaritems,N}(s_{i,N} * R_{u,N})}{\sum_{allsimilaritems,N}(|s_{i,N}|)} \qquad (2)$$

ここで，$P_{u,i}$ はユーザ u のアイテム i に対する予測評価値を表し，N は類似度計算でアイテム i と類似度が高いと判断されたアイテムである．$s_{i,N}$ は前節で述べた類似度計算式を用いて求められるアイテム i とアイテム N の類似度を示す．2式の分子はアイテム i と類似度が強いアイテムへの評価は予測値への影響度が高く，逆に類似度が低いアイテムへの評価は小さく反映されることを意味する．

3.5 推薦アイテムの選択

ルートノードは情報推薦の要求をしてきたアクティブユーザの未評価アイテムに対して，保持している評価履歴のコンテンツ ID 空間で2式の予測評価値計算を行い，最も予測評価値が高かったアイテムを推薦アイテムとする．

4. 評　　価

本研究の目的を達成するするため，本提案手法の評価項目は以下の2点である．

- スケーラビリティがあるか
- 推薦精度は十分な高さを持っているか

4.1 評価の方法

推薦精度を評価するために，実際に映画やコンテンツに対するユーザの評価データが必要となる．そこで本研究の評価に MovieLens[4] が提供しているデータセットを用いた．本データセットを用いた推薦精度評価は協調フィルタリングに関する研究[5] や PipeCF[3] などでも行なわれており，他の研究成果と比較する上でも有用であるといえる．MovieLens のデータセットの特徴について簡単に述べる．

- 全943ユーザによる1682タイトルの映画に対する5段階評価（rating）が100,000件収録されている
- それぞれのユーザは1人あたり20件以上の評価を行なっている
- それぞれのユーザの属性（年齢，性別，職業）な

ども収録されている

推薦精度の評価実験には，投入する全 rating データのうち 80%を training データとして投入し，残りの 20%の rating データに対する予測評価値を求める．本評価実験では，全 100,000 件の rating データのうちランダムに選ばれた 80,000 件の training データとして投入し，残りの 20,000 件を test データとして予測評価実験に用いた．training データと test データの分類には，MovieLens のデータセットが提供しているランダムな分類のデータを用いた．

4.2 評価の尺度

推薦精度の評価尺度としては平均絶対誤差 MAE(Mean Absolute Error) を用いる．MAE は，予測評価値と実測評価値の差の絶対値を用いて，予測評価値と実測評価値の近さを示す．以下に，MAE の計算式を示す．

$$MAE = \frac{\sum_{t \in T} | p_t - q_t |}{T} \quad (3)$$

ここで，T は training データの集合を示し，p_t，q_t はそれぞれ training データ t に含まれるユーザ u のアイテム i に対する予測評価値と実測評価値を示す．MAE は値が小さいほどその推薦精度が高いことを示す．

5. 実　　験

オーバーレイネットワークに参加するノードを次のように 2 種類に分けた．training 用の rating データの投入および予測評価値クエリの送信を行なうノードを投入ノードとした．投入ノードは 1 台のマシン上に起動させ，その他のノードは 1 台のマシンに 10〜120 ノード起動させた．投入ノードは training データを **rate** メソッドを用いてオーバーレイネット上に転送する．転送された評価情報は前述のルールに従ってそのコンテンツ ID のルートノードへ転送される．全 80,000 件の評価情報が投入ノード以外の Sommelier ノードに蓄積した状態で，投入ノードは残りの 20,000 件の test データに関する予測評価値クエリを **getPrediction** メソッドを用いて転送する．予測評価値クエリを受信したノードはそのノードが保持する評価履歴情報を元に，コンテンツ間の類似度計算を行い，予測評価値を求める．実験項目は次の 2 通りである．

実験 1 オーバーレイネットワークに参加するノードを 10〜120 ノード（10 ノード間隔）とし，80%の rating データを投入した後，20%の rating データの予測評価値を求め，予測評価値計算に要した時間を測定する．

実験 2

オーバーレイネットワークに参加するノードを 10〜120 ノード（10 ノード間隔）とし，80%の rating データを投入した後，20%の rating データの予測評価値を求め MAE 値を測定する．

5.1 実験結果

本節では，評価実験の結果について述べる．図 6 は実験 1 の結果を示す．

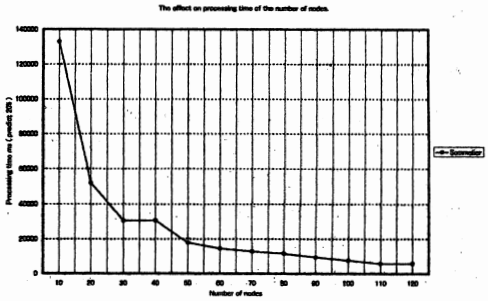


図 6 推薦処理ノード数と予測評価値計算時間（ユーザ数 943, コンテンツ数 100）コンテンツ間類似度の事前計算なし

図 6 の縦軸は予測評価値の計算に要した時間（ミリ秒），横軸はオーバーレイネットワークを構成するノード数である．予測評価値の計算には，予測評価値計算を要求されたルートノードで事前にコンテンツ間の類似度 $sim(i,j)$ を計算する必要がある．図 6 の縦軸で示した時間は，コンテンツ間の類似度計算と，予測評価値の計算に要した時間を足した時間を表している．図 6 はオーバーレイネットワークを構成するノード数の増加によって予測評価値の計算時間がどのように変化をするかを示している．図 6 から明らかなように，ノード数が増加するにつれて予測評価値計算に要する時間が大幅に減少するのが分かる．

次に図 7 では，実験 2 の結果を示す．

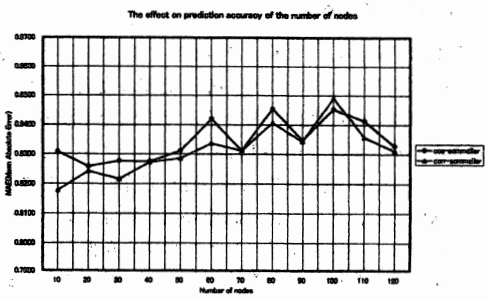


図 7 推薦処理ノード数と MAE（ユーザ数 943, コンテンツ数 100）

図7の縦軸は20%のratingデータで予測評価値を求めた場合のMAE値，横軸はオーバーレイネットワークを構成するノード数である．cos-sommelierは類似度計算にコサイン法を用いた場合，corr-sommelierは類似度計算に相関係数法を用いた場合である．図7より，ノード数が10〜120に増加する間にMAE値はほぼ横ばいで0.82〜0.84の間を推移していることが分かる．

6. 考　　察

グラフから明らかなように，推薦処理ノード数が増加すると1ノードあたりの処理は大きく減少している．これは，ノード数が増加するにつれ，1ノードあたりが対応するクエリの量が減少するからだと考えられる．P2Pモデルを採用することにより，1つの処理マシンへ負荷が集中することが回避されている．図8のようにルートノードに集まる情報量が減ってはいるが，図7が示すように推薦精度はほぼ一定である．これは，3.2節で述べたようにルートノードに情報推薦に必要な評価履歴が効率的に集積するからだと考えられる．

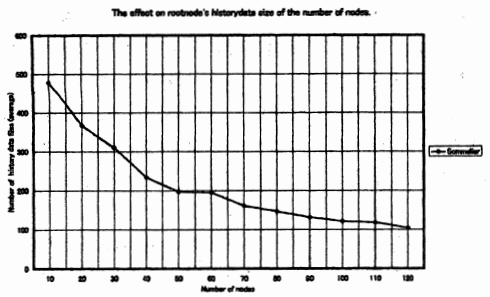


図8　1ノードが保持していた平均ユーザプロファイル数

7. ま と め

図7の結果から，本手法でのMAE値がおよそ0.82〜0.84の間にあることが分かる．既存の協調フィルタリングでは多少の条件が異なるので正確な比較検討はできないが，同じMovieLensのデータセットを用いた場合にMAE値が0.75から0.85で推移してることを加味すると，十分それと同等の推薦精度であるといえる．また，本実験ではコンテンツ間の類似度計算にコサイン法と相関係数法を採用しているが，さらに拡張コサイン法などを用いることでこの結果を超える成果をあげることが期待できる．以上の結果より次のことが言える．

- 実験1の結果，図6と図??より十分なスケーラビリティをもった情報推薦システムが実現できた
- 実験2の結果，図7より十分な高い精度をもった情報推薦システムが実現できた

今後は数千，数万ノードでの評価実験を行い本手法の有効性を確認し、実サービスへの利用を検討したい．

参 考 文 献

1) Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", In the Proceedings of ACM SIGCOMM 2001, San Deigo, CA, August (2001).
2) A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November, 2001.
3) Xie Bo, Han Peng, Yang Fan, Shen Ruimin, "PipeCF:A Scalable DHT-based Collaborative Filtering Recommendation System", WWW 2004, May 17-22,2004,New York(2004)
4) Eachmovie collaborative filtering data set,1997.
5) Badrul Sarwar,George Karypis, Joseph Konstan, John Riedl,"Item-based Collaborative Filtering Recommendation Algorithms", Proceedings of the WWW10 Conference,May 1-5 2001