

E-13

クラスタ生成に基づく電子メール文書の重要度ランク付け手法 A Ranking Method for E-mail Documents based on Creating Clusters

小川 恭幸†
Yasuyuki OGAWA

石田 崇†
Takashi ISHIDA

後藤 正幸‡
Masayuki GOTO

平澤 茂一†
Shigeichi HIRASAWA

1. はじめに

近年のインターネットの普及にともない、電子メールがコミュニケーションの1つの手段として確立されてきた。電子メールによって情報の伝達が可能になった反面、各個人向けの計算機にも大量のメール文書が送信され、重要なメールと商用メールに代表される重要性が低いメールとを自動分類する処理、すなわち情報フィルタリング技術の重要性が高まっている[1]。

従来のフィルタリング手法[2]は、各個人があらかじめ優先度付けした既存のメール文書からメールの重要性を決定する属性を取り出し、それを基に新規メールの重要度を算出する。しかし、従来法では、新規メールの属性値と同一のものが、既存のメール中に存在しない場合に、正確に重要度を算出することができない。

本研究では、各属性に距離概念を導入し、凝集法によるクラスタリングを適用することにより、新規メールの属性値が既存のメールの属性値と一致しない場合でも、重要度の推定精度を向上させる手法を提案する。また、シミュレーションにより、フィルタリング性能の評価実験を行ない有効性を示す。

2. 従来の電子メールのフィルタリング手法[2]

2.1 フィルタリング手法の概念

「他のメール文書よりも早く読む必要がある」メール文書を重要性が高いと定義する。学習部では、各ユーザによって優先度(ユーザが前もって付けた重要性)が付与された既存の受信済み文書から各属性を抽出し、そこからプロファイルを作成する。解析部では、そのプロファイルを用いて新規メールの重要度を算出する。

2.2 学習部

2.2.1 重要度の要因(属性)

メールから得られる情報で、フィルタリングに有効と思われる属性として以下の4種類を取り上げる。

- ①送信元 α ; (例) 小川, 佐藤
(メール文書内ヘッダーFromの項目から取得)
- ②文の種類 β ; (例) 依頼, 勧告, 義務
(メール文書の本文内に含まれる叙述表現[3]から取得)
- ③時間的制限 γ ; (例) 3日以内, 1週間以内
(メール文書の本文内に含まれる時間表現[4]を表わす副詞句から取得)
- ④テーマ θ ; (例) 会議, ゼミ, 試験
(既存のメールに対して、ユーザが与える。新規のメールに対しては、テーマを特徴づける名詞類を登録した辞書を用いる)

2.2.2 プロファイルの作成

学習用データを各属性値、優先度 κ_i から $x_i = (\alpha_i, \beta_i, \gamma_i, \theta_i, \kappa_i)$ と表現し、学習用データ集合 $D = \{x_1, x_2, \dots, x_i, \dots, x_n\}$ を作成する。D をプロファイルと呼ぶ。

$\text{freq}(p < \alpha, \beta, \gamma, \theta, \kappa >)$: プロファイル内の多重組 $(\alpha, \beta, \gamma, \theta, \kappa)$ の出現頻度

$\text{freq}(p < \alpha, \beta, \gamma, \theta >)$: プロファイル内の多重組 $(\alpha, \beta, \gamma, \theta)$ の出現頻度

ただし, $\text{freq}(p < \alpha, \beta, \gamma, \theta >) = \sum_{\kappa} \text{freq}(p < \alpha, \beta, \gamma, \theta, \kappa >)$

2.3 解析部

2.3.1 重要度推定アルゴリズム

新規メールの属性ベクトルを $y_j = (\alpha_j, \beta_j, \gamma_j, \theta_j)$ とし $T = \{y_1, y_2, \dots, y_j, \dots, y_n\}$ を新規メール集合とする。新規メールの重要度 R_j の算出アルゴリズムを以下に示す。

Step1 y_j と同一のプロファイルデータがあるか否かを判定。あ

ればStep2へ、なければStep3へ。

Step2 通常処理により y_j の重要度を算出。(2.3.2節参照)

Step3 近似処理により y_j の重要度を推定。(2.3.3節参照)

2.3.2 通常処理

y_j と同じ属性値をもつプロファイルデータの、優先度の期待値を求め、その値を y_j の重要度 R_j とする。

$$R_j = \sum_{\kappa} \kappa \times P(\kappa | \alpha_j, \beta_j, \gamma_j, \theta_j) \quad (1)$$

ただし,

$$P(\kappa | \alpha_j, \beta_j, \gamma_j, \theta_j) = \frac{\text{freq}(p < \alpha_j, \beta_j, \gamma_j, \theta_j, \kappa >)}{\text{freq}(p < \alpha_j, \beta_j, \gamma_j, \theta_j >)} \quad (2)$$

2.3.3 近似処理

y_j と同じプロファイルデータが存在しない場合は、プロファイル内に存在する他の多重組に置き換えて重要度を近似的に計算する。置換によるノイズ混入量の違いを考慮し、属性の置き換えを次のように行う。ただし、「文の種類(β)」については、言語学的に各属性値間に類似性が少ないことから、近似の対象外とする。

①「時間的制限」に対する近似処理

$$R_j = Q_j + (\gamma_j - \gamma_{\text{ave}}) \times \Delta\kappa / \Delta\gamma \quad (3)$$

γ_{ave} : 基準重要度の計算で用いたプロファイル内の多重組に含まれる時間属性値の平均

$\Delta\kappa / \Delta\gamma$: 時間属性値の変化に対する優先度の変化率

ただし,

$$\text{基準重要度 } Q_j = \sum_{\kappa} \sum_{\gamma} \kappa \times P(\kappa | \alpha_j, \beta_j, \gamma, \theta_j) \quad (4)$$

②「送信元」に対する近似処理

①で近似処理できなかった場合、「送信元」に対する近似処理を行う。同一の文の「類型」「時間的制限」「テーマ」から成る多重組に、値が近い優先度が付与されている送信元ほど、類似していると判断する。そして、多重組 $y_j = (\alpha_j, \beta_j, \gamma_j, \theta_j)$ に対して、最も類似している送信元 α_{rep} に置換し、重要度を算出する。

③「テーマ」に対する近似処理

②で近似処理できなかった場合、「テーマ」に対する近似処理を②と同様な方法で行う。

<近似処理①~③全てに失敗した場合>

$$R_j = \sum_i \kappa_i / B \quad (5)$$

2.4 従来手法の問題点

従来法では、新規メールの送信元がプロファイル内に存在しない場合、近似する事ができない。また、テーマが送信元に依存している場合が多く、そのような状況では各テーマ間にまたがって出現する送信元が少なくなるため、各テーマ間の類似性が測りにくくなり、多くの場合近似する事ができない。これにより、全てのプロファイルデータの優先度の平均値を重要度としてしまう為、信頼性に欠ける結果になるという問題がある。

3. 提案手法

各属性値間の類似性に基づき、各属性に距離概念を導入する。それを用いて、凝集法によるクラスタリングを行う手法を提案する。

[提案手法1]

各属性値をユーザが各属性値間の類似性に基づき数直線上に数値化する。

Step1 「送信元」の属性値に関しては、どのようなグループに属しているかというタグ(例: 友人, 学会)をユーザが与える。そして、タグ間の類似性に基づきユーザが数直線上に数値化を行う。ただし、数値間が近いものほど類似性

† 早稲田大学理工学部経営システム工学科

‡ 武蔵工業大学環境情報学部情報メディア学科

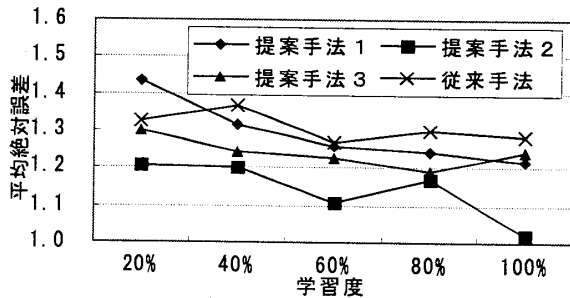


図1 学習度と平均絶対誤差の関係

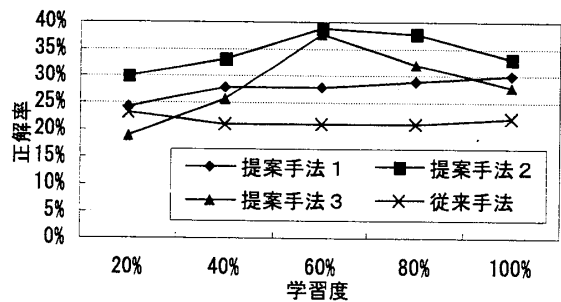


図2 学習度と正解率の関係

が高い。
また、他の属性に関しては属性値間の類似性に基づきユーザが数値化を行う。この値に基づいてプロフィール内の多重組 x_i に対し、ベクトル d_i を作成する。

[例] プロファイルデータのベクトル化
 $x_i = (\text{小川, 依頼, 3日以内, 大学内})$
 $\rightarrow d_i = (0.075, 0.8, 0.7, 0.1)$

Step2 ベクトル d_i をユークリッド距離の近いものから順に結合し、クラスタを生成する。
各クラスタについて、所属する全プロフィールデータの重心ベクトルを、そのクラスタの特徴ベクトルとする。
クラスタの重心ベクトル C は次式で与える。

$$C = (d_1 + d_2 + \dots + d_n) / n \quad (6)$$

n : クラスタ内のプロフィールデータ数
クラスタ数がユーザの与えた数に達するまでこの作業を行う。

Step3 Step1 の方法で新規メールの属性値を数値化する。そして最もユークリッド距離が近いクラスタを求め、そのクラスタに属しているプロフィールデータの優先度の平均値を新規メールの重要度とする。 □

【提案手法2】

各属性値間の類似性に基づき、各属性値間の距離データをマトリックス上に数値化する。

Step1 各属性値間の類似性に基づき、ユーザがマトリックス上に各属性値間の距離の数値化を行う。

Step2 群平均距離の近いものから順にプロフィールデータを結合し、クラスタを生成する。クラスタ数がユーザの与えた数になるまでこの作業を行う。

Step3 Step1 のマトリックスの知識を用いて、新規メールの属性値に対して最も群平均距離が近いクラスタを求め、そのクラスタに属しているプロフィールデータの優先度の平均値を新規メールの重要度とする。 □

【提案手法3】

提案手法1の各属性値の数値化をプロフィールを用いて行う。「送信元」の属性値に関しては、同一のタグからなるプロフィールデータに付与されている優先度の平均値を各タグの数値ベクトルとする。また、他の属性についても同様に行う。 □

4. 提案手法の評価

4.1 利用データ

シミュレーションには、既存のメールとしてユーザが予め優先度付けた200通を準備した。表1にその内容を示す。

表1 学習用データの内容

種類 (背景知識)	属性値数
送信元数	86 (タグ16種類)
文の種類	221
時間的制限	81
テーマ	279

そして、各テーマ内の文書数に比例して無作為に抽出した30通のメールを評価用データとし、残りの170通を学習用データとした。また、学習用、評価用データの優先度の平均、分散共に極

端に偏った分布にはなっていない。これらのデータを用いてプロフィールを作成した後、評価用データの重要度を求めた。なお、優先度、重要度ともに5段階評価(5が最高値)とした。

4.2 従来手法と提案手法の比較と考察

【評価1】 提案手法、従来手法で求めた重要度と、ユーザが予め付けた優先度との平均絶対誤差を求めた結果を図1に示す。ここで学習度は用意した学習用データに対し、学習に用いたデータ数の割合である。また、提案手法で用いるクラスタ数は学習に用いたデータ数の40%として、従来手法と比較した。

【評価2】 学習度と正解率の関係を図2に示す。ただし、小数点以下を四捨五入した重要度と、優先度が一致する割合を正解率とする。

【考察】 提案手法1~3は従来手法に比べ全体的に、平均絶対誤差が小さく、高い正解率を示した。これは、従来手法では近似処理①~③を行えなかった評価用データに対し、提案手法ではクラスタを求める事で、より正確に重要度を算出できた為であると考えられる。提案手法3が、提案手法1に比べ、全体的に平均絶対誤差が小さくなった原因としては、提案手法1ではユーザが属性値間の類似性(距離)を無理に数直線上に与えているのに対し、提案手法3ではプロフィールデータに付与されている優先度に基づいて、数直線上に数値化することで、より正確に重要度が算出できたためであると考えられる。また、学習データ数が少ない時(学習度20%)は、従来手法では近い学習用データをもとに重要度が算出されるが、提案手法では、クラスタの信頼性が低いにも関わらず、最も近いクラスタ内のプロフィールデータの優先度の平均値を取る為、評価用データから遠い位置にある学習用データの影響も受けてしまう為に提案手法が従来手法に比べ精度が変わらなかったと考えられる。さらに学習度を上げれば、提案手法が従来手法に比べより正確な重要度算出を行うことが期待できる。

5. おわりに

各属性値間の類似性に基づき、凝集法によるクラスタリングを適用することによって、より正確な重要度算出を行うことができた。

今後の課題として、プロフィールの規模を大きくしてシミュレーションを行い、マトリックスの中身の計算を行うためのアルゴリズムを考えていきたい。

謝辞

本研究を行うにあたり、数多くのご助言、ご協力をいただきました早稲田大学平澤研究室の各氏に心より感謝いたします。

【参考文献】

[1] 森田昌宏, 速水治夫: “情報フィルタリングシステム”, 情報処理 Vol. 37, No. 8, pp. 751-758, 1996.
[2] 獅々堀正幹, 藤井誠, 安藤一秋, 青江順一: “多属性項目の履歴情報に基づく電子メール文書のフィルタリング手法”, 情報処理学会論文誌 Vol. 41, No. 8, pp. 2299-2308, 2000.
[3] 首藤公昭: “文節構造モデルによる日本語の機械処理に関する研究”, 福岡大学研究所報, pp. 1-121, 1980.
[4] 溝沢昭二, 住友徹, 弘田正雄, 青江順一: “日本語時間表現の一般解法”, 情報処理学会論文誌 Vol. 40, No. 9, pp. 3408-3419, 1999.