機械学習を用いたWWWからの製品性能表の分類と抽出

林 晃司† 嶋田和孝‡ 遠藤 勉‡

† 九州工業大学大学院情報工学研究科情報科学専攻 ‡ 九州工業大学情報工学部知能情報工学科

1 はじめに

WWWの急速な普及に伴い，これまで紙面で伝えられていた情報が電子文書化され，カタログ等の製品情報もインターネットを介して入手可能になった．しかしながら，WWWには多種多様の情報が共存する為，必ずしも欲しい情報のみを正しく検索・収集出来る訳では無い．また，仮に情報を収集出来たとしても，その量が膨大であれば閲覧は困難を極める．WWW上の情報検索においては単なる情報収集技術のみでは無く，収集した情報をいかに効率良く利用出来るかについても考慮されるべきである．

この要求が高い意義を持つものの一種に「製品性能表」がある．製品性能表とは製品の仕様書を電子化し，テーブル形式で記述したものを指す．図1に例を示す．

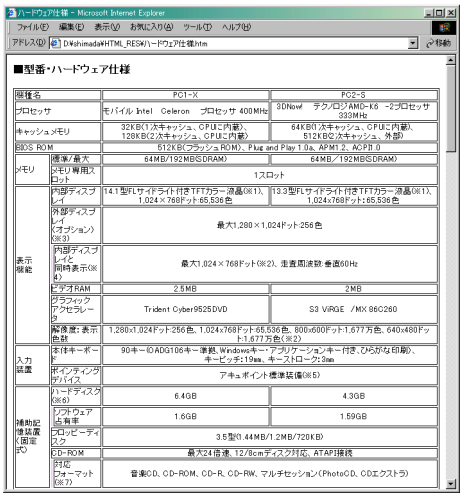

図 1: 製品性能表

製品性能表には，その製品に関する具体的なデータが掲載されている．しかし，データを閲覧しただけでどの項目がその製品の特徴となるかは一概には判断し難い．その理由としては，

1. 各サイトでは自社製品の特徴は述べられているが，他社製品との比較は余りなされていない．
2. 各サイトごとに様々な表現方法がある．
3. 要求と製品の特徴を関連づけるには，その製品に対してある程度知識が必要である．

等が挙げられる．ユーザの要求を満足するには，複数のサイトから情報を抽出し，統合する必要がある．また，各々の製品の相対的な特徴を正しく抽出出来たとしても，性能表を提示するのみではユーザにとって可読性は低く，効率の良い閲覧は臨めない．

これらを基盤に，我々は現在，複数のパソコン(PC)の製品性能表を解析し，各々のPC製品の特徴を抽出・比較する事によりユーザの要求に合致したPC製品選択を支援するシステムの構築を進めている[3]．

本システムの流れを説明する．製品メーカのウェブサイトより収集されたHTMLドキュメント群内に存在する製品性能表の抽出を行う．製品性能表はメーカ毎に表記が異なる為，表構造と呼ばれるデータ構造へと正規化を行う．得られた表構造中の数値データや文字データを比較し，各製品の相対的な特徴にユーザの意図を反映したスコアリングを行う．抽出された特徴データ及びスコアリングの結果を基に文章生成や表の再構築，グラフ生成等を行い，複数の形式を統合させた要約形式の製品情報をスコアの序列に準じてユーザへと提示する．図2は我々が開発した製品選択支援システムの概観である．

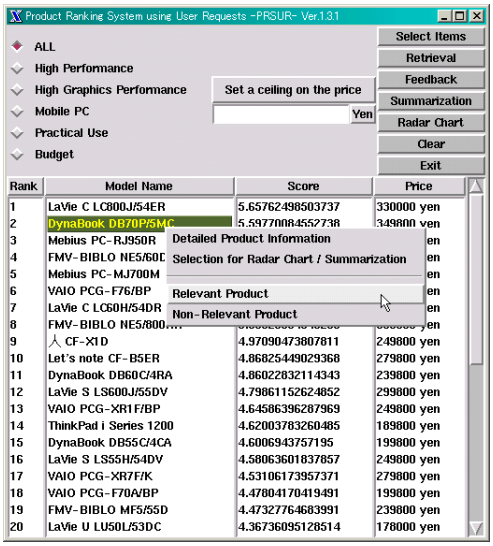

図 2: 製品選択支援システム

本稿で対象とするのはシステムの入力データである製品性能表の自動獲得である．図3に本研究の流れを示す．製品性能表の抽出は，収集した文書群に対し，製品性能表が含むと推測される候補文書の選出を行なうフィルタリング処理，及び候補文書に対して製品性能表の領域抽出を行なう表領域抽出理の2タスクに分割される．いずれの処理においても，文書内に性能表が含まれるか，もしくはテーブルが性能表かといった同定を要する．その手掛かりとなるキーワードは収集した文書群より学習し，これを参照しながら同定を行な

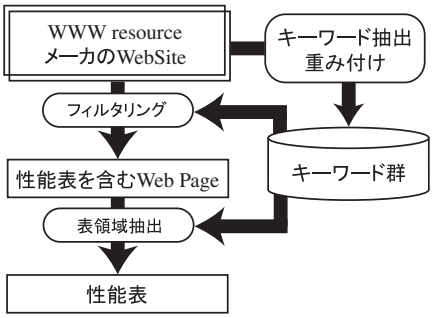

図 3: 製品性能表抽出

う．本稿はこのキーワード学習，フィルタリング処理，及び表領域抽出処理の手法とその成果について報告するものである．

2 素性選択

2.1 テーブルの氾濫

HTML ドキュメント (以下，文書) は HTML にて記述されており，性能表等のテーブル領域も同様，<TABLE>〜</TABLE>タグにて描画される．要するに，性能表の抽出とは<TABLE>タグでマークアップされた領域 (T wrapper) を抜き出す作業である．

T wrapper は HTML ソース内からの領域把握が容易であるが，一方で<TABLE>タグはレイアウト補正等にも多用される為，<TABLE>タグのみで T wrapper がテーブルであるか否かの判断は出来ない．また，文書 1 件あたり 2.35 個の T wrapper が存在し，実際にテーブルとして用いられているものは 3 割に満たないという報告もある [1]．この事から，文書群から正しく製品性能表を抽出するには，フィルタリングによって文書内の性能表の有無を推定しなければならない．単一ドキュメント内に複数存在するテーブルの中から性能表を同定する必要もある．

2.2 キーワード候補の抽出

本研究では性能表同定の手掛かりとして，テーブル内に生起する単語に着目した．中でも，性能表の項目名は非常に特徴的な情報となり得ると推測される．よって，性能表抽出に必要な特徴は「項目欄に位置する，テーブルの最左列」及び「一定長以内の文字列中」において顕著，または限定的に出現すると定義される．この定義に基づき，文書群キーワード候補の抽出を行う．候補抽出の手順を以下に示す．

1. 文書より T wrapper を抽出する．
2. T wrapper 中の各<TR> wrapper について，最初の<TD> wrapper の内容を抽出する．
3. タグ抜きの文字列長が一定長以内であれば形態素解析により名詞単語を抽出，これを候補とする．

尚，形態素解析には奈良先端科学技術大学で開発された「茶筌」[4] を用いた．

2.3 候補の重み付け

次に，得られたキーワード候補群に対して重みを計算し，その重みを基にキーワードを抽出する．重み付け手法としては，$normalized \cdot idf$ を用いる手法，及びベイズの定理を用いる手法の二種類を比較・評価する．

normalized $tf \cdot idf$

$tf \cdot idf$ は最も有名な重み付け手法の一つであり，文書群 $D = \{d_1, ..., d_N\}$ について，文書 d におけるキーワード候補 t の生起数 $tf(t,d)$，及び候補が生起する文書数 $df(t)$ を基に重み付けを行う．これを学習用に拡張し，$D = \{D_{real}, D_{no}\}$ とする．ここで，D_{real} は製品性能表を含む正解文書群，及び D_{no} は製品性能表を含まない，もしくは製品性能表以外のテーブルを含むノイズ文書群である．各々の文書群に生起する単語 t について，Wang ら [6] が表抽出で用いた以下の式にて重み付けを行う．

$$w_t^{real} = \sum_{d_i \in D_{real}} tf(t,d_i) \times \log(\frac{df_t^{real}}{|D_{real}|}\frac{|D_{no}|}{df_t^{no}} + 1)$$

$$w_t^{no} = \sum_{d_i \in D_{no}} tf(t,d_i) \times \log(\frac{df_t^{no}}{|D_{no}|}\frac{|D_{real}|}{df_t^{real}} + 1)$$

ここで，df_t^{real}, df_t^{no} は D_{real} 及び D_{no} における単語 t の df 値である．最終的な重みは以下の式にて求める．

$$ws_t^{real} = \frac{w_t^{real}}{Norm_{real}} \quad Norm_{real} = \sqrt{\sum_{t \in D_{real}} (w_t^{real})^2}$$

$$ws_t^{no} = \frac{w_t^{no}}{Norm_{no}} \quad Norm_{no} = \sqrt{\sum_{t \in D_{no}} (w_t^{no})^2}$$

正解文書内，ノイズ文書内にて高生起率となる語をそれぞれキーワード，及びノイズワードと呼ぶ．各々について，重みが閾値以上となる語を素性とする．

Bayes' rule

Bayes' rule はパターン認識・分類の分野にて広く知られる確率である．事象 $C = [C_i]_{i=1}^M$ において，$P(C_i)$ $(\sum_{i=1}^M P(C_i) = 1)$ は事前確率と呼ばれる．事前確率と条件付き確率密度分布 $p(t|C_i)$ （ $\int p(t|C_i)dt = 1$ ）が事前に得られる場合，単語 t が C_i に属する事後確率 $P(C_i|t)$ は次の式で求められる．

$$P(C_i|t) = \frac{P(C_i)p(t|C_i)}{\sum_{j=1}^M P(C_j)p(t|C_j)}$$

ここで，$C = \{D_{real}, D_{no}\}$ である．全単語に対して各クラスでの事後確率を求め，それらを単語の重みとする．即ち，$ws_t^{real} = P(D_{real}|t)$，及び $ws_t^{no} = P(D_{no}|t)$ である．$ws_t^{real} > 0.25$，かつ $df_{real}(t)/|D_{real}| > \frac{2}{10}$ を満たす語を正の素性に，$ws_t^{no} > 0.25$，かつ $df_{no}(t)/|D_{no}| > \frac{1}{10}$ を満たす語を負の素性とする．

3 フィルタリング

3.1 Transductive SVM

ここではフィルタリングを正解文書，及びノイズ文書を選別する二値分類問題と捉え，分類器による解決を試みる．一般に，高精度の分類器生成には多量の訓練サンプルを要するが，十分な量の訓練サンプルをラベリングするのは非常に高コストな作業となる．そこで，少量のサンプルで高精度の分類器を生成する手法が期待される．

Vapnik[5] の理論を基に Joachims[2] によって提案された Transductive SVM (TSVM) は，学習時にラベル無データの分布を考慮する事で分類精度を上げる手法である．以下に TSVM のアルゴリズムを示す．

1. 訓練サンプルのみで SVM により分類器を生成する．
2. 生成された分類器により全てのラベル無データを判別し，仮の分類クラスを与える．
3. 仮のクラスが付与されたラベル無データを訓練サンプルに含め，SVM による学習を行う．
4. マージン内に存在するラベル無データのうち，各々の仮クラスを入れ替える事でマージンを最大化出来るペアを見つけ入れ替え，再度 SVM による学習を行う．
5. 入れ替えるラベル無データのペアが無くなるまでクラスの入れ替えと学習を繰り返す．

TSVM の識別関数及び制約条件を下式に示す．

線形分離可能

$$\begin{aligned} \min \quad & \frac{1}{2}||\boldsymbol{w}||^2 \\ subject\ to: \quad & \forall_{i=1}^{n} : y_i[\boldsymbol{w}\boldsymbol{x}_i + b] \geq 1 \\ & \forall_{i=1}^{n} : y_i^*[\boldsymbol{w}\boldsymbol{x}_i^* + b] \geq 1 \end{aligned}$$

線形非分離 (ソフトマージン適用)

$$\begin{aligned} \min \quad & \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=0}^{n}\xi_i + C^*\sum_{j=0}^{k}\xi_i^* \\ subject\ to: \quad & \forall_{i=1}^{n} : y_i[\boldsymbol{w}\boldsymbol{x}_i + b] \geq 1 - \xi_i \\ & \forall_{j=1}^{k} : y_j^*[\boldsymbol{w}\boldsymbol{x}_j^* + b] \geq 1 - \xi_j^* \\ & \forall_{i=1}^{n} : \xi_i > 0 \\ & \forall_{j=1}^{k} : \xi_j^* > 0 \end{aligned}$$

ここで，$\boldsymbol{x}^*{}_j$ 及び y_j^* はそれぞれラベル無データにおける入力ベクトル及び仮クラスである．また，C^* 及び ξ_j^* は，ラベル無データにおける制約条件緩和パラメータ及びスラック変数である．

3.2 実験と考察

PC，デジカメ，プリンタ 3 種類の製品カテゴリについて，製品メーカのウェブサイトより収集した HTML 文書をテストセットとした．また，TSVM は訓練セットにおける正例，負例の分布を基にしてラベル無データへの仮クラス付与を行なう．そこで，訓練セットはテストセット数の 10%，及び 1%をテストセットから無作為にサンプリングした．データセットの内訳を表 1 に示す．実験は各カテゴリの訓練セットの正解文書，及びノイズ文書より素性選出及び分類器生成を行い，訓練セット数の減少に伴う分類精度の変化を観察した．また，同じ訓練セット数における SVM と TSVM の分類精度を比較した．分類器生成には SVM^{light} [3]を採用した．実験結果を表 2 及び表 3 に示す．太字部分は同じ訓練数において SVM または TSVM の精度が上回っている結果である．精度の評価には F 値 ($\alpha = 0.4$) を用いた．尚，P 及び R はそれぞれ適合率，及び再現率である．

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

結果について考察する．等しい訓練数においては，TSVM の適用により平均 38%，最大 60%の精度向上が確認された．また，一部のカテゴリにおいて，TSVM は SVM の 1/10 の訓練数にて SVM 同等の精度を達成し，他カテゴリについても訓練数の減少に伴う精度低下を平均 5%にまで回避出来た．これらの結果から，フィルタリングにおける TSVM の有用性が実証された．

一方で，幾つかの課題点も存在する．サンプル数の増加に伴い，訓練セットの分布はテストセットの分布へと接近するが，収集したサンプルへのラベリングにかかるコストもそれに準じて増大する．しかし，サンプルが少なければ分布の信頼性は薄い．このトレードオフの解決にはテストセットの確率分布の予測が必要であるが，未知の母集団からの確率分布推定はパラメトリックな分布の仮定，もしくは事前確率の均一化といった様な経験則に頼らざるを得ない．

また，素性選択に関する課題点も残る．誤分類の多くは製品カテゴリに類似した性能表を含む文書である．テストセットにおける正解数及びノイズ数の比が大きく，加えて製品カテゴリに類似した性能表が D_{no} に多数含まれている場合，D_{real} における素性が一般化され，重みが減少する．さらに，項目欄はメーカ間で表記に違いがある為，少量の訓練セットにおいて頻度情報，特に df 値は効果的でない．これらは，訓練サンプル 1%における Bayes' rule 素性の精度の低さからも分かる．

最後に，TSVM 学習における誤学習について述べる．TSVM はその特性から，仮クラス内にて生起率の高い素性の次元による分類を狙おうとする．その為，本来分離すべき次元を超えて，全く意図しない素性での分離を行い，結果として誤学習に陥る可能性もある．ラ

[3] http://svmlight.joachims.org/

表 1: 実験データの内訳

製品カテゴリ	訓練 (正解:ノイズ)		テスト (正解:ノイズ)
	1% sampled	10% sampled	
PC	4 : 96	38 : 962	395 : 9605
デジカメ	2 : 98	22 : 978	210 : 9790
プリンタ	2 : 98	23 : 977	233 : 9767

表 2: 分類結果 (tf・idf)

製品カテゴリ	1% sampled		10% sampled	
	SVM	TSVM	SVM	TSVM
PC	88.83	**92.18**	**96.21**	94.53
デジカメ	53.61	**81.24**	**89.55**	89.39
プリンタ	38.18	**89.24**	88.76	**93.19**

表 3: 分類結果 (Bayes)

製品カテゴリ	1% sampled		10% sampled	
	SVM	TSVM	SVM	TSVM
PC	87.56	**93.05**	**96.91**	96.87
デジカメ	19.74	**80.60**	91.21	**92.94**
プリンタ	12.88	**78.93**	85.56	**90.18**

ベル無データから訓練サンプルを得るのは，サンプルとラベル無データとに同じ特徴分布を持たせる事で誤学習を回避する為であるが，サンプル，ラベル無データ間の比が大きくなる程に誤学習の危険性も高まる．高精度を維持しつつ訓練数の減少を目指すには，様々な課題に取り組む必要がある．

4 表領域抽出

ここではフィルタリング処理より渡された，文書内に性能表を含むと推測される候補文書群に対し，性能表領域の抽出を行う．抽出処理はさらに幾つかの処理に細分されるが，主として，候補群より生成した規則を基に文書内の各 T wrapper についてスコアリングを行い，最もスコアの高い T wrapper を製品性能表として抽出する．

フィルタリング処理にて渡された候補文書群をデータセットとし，表領域抽出実験を行った．実験は closed test にて抽出規則の最適化を行い，open test にて規則の妥当性を評価した．正解文書から一件ないし複数件の製品性能表を抽出出来た場合は抽出成功とし，ノイズ文書からテーブルを抽出した場合は誤抽出，正解文書から製品性能表を抽出出来なかった場合を抽出洩れとした．実験結果を表 4 に示す．

closed test について，PC に関する抽出洩れは全て規則の不適合性によるものだった．open test における抽出洩れについては，33%が規則の不適合，67%が HTML タグの記述誤りによる解析失敗だった．表領域抽出では HTML タグの欠落や記述誤り，特殊な構造を持つ性能表をある程度許容しながら解析を行うが，これらの解析失敗については規則の最適化により解決可能である．また，誤抽出の 96%は製品カテゴリに類似した性能表だった．フィルタリングにて棄却出来なかったノイズ文書は表抽出処理にも影響する為，フィルタリングの問題解決が誤分類の減少に繋がる．

表 4: 表領域抽出結果

カテゴリ	適合率	再現率	F-$measure$
PC	98.93 %	93.45%	96.11%
デジカメ	91.78%	90.95%	91.36%
プリンタ	86.31%	87.39%	86.65%

5 おわりに

本稿では，製品選択支援システムの構築に向け，システム入力部にあたる，HTML 文書からの製品性能表の抽出処理について述べた．HTML 文書はメーカのウェブサイトより自動収集し，フィルタリング処理は機械学習を用いた分類器生成による解決を試みた．実験は実際の製品メーカからの抽出を想定し，10000 件のテストセットからフィルタリングを行なった．また，高精度の分類器生成と学習コストとの関係を考慮し，ラベル無セットの分布を学習に組み込む事で少量の訓練セットにて高精度を達成出来る TSVM を分類器生成に用いた．結果，最大 60%の精度向上，並びに従来の 1/10 の学習数にて同等の精度を達成出来た．TSVM に対する課題点も多い．しかし，少量のラベル付データ及び多量のラベル無データを学習に用いる TSVM は，WWW をリソースとする本研究において有用性が高い．問題解決に臨み，さらなる発展を図りたい．

HTML は他の構造言語に対し比較的柔軟なコーディングが行える一方，ウェブドキュメントにおける構造の曖昧性をもたらしている．WWW をリソースとした情報検索やデータマイニングに関する研究は盛んではあるが，ウェブドキュメントを対象とした研究では人手にて正規化された文書を入力に用いたものが多く，構造の曖昧性を考慮するといささか現実的ではない．表領域抽出処理では構造の曖昧性をいかに許容するかを展望とし，より堅牢性のある性能表抽出を目指す．

参考文献

[1] H. H. Chen, S. C. Tsai and J. H. Tsai: Mining tables from large scale HTML texts, Proc. of the COLING2000, pp.166-172, 2000.

[2] T. Joachims. : Transductive Inference for Text Classification using Support Vector Machines, Proc. ICML-1999, pp.200-209 (1999).

[3] K. Shimada et al. : Information Extraction from Personal Computer Specifications on the Web Using a User's Request, IEICE Transactions on Information and Systems, Vol E86-D, No.8, pp. 1386-1395,2003.

[4] 松本裕治，北内啓，山下達雄，平野喜隆，松田寛，高岡一馬，浅原正幸: 日本語形態素解析システム「茶筌」 http://chasen.aist-nara.ac.jp/

[5] Vapnik, V. : Statistical Learning Theory. Wiley, 1998.

[6] Y. Wang and J. Hu : A machine learning based approach for table detection on the Web, Proc. of The Eleventh International World Web Conference, 2002.