SVM および Transductive SVM を用いた製品スペック情報の抽出

嶋田　和孝[†]　林　晃司[††]　遠藤　勉[†]

ネットワークの普及により，今までは紙面で伝えられていた情報の電子化が進んでいる．本稿では，それら電子化された情報の一つである，製品のスペック情報の抽出について議論する．現在，製品情報を収集し，利用しているポータルサイトが数多く存在するため，膨大な Web ページの中から製品のスペック情報を的確に抽出することは，そのようなポータルサイトの自動構築のために大きな意義を持つ．製品のスペック情報は，殆どの場合，表形式で記述されている． Web 上の表は HTML の<TABLE> タグを用いて記述されるが，<TABLE> タグは表を記述する以外にも，レイアウトを整えたりする場合に頻繁に用いられる．ある特定の領域においては，<TABLE> の 70% がレイアウト目的で使われているとの報告もある．そのため， HTML 文書中の<TABLE> タグが表なのか，それとも他の目的で使用されているのかを判別する必要がある．提案手法では， Support Vector Machines (SVM) を用いて， Web ページ中に存在する表領域が製品スペックかどうかの判定を行う． Transductive SVM を用いて，訓練データの削減についても考察する．パソコン，デジタルカメラ，プリンタの 3 種類の製品について，実験を行い，それぞれの製品について高い再現率と適合率を得た．訓練データが少ない場合， Transductive SVM を用いた手法の方が，通常の SVM と比べ，精度が改善されることを確認した．

キーワード:　製品スペック，表抽出，分類，　*Transductive SVM*

Product Specification Extraction Using SVM and Transductive SVM

Tables are an efficient way to express relational information. Most of information about products is written in tabular form. Table (specification) extraction is a significant task to handle product information written in tabular form such as specifications. We are developing a multi-specifications summarization system. The specifications are written in <TABLE> tags. The presence of the <TABLE> tags in an HTML document does not necessarily indicate the presence of specifications. Less than 30% of HTML <TABLE> tags are real tables in one particular domain. In this paper, we propose a method for specification extraction using SVMs. To reduce the training data, we also evaluate this task by using transductive SVMs. For PC, digital still camera and printer specifications, we evaluate the performance of SVMs and transductive SVMs. Experimental results show the effectiveness of our methods.

KeyWords:　*Product Specifications, Table Extraction, Classification, Transductive SVM*

† 九州工業大学 情報工学部 知能情報工学科, Department of Artificial Intelligence, Kyushu Institute of Technology

†† 九州工業大学 情報工学部 知能情報工学科. 現在は富士ゼロックス株式会社, Department of Artificial Intelligence, Kyushu Institute of Technology. He is now with Fuji Xerox Co.

1　はじめに

近年の Wrold Wide Web (WWW) の急速な普及により，世界中から発信された膨大な電子化文書へのアクセスが可能になった．しかしながら，そのような膨大な情報源から，必要な情報のみを的確に得ることは困難を極める．的確な情報を得るために，テキストを対象とした文書分類や情報の抽出などの様々な技術が注目され，研究されている．しかしながら， Web 上に存在するのはテキスト情報だけではなく，表や画像など様々な表現形式が使用されている．ここで，表形式で記述された情報について着目する．従来の情報検索システムなどでは，表はテキストとして扱われることが多かった．表は属性と属性値によって構造化された情報であり，その特性を考えると，表をテキストとして扱うのではなく，テキスト部分と切り離し，表として認識し，利用することが情報検索システムなどの精度向上に繋がる．また表は情報間の関係を記述するのに適した表現形式であり， Web 上に存在する文書から表を抽出することは， Web Mining や質疑応答システム，要約処理などのための重要なタスクの一つである (Hurst 2001; Itai, Takasu, and Adachi 2003; Pinto, McCallum, Wei, and Croft 2003; Shimada, Ito, and Endo 2003; Wang and Hu 2002, など).

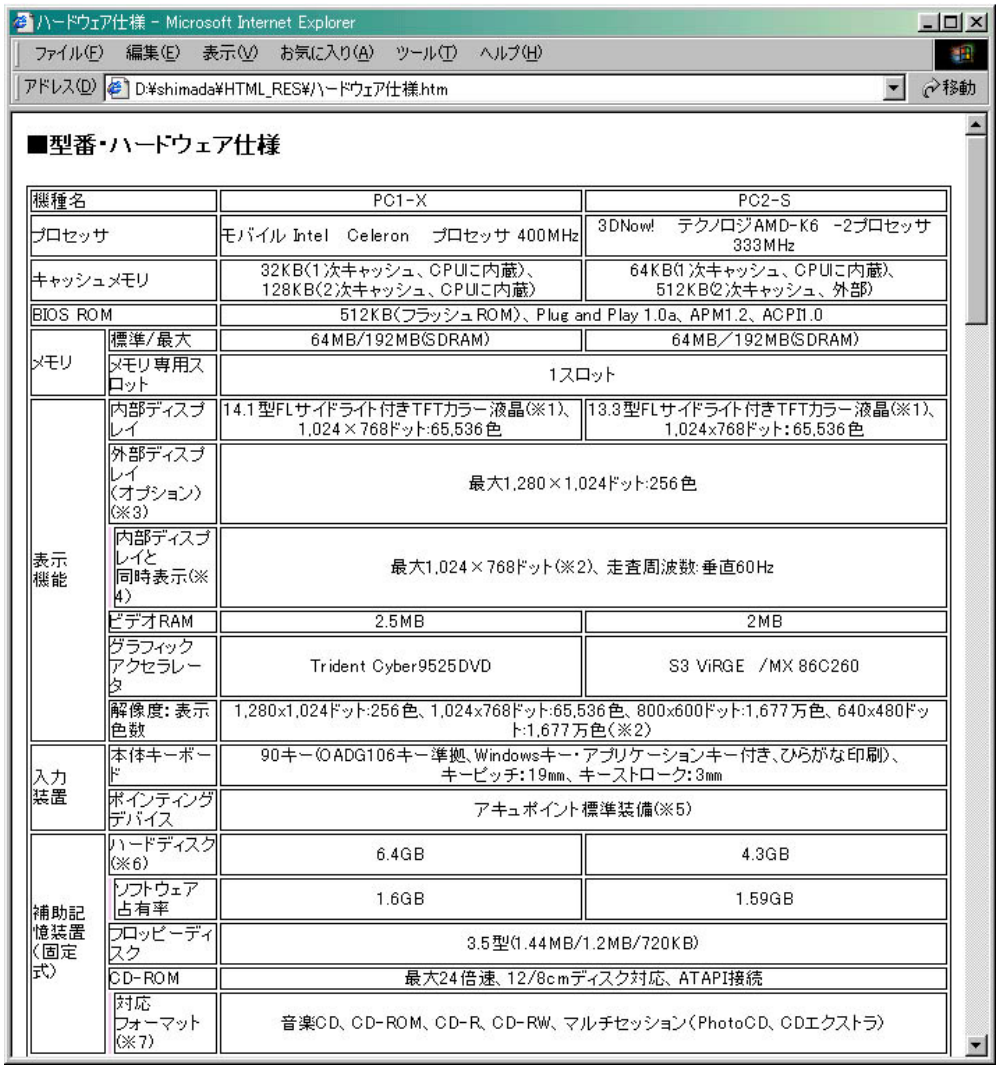
本稿では，電子化された情報の一つである，製品のスペック情報の抽出について議論する．一般に，パソコンやデジタルカメラ，プリンタなどの製品の機能や装備などのスペック情報は表形式で記述される．本稿ではこれらの表形式で記述されたスペック情報を性能表と呼ぶことにする．その例を図 1に示す．性能表を扱う理由としては，

- ポータルサイトの存在
  現在， Web 上には，数多くの製品情報に関するポータルサイトやオンラインショッピングサイトが存在する[1]．これらのサイトで，ユーザが製品を比較する際に最も重要な情報の一つが性能表である．多くの製品は頻繁に最新機種が発表され，その度に性能表を人手で収集するのはコストがかかる．膨大な Web ページの中から製品のスペック情報を的確に抽出することは，そのようなポータルサイトの自動構築のために大きな意義を持つ.
- 製品情報のデータベース化
  性能表は表形式で記述されているので，表領域が正しく特定されれば，属性と属性値の切り分けや対応付けなどの解析が比較的容易で，製品データベースの自動獲得が可能になる．これらのデータを利用し，ユーザの要求に合致した製品を選択するシステムなどの構築が可能になる (Shimada and Endo 2003).

などが挙げられる.

Web 上での表の記述に関しては，いくつか問題点がある．その一つが，<TABLE> タグの一般的な使用方法である． Web 上の表は HTML の<TABLE> タグを用いて記述されるが，<TABLE> タグは表を記述する以外にも，レイアウトを整えたりする場合に頻繁に用いられる．ある特定の領域においては，<TABLE> の 70% がレイアウト目的で使われているとの報告もある (Chen, Tsai,

1 価格.com (http://www.kakaku.com/) や Yahoo! Shopping (http://shopping.yahoo.co.jp/) など.



ハードウェア仕様 - Microsoft Internet Explorer

ファイル(F)　編集(E)　表示(V)　お気に入り(A)　ツール(T)　ヘルプ(H)

アドレス(D)　D:¥shimada¥HTML_RES¥ハードウェア仕様.htm　移動

■型番・ハードウェア仕様

機種名		PC1-X	PC2-S
プロセッサ		モバイル Intel　Celeron　プロセッサ 400MHz	3DNow!　テクノロジAMD-K6　-2プロセッサ 333MHz
キャッシュメモリ		32KB(1次キャッシュ、CPUに内蔵)、128KB(2次キャッシュ、CPUに内蔵)	64KB(1次キャッシュ、CPUに内蔵)、512KB(2次キャッシュ、外部)
BIOS ROM		512KB(フラッシュROM)、Plug and Play 1.0a、APM1.2、ACPI1.0	
メモリ	標準/最大	64MB/192MB(SDRAM)	64MB／192MB(SDRAM)
	メモリ専用スロット	1スロット	
表示機能	内部ディスプレイ	14.1型FLサイドライト付きTFTカラー液晶(※1)、1,024×768ドット:65,536色	13.3型FLサイドライト付きTFTカラー液晶(※1)、1,024x768ドット:65,536色
	外部ディスプレイ(オプション)(※3)	最大1,280×1,024ドット:256色	
	内部ディスプレイと同時表示(※4)	最大1,024×768ドット(※2)、走査周波数:垂直60Hz	
	ビデオRAM	2.5MB	2MB
	グラフィックアクセラレータ	Trident Cyber9525DVD	S3 ViRGE　/MX 86C260
	解像度:表示色数	1,280x1,024ドット:256色、1,024x768ドット:65,536色、800x600ドット:1,677万色、640x480ドット:1,677万色(※2)	
入力装置	本体キーボード	90キー(OADG106キー準拠、Windowsキー・アプリケーションキー付き、ひらがな印刷)、キーピッチ:19mm、キーストローク:3mm	
	ポインティングデバイス	アキュポイント標準装備(※5)	
補助記憶装置(固定式)	ハードディスク(※6)	6.4GB	4.3GB
	ソフトウェア占有率	1.6GB	1.59GB
	フロッピーディスク	3.5型(1.44MB/1.2MB/720KB)	
	CD-ROM	最大24倍速、12/8cmディスク対応、ATAPI接続	
	対応フォーマット(※7)	音楽CD、CD-ROM、CD-R、CD-RW、マルチセッション(PhotoCD、CDエクストラ)	

図 1　パソコンの性能表の例

and Tsai 2000)．そのため，HTML 文書中の<TABLE> タグが表なのか，それとも他の目的で使用されているのかを判別する必要がある．また，実際の Web 文書では，<TABLE> の入れ子構造が頻繁に見られる．性能表抽出のタスクでは，入れ子構造になった<TABLE> の中で，どこまでが性能表を表しているかという表領域を特定する必要がある．

提案手法では，(1) フィルタリング，(2) 表領域抽出，の 2 つのプロセスによって Web 文書群から性能表を獲得することを試みる．処理の流れを図 2に示す．ここで，フィルタリングとは，製品メーカのサイトから HTML ダウンローダで獲得した Web 文書群を対象とし，その中から性能表

図 2　処理の概要

を含む文書を抽出することを指す．フィルタリング処理では，文書分類などのタスクで高い精度を収めている Support Vector Machines (SVM) を用いる．また，少ない訓練データでも SVM と比較して高い精度を得ることができるといわれている Transductive SVM (TSVM) と SVM を比較する．一方，表領域抽出とは，フィルタリング処理で得られた文書中から，性能表の領域のみを抽出することを意味する．表領域抽出処理では，フィルタリングの際に SVM および TSVM のための素性として選ばれた語をキーワードとし，それらを基に表領域を特定する．

以下では，まず，2 節で，本稿で扱う性能表抽出のタスクに最も関連のある表認識などの関連研究について説明する．3 節では，フィルタリングに用いる SVM と TSVM について述べ，学習に用いる素性選択の手法について説明する．続いて，4 節で，各 Web 文書から表領域を特定する手法について述べ，5 節で提案手法の有効性を検証し，6 節でまとめる．

2　関連研究

本節では，本稿で扱う性能表抽出のタスクに最も関連のある表認識・抽出などについての先行研究について述べる．表やレイアウト構造を持つ文書からの構造解析および情報抽出に関する研究は，古くからなされている．しかしながら，従来の研究では，画像中の表領域の認識，箇条書きやプレインテキストで書かれた表の認識などが主な研究対象だった (Hu, Kashi, Lopresti, and Wilfong 2000; 河合, 塚本, 山本, 椎野 1998; Ng, Lim, and Koo 1999; Pinto et al. 2003; 佐藤, 佐藤, 篠田 1997, など).

一方で，近年，HTML で記述された文書を対象とした表認識や表抽出に関する研究がなされている．Chen ら (Chen et al. 2000) は，HTML 文書中の表の認識手法について提案したが，表

認識のためのルールが人手で作成されており，汎用性や拡張性に問題がある．Itai ら (Itai et al. 2003) は，表を対象とした HTML 文書からの情報抽出とその統合について報告している．しかし，表領域の抽出手法については十分に議論されていない．Wang ら (Wang and Hu 2002) は，決定木や SVM などを用いて表抽出を試みている．しかし，これらの手法は，学習のために十分な訓練データが必要となる．Yoshida ら (Yoshida, Torisawa, and Tsujii 2001) は，EM アルゴリズムを用いることで，この問題を解消しているが，精度は，Wang らの手法の方が高い．本稿では，文書分類で高い精度を収めている SVM を性能表抽出のタスクに適用し，少ない訓練データでも比較的高い精度が得られるといわれる TSVM と SVM の精度を比較する．

上に述べた従来の HTML からの表抽出に関する研究は，一般的な表を抽出することを目的としていることが多い．すなわち，<TABLE> タグで記述された領域が表であるか否かのみを判定することである．このような一般的な表抽出タスクでは，その<TABLE> タグ中にどのような内容が記述されているかを対象としないため，言語情報よりも構造情報 (例えば，縦および横のセルの一貫性など) を重視する．一方で，本タスクは表という構造情報を利用しながら，「ある特定の内容が記述された表」を抽出することを目的としている．内容にまで踏みいった抽出を行うには，構造情報だけではなく，言語情報も重要な手がかりとなる．本タスクが従来の表抽出と大きく異なる点は，上記の理由から，言語情報を重視した抽出処理を行うことである．

先行研究において，Yoshida ら (Yoshida et al. 2001) は表抽出ののちに表のクラスタリングを行なっている．しかし，一般に 1 つの文書中に膨大な数の<TABLE> タグが存在するため，機械学習などのための訓練データとして，全ての表にその表の内容が何であるかというラベルを付ける作業が高コストとなる．この問題点を解消するために，本研究では文書をフィルタリングしたのちに特定の表を抽出するという手法を取る．この手法により，正例および負例のラベル付けは，<TABLE> タグ単位ではなく，ページ単位となり，人手によるコストを最小限に抑え，実用的な精度を得ることができるという利点がある．

3　フィルタリング

本節では，フィルタリング処理について述べる．フィルタリングとは，製品メーカのサイトから HTML ダウンローダで獲得した Web 文書群から，性能表を含む文書を抽出することを指す．フィルタリング処理では，SVM および TSVM を用いる．

3.1　Support Vector Machines

SVM は Vapnik らが考案した Optimal Separating Hyperplane を起源とする，超平面による特徴空間の分割法であり，現在，二値分類問題を解決するための最も優秀な学習モデルの一つとして知られている (Vapnik 1999)．SVM は訓練サンプル集合からマージン最大化と呼ばれる戦略を

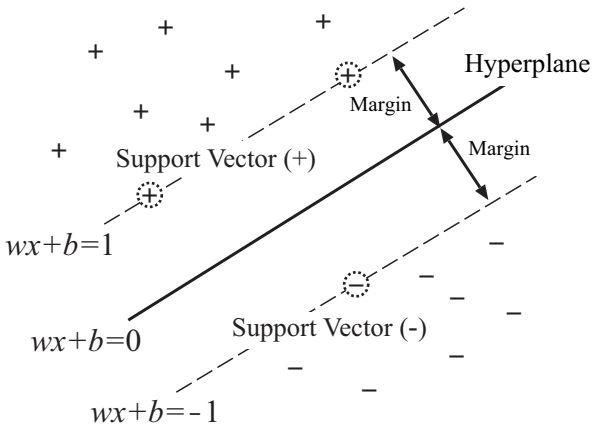


図 3　SVM の学習モデル

用いて，線形識別関数

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b \tag{1}$$

のパラメータを学習する．ここで，$\boldsymbol{x}$は入力ベクトルである．$\boldsymbol{w}$ と b がマージン最大化戦略の際に学習されるパラメータであり，$f(\boldsymbol{x}) \in \{+1, -1\}$ となる．

図 3に SVM の学習モデルを示す．＋は正のサンプル，－は負のサンプルである．図中の実線は $y = f(\boldsymbol{x}) = 0$ となる点の集合であり，分離超平面 (hyperplane) と呼ばれる．サンプルは，この超平面を境界として 2 つのクラスに分類される．すなわち，識別関数は分離超平面によって入力素性空間を二分する．また，超平面に対して最近傍のサンプル間の距離をマージンと呼び，$|\boldsymbol{w} \cdot \boldsymbol{x} + b|/||\boldsymbol{w}||$ で表す．図中の 2 つの破線上にある，分類を決定づける事例をサポートベクタと呼ぶ．

訓練データが線形分離可能な場合，$\boldsymbol{w}$ および b は複数存在することから，以下のような制約を与える．

$$\min_{i=1,\ldots,n} |\boldsymbol{w} \cdot \boldsymbol{x}_i + b| = 1 \tag{2}$$

この制約により，距離は $1/||\boldsymbol{w}||$ となり，結論として識別関数は

$$\min \quad \frac{1}{2}||\boldsymbol{w}||^2 \tag{3}$$

$$subject \quad to : \forall_{i=1}^{n} : y_i[\boldsymbol{w} \cdot \boldsymbol{x}_i + b] \geq 1$$

となる．本研究では，線形カーネルを利用した．

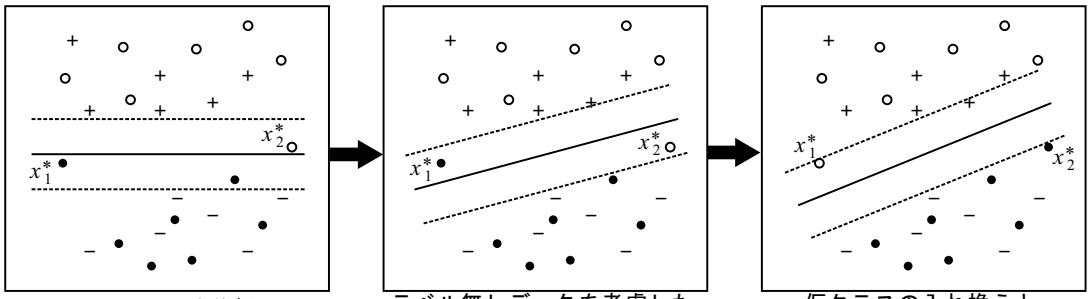


図 4　TSVM の学習モデル

3.2　Transductive SVM

一般に，高精度の分類器生成には多量の訓練サンプルを必要とする．しかし，十分な量の訓練データを人手によってラベリングするのは非常に高コストな作業といえる．そこで，少量の訓練データで高精度の分類器を生成する手法が期待される．

Vapnik (Vapnik 1999) が提案した理論を基に Joachims (Joachims 1999) によって具体化された Transductive SVM (TSVM) は， Transductive 法と呼ばれる，与えられたラベル無しデータの分布に注目し，ラベル無しデータの誤分類の最小化を目的とする学習方法を SVM に適用し，拡張したもので，学習時にラベル無し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．

Step 1　訓練データを基に SVM で分類器を生成する．

Step 2　得られた分類器を用いてラベル無しデータを分類する．得られた分類結果をそれぞれのラベル無しデータの仮クラスとする．

Step 3　仮クラスの付与されたラベル無しデータを訓練データに含め， SVM によって分類器を再生成する．

Step 4　マージン内のラベル無しデータのうち，各々の仮クラスを入れ替えることでマージンを最大化できるペアを見つけ，入れ換える．入れ換えられたデータセットを用いて， SVM による再学習を行う．この処理の際に，ラベル無しデータ中の正例および負例の分布を考慮する[2]．

Step 5　入れ換えるペアがなくなるまで **Step 4** を繰り返す．

図 4は， TSVM の学習過程の例である．図中の＋と－は，通常の SVM が分離超平面を生成する際に使用した正のサンプルと負のサンプルを表す (すなわち図 3における＋および－と同じ意味を持

2 一般には，ラベル無しデータ中の正例および負例の分布比率は未知なため，訓練データ中の正例と負例の比率などを参考にして求められた予測比率を利用し，パラメータが調整されることが多い．

つ)．ここで，$\circ$ および $\bullet$ はそれぞれ最初の SVM による分離超平面によって正例および負例と判断されたラベル無しデータを表す．例では，マージン内にある $\boldsymbol{x}_1^*$ と $\boldsymbol{x}_2^*$ がアルゴリズム中の **Step 4** の部分で入れ換えられ，再学習の結果，マージンが最大化された新しい分離超平面が得られる過程を示している． TSVM においても，通常の SVM と同様に線形カーネルを使用した．線形分離可能な場合， TSVM の識別関数および制約条件は下式に拡張される．

$$\min \quad \frac{1}{2}||\boldsymbol{w}||^2 \tag{4}$$

$$subject \quad to : \forall_{i=1}^{n} : y_i[\boldsymbol{w} \cdot \boldsymbol{x}_i + b] \geq 1$$

$$\forall_{i=1}^{n} : y_i^*[\boldsymbol{w} \cdot \boldsymbol{x}_i^* + b] \geq 1$$

ここで， $\boldsymbol{x}_j^*$ および y_j^* は，それぞれ仮クラスが与えられたラベル無データにおける入力ベクトルおよび仮クラスである．

3.3　素性選択

続いて， SVM および TSVM のための素性選択について述べる．本研究では，以下の条件を全て満たすものを素性候補とした．

(1)　表の属性欄中に出現する単語

(2)　一定長以内の文章中に出現する単語

(3)　性能表が存在する文書および性能表が存在しない文書内で顕著または限定的に出現する単語

これらの条件に基づき，素性となる候補を Web 文書から抽出する．条件 **(1)** では表中の要素を属性および属性値に切り分ける必要がある．ここでは，一般に殆どの性能表は第 1 列目 (最左列) に属性が現れ，それより右側の列に属性値が存在するという経験則から，最左列の要素を属性だと解釈する．表の属性部分を素性に使い，属性値を素性として用いない理由は，製品の属性 (例えば，パソコンなら CPU やメモリなど) は，新しい機種が発売されても変更されにくいのに対し，属性値 (例えば， CPU でいえば， 800MHz， 2GHz など) は，その値や表現に揺れが生じやすいためである．素性候補の抽出は，以下の手順で行われる．

(1)　HTML 文書から<TABLE> タグで記述された領域を抽出する．

(2)　<TABLE> タグ中の各<TR> タグ中の初めの<TD> タグの内容を抽出する (図 5)．

(3)　得られた文字列が 25 文字以内であれば，形態素解析[3]を行い，素性候補を抽出する． 25 文字という制約は経験的に定められた．

続いて，素性候補について重み付けを行い，素性を選択する．本稿では， (1) 正規化 $tf \cdot idf$, (2) ベイズの定理 の 2 種類を用いて，その精度を比較，考察する．ここで，性能表を含んでいる文書中の<TABLE> タグ内で顕著に生起する語と，性能表を含んでいない文書中の<TABLE> タグ内で顕著に生起する語を素性とする．以下に各手法での素性選択の流れを示す．

3 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」を用いた．http://chasen.naist.jp/hiki/ChaSen/

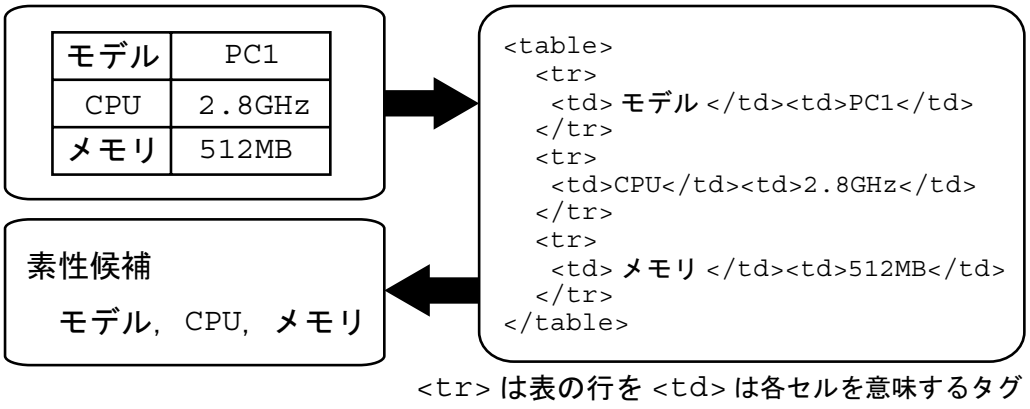


図 5　素性候補の例

正規化 $tf \cdot idf$

$tf \cdot idf$ は，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 $tf(t,d)$，および候補が生起する文書数 $df(t)$ を基に重み付けを行う最も有名な手法の一つである．ここで，本研究では，素性候補の抽出条件を考慮する．すなわち，素性の候補となる単語 t としては，文書 d 中の<TABLE> タグにおける最左列の単語のみを利用する．さらに，これを学習用に拡張し，$D = \{D_{real},\ D_{no}\}$ とする．ここで，D_{real} は性能表を含む文書群，D_{no} は求めている製品の性能表を含まないもしくは性能表以外のテーブルを含む文書群である．各々の文書群に生起する単語 t について，Wang ら (Wang and Hu 2002) が表抽出で用いた式を基に重み付けを行う．

$$w_t^{real} = \sum_{d_i \in D_{real}} tf(t,d_i) \times \log(\frac{df_t^{real}}{N_{real}} \frac{N_{no}}{df_t^{no}} + 1) \tag{5}$$

$$w_t^{no} = \sum_{d_i \in D_{no}} tf(t,d_i) \times \log(\frac{df_t^{no}}{N_{no}} \frac{N_{real}}{df_t^{real}} + 1) \tag{6}$$

ここで，df_t^{real}, df_t^{no} は D_{real} および D_{no} における単語 t の df 値である．また，N_{real} および N_{no} は，D_{real} および D_{no} に属する文書の総数を表す．最終的な重みは以下の式で求める．

$$ws_t^{real} = \frac{w_t^{real}}{Norm_{real}},\quad ws_t^{no} = \frac{w_t^{no}}{Norm_{no}} \tag{7}$$

ただし，

$$Norm_{real} = \sqrt{\sum_{t \in D_{real}} w_t^{real} \times w_t^{real}},\quad Norm_{no} = \sqrt{\sum_{t \in D_{no}} w_t^{no} \times w_t^{no}} \tag{8}$$

ここで閾値以上の値を持つ ws_t^{real} および ws_t^{no} を SVM および TSVM のための素性として扱う．

ベイズの定理

素性選択のためのもう一つの手段として，パターン認識・分類の分野で広く知られているベイズの定理を用いる．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．ここで，正規化 $tf \cdot idf$ と同様に素性候補の抽出条件を考える．すなわち，単語 t としては，文書 d 中の<TABLE> タグにおける最左列に生じるもののみを利用する．事前確率と条件付き確率密度分布 $p(t|C_i)$ 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．

$$P(C_i|t) = \frac{P(C_i)p(t|C_i)}{\sum_{j=1}^M P(C_j)p(t|C_j)} \tag{9}$$

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考える．すなわち，$ws_t^{real} = P(D_{real}|t)$，および $ws_t^{no} = P(D_{no}|t)$ である．ここで，$ws_t^{real} > 0.75$ を満たす語，$ws_t^{no} > 0.75$ でかつ 5 回以上生起した語を素性とする．

4　表領域抽出

本節では，表領域抽出処理について述べる．表領域抽出処理とは，フィルタリング処理によって得られた性能表を含んでいる文書から性能表の領域を特定する処理を指す．一般に，1 つの HTML 文書中には複数の<TABLE> タグが存在するため，それらの中から特定の表のみを抽出する処理が必要となる[4]．

4.1　スコアリング

まず，文書内の全ての<TABLE> タグについて，それぞれにユニークな ID とその<TABLE> タグの深さに関する情報を付加する．深さは 1 から始まり[5]，<TABLE> タグが入れ子構造になれば，その値は大きくなる．例を図 6に示す．

次に，各<TABLE>$_{id}$ についてスコアリングを行う．スコアリングには，前節で素性として選ばれた語 t とその値 ws_t^{real} を用いる．各<TABLE>$_{id}$ の最左列の要素について，以下の式でスコアを計算する．

$$Score_{id} = \sum_{t \in W_{list}} ws_t^{real} \times \log(s) \tag{10}$$

ここで，W_{list} は各<TABLE>$_{id}$ の最左列のセル中に存在する単語のリストを表し，s は，<TABLE>$_{id}$

4 我々が実験で用いたデータでは，1 つの HTML 文書中に含まれる<TABLE> タグの数の平均は 27.4 個だった．

5 深さ 1 は，<HTML> ··· </HTML> のレベル．

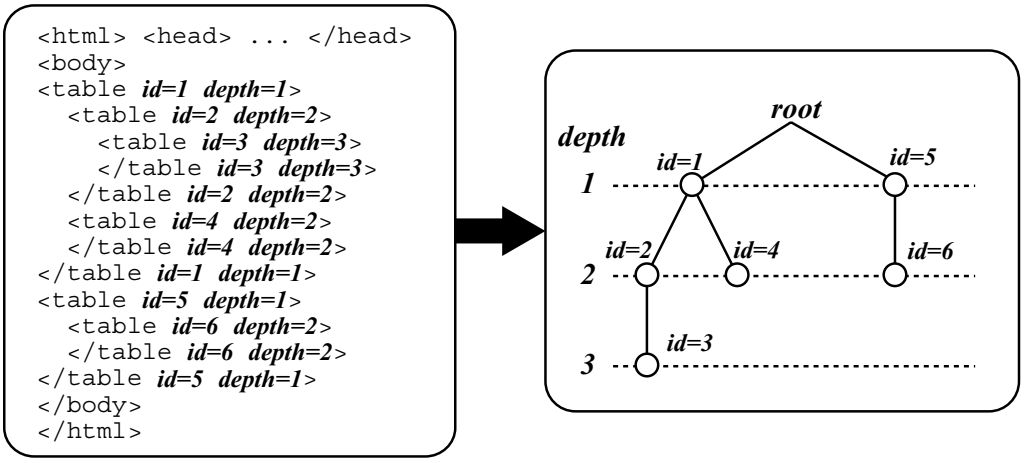


図 6　ID と深さ

の最左列の要素に生起したキーワード t の総数を表す．この $Score_{id}$ が最大になる<TABLE>$_{id}$ を性能表であると見なし，抽出する．

また，1 つの文書に複数の性能表が含まれていることもある[6]．そこで，$Score$ が最大になる<TABLE> にマッチしたキーワードの $\frac{4}{5}$ がマッチする<TABLE> も性能表だとして抽出する．

4.2　特殊な構造への処理

性能表が必ずしも 1 つの<TABLE> タグで構成されているとは限らない．実際に複数の表が入れ子構造になった性能表や複数の<TABLE> タグで分割されている性能表が多く存在する．前者の例は，図 6で $id = 5$ および $id = 6$ がまとまって 1 つの性能表である場合であり，後者の例は $id = 1$ と $id = 5$ が 1 つの性能表である場合である．

入れ子構造になった<TABLE> タグの場合，ある<TABLE>$_{id}$ が性能表と見なされたとすると，その<TABLE> より深さの深い<TABLE> は，性能表の一部だとして抽出する．さらに特殊な入れ子構造の例として，ブラウジングの際の視覚効果を狙い，<TABLE> タグ中の各<TD>...</TD> 内の要素が単一の<TABLE> タグで構成されている場合がある．このような場合は入れ子構造になっている<TABLE> タグ部分を通常の単一セルと見なして処理する．図 7に例を示す．

続いて，1 つの性能表が複数の<TABLE> タグによって構成されている場合の処理について述べる．まず，次の条件を満たす<TABLE>$_{id}$ を抽出する．

- <TABLE> タグの深さが等しい．
- 同じ親を持つ<TABLE> タグである．

6 実験データでは，1 つの文書に複数の性能表が存在した割合はデジタルカメラとプリンタの場合は 1% 以下だったが，パソコンの場合は 6% であった．

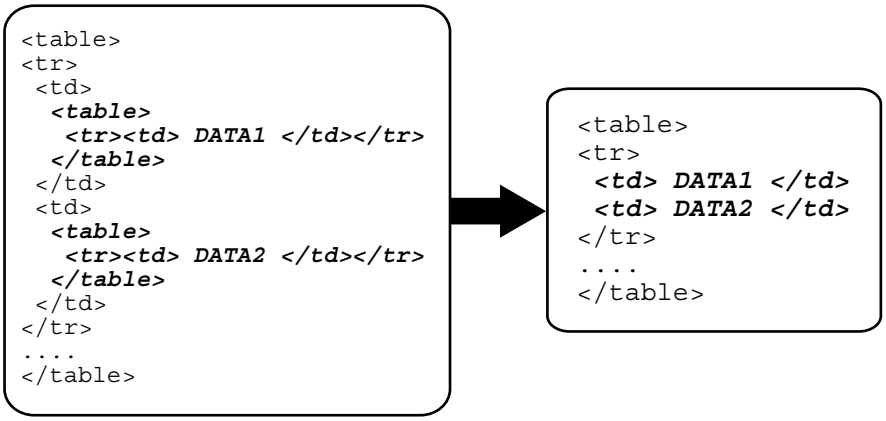


図 7　単一セルと見なされる入れ子構造

表 1　データセット

	パソコン	デジタルカメラ	プリンタ
性能表を含んでいる文書数	2090	236	520
性能表を含んでいない文書数	50621	11215	22055

図 6でいえば，$id = 1$ と $id = 5$ のペア，$id = 2$ と $id = 4$ のペアがこれにあたる．次に抽出された<TABLE> タグ群について，次の項目をチェックする．

(1)　各行のセルの数が一致するか

(2)　<TABLE> タグに幅 (width) が指定されている場合，その値が一致するか

(3)　<TABLE> タグの<TD> タグについて，背景色 (bgcolor) が指定されている場合，その使用パターンが一致するか

これらの項目のうち，2 つ以上の項目に，抽出された<TABLE> 群がマッチする場合は，それらを 1 つの<TABLE> として捉え，スコアリングの際に，それぞれのスコアの和をその<TABLE> のスコアとする．

5　実験

本節では，提案したフィルタリングおよび表領域抽出処理に関する評価実験について述べる．実験対象となる製品は，パソコン，デジタルカメラおよびプリンタの 3 種類とした．28 の製品メーカサイトから HTML ダウンローダを用いて獲得した HTML 文書群を評価実験の対象とした．総データ数は 86737 文書であり，それら文書の製品ごとの内訳を表 1に示す．但し，性能表を

含んでいない文書群中には，別の製品の性能表が含まれている．例えば，デジタルカメラの場合は，性能表を含んでいない文書にフィルムカメラやビデオカメラなどの性能表が含まれていることがある．

5.1 フィルタリングに関する実験

まず，フィルタリング処理について評価する．実装には，SVM^{light} を使用した[7]．評価の基準として，適合率，再現率，F 値を用いる．それぞれの値は以下の式で算出される．

$$\text{適合率}(P) = \frac{\text{抽出された文書中で性能表を含んでいる文書の数}}{\text{抽出された文書の総数}} \tag{11}$$

$$\text{再現率}(R) = \frac{\text{抽出された文書中で性能表を含んでいる文書の数}}{\text{性能表を含んでいる文書の総数}} \tag{12}$$

$$F\text{値}(F) = \frac{1}{\frac{1}{2P} + \frac{1}{2R}} \tag{13}$$

フィルタリング処理では，訓練データ数を 100 文書，300 文書，500 文書，1000 文書とした，4 つの場合について評価した．それぞれの訓練データは全データからランダムに 5 セットずつ抽出され，適合率および再現率は 5 セットの実験結果の平均値とした．評価データは，ダウンローダによって獲得された全データから，サンプリングされた訓練データを除いたもので，訓練データと評価用のデータは重複しない．TSVM のためのラベル無しデータとしては，評価データからサンプリングした 1000 文書を用いた．ラベル無しデータと評価データは重複している．素性選択としてベイズの定理を用いた場合の実験結果を表 2に，正規化 $tf \cdot idf$ を用いた場合の実験結果を表 3に示す．図 8にベイズの定理もしくは正規化 $tf \cdot idf$ によって素性として選択された単語の例を示す．単語とともに記述されている数値は，それぞれの手法によって算出された，性能表を含む文書群 (D_{real}) もしくは性能表を含まない文書群 (D_{no}) における，その単語の重みを表す．例えば，図中の正規化 $tf \cdot idf$ の例では，メモリやスロットという単語は D_{real} で顕著に出現し，方法やロードなどは D_{no} で顕著に出現したことを表している．この値を基に SVM のための素性が選択された．

フィルタリング処理で抽出された文書の例を示す[8]．図 9は，デジタルカメラを対象とした場合の正解例である．表の最左列に細分化された多くの属性が存在するため，訓練データから抽出された素性とマッチする．このように最左列に多く属性が存在する場合は，正しく分類される．一方で，図 10は同じくデジタルカメラを対象とした場合の失敗例である．この失敗例は，ある製品で撮影した写真の画像サンプルに関する文書であり，この文書には性能表が含まれていない．しかし，文書中にあるいくつかの<TABLE> 中の最左列に訓練データで抽出した素性がマッチしてしまい，誤抽出となった．図 11は，性能表を含んでいるにもかかわらず，フィルタリングで獲得できなかった例である．この文書に含まれる性能表は，属性部分が細かく分類されておらず，属性値の欄に箇

7 http://svmlight.joachims.org

8 訓練データ数が 1000 文書の場合の実験結果からの抜粋.

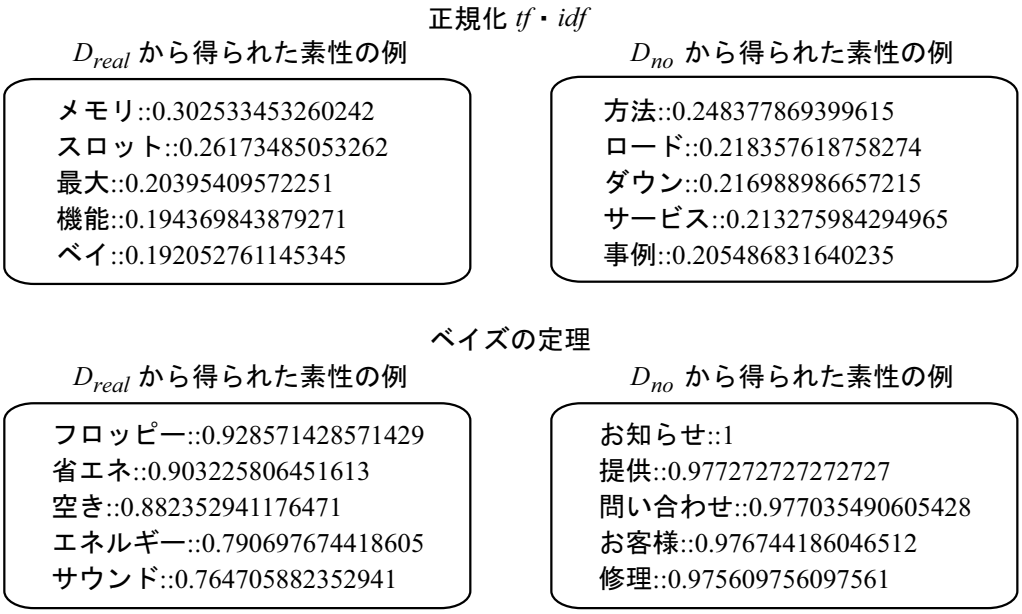
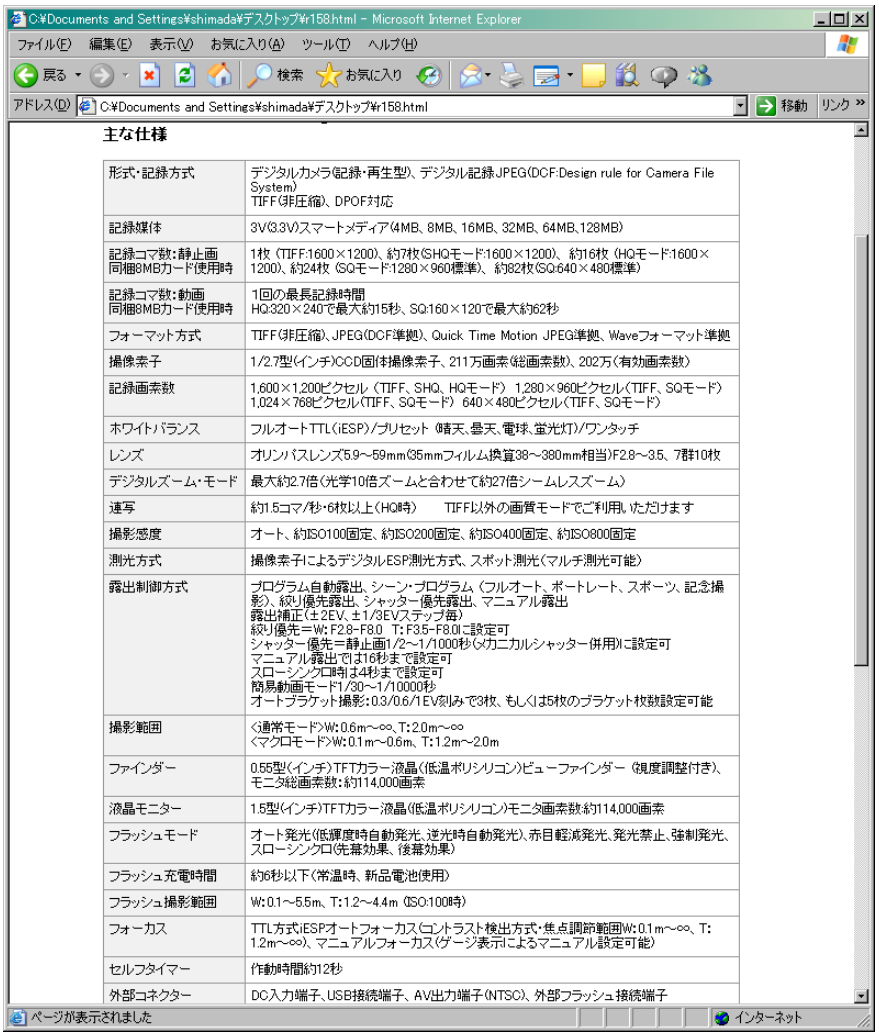


図 8　選択された素性とその値の例 (パソコンの場合)

条書きで細分化された属性が記述されている．提案手法では，分類器のための素性を表の最左列に限定しているため，このような性能表は正しく獲得できない場合がある．

ベイズの定理と正規化 $tf \cdot idf$ によって選ばれた素性を比較すると，多くの場合，正規化 $tf \cdot idf$ の方が高い F 値を収めた．製品種別で比較すると，デジタルカメラの精度が低くなる傾向があり，プリンタもパソコンに比べると精度が落ちる．この理由としては，(1) パソコンの性能表は比較的大きな表であることが多く，有効なキーワードが得やすいこと，(2) デジタルカメラのメーカは，フィルムカメラやビデオカメラも扱っていることが多く，プリンタの場合にもコピー機やスキャナのような対象となる性能表によく似たノイズが同じサイト内に存在すること，などが挙げられる．しかしながら，このように非常に似た性能表が混在しているにもかかわらず，比較的高い F 値を得ることができている．

SVM と TSVM について比較すると，ベイズの定理を用いて素性を選択した場合は殆どの実験で SVM に比べ，TSVM の方が高い F 値を得た．正規化 $tf \cdot idf$ を用いた場合は，TSVM の F 値の方が低くなることが多いが，両方の素性とも，訓練データが少ない場合は，TSVM の F 値が SVM の F 値を上回る傾向がみられた．これは，訓練データが少ない場合の TSVM の有効性を示している．TSVM が SVM の F 値を上回っている殆どの場合では，再現率が大幅に向上している．これは，TSVM が正例と負例の分布に基づいて再学習を行うためである．今回の実験では，ラベル無しデータの正例と負例の分布については，訓練データ中の正例と負例の分布を用いた



形式・記録方式	デジタルカメラ(記録・再生型)、デジタル記録JPEG(DCF:Design rule for Camera File System) TIFF(非圧縮)、DPOF対応
記録媒体	3V(3.3V)スマートメディア(4MB、8MB、16MB、32MB、64MB、128MB)
記録コマ数:静止画 同梱8MBカード使用時	1枚 (TIFF:1600×1200)、約7枚(SHQモード:1600×1200)、約16枚 (HQモード:1600×1200)、約24枚 (SQモード:1280×960標準)、約82枚(SQ:640×480標準)
記録コマ数:動画 同梱8MBカード使用時	1回の最長記録時間 HQ:320×240で最大約15秒、SQ:160×120で最大約62秒
フォーマット方式	TIFF(非圧縮)、JPEG(DCF準拠)、Quick Time Motion JPEG準拠、Waveフォーマット準拠
撮像素子	1/2.7型(インチ)CCD固体撮像素子、211万画素(総画素数)、202万(有効画素数)
記録画素数	1,600×1,200ピクセル (TIFF、SHQ、HQモード) 1,280×960ピクセル(TIFF、SQモード) 1,024×768ピクセル(TIFF、SQモード) 640×480ピクセル(TIFF、SQモード)
ホワイトバランス	フルオートTTL(iESP)/プリセット (晴天、曇天、電球、蛍光灯)/ワンタッチ
レンズ	オリンパスレンズ5.9~59mm(35mmフィルム換算38~380mm相当)F2.8~3.5、7群10枚
デジタルズーム・モード	最大約2.7倍(光学10倍ズームと合わせて約27倍シームレスズーム)
連写	約1.5コマ/秒・6枚以上(HQ時)　TIFF以外の画質モードでご利用いただけます
撮影感度	オート、約ISO100固定、約ISO200固定、約ISO400固定、約ISO800固定
測光方式	撮像素子によるデジタルESP測光方式、スポット測光(マルチ測光可能)
露出制御方式	プログラム自動露出、シーン・プログラム (フルオート、ポートレート、スポーツ、記念撮影)、絞り優先露出、シャッター優先露出、マニュアル露出 露出補正(±2EV、±1/3EVステップ毎) 絞り優先=W:F2.8~F8.0　T:F3.5~F8.0に設定可 シャッター優先=静止画1/2~1/1000秒(メカニカルシャッター併用)に設定可 マニュアル露出では16秒まで設定可 スローシンクロ時は4秒まで設定可 簡易動画モード1/30~1/10000秒 オートブラケット撮影:0.3/0.6/1EV刻みで3枚、もしくは5枚のブラケット枚数設定可能
撮影範囲	<通常モード>W:0.6m~∞、T:2.0m~∞ <マクロモード>W:0.1m~0.6m、T:1.2m~2.0m
ファインダー	0.55型(インチ)TFTカラー液晶(低温ポリシリコン)ビューファインダー (視度調整付き)、モニタ総画素数:約114,000画素
液晶モニター	1.5型(インチ)TFTカラー液晶(低温ポリシリコン)モニタ画素数:約114,000画素
フラッシュモード	オート発光(低輝度時自動発光、逆光時自動発光)、赤目軽減発光、発光禁止、強制発光、スローシンクロ(先幕効果、後幕効果)
フラッシュ充電時間	約6秒以下(常温時、新品電池使用)
フラッシュ撮影範囲	W:0.1~5.5m、T:1.2~4.4m (ISO:100時)
フォーカス	TTL方式iESPオートフォーカス(コントラスト検出方式・焦点調節範囲W:0.1m~∞、T:1.2m~∞)、マニュアルフォーカス(ゲージ表示によるマニュアル設定可能)
セルフタイマー	作動時間約12秒
外部コネクター	DC入力端子、USB接続端子、AV出力端子(NTSC)、外部フラッシュ接続端子

図 9　正しく獲得できた性能表を含む文書の例

が，ここに全データから算出された正例と負例の比を適用すると実験結果は表 4のようになる[9]．TSVM で使用する正例と負例の比が正確であれば，少数の訓練データの場合，さらなる精度向上に繋がることが確認できた．実験結果より，本タスクでは，訓練データが少ない場合においてTSVM が有効に機能することが確認された．

続いて，素性選択に使用した条件について考察する． 3.3節で示したように，素性は表の最左列のみを使用している．この条件の有効性を検証するために，素性選択に最左列という条件を除いた場合の結果を表 5に示す．訓練および評価データは最左列に限定したものと同じものを使用してお

9 素性には正規化 $tf \cdot idf$ で選ばれたものを利用した．使用した正例と負例の比は表 1の文書数の比である．

図 10　誤って獲得した性能表を含まない文書の例

り，素性選択の手法には，正規化 $tf \cdot idf$ を用いた．表は，通常の SVM に関する実験結果である．一部で，表中の全ての要素を素性選択に用いた場合の方が良い F 値を得ることがあるが，平均で 3 ～ 4% 程度，最大で約 13%，素性選択を最左列に限定した方が良いという実験結果が得られた．素性選択を最左列に限定しない場合に F 値が落ちる原因は，1 つの文書に含まれる<TABLE> タグが多いことが考えられる．我々が用いた実験データでは，1 つの文書に 30 個程度の<TABLE> タグが存在する．最左列という条件を除くと，性能表を含む文書中の関係のない<TABLE> タグの中身まで素性候補としてしまう可能性が高くなり，それが精度に影響するものと考えられる．実験結果より，素性選択に関する条件の有効性も確認できた．

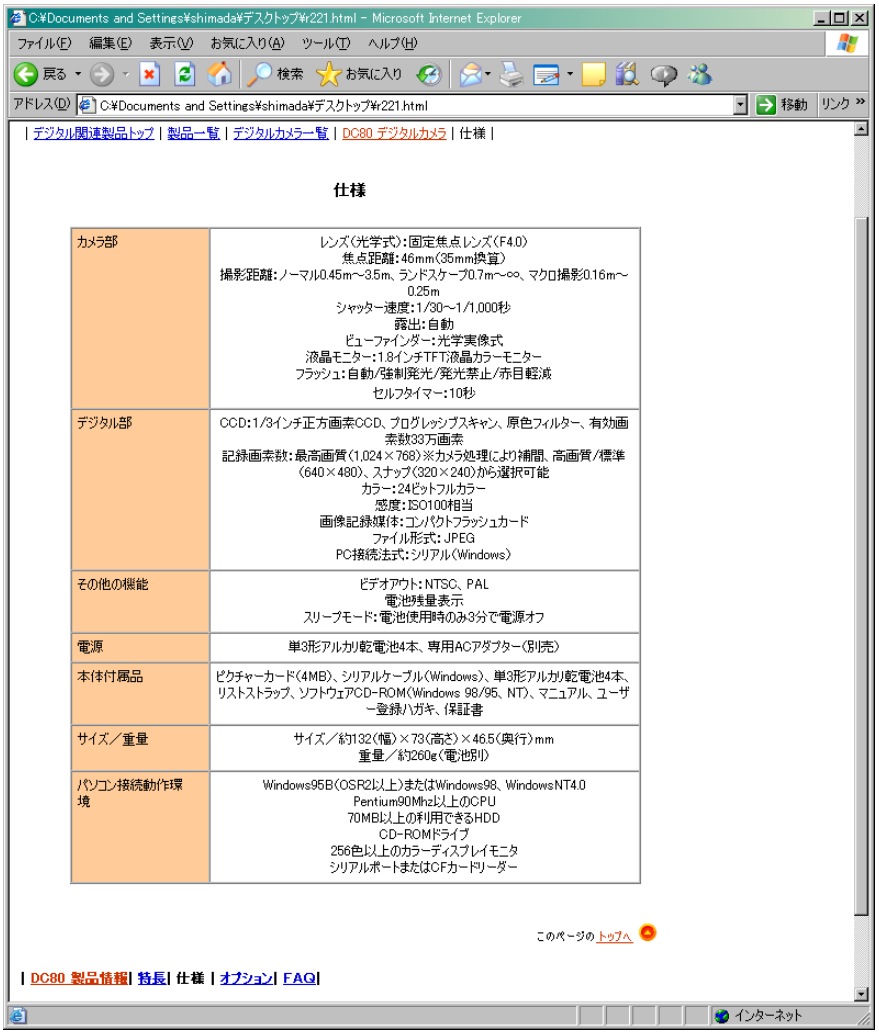


図 11　正しく獲得できなかった性能表を含む文書の例

表 2　フィルタリング実験結果 - ベイズの定理 -

製品	訓練データ数	SVM			Transductive SVM		
		適合率	再現率	F 値	適合率	再現率	F 値
パソコン	100	0.9121	0.5318	0.6719	0.7946	0.7848	**0.7897**
	300	0.8934	0.8885	0.8909	0.8507	0.8870	0.8685
	500	0.9226	0.8284	0.8730	0.8924	0.8870	**0.8897**
	1000	0.9200	0.9028	0.9113	0.9037	0.9191	**0.9113**
デジカメ	100	0.8845	0.4383	0.5862	0.6426	0.8580	**0.7348**
	300	0.8405	0.6668	0.7437	0.7389	0.8556	**0.7930**
	500	0.8751	0.7197	0.7898	0.8232	0.8196	**0.8214**
	1000	0.8830	0.8267	0.8539	0.8171	0.8785	0.8466
プリンタ	100	0.8772	0.2247	0.3577	0.6099	0.6077	**0.6088**
	300	0.9327	0.5132	0.6621	0.7647	0.7494	**0.7570**
	500	0.7263	0.5949	0.6540	0.8024	0.9098	**0.8527**
	1000	0.9245	0.8628	0.8926	0.8946	0.9147	**0.9045**

表 3　フィルタリング実験結果 - $tf \cdot idf$ -

製品	訓練データ数	SVM			Transductive SVM		
		適合率	再現率	F 値	適合率	再現率	F 値
パソコン	100	0.8532	0.7510	0.7988	0.7595	0.8427	**0.7990**
	300	0.8792	0.9318	0.9047	0.8564	0.9144	0.8845
	500	0.9142	0.8816	0.8976	0.9223	0.8776	**0.8994**
	1000	0.9293	0.9329	0.9311	0.9140	0.8946	0.9042
デジカメ	100	0.7510	0.7145	0.7323	0.6049	0.8211	0.6966
	300	0.8141	0.7870	0.8003	0.6856	0.8212	0.7473
	500	0.8793	0.7948	0.8349	0.7776	0.8392	0.8072
	1000	0.8532	0.8935	0.8729	0.8041	0.9200	0.8582
プリンタ	100	0.8684	0.5664	0.6856	0.6657	0.8145	**0.7326**
	300	0.8755	0.7926	0.8320	0.7915	0.8859	**0.8360**
	500	0.8620	0.9110	0.8859	0.8141	0.8907	0.8506
	1000	0.8973	0.9471	0.9215	0.8995	0.9040	0.9017

表 4　テストデータの分布を使用した場合の実験結果

製品	訓練データ数	適合率	再現率	F 値
パソコン	100	0.8239	0.8763	0.8493
	300	0.8518	0.9312	0.8897
	500	0.8723	0.9106	0.8910
	1000	0.9185	0.9392	0.9287
デジカメ	100	0.7173	0.7548	0.7356
	300	0.8080	0.7904	0.7991
	500	0.8065	0.7697	0.7877
	1000	0.8158	0.9198	0.8647
プリンタ	100	0.7280	0.7540	0.7408
	300	0.7983	0.8893	0.8413
	500	0.8438	0.8788	0.8609
	1000	0.9151	0.9046	0.9098

表 5　全てのセルを使用した場合の SVM の結果

製品	訓練データ数	適合率	再現率	F 値
パソコン	100	0.8336	0.7679	0.7994
	300	0.8665	0.8859	0.8761
	500	0.9029	0.8651	0.8836
	1000	0.9266	0.8742	0.8996
デジカメ	100	0.7030	0.5297	0.6041
	300	0.7817	0.8461	0.8126
	500	0.8004	0.7544	0.7767
	1000	0.8269	0.8834	0.8542
プリンタ	100	0.7692	0.5640	0.6511
	300	0.8463	0.7431	0.7913
	500	0.8324	0.8557	0.8439
	1000	0.8751	0.8545	0.8646

5.2 表領域抽出に関する実験

続いて，表領域抽出処理について実験する．ここでは，SVM および TSVM の素性選択に使用した，性能表を含む文書群 (D_{real}) 中で高い重みを持つ単語 t をキーワードとし，その値 ws_t^{real} を用いる．それらのキーワードによって，性能表を含む文書からどれだけの性能表の領域を正しく特定できるかを評価する．すなわち，実験データは，表 1の各製品の「性能表を含んでいる文書」に含まれる文書となる．実験には，フィルタリング処理で最も F 値が良かった実験結果の素性をキーワードとして用いた．実験結果を表 6に示す．正解率は以下の式で算出される．

$$\text{正解率} = \frac{\text{正しく抽出された性能表の数}}{\text{全文書に含まれる性能表の数}} \tag{14}$$

表中で，部分成功とは，ある製品の性能表が複数の並列した<TABLE> タグで構成されており (例えば，図 6の $id = 1$ と $id = 5$ が 1 つの性能表である場合)，その内のどれかが欠けている場合を指す．過抽出は，正解領域だけでなく，別の<TABLE> タグも併せて抽出した場合を表しており，誤抽出は，性能表ではない<TABLE> タグを抽出した場合である．

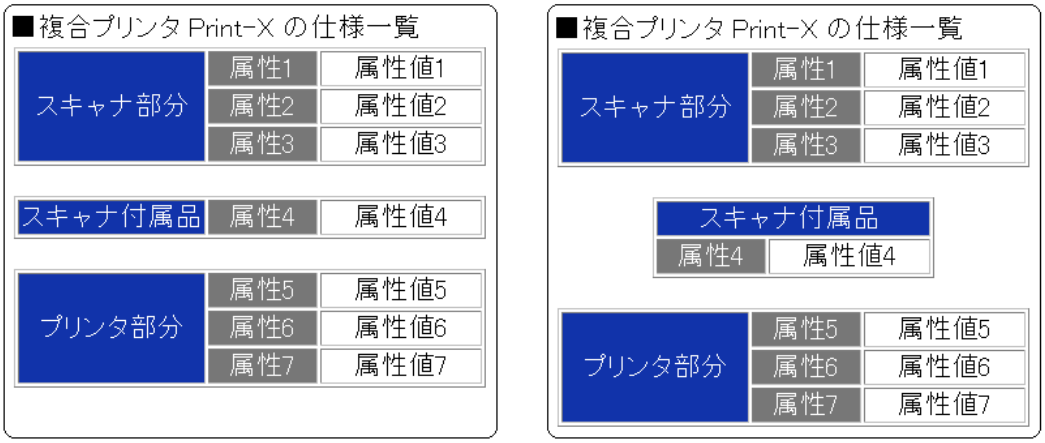
実験結果より，パソコンの場合は非常に高い精度で表領域を特定できることがわかる．パソコンの性能表の領域抽出の精度が高い理由としては，一般にパソコンの性能表が (1) 比較的大きな表であること，(2) 性能表が複数の並列した<TABLE> タグで記述されることが少ないこと，などが挙げられる．それと比較すると，デジタルカメラとプリンタの表領域抽出精度は若干落ちる．デジタルカメラやプリンタは，(3) 性能表を含む文書中に存在する<TABLE> タグの数がパソコンに比べて若干多いこと[10]，(4) 複数の並列した<TABLE> タグで一つの性能表が記述されることが多いこと，などが精度が劣る原因である．プリンタの場合で，誤抽出が多く見られるのは，(3) が大きな原因だと考えられる．(4) に対しては，表領域抽出処理で，各<TABLE> の構造的な類似度を用いて結合処理を行っているが，並列する<TABLE> 間に書式の異なる<TABLE> や性能表と関係のない<TABLE> が挿入されると，条件を満たさなくなり，<TABLE> が結合されない．その例を図 12に示す．図 12(b) の例では，3 つの<TABLE> は同じ深さで存在するが，[スキャナ付属品] の表が [スキャナ部分] と [プリンタ部分] の表の構造 (セルの数とセルの幅，対応するセルへの背景色) と異なるため，結合されず，最もスコアが高い<TABLE> のみが抽出されることになる．

誤抽出や結合に失敗する場合の対処法として，スコアがある閾値以上の<TABLE> のみを性能表として抽出するという手法が考えられるが，一般的な閾値を見つけることは難しい．また，現在我々が対象としているデータは，負例 (D_{no}) に比べて，正例 (D_{Real}) の数が極端に少ないため，サンプリングする訓練データの数によっては，十分な正例が得られず，必ずしも十分なキーワードが得られるとは限らない．その結果，設定した閾値によっては多くの性能表が棄却される場合もある．精度向上のためには，訓練データ中の正例の数をいかに多く獲得するかが課題となる．

10 実験データにおいて，パソコンは 1 文書中の<TABLE> タグの数が平均 24.8 個であり，デジタルカメラとプリンタはそれぞれ 29.4 個，36.6 個であった．

表 6　表領域抽出実験の結果 (フィルタリングの素性を利用)

製品	正解率	不正解の内訳 (文書数)		
		部分成功	過抽出	誤抽出
パソコン	0.991	5	5	8
デジカメ	0.907	17	0	5
プリンタ	0.887	24	5	30



(a) 正しく抽出される場合　　(b) 一部しか抽出されない場合

図 12　複数の<TABLE> による性能表

性能表には製品ごとやメーカごとに，その書式や使われている用語にある程度一貫性がある場合が多い．精度向上のための別の手法としては，抽出処理に利用するキーワードを全データから獲得するのではなく，メーカごとに獲得し，それらを用いて表領域抽出処理を行うことなども今後の課題として考えられる．

また，この実験とは別に，各製品に対して 5 分割交差検定を行った．すなわち，全データを 5 分割し，そのうち 4 つを性能表抽出のためのキーワード抽出処理の訓練データとし，残りの 1 つを評価データとした．実験結果を表 7に示す．5 分割にして実験を行ったことにより，フィルタリングに使用した素性の場合に比べ，訓練データの数が多くなるため[11]，有効なキーワードが獲得できた結果，全体的に誤抽出の数が減少した．部分成功の数が増加したのは，フィルタリングに使用した素性ではキーワード不足で誤抽出となっていたものが，キーワードの増加によって抽出され，部分成功になったためである．

11 例えば，PC では，1000 文書をサンプリングした場合 (フィルタリングで使用した素性をキーワードとする場合) の平均正例数は 41 文書だが，5 分割交差検定の場合，1672 文書の正例から性能表抽出のためのキーワードを獲得したことになる．

表 7　表領域抽出実験の結果 (5 分割交差検定)

製品	正解率	不正解の内訳 (文書数)		
		部分成功	過抽出	誤抽出
パソコン	0.993	3	5	6
デジカメ	0.911	19	0	2
プリンタ	0.923	27	8	5

表 8　単一セルと複数テーブルの数と全体に占める割合

	パソコン	デジカメ	プリンタ
性能表を含んでいる文書の数	2090	236	520
単一セルの数 (割合)	6 (0.3%)	32 (13.6%)	0 (0%)
複数テーブルの数 (割合)	10 (0.4%)	23 (9.7%)	69 (13.3%)

続いて，特殊な構造を持った文書について考察する．<TD>...</TD> 内の要素が単一の<TABLE> タグで構成されている場合 (以下，単一セルと呼ぶ) と複数の<TABLE> によって 1 つの性能表が構成されている場合 (以下，複数テーブルと呼ぶ) の内訳と全体に占める割合を表 8に示す．パソコンの場合は，単一セル，複数テーブルが存在する文書は正例中の 1% 以下だが，プリンタの場合は 13% の文書が，例え，抽出のための十分なキーワードを獲得できていたとしても，特殊構造への処理を行わないと根本的に抽出できないことになる．デジタルカメラの場合は単一セルと複数テーブルに重複があり，それを考慮すると，17% の文書 (42 文書) が同じく根本的に抽出できないことになる．特殊構造への処理を行わなかった場合と行った場合の正解率を表 9に示す．実験では，表 6の実験で使用したキーワードを利用した．パソコンについては，単一セルと複数テーブルの数が全体に対して少ないため，正解率の上昇は 1% 以下であった．一方で，特殊構造に対する処理を行わなかった場合，デジタルカメラの正解率は 0.800，プリンタの場合は 0.833 となった．すなわち，提案手法による特殊構造への処理は，プリンタの場合で 5% 程度，デジタルカメラの場合は約 10% の正解率の向上に繋がっている．特殊構造への対応にはいくつかの課題が残るが，この実験結果より，提案手法の有効性は確認できた．

本研究では，フィルタリングで性能表を含んでいる文書を絞り込み，続いて性能表を抽出するという流れを取った．提案手法以外にも，まず従来の表抽出の研究に基づき，一般的な表抽出を実行し，その表から特定の内容を含んだ表を抽出するという手法も考えられる．しかし，この手法を用いる場合，訓練データの獲得のために，全ての<TABLE> タグを人手でチェックし，正例もしくは負例のラベル付けをする必要がある．我々の使用した実験データでは，1 つの文書中に 30 前後

表 9　特殊構造への処理の有効性

製品	特殊構造への処理を行わない場合の正解率	特殊構造への処理を行う場合の正解率 (表 6の正解率)
パソコン	0.988	0.991
デジカメ	0.800	0.907
プリンタ	0.833	0.887

の<TABLE> タグが存在する．すなわち，膨大な数の<TABLE> タグへのチェックが必要になり，実用面を考えればコストが高い．一方で，提案手法は，そのページに性能表が含まれているかもしくは含まれていないかのチェックをするだけで良いという利点がある．また，表抽出処理では，フィルタリング処理で用いた SVM のような機械学習のアルゴリズムを使用しなかった．これは，上記の表抽出を行い，内容を分類するという手法における問題点と同様に，性能表の正確な領域を膨大な<TABLE> タグをチェックしながら，人手で正例のラベルを付けることが，高コストなためである．このように提案手法には，訓練データの作成に関して，実用的な面での大きな利点がある．

6　おわりに

本稿では， Web から製品のスペック情報を記述した表 (性能表) の抽出方法について述べた．提案手法は， Web からの製品データベースの自動獲得や，オンラインショッピングサイトの自動構築などのために有効である．

提案手法では， (1) フィルタリング， (2) 表領域抽出，の 2 つのプロセスによって Web 文書群から性能表を獲得することを試みた．フィルタリングでは， SVM と TSVM を用い，その精度を検証した．訓練データが少ない場合， TSVM が有効に機能することを確認した． TSVM の精度を向上させるには，ラベル無しデータ中の正例と負例の正確な比を推定することが有効である．少ない訓練データで，いかにラベル無しデータの正例と負例の分布を推測するかが今後の課題の一つとなる．表領域抽出処理では，パソコンの場合で，非常に高い抽出正解率を得た．デジタルカメラとプリンタの場合においても 90% 程度の精度を得ている．並列した複数の<TABLE> タグからなる性能表をより正確に抽出するためには，訓練データ中の正例をいかに多く獲得できるかや，構造的類似度に関する新たな尺度の導入が課題となる． 2 つのプロセスにおいて，現在は一括して素性選択やキーワード抽出を行っているが，メーカや製品ごとの表の記述方法の一貫性などを利用することで，より高い抽出精度が得られる可能性がある．その実装と評価は，今後の課題の一つである．

両プロセスとも 90% 程度の精度を得ており，実験結果から本手法の有効性と実用性を確認できた．

謝辞 実験の説明において，図 9はオリンパス株式会社[12]，図 10は松下電器産業株式会社[13]，図 11はコダック株式会社[14]に，論文中でのデータの利用についてご許可いただきました．但し，図 10では肖像権上の都合により一部の画像を差し替えている．また，本稿の改善に対して，査読者の方から数多くの有益なコメントをいただきました．ここに深く感謝いたします．

参考文献

Chen, H., Tsai, S., and Tsai, J. (2000). "Mining tables from large scale HTML texts." In *Proceedings of COLING2000*, pp. 166–172.

Hu, J., Kashi, R., Lopresti, D., and Wilfong, G. (2000). "Medium-independent table detection." In *Proceedings of Document Recognition and Retrieval VII*, pp. 23–28.

Hurst, M. (2001). "Layout and language: Challenges for table understanding on the web." In *Proceedings of Workshop on Web Document Analysis, WDA01*, pp. 27–30.

Itai, K., Takasu, A., and Adachi, J. (2003). "Information extraction from HTML pages and its integration." In *Proceedings of the 2003 Symposium on Application and the Internet Workshops (SAINT03)*, pp. 276–281.

Joachims, T. (1999). "Transductive inference for text classification using Support Vecor Machines." In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 200–209.

河合敦夫, 塚本雄之, 山本勝紀, 椎野務 (1998). "文書構造を利用した箇条書きや表形式文書からの内容抽出." 信学論 (D-II), **J81-D2** (7), 1609–1619.

Ng, H., Lim, C., and Koo, J. (1999). "Learning to recognize tables in free text." In *Proceedings of the 37th Annual Meeting of ACL*, pp. 443–450.

Pinto, D., McCallum, A., Wei, X., and Croft, W. B. (2003). "Table extraction using conditional random feilds." In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 235–242.

佐藤円, 佐藤理史, 篠田陽一 (1997). "電子ニュースのダイジェスト自動生成." 情報処理学会論文誌, **36** (10), 2371–2379.

Shimada, K. and Endo, T. (2003). "Product Specifications Summarization and Product Ranking System using User's Requests." In *Information Modelling and Knowledge Bases XV, IOS Press*, pp. 315–331.

12 http://www.olympus.co.jp/jp/news/2001a/nr010321c700uzspj.cfm

13 http://panasonic.jp/dc/gallery/fz3.html

14 http://wwwjp.kodak.com/JP/ja/digital/cameras/dc80/spec.shtml

Shimada, K., Ito, T., and Endo, T. (2003). "Multiform Summarization from Product Specifications." In *Proceedings of PACLING 2003*, pp. 83–92.

Vapnik, V. N. (1999). *Statistical Learning Theory*. Wiley.

Wang, Y. and Hu, J. (2002). "A machine learning based approach for table detection on the Web." In *Proceedings of The Eleventh International World Web Conference*.

Yoshida, M., Torisawa, K., and Tsujii, J. (2001). "Extracting ontologies from World Wide Web via HTML tables." In *Proceedings of PACLING 2001*, pp. 332–341.

略歴

嶋田和孝: 1997 年大分大学工学部知能情報システム工学科卒．1999 年同大大学院博士前期課程了．2002 年同大大学院博士後期課程単位取得退学．同年より九州工業大学情報工学部助手．博士 (工学)．表抽出，テキスト処理などの研究に従事．言語処理学会，電子情報通信学会，情報処理学会各会員．

林 晃司: 2002 年九州工業大学情報工学部卒．2004 年同大大学院修士課程了．現在は富士ゼロックス株式会社．在学中は表の分類・抽出に関する研究に従事．

遠藤 勉: 1972 年九州大学工学部電子工学科卒業．1974 年同大学院修士課程修了．1977 年同博士課程単位取得退学．同大助手を経て，1980 年大分大学工学部講師．同大助教授，教授を経て，2000 年より九州工業大学情報工学部知能情報工学科教授．工学博士．自然言語処理，コンピュータビジョンの研究に従事．電子情報通信学会，情報処理学会，人工知能学会，日本ロボット学会，日本ソフトウェア科学会，IEEE Computer Society 各会員．

(2004 年 9 月 28 日 受付)
(2004 年 11 月 30 日 再受付)
(2005 年 1 月 11 日 採録)