データベースシステム 119−30
（1999. 7. 22）

# Support Vector Machine を用いた Text Categorization

矢田裕之† 上原邦昭††

本稿では，Support Vector Machine（SVM）を用いて，テキストデータをカテゴリに分類する手法を提案する．SVM は，入力空間よりも非常に高次元な特徴空間において識別を行なうことで，高い認識率が得られる識別器である．また，SVM の精度向上のために，訓練事例のチューニングや，索引語の選択の方法，およびベイジアンネットワークを用いた属性値の補完について提案する．得られた結果は，Textual CBR の事例ベースとして利用することを目的としている．

# Text Categorization using Support Vector Machine

HIROYUKI YADA † and KUNIAKI UEHARA††

In this paper, we will propose the method to categorize the text data by using *Support Vector Machine (SVM)*. In order to improve recall and precision of categorization, we will also propose 3 methods: modification of training set, selection of indexes and completion of attribute value using Bayesian network. The result of recognition is used as the case base of textual CBR.

## 1. はじめに

近年，情報検索（IR）と事例ベース推論（CBR）を融合させた Textual CBR という考え方が提案されている．Textual CBR は，事例につけられた索引をたよりに事例の検索を行い，検索された事例を修正して問題解決するアルゴリズムである．Textual CBR の実装例として，FAQ Finder[1] があげられる．FAQ Finder では，各々の事例がカテゴリに分けられて事例ベースに蓄えられている．ユーザがカテゴリを選択すると，FAQ Finder はそのカテゴリに含まれる事例の中から，ユーザの質問に適するものを選択し出力する．

本研究では，UNIX コマンドの利用支援システムの開発を行なっている．このシステムは，ユーザの質問に対して適切なコマンドを提示することを目標の一つとしている．この目標をみたすために，Textual CBR の枠組みで開発している．実際には，事例ベースに UNIX コマンドのマニュアルを蓄えておき，各々のマニュアルとユーザの質問の類似度を計算し，適すると思われるコマンドをユーザに提示している．

ここで FAQ Finder と同様に，事例ベースをカテゴリに分類することを考える．コマンドマニュアルをカテゴリに分類すれば，システムはすべての事例に対する検索を行なう必要はなくなり，ユーザの選択したカテゴリ内の事例に対してのみ検索を行なうことができる．このような機能を実現するためには，事例ベースのカテゴリへの分類を自動化できることが望ましい．

本稿では Support Vector Machine（SVM）[2] を用いてテキストデータを分類する手法を提案する．SVM は，入力空間よりも非常に高次元な特徴空間において識別を行なうことで，高い認識率が得られる識別器である．SVM は，本来 2 クラスの識別器であるため，複数のクラスの分類に用いるためには工夫が必要になる．そこで，カテゴリに 2 分木構造をもたせ，各々の節点に SVM を用意することにより，複数のカテゴリへの分類を実現している．また，訓練事例のチューニングや，索引語の選択，ベイジアンネットワークによる属性値の補完を行い，SVM の精度の向上させるための手法を提案する．

## 2. Textual CBR

Textual CBR は，IR と CBR の 2 つのアルゴリズムを融合させたアルゴリズムとして提案された．IR は膨大な情報から必要とする情報のみを抽出するアル

† 神戸大学大学院自然科学研究科
Graduate School of Science and Technology, Kobe University
†† 神戸大学都市安全研究センター
Research Center for Urban Safety and Security, Kobe University

ゴリズムであり，CBR は与えられた問題に最も類似する問題を過去の事例から検索し，検索された事例を修正して問題解決を行うアルゴリズムである．

Textual CBR において，事例は加工されていないデータ（Raw データ）のまま蓄えられる．各々の事例は索引づけされており，その索引は該当する Raw データと対応づけられている．新たな問題の解を得る際には，その問題に対して索引づけを行い，各々の事例の索引と照合し，最も類似した既知の問題を検索する必要がある．たとえば，FAQ Finder では，ユーザの質問に対して，Q&A の Question を索引として照合し，最も類似度の高い Question を事例ベースから検索し，その Question に該当する Answer を質問の解としている．

Textual CBR では，索引による類似性の解釈において IR の照合手法を用い，事例の事例ベースへの蓄積や事例の修正に CBR の枠組みを用いている．そのため CBR と同様に，Textual CBR は索引と Raw データを一組にしたデータを事例としてもっている．しかしながら，データを構造化せずに事例ベースに蓄えるだけでは，検索の際にすべての事例に対してユーザの質問に対する類似度を計算することが必要になる．そこで，事例ベースを図 1 の様にカテゴリに分類し構造化することを考える．本稿で提案する SVM を用いた事例のカテゴリへの分類手法を用いれば，構造化された事例ベースを自動的に構築することができる．

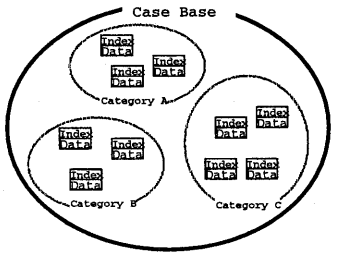


図1　Category in Case Base.

## 3. Support Vector Machine

入力空間を非常に高次元な特徴空間に写像し，その特徴空間において識別を行い，精度の高い識別を可能にしているのが Support Vector Machine（SVM）と呼ばれる 2 クラスの識別器である．本章では，SVM について説明する．

入力空間 $\Re^m$ に属する訓練サンプル集合を $\{\mathbf{x_i}\}_{i=1}^N$ とし，訓練サンプルのラベルを $y_i$ で表現する．ここで，$y_i = -1$ のとき $\mathbf{x_i}$ はクラス A に属し，$y_i = 1$ のときクラス B に属するとする．まず，訓練サンプル集合に対して，Optimal Hyperplane Classifier (OHC) を適用して学習を行う．次に，得られた OHC の関数を用いて，未知のサンプルをクラス A かクラス B にクラス分けする．なお，OHC は線形識別器であり，式 (1) で表される線形識別関数を用いている．ここで，$\mathbf{w} \in \Re^m$，$b \in \Re$ である．

$$f(\mathbf{z}) = \mathbf{w}^T\mathbf{z} + b \tag{1}$$

式 (2) に示すとおり，カーネル関数の固有関数 $\varphi_i$ と固有値 $\lambda_i$ を用いて特徴空間 $\Re^n$ を定義する．

$$\phi(\mathbf{x_i}) = (\sqrt{\lambda_1}\varphi_1(\mathbf{x_i}), \ldots, \sqrt{\lambda_n}\varphi_n(\mathbf{x_i}))^T \tag{2}$$

一般に，$n$ は $m$ より大幅に大きくなるため，特徴空間において識別を行えば，より精度の高い識別が可能になる．非線型写像 $\phi$ により，訓練サンプル集合を特徴空間に写像した集合に対して，式 (3) を満たす $\mathbf{w}$, $b$ を得る．この不等式の解は一意に定まらないが，得られた解の中で VC 次元☆ が最大になるものを解とする．VC 次元の上限は $\left|\left|\mathbf{w}\right|\right|^2$ に比例するため，$\mathbf{w}$, $b$ は，式 (3) の条件のもとで $\left|\left|\mathbf{w}\right|\right|^2$ を最小にするという二次計画問題（quadratic programming problem）の解として与えられる．

$$\begin{cases} f(\phi(\mathbf{x_i})) \geq 1 & (y_i = +1) \\ f(\phi(\mathbf{x_i})) \leq -1 & (y_i = -1) \end{cases} \tag{3}$$

この問題を解くことは困難であるため，ラグランジュ乗数 $\alpha_i$ を導入して，$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x_i})$ とすると，$\sum_{i=1}^N \alpha_i y_i = 0$，$\alpha_i \leq 0$ の条件のもとで，式 (4) の値を最大にするという等価な問題に変換できる．

$$\sum_{i=1}^N \alpha_i + \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \alpha_i\alpha_j y_i y_j \phi(\mathbf{x_i})^T\phi(\mathbf{x_i}) \tag{4}$$

この問題から得られた最適解を $\alpha_i^*$ とすると，OHC の識別関数は式 (5) で表される．ここで，$\alpha_i^* \neq 0$ の訓練サンプル $\mathbf{x_i}$ は Support Vector（SV）と呼ばれる．

$$f(\phi(\mathbf{z})) = \sum_{i=1}^N \alpha_i^* y_i \phi(\mathbf{z})^T\phi(\mathbf{x_i}) + b \tag{5}$$

カーネル関数 $K(\mathbf{x}, \mathbf{y})$ は固有関数，固有値に対して式 (6) の関係が成り立つため，式 (2)，式 (6) より，$K(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i})^T\phi(\mathbf{x_j})$ が成り立つ．

☆ VC-dimension

$$K(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \lambda_i \varphi_i(\mathbf{x}) \varphi_i(\mathbf{y}) \qquad (6)$$

以上のことから，OHC の識別関数は式 (7) で表すことができる．未知のサンプル $\mathbf{z}$ が与えたとき，識別関数の値が $\leq 0$ の時は $\mathbf{z} \in$ クラス A，逆に $\geq 0$ の時は $\mathbf{z} \in$ クラス B として，未知のサンプルをクラス分けすることができる．

$$f(\phi(\mathbf{z})) = \sum_{i=1}^{N} \alpha_i^* y_i K(\mathbf{z}, \mathbf{x_i}) + b \qquad (7)$$

## 4. カテゴリの階層化

3 章で説明したとおり，SVM は本来 2 クラスの識別器であるため，多クラスの分類を行うときには工夫が必要になる．本研究では，カテゴリに 2 分木構造をもたせて，SVM による複数個のカテゴリへの分類を実現している．

本稿でカテゴリにもたせる階層構造は，図 2 のような汎用性の高い 2 分木である．さらに，2 分木の葉 (leaf) に各々のカテゴリを配置し，節点 (node) に SVM を配置している．各 SVM は，配置する節点の右の子に含まれるカテゴリに属するすべての訓練事例を正事例，左の子に含まれるカテゴリに属するすべての訓練事例を負事例として作成している．

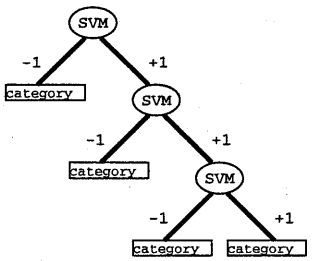


図2 Binary Tree Structure.

本手法で作成した分類木を，UNIX コマンドのマニュアルのカテゴリ分類に適用することを議論する．まず，提案手法を用いて，訓練事例から分類木を作成する．試行を行う際には，はじめに 2 分木の根 (root) に配置されている SVM が，試行するマニュアルが正事例であるか負事例であるかを判定する．正事例であった場合には，右の子節点に配置されている SVM が，試行するマニュアルが正事例であるか負事例であるかを判定する．この操作を繰り返して，最終的にたどり着いた葉に配置されているカテゴリが試行したマニュアルの属するカテゴリとなる．

この分類で最も重要となるのが SVM の精度である．SVM の精度は，訓練事例のカテゴリ分けの正確さと，訓練事例のデータ表現に依存している．5 章では，それらを改善することにより SVM の精度をさらに向上させることを議論する．

## 5. マニュアルのカテゴリへの分類

まず，事例として与えるマニュアルのデータ表現について説明する．事例として与えるマニュアルは，属性–値対の組で記述している．その際には，該当するマニュアルに索引語が含まれていれば属性の値を 1，含まれていなければ 0 とする．この操作の結果得られたベクトルを正規化して，マニュアルを特徴ベクトルとして表現している．なお，本稿において重要語とは，機能語を除いた名詞と動詞のこととする．また索引語とは，特徴ベクトルを作成する際に属性とする重要語のことである．分類された訓練事例を特徴ベクトルで表現し，そのベクトルから SVM を作成している．

### 5.1 訓練事例のチューニング

訓練事例は人手でカテゴリ付けされているので，作業を行なった者の感覚に依存するところが大きい．このため，どうしても結果にはばらつきが生じてしまう．このような訓練事例から作成された SVM では十分な精度を得ることはできない．そこで，以下の手順で訓練事例のカテゴリ付けを調整することを考える．

(1) 訓練事例に対して交差検定を行い，失敗した訓練事例の中で，しきい値以上の値が SVM から出力された訓練事例の集合を，事例集合 Failure とする．

(2) Failure の中で，分類の際に最大の値が SVM から出力された訓練事例をとりだす．そして，その事例のカテゴリを変更した後に，訓練事例に対してもう一度交差検定を行う．この際に，定められた評価尺度による評価が向上した時は手順 (1) に戻る．また，向上しなかったときは手順 (2) を繰り返す．

手順 (1) における，SVM から出力された値が大きい訓練事例とは，分類を失敗している可能性の高い訓練事例である．以上の操作は，この様な訓練事例のカテゴリを変更して，より正確な訓練事例の分類をめざしたものである．この操作を実行して訓練事例のチューニングを行えば，SVM の精度を向上させることができる．

### 5.2 索引語の選択

すべての重要語を索引語としてあつかうと計算量が

膨大になってしまい効率が悪い．このため，TFIDFを用いたアルゴリズムを用いて，重要語の中でも特に重要であると思われる語を選択し索引語とする．そして，索引語を属性として特徴ベクトルを作成する．索引語を選択するためのアルゴリズムとして，以下の2つのアルゴリズムを用いている．

(1) ある単語の文書中での出現頻度と分散を用いてその単語の重要度を計算するTFIDF☆を，索引語の選択に用いる．TFIDFのアルゴリズムから重要度を得て，重要語の重要度の大きいものから，順に定めた個数を索引語として選択する．TFIDFによる重要語の重要度 $t_i$ は，式(8)より求められる．

$$t_i = n_i \times log \frac{M}{m_i} \tag{8}$$

なお，$M$ はマニュアル数，$n_i$ は該当する重要語のすべてのマニュアルにおける出現回数，$m_i$ はそれぞれ該当する重要語が含まれるマニュアル数である．

(2) 訓練事例の正事例集合と負事例集合の各々でTFIDF用いて重要語の重要度を求め，その重要度の差の絶対値を単語の重要度とする．重要語の重要度の大きいものから，順に定めた個数を索引語として選択する．このアルゴリズムによる重要語の重要度 $t_i$ は，式(9)より求められる．

$$\begin{cases} t_i^+ = n_i^+ \times log \frac{M^+}{m_i^+} \\ t_i^- = \frac{M^+}{M^-} \times n_i^- \times log \frac{M^-}{m_i^-} \\ t_i = |t_i^+ - t_i^-| \end{cases} \tag{9}$$

なお，$M^+$，$M^-$ は，それぞれ正事例集合，負事例集合に含まれるマニュアル数，$n_i^+$，$n_i^-$ は，それぞれ正事例集合，負事例集合に含まれるマニュアルにおける該当する重要語の出現回数，$m_i^+$，$m_i^-$ は，それぞれ正事例集合，負事例集合に含まれるマニュアルにおける該当する重要語が含まれるマニュアル数である．

1つめのアルゴリズムは，正事例集合と負事例集合を区別せずに重要語の重要度を算出しているのに対して，2つめのアルゴリズムは，ある重要語の正事例集合と負事例集合における重要度の差を重要度としている．このことにより，より正事例集合と負事例集合を明確に分割することができる重要語の重要度が大きくなる．重要度の大きい重要語は，索引語として選ばれる確率が高くなる．このような索引語を属性として訓練事例を特徴ベクトル表現すれば，より精度の高いSVMを作成することが期待できる．

☆ Term Frequency × Inverse Document Frequency

### 5.3 特徴ベクトルの補完

5.2節で説明したアルゴリズムを用いて，索引語の選択を行う．この過程で，精度を向上させるために索引語の個数を増やそうとすると，0値をとる特徴ベクトルの属性もまた増加する．例えば，索引語を100個選択して特徴ベクトルを作成したときには，0値をとる属性は約90％であったが，索引語を1000個選択して特徴ベクトルを作成したときには，0値をとる属性は約96％にまで増加した．このように，0値をとる属性が増加すると，1値の影響よりも0値の影響が大きくなり，SVMの精度が低下する．そこで，重要語間の共起関係より属性値の補完を行う．

実際には，重要語間の共起関係よりベイジアンネットワークを構築して，特徴ベクトルの属性値の補完を行う．ベイジアンネットワークとは，変数間の依存関係を表現するデータ構造である．ベイジアンネットワークでは，いくつかの証拠変数の正確な値が与えられると，質問変数の集合の事後確率分布を推論する．本稿では，問題領域に応じたデータ構造をもつ領域知識を構築し，特徴ベクトルの属性値を補完している．ベイジアンネットワークは，以下の手順で各カテゴリごとに構築している．

(1) カテゴリに含まれるすべての重要語のTFIDFを計算し，TFIDFの値が最大である重要語をベイジアンネットワークのルートノードとする．

(2) ルートノードに保持された重要語と，その他の重要語との相互情報量 $log(p(x\&y)/(p(x)p(y)))$ を計算する．ただし，$p(x\&y)$ は，カテゴリ内で同じマニュアルに重要語 $x$ と $y$ が共起して出現する確率であり，$p(x)$ および $p(y)$ は，カテゴリ内でマニュアルに重要語 $x$ と $y$ が出現する確率である．相互情報量の値がしきい値以上である重要語をルートノードの子ノードとする．

(3) 子ノードとして保持された重要語を親ノードの重要語とする．手順(2)と同様にして，親ノードに保持されている重要語とその他の重要語の相互情報量を計算し，しきい値以上の値をとる重要語を子ノードとする．この操作を繰り返して行い，トップダウン的にベイジアンネットワークを構築していく．

以上の手順で，カテゴリのベイジアンネットワークが構築できる（図3）．

ベイジアンネットワークを用いて重要語の出現頻度を補完するということは，すなわち特徴ベクトルを補

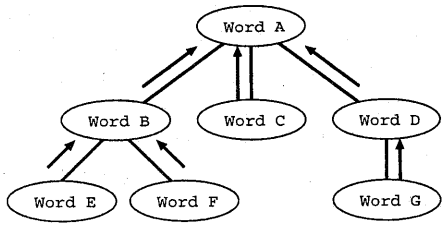


図3 Bayesian Network.

完することになる．出現頻度の補完は，ベイジアンネットワークの葉に保持される重要語から，ボトムアップ的に行なっている．親ノードの値が 0，かつ子ノードの出現頻度の平均がしきい値以上であるときに，親ノードの値をその平均値で補完している．例えば，図 3 の Word B の出現頻度が 0 である時，Word E と Word F の出現頻度の平均値がしきい値以上であれば，その平均値を Word B の出現頻度として補完している．

## 6. 実　　験

本実験では，120 種類の UNIX コマンドのマニュアルを，あらかじめ表1 の様に 6 つのカテゴリに分類して実験データとしている．

表1 Category of Manuals.

| Category | Manuals | Category | Manuals |
|---|---|---|---|
| Hard | 16 | Information | 15 |
| Output | 16 | Network | 18 |
| File | 28 | Others | 27 |

### 6.1 実　験 1

実験 1 では，5.1 節で説明した訓練事例のチューニングの実験を行う．SVM のカーネル関数には，式 (10) に示す 1 次多項式カーネル関数を用い，式 (3) におけるラグランジュ乗数 $\alpha_i$ の上限値は 10 としている．

$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}) + 1 \qquad (10)$$

訓練事例のカテゴリ付けには，分割数 10 の交差検定を用いている．索引語の選択には，5.2 節で説明した TFIDF を用いている．また，評価尺度には再現率（Recall）と適合率（Precision）の平均値を用いている．なお，再現率と適合率は，式 (11) より求められる．

$$\begin{aligned}\text{Recall} &= \frac{\text{retrieved relevant manuals}}{\text{all relevant manuals}} \\ \text{Precision} &= \frac{\text{retrieved relevant manuals}}{\text{all retrieved manuals}}\end{aligned} \qquad (11)$$

訓練事例のチューニングによる再現率と適合率の変化を図 4 に示す．また，訓練事例のカテゴリ変更の操作の順序を表 2 に示す．さらに，チューニング終了時の各々のカテゴリの様子を表 3 に示す．

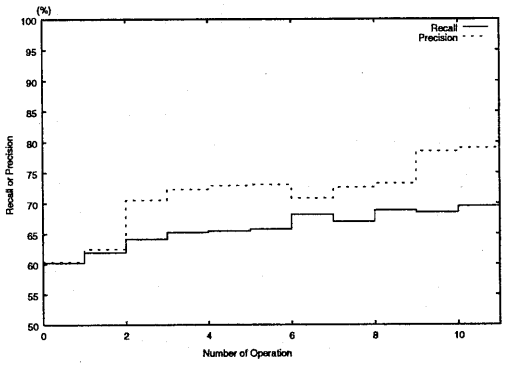


図4 Recall and Precision by Data Tuning.

訓練事例のチューニングにより，十分な再現率と適合率の向上が確認できた．しかしながら，評価尺度に訓練事例のカテゴリ付けを行なった者の意図は反映されないため，その者の意図と反して訓練事例がカテゴリ付けされるという操作が，more のカテゴリ変更の際にみうけられた．その他のカテゴリの変更は，納得できるものであった．

また，カテゴリ変更の行き先に Others が多く，カテゴリ付けに偏りができてしまった．この理由は，カテゴリに Others という，その他のカテゴリに比べて明確な基準をもたないカテゴリをもうけたためであると考えられる．

表2 Operation of Data Tuning.

| No. | Manual | From | To |
|---|---|---|---|
| 1 | ln | File | Others |
| 2 | lptest | Hard | Output |
| 3 | whatis | Information | Others |
| 4 | df | Hard | Others |
| 5 | hostname | Hard | Others |
| 6 | more* | Output | Others |
| 7 | mpstat | Hard | Others |
| 8 | touch | File | Others |
| 9 | du | Hard | File |
| 10 | uusend | File | Network |

### 6.2 実　験 2

実験 2 では，5.2 節で説明した索引語の選択と，5.3 節で説明した特徴ベクトルの補完の実験を行う．訓練

表3 Result of Data Tuning.

| Category | Manuals | Category | Manuals |
|---|---|---|---|
| Hard | 11 | Information | 14 |
| Output | 16 | Network | 19 |
| File | 26 | Others | 34 |

事例には，各々のアルゴリズムを用いてチューニングした結果を用いている．なお，実験1と同様に分割数10の交差検定を用い，評価尺度には，再現率（Recall）と適合率（Precision）を用いている．また，SVMのカーネル関数も実験1と同様，1次多項式カーネル関数を用いている．

5.2節で説明した前者のアルゴリズムをTFIDF-1，後者のアルゴリズムをTFIDF-2とする．索引語の選択の2つのアルゴリズム（TFIDF-1，TFIDF-2）と，ベイジアンネットワークによる特徴ベクトルの補完（Bayesian）の適用による再現率の変化を図5に，適合率の変化を図6に示す．

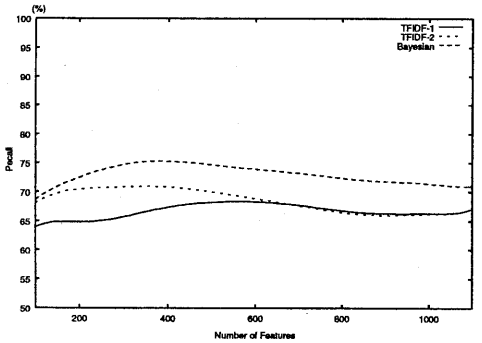


図5 Recall by 3 Methods.

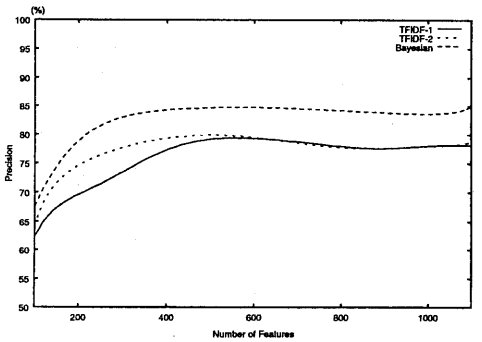


図6 Precision by 3 Methods.

索引語の選択される数が少ないときに，TFIDF-2はTFIDF-1と比較して高い精度が得られている．これは，TFIDF-1と比較して，TFIDF-2によってつけられた重要語の優先順位において，SVMの作成に適した重要語が上位に順位づけされていることを示している．このことから，良好な結果が得られたといえる．また，ベイジアンネットワークによる特徴ベクトルの補完を適用した結果，適用しなかった場合と比較して，再現率，適合率の双方で5％程度の精度の向上が確認できた．

## 7. おわりに

本稿では，SVMを用いたUNIXコマンドのマニュアルの分類を行った．この分類において最も重要となることは，訓練事例からより精度の高いSVM作成することである．そこで，まず訓練事例をチューニングすることにより，人手によって行われた訓練事例のカテゴリ付けのばらつきをおさえた．さらに，訓練事例の特徴ベクトルの属性となる索引語の選択と，ベイジアンネットワークによる属性値の補完によりSVMの識別精度の向上をめざした．

本稿では，SVMを用いたカテゴリへの分類手法に焦点をしぼって説明した．これらの手法を用いて，得られた分類木からUNIXコマンドのマニュアルをカテゴリに分類し，その結果をTextual CBRの事例ベースとして用いれば，ユーザはより効率的な事例検索を行うことができると思われる．

## 参考文献


1) R. D. Burke, et al., "Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System," Technical Report TR-97-05, Intelligent Information Laboratory, University of Chicago (1997).
2) 津田宏治, "可変カーネル関数を用いた Support Vector Machine," 信学技報, PRMU98-175, pp. 195–202 (1998).
3) J. T. Kwok, "Automated Text Categorization Using Support Vector Machine," Proc. of the International Conference on Neural Information Processing (ICONIP), pp. 347–351 (1998).
4) L. Y. Savio Lam and D. L. Lee, "Feature Reduction for Neural Network Based Text Categorization," Proc. of Sixth International Conference on Database Systems for Advanced Applicaitons, pp. 195–202 (1999).
5) M. Brown, C. Förtsch, and D. Wißmann, "Combining Information Retrieval and Case-Based Reasoning for „Middle Ground" Text Retrieval Problem," *AAAI Workshop on Textual Case-Based Reasoning, WS-98-12* pp. 3–7 (1998).