

異なる例からの素性の組合せを用いた ペアワイズ分類器の学習

Using Feature Conjunctions across Examples for Learning Pairwise Classifiers

小山 聡

Satoshi Oyama

京都大学大学院情報学研究科社会情報学専攻

Department of Social Informatics, Graduate School of Informatics, Kyoto University
oyama@kuis.kyoto-u.ac.jp, <http://www.dl.kuis.kyoto-u.ac.jp/~oyama/>

クリストファー D.
マニング

Christopher D. Manning

スタンフォード大学計算機科学科

Department of Computer Science, Stanford University

manning@cs.stanford.edu, <http://www-nlp.stanford.edu/~manning/>

keywords: kernel methods, support vector machines, learning similarity measures, entity matching

Summary

We propose a kernel method for using combinations of features across example pairs in learning pairwise classifiers. Pairwise classifiers, which identify whether two examples belong to the same class or not, are important components in duplicate detection, entity matching, and other clustering applications. Existing methods for learning pairwise classifiers from labeled training data are based on string edit distance or common features between two examples. However, if two examples from the same class have few common features, these methods have difficulties in finding these pairs and achieving high recall. One typical example is to check whether two abbreviated author names in different citations refer to the same person or not. Since similarities between examples from the same class become close to zero, classifiers fail to distinguish positive pairs from negative pairs. One approach to avoiding the problem of zero similarities is using conjunctions of different features across examples, but implementing this idea straightforwardly makes the computational cost prohibitive for practical problems. Using a kernel on pair instances, our method can use feature conjunctions across examples without actually doing feature mappings, which are computationally expensive. The kernel is a tensor product of two inner products on the original feature space. The corresponding feature mapping generates conjunctions of features only across the two different examples while that of the conventional polynomial kernel also generates conjunctions of features from the same example, which are irrelevant to pairwise classification and cause deterioration of accuracy. Our experiments on the author matching problem show that this method can give a precision 4 to 8 times higher than that of previous methods at medium recall levels.

1. はじめに

与えられた二つの例が同じクラスに属するか否かを判定するペアワイズ分類 (pairwise classification) は、データベースにおける重複発見やエンティティのマッチング、また、より一般的なクラスタリング問題においても、重要な要素技術である。例えば、引用マッチング (citation matching) [Lawrence 99] は、図 1 の様に二つの引用を比較し、同じ論文を参照しているか否かを決定する問題である。

従来は、これらの分類は固定もしくは人手で調整された、二つの例の間の距離尺度に基づいて行われてきた。近年、機械学習の技術を用いて、人手でラベル付けされた訓練集合から、与えられた二つの例が同じクラスに属する

か属さないかを分類する分類器を学習する試みが幾つか行われてきている [Bilenko 03a, Bilenko 03b, Sarawagi 02, Tejada 02]。これらの手法の多くは、二つの例の間の文字列編集距離 (string edit distance) や共通な素性 (feature) に基づいている。ここで共通の素性とは、例えば二つの例に同じ単語が現れるか否かという情報である*1。

これらの手法は、同じ論文への二つの引用を発見する引用マッチング問題の様に、同じクラスに属する二つの例が共通の素性 (共通の単語等) を多く持つ場合には、効果的な方法である。しかしながら、同じクラスの二つ

*1 feature は特徴とも訳される。また、属性 (attribute) は対象に本来備わっている性質であり、素性は複数の属性を組み合わせる等の処理を行ったものという区別がされる場合もあるが、本論文では全てを素性という用語で統一する。

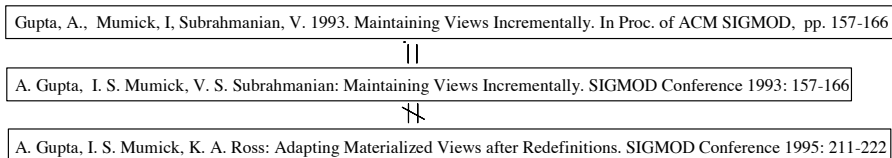


図 1 引用のマッチング

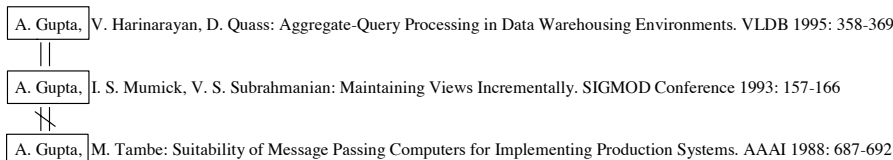


図 2 著者のマッチング

の例が共通の素性を僅かしか持たない場合には、従来の手法ではこれらの例の組を同じクラスであると判別することが出来ず、十分な再現率を実現することが困難である。一つの例として、以下に述べる異なる引用に現れる同じ著者を判別する問題（著者のマッチング）がある。

引用においては多くの場合、著者のファーストネームはイニシャルに省略される。図 2 の様に、省略された著者名が同じ人物を指しているか否かを判別することは、引用解析や研究者の評価を行う際に重要である。しかしながら、代表的な引用データベースである ISI の Citation Index^{*2}や CiteSeer の著者別の引用数のページ（Most Cited Authors in Computer Science）^{*3} おいても、共通のファーストネームのイニシャルとラストネームをもつ著者は、全て混同されてしまっている。これらの著者を区別することは、引用データベースの中で著者を独立のエンティティとして扱い、著者毎に研究評価を行うためには必須である。

著者のマッチングは、引用のマッチングよりもある意味で困難な問題である。図 1 の例の様に、引用のマッチングにおいては、同じ論文への引用は多くの共通の単語を持っている。引用を表す二つの文字列が一致していたり、多くの単語を共通に持つ場合には、これらの引用は同じ論文を指していると推測することができる。しかしながら、著者のマッチングの場合、たとえ著者を表す二つの文字列が完全に一致していたとしても、実世界の同じ人物を参照しているとは限らない。それ故、著者名の曖昧性を解消するためには、引用の中の著者名以外の他の情報を参照することが必要である。

また、図 2 の著者のマッチングの例において、最初の二つの文献の第一著者（A. Gupta）は同じ人物であるが、二つの例はこの著者名の他に共通の単語を一つも持たない。人間は、二つの論文のタイトルの表す研究分野の類似性や、二つの会議の研究コミュニティの類似性に基づいて、この二人の A. Gupta が同じ人物であると推測することができる。しかしながら、従来の文字列の編集距離や共通素性に基づく方法では、この二つの例は類似度

がゼロになり直交するため、同じく類似度がゼロとなる、異なったクラスからの例の組（例えば、図 2 の 2 番目と 3 番目の論文の組）と区別することができない。（ここでは、二つの“A. Gupta”という省略名が同じ人物に対応するか否かを判定する問題を考えているので、“A. Gupta”という文字列自体は素性としては考えていない。）それ故、共通の素性を僅かしか持たないが、同じエンティティを指している例を判別することは、従来の手法では困難であった。

この問題を解決するための一つの方法は、組を構成する異なる例からの同じ素性の組合せだけではなく、異なる例からの異なる素性の組み合わせも素性として学習アルゴリズムに与えることである。著者のマッチングの問題においては、分類器は二つの例に現れる共通の単語だけではなく、二つの例に現れる異なった単語の組にも重みを与え、全体での類似度を計算する。これによって、直交性を破り、ゼロ類似度の問題を回避することができる。例えば、ラベル付けされた正例（同じ著者による論文の組）の中に、片方に“SIGMOD”という単語と、もう一方に“VLDB”という単語を含む例が多く現れれば、学習アルゴリズムがこの二つの単語の組に重みを与え、分類に利用することが期待される。しかしながら、単純に n 個全ての素性の組を用いると、素性空間の次元が n^2 となり計算の負荷が大きくなるため、素性の数の大きな実問題への適用が困難になる。

そこで本研究では、カーネル法を用いることで、計算の負荷を大きくすることなく、異なる例からの素性の組をペアワイズ分類器の学習に用いる方法を提案する。我々の手法は、異なる例からの素性の組合せの空間へ仮想的に写像を行うことで、従来の共通の素性だけにに基づく方法では困難であった問題に対しても、高い性能を上げることが可能となる。

以下では、2 章でペアワイズ分類を定式化し、既存の手法の問題点について述べる。3 章で異なる例からの素性の組合せを用いるためのカーネル法を提案し、4 章で著者のマッチングと引用のマッチングの問題における評価実験の結果を示す。5 章で関連研究について述べ、6 章で

*2 <http://isiknowledge.com/>

*3 <http://citeseer.ist.psu.edu/mostcited.html>

今後の研究課題を挙げる．7章で本論文の結論を述べる．

2. ペアワイズ分類

2.1 問題の定式化

本論文では、ペアワイズ分類を、例の組 $\mathbf{x}^\alpha$ と $\mathbf{x}^\beta$ (以降では、**ペアインスタンス** (pair instance) と呼ぶ) が与えられた時に、二つの例が同じクラスに属するか否かを判別する問題と定義する。(以降では、上付きの添え字で例を表し、下付きの添え字で例の素性を表す。) すなわち、二値分類の場合には、

$$f(\mathbf{x}^\alpha, \mathbf{x}^\beta) = \begin{cases} 1 & (\mathbf{x}^\alpha \text{ と } \mathbf{x}^\beta \text{ が同じクラスに属する場合}), \\ -1 & (\text{それ以外の場合}). \end{cases}$$

を充たす様な分類器 f を構築することが問題となる。

ペアワイズ分類器と類似度は関連が深い．分類器が $f(\mathbf{x}^\alpha, \mathbf{x}^\beta) \in [0, 1]$ の形で、類似度に対応する連続値を出力する問題も考えることができる．連続値の分類器は、閾値を用いることで二値の分類器に変換可能であるし、多くの二値分類器も、容易に連続値を出力する形に変換できる [Bilenko 03b]．それ故、我々は以下でペアワイズな分類器とペアワイズな類似度を区別せずに用いる場合がある．

ペアワイズ分類はデータベースにおける重複発見やエンティティのマッチング、更にはより広い意味でのクラスタリングにおいても重要な要素技術である．ペアインスタンスが同じクラスに属するか否かがペアワイズな分類器を用いて決定されれば、同じクラスに属すると判定された例の推移閉包を取ることによって、大域的なクラスタリングを決定することが可能である [Bilenko 03b]．

2.2 元のデータからのペアインスタンスの作成

精度の高い分類器を、固定もしくは人手で調整された類似度で実現することは困難な問題である．そこで、機械学習を用いてこの様な分類器を実現する試みが近年幾つか行われて来ている．多くの場合には、まず母集団から二つの例 $\mathbf{x}^\alpha$ と $\mathbf{x}^\beta$ をサンプルしてペアインスタンスを作成し、それらが同一のクラスに属するかを人手で判別してラベル y を与える．(例えば、二つの例が同じクラスに属する場合 $y = 1$ とし、異なるクラスに属する場合 $y = -1$ とする．) そして、ペアインスタンスとラベルの組 $(\mathbf{x}^\alpha, \mathbf{x}^\beta, y)$ を訓練例として扱い、サポートベクターマシン [Vapnik 99] 等の二値分類器の学習アルゴリズムを適用する．

例えば Bilenko と Mooney の方法 [Bilenko 03b] では、文献の例を $\mathbf{x}^\alpha = (x_1^\alpha, x_2^\alpha, \dots, x_n^\alpha)$ の形の素性のベクトルで表現している．ここで、各素性は、単語が引用の文字列中に現れるか否かに対応し、ベクトルの次元 n は訓練集合の語彙の数である．このとき、二つの例 $\mathbf{x}^\alpha =$

$(x_1^\alpha, x_2^\alpha, \dots, x_n^\alpha)$ および $\mathbf{x}^\beta = (x_1^\beta, x_2^\beta, \dots, x_n^\beta)$ から、 n 次元の新たな特徴ベクトル

$$\hat{\mathbf{x}}_{common} = (x_1^\alpha x_1^\beta, x_2^\alpha x_2^\beta, \dots, x_n^\alpha x_n^\beta) \quad (1)$$

を作成し、これでペアインスタンス $\hat{\mathbf{x}} = (\mathbf{x}^\alpha, \mathbf{x}^\beta)$ を表現している．そして、これにラベル y を与えた上で、SVM を適用している．(彼らは $|\mathbf{x}^\alpha| |\mathbf{x}^\beta|$ で各素性を割ることで規格化も行っている．)

この手法は、同じ論文への二つの引用の様に、同じクラスに属する二つの例が共通の素性を多く持つ場合には適した方法である．問題点は、二つの例で共に現れる素性が少ない場合に、十分な精度を達成することが出来ないことである．図2の著者のマッチングの例において、最初の二つの論文は同一の A. Gupta によって書かれているにもかかわらず、A. Gupta の他に共通の単語がない．それ故、式 (1) によるこのペアインスタンスの表現はゼロベクトルとなる．この現象は同じ著者に書かれた論文の間で稀ではない．共通の素性に基づく方法は、これらと同じくゼロベクトルを表現として持つ多くの負例 (例えば、別々の A. Gupta で書かれた論文である図2の2番目と3番目の論文の組からなるペアインスタンスのベクトル表現) と区別することができない．

これに対して考えられる一つの方策は、異なる元の例からの異なる素性の組 $x_i^\alpha x_j^\beta$ も利用して、

$$\begin{aligned} \hat{\mathbf{x}}_{Cartesian} = & (x_1^\alpha x_1^\beta, \dots, x_1^\alpha x_n^\beta, \\ & x_2^\alpha x_1^\beta, \dots, x_2^\alpha x_n^\beta, \\ & \dots \\ & x_n^\alpha x_1^\beta, \dots, x_n^\alpha x_n^\beta) \end{aligned} \quad (2)$$

の形でペアインスタンスを表現することである．即ち、新しい素性の集合は、 $\mathbf{x}^\alpha$ の素性の集合と $\mathbf{x}^\beta$ の素性の集合のデカルト積 $\{x_i^\alpha x_j^\beta | i = 1, \dots, n; j = 1, \dots, n\}$ であり、素性空間は n^2 次元である．この表現では、ペアインスタンスの素性ベクトルがゼロベクトルになるのは、元の例の少なくとも一方がゼロベクトルになる (素性が一つも成立しない) 場合だけであり、これは通常は起こらない．これにより、上で述べたゼロベクトルの問題を回避できる．また、片方の例に“VLDB”，もう一方に“SIGMOD”という単語が現れるペアインスタンスが正例に頻繁に現れれば、学習アルゴリズムがこの素性の組合せに重みを置いた分類器を生成することで、分類の精度が向上することが期待される．

しかしながら、この方法を直接実現するには、以下の問題がある．一つ目として、素性空間の次元が n^2 となり、計算量が大きくなるため、実問題に適用することが困難になる．二つ目として、次元の呪いと呼ばれる、素性空間の次元が大きくなることで、過学習 (over-fitting) に陥る危険性が大きくなるという問題がある．

3. 異なる例からの素性の組合せを用いるためのカーネル法

3.1 カーネル法

サポートベクターマシン, パーセプトロン, ロジスティック回帰モデル (最大エントロピーモデル) 等, 幾つかの学習アルゴリズムは, その中で例が単独では現れずに常に二つの例 $\mathbf{x}$ と $\mathbf{z}$ の内積 $\langle \mathbf{x} \cdot \mathbf{z} \rangle$ として現れる形に書ける [Schölkopf 02, Shawe-Taylor 04]. カーネル法とは, これらのアルゴリズムにおいて, 例の間の内積 $\langle \mathbf{x} \cdot \mathbf{z} \rangle$ が現れる箇所をカーネル関数 $K(\mathbf{x}, \mathbf{z})$ で置き換えることで, より高次元空間での分類学習を可能とする手法である.

例えば,

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x} \cdot \mathbf{z} \rangle^2 \quad (3)$$

というカーネル関数を考える. このカーネル関数を用いて学習を行うことは,

$$\begin{aligned} \phi(\mathbf{x}) = & (x_1 x_1, \dots, x_1 x_n, \\ & x_2 x_1, \dots, x_2 x_n, \\ & \dots \\ & x_n x_1, \dots, x_n x_n) \end{aligned}$$

の様に, 高次元の空間に例の素性ベクトルを写像して, 学習アルゴリズムを適用することと等価である. このことは, 以下の様に示すことができる.

$$\begin{aligned} \langle \mathbf{x} \cdot \mathbf{z} \rangle^2 &= \left(\sum_{i=1}^n x_i z_i \right) \left(\sum_{j=1}^n x_j z_j \right) \\ &= \sum_{i=1}^n \sum_{j=1}^n x_i z_i x_j z_j \\ &= \sum_{i=1}^n \sum_{j=1}^n (x_i x_j) (z_i z_j) \\ &= \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle. \end{aligned}$$

式 (3) のカーネルは, 2 次の多項式カーネルと呼ばれる. 他にも無限次元の素性空間への写像に対応する

$$K(\mathbf{x}, \mathbf{z}) = \exp \left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2} \right) \quad (4)$$

のガウシアンカーネルもしばしば用いられる.

カーネルを用いることで, 計算量のかかる高次元の素性空間への写像と, 写像された高次元のベクトル同士の内積を実際には計算せずに, 高次元の素性空間での分類学習を行うことが可能となる. さらに, カーネルを用いた SVM は高次元空間での学習にもかかわらず, 過学習の問題に対して頑健であることが知られている.

3.2 カーネルによる異なる例からの素性の組合せの実現

異なる例からの素性を組み合わせる単純な方法の 1 つは, 上で紹介した多項式カーネルを用いることである. ペアインスタンスを

$$\hat{\mathbf{x}} = (\mathbf{x}^\alpha, \mathbf{x}^\beta) = (x_1^\alpha, \dots, x_n^\alpha, x_1^\beta, \dots, x_n^\beta) \quad (5)$$

の形の $2n$ 次元のベクトルで表し, 式 (3) のカーネルを適用することで, 仮想的な写像で生成される素性の集合は, $\{x_i^\alpha x_j^\alpha\} \cup \{x_i^\alpha x_j^\beta\} \cup \{x_i^\beta x_j^\alpha\} \cup \{x_i^\beta x_j^\beta\}$ となる. これは, 式 (2) における素性の集合 $\{x_i^\alpha x_j^\beta\}$ を含んでいる. しかし, それ以外にも, 単一の例からの素性同士の組み合わせ $\{x_i^\alpha x_j^\alpha\}$ および $\{x_i^\beta x_j^\beta\}$ も含んでしまっている. これらの素性は一方の例のみに依存するという点で, 二つの例が同じクラスか否かを判別するペアワイズ分類には明らかに不適切である. 正例と負例の中で, 元の例の出現頻度に差がある場合, 学習アルゴリズムがこれらの例で成立する素性を結果の分類器で重視すると, 汎化能力の低下を招く恐れがある.

我々が必要とするのは, ペアインスタンスを構成する二つの異なる例からの素性の組合せだけを生成する

$$\begin{aligned} \phi(\hat{\mathbf{x}}) &= \phi((\mathbf{x}^\alpha, \mathbf{x}^\beta)) \\ &= (x_1^\alpha x_1^\beta, \dots, x_1^\alpha x_n^\beta, \\ &\quad x_2^\alpha x_1^\beta, \dots, x_2^\alpha x_n^\beta, \\ &\quad \dots \\ &\quad x_n^\alpha x_1^\beta, \dots, x_n^\alpha x_n^\beta) \end{aligned} \quad (6)$$

の写像である. 我々は, この写像を実現する以下のカーネルを提案する.

$$\begin{aligned} K(\hat{\mathbf{x}}, \hat{\mathbf{z}}) &= K((\mathbf{x}^\alpha, \mathbf{x}^\beta), (\mathbf{z}^\alpha, \mathbf{z}^\beta)) \\ &= \langle \mathbf{x}^\alpha \cdot \mathbf{z}^\alpha \rangle \langle \mathbf{x}^\beta \cdot \mathbf{z}^\beta \rangle. \end{aligned} \quad (7)$$

このカーネルは, まず $\mathbf{x}^\alpha$ と $\mathbf{z}^\alpha$ および $\mathbf{x}^\beta$ と $\mathbf{z}^\beta$ 同士の内積を計算し, その後でこれらの内積のスカラー積を計算する.

このカーネルが式 (6) の写像を実現することは, 以下の様にして示すことができる.

$$\begin{aligned} \langle \mathbf{x}^\alpha \cdot \mathbf{z}^\alpha \rangle \langle \mathbf{x}^\beta \cdot \mathbf{z}^\beta \rangle &= \left(\sum_{i=1}^n x_i^\alpha z_i^\alpha \right) \left(\sum_{j=1}^n x_j^\beta z_j^\beta \right) \\ &= \sum_{i=1}^n \sum_{j=1}^n x_i^\alpha z_i^\alpha x_j^\beta z_j^\beta \\ &= \sum_{i=1}^n \sum_{j=1}^n (x_i^\alpha x_j^\beta) (z_i^\alpha z_j^\beta) \\ &= \langle \phi((\mathbf{x}^\alpha, \mathbf{x}^\beta)) \cdot \phi((\mathbf{z}^\alpha, \mathbf{z}^\beta)) \rangle \\ &= \langle \phi(\hat{\mathbf{x}}) \cdot \phi(\hat{\mathbf{z}}) \rangle. \end{aligned}$$

このカーネルは, 二つの元の例の上でのナイーブカーネル (単純な内積) の間のテンソル積 [Haussler 99] である. このカーネルの直感的な解釈は以下の通りである. カーネ

ル関数は、入力空間の例の間の類似度（内積）を定義する。我々の場合は、二つのペアインスタンス $\hat{x} = (x^\alpha, x^\beta)$ と $\hat{z} = (z^\alpha, z^\beta)$ の間の類似度である。

例えば SVM において、 $\{(\hat{x}^{(1)}, y^{(1)}), \dots, (\hat{x}^{(m)}, y^{(m)})\}$ をラベル付きの訓練例とし、これを用いて学習された重みを $\{\lambda_1, \dots, \lambda_m; \gamma\}$ とすると、新しい例 $\hat{x}$ は

$$y = g\left(\sum_{l=1}^m \lambda_l K(\hat{x}^{(l)}, \hat{x}) y^{(l)} + \gamma\right)$$

に従って分類される。（ $g(u)$ は $u \geq 0$ の時 $+1$ 、 $u < 0$ の時に -1 を取る関数である。）これは、新しい例 $\hat{x}$ とラベルが既知の例 $\hat{x}^{(l)}$ の間のカーネルにより計算される類似度 $K(\hat{x}^{(l)}, \hat{x})$ に学習された重み λ_l を掛けた値で、既知のラベル $y^{(l)}$ の加重平均を取る（更に、バイアス γ を加える）ことで、未知のラベル y を決定すると解釈できる。

式 (7) のカーネルは、ペアインスタンスを構成する対応する（上付の添字が同じ）元の例同士の類似度が両方とも高い（内積が大きい）場合のみ全体としての類似度が高くなる。例えば、 x^α と z^α の内積がゼロであれば、 x^β と z^β の内積が大きな値であっても、式 (7) の全体としての類似度はゼロになる。ペアワイズ分類は片方の例だけではなく、両方の例に基づいてなされるべきであるので、この類似度の定義は自然である。

著者マッチングの問題において、提案手法のカーネルを用いた分類器は、著者が省略名で与えられた 2 つの引用（これらの直接の内積はゼロになる場合がある）の間の著者の一致に関する類似度を、著者のフルネームが既知で（同姓同名を除けば）著者の一致が明らかな訓練集合内の引用の組と比較をすることにより計算する。テスト例の引用の組と各々の引用が類似した（内積の大きな）ペアインスタンスが訓練集合に存在すれば、カーネル関数の値が大きくなり、そのペアインスタンスの（著者の一致を表す）ラベルがテスト例の引用間の類似度の計算において重視される。すなわち、提案手法は、引用間の大雑把な類似度（内積）と著者の一致を表す二値のラベルから、引用間での著者の一致に関するより正確な類似度を計算することに対応する。この類似度を用いることで、未知の著者による論文や多くの訓練例が集められない著者の論文への引用に関しても、他の著者による論文と区別することが可能になるという利点がある。

3.3 ペアインスタンスの構成順序に関する対称性

注意したいのは、式 (6) の写像および式 (7) のカーネル関数が、ペアインスタンス内の例の順序に依存することである。即ち、 $\phi((x^\alpha, x^\beta)) \neq \phi((x^\beta, x^\alpha))$ であり、 $K((x^\alpha, x^\beta), (z^\alpha, z^\beta)) \neq K((x^\alpha, x^\beta), (z^\beta, z^\alpha))$ である。よって、学習された分類器を f とした時、新しい例の分類結果が $f((x^\alpha, x^\beta)) \neq f((x^\beta, x^\alpha))$ の様に、ペアインスタンスの構成順序によって異なることが考えられる。ただし、訓練集合において、 (x^α, x^β) と (x^β, x^α) の両方を

用いて SVM の学習を行えば、分類結果はペアインスタンスの順序に依存しないことを示すことができる（付録 A）。

また、ペアインスタンス同士の掛け算を行う別の方法として、 $\langle x^\alpha \cdot z^\beta \rangle \langle x^\beta \cdot z^\alpha \rangle$ の様に、異なった上付の添字同士の例を掛け合わせる方法も考えられ、これを用いてカーネルを

$$\begin{aligned} K(\hat{x}, \hat{z}) &= K((x^\alpha, x^\beta), (z^\alpha, z^\beta)) \\ &= \frac{1}{2} (\langle x^\alpha \cdot z^\alpha \rangle \langle x^\beta \cdot z^\beta \rangle + \langle x^\alpha \cdot z^\beta \rangle \langle x^\beta \cdot z^\alpha \rangle) \end{aligned}$$

と対称化できる。このカーネルはペアインスタンスの順序に対して $K((x^\alpha, x^\beta), (z^\alpha, z^\beta)) = K((x^\alpha, x^\beta), (z^\beta, z^\alpha))$ と対称になる。実は、このカーネルの対称化は上で述べた訓練集合の対称化と等価であることを示すことができる（付録 B）。

4. 評価実験

4.1 データセットとプログラム

本実験では 2 種類のデータセットを用いた。1 つは、400,000 件以上の計算機科学の文献目録である DBLP*4 から取得した。DBLP のデータは XML ファイルでダウンロード可能であり、個々の論文の情報は、XML のタグによって、author, journal 等のフィールドに分割されている。我々は、この中から、ジャーナル論文と国際会議論文を実験に用いた。DBLP データは人手で入力されており、多くの著者名はフルネームで与えられている。我々はフルネームで与えられている著者名のファーストネームをイニシャルに変換し、ミドルネームを除くことで、訓練集合およびテスト集合を作成し、省略された二つの著者名が同じ人物を指すか否かを判定する著者のマッチングの実験に用いた。（この実験では、フルネームが一致している著者は同一の人物であると想定している。）素性としては、共著者の名前、ジャーナル名、国際会議名、およびタイトルに含まれる単語を用いている。

もう 1 つは、Andrew McCallum が提供している Cora Citation Matching データセット*5であり、引用マッチングの実験に用いた。この中には、計算機科学論文の検索エンジン Cora から採られた 191 本の論文への 1,879 件の引用が含まれている。我々は、引用にそれぞれの単語が含まれるか否かを、素性として定義した。このデータセットは、先行研究 [Bilenko 03b, Bilenko 03c] においても用いられている。

我々は、Joachims によって開発された SVM^{light} [Joachims 99] を実験に用いた。SVM^{light} では、多項式カーネル、ガウシアンカーネル等の基本的なカーネルの他に、ユーザが定義したカーネルを用いることが可能である。我々は、式 (7) のカーネルを実装し、実験に用いた。カーネ

*4 <http://dblp.uni-trier.de/>

*5 <http://www.cs.umass.edu/~mccallum/code-data.html>

表 1 DBLP から収集したデータセット

Abbreviated Name	Number of Papers	Number of Distinct Authors
D. Johnson	222	26
A. Gupta	401	39
J. Smith	345	56
R. Johnson	122	29

ルを指定するパラメータ以外の SVM^{light} のパラメータは、全てデフォルト値を用いた。

4.2 実験手法

著者のマッチング問題においては、CiteSeer の計算機科学分野で最も多く引用された著者のリスト上位 20 人*6)の中から、多くの異なった著者が区別されずに一つの項目になってしまっている例を 4 つ選択した。(すなわち、*J. Ullman* ではなく *J. Smith* の様な名前を選択した。)それぞれの省略名に対する訓練集合とテスト集合を作成するために、同じファーストネームのイニシャルとラストネームとを持つ著者が書いた論文を DBLP データから検索した。表 1 に省略名毎の論文数および異なった著者(フルネーム)の数を示した。

このようにして収集した全ての論文の組合せをペアインスタンスとして生成すると、同じ著者によって書かれた論文の組よりも、別の著者によって書かれた論文の組の数が大きい。正例よりも負例の数が遥かに大きくなる。正例と負例の数の違いが分類精度に与える影響を調べるため、我々は 2 種類のデータセットを用意した。一つは不均一なデータセット (imbalanced data sets) であり、全ての論文の組を例として生成した。もう一つは均一なデータセット (balanced data sets) であり、まず同じクラスに属する例同士の組を全て作成して正例を作成し、次に異なったクラスに属する例の組をランダムにサンプルすることで正例と同数の負例を作成した。均一なデータセットと不均一なデータセットそれぞれにおいて、2 分割交差検定を行った。2 分割交差検定を用いた理由は、文献 [Bilenko 03b] にも述べられているように、分割数を多くし過ぎると分割内の同じ著者の数が少なくなり過ぎるからである。また、学習される分類器がペアインスタンス内の例の順序に依存しないようにするために、訓練集合において、例の順序を反転したペアインスタンスを両方用いた。表 2 に各分割でのデータセットの大きさを示す。

テスト集合において、二つの例が同じクラスに属するか否かを分類器が判定した結果と、実際にそれらの例が同じクラスから採られたものか否かを表すラベルを比較し、適合率と再現率を評価した。文献 [Bilenko 03b] においては、同じクラスに属すると判定された例の組の推測閉包を計算し、その上で適合率と再現率を評価している。

表 2 データセットの大きさ

Abbreviated Name	Balanced Data		Imbalanced Data	
	Fold 1	Fold 2	Fold 1	Fold 2
D. Johnson	5160	4944	12210	12210
A. Gupta	7556	7648	40200	39800
J. Smith	5504	5292	29756	29412
R. Johnson	760	756	3660	3660

我々はペアワイズ分類の精度を検証することが目的のため、単純に個々のペアインスタンスの分類結果だけに基づいて適合率、再現率を評価した。文献 [Bilenko 03c] で主張されている様に、エンティティのマッチングにおいては、適合率と再現率のどちらかを重視するかは状況によって異なるため、我々は SVM による分類結果の適合率と再現率だけではなく、適合率-再現率曲線でも評価を行っている。そのために、SVM の決定境界を平行移動し、異なった 20 の再現率レベルでの適合率を求めている。

我々は、以下に述べる異なったカーネルを用いて学習した分類器の性能を比較した。

Common 式 (1) の共通の素性のみに基づくペアインスタンスのベクトル表現に、式 (4) で $\sigma = 10$ としたガウシアンカーネルを適用したもの*7。

Polynomial 式 (5) によるペアインスタンスの素性ベクトル表現に、式 (3) の多項式カーネルを用いたもの。

Cartesian 我々が提案した式 (7) のカーネルを用いたもの。

4.3 実験結果

表 3 および図 3 に、著者のマッチング問題における均一なデータセットの場合の結果を示す。低い再現率の領域では、**Common** も高い適合率を示しているが、再現率がある値を超えると、急激に適合率が低下する。これは、式 (1) で表現したペアインスタンスは多くがゼロベクトルもしくはそれに近いベクトルになり、この中では正例と負例を区別することができないためと考えられる。それに対して、**Polynomial** と **Cartesian** は、異なる例からの異なる素性の組を用いることで、ゼロベクトルの問題を回避し、高い再現率の領域でも適合率を維持することが可能となっている。これらの二つのカーネルの中では、一般に **Cartesian** が **Polynomial** よりも高い適合率を達成している。表 4 および図 4 に、不均一なデータセットの場合を示す。均一なデータセットの場合と同様に、ある程度以上の再現率の領域では、異なる素性の組合せを用いた方法が共通の素性を用いた方法に大きく勝っており、特に中程度の再現率の領域では **Cartesian** は **Common** に対して 4-8 倍高い適合率を達成している。

*7 このパラメータの値は、文献 [Bilenko 03b]) に基づいている。実際には、 σ の値を変化させても、分類性能は大きく変化しなかった。これは式 (1) の形でペアインスタンスを表現した時点で、分類に必要な情報が失われてしまっているからではな

*6 <http://citeseer.ist.psu.edu/mostcited.html>

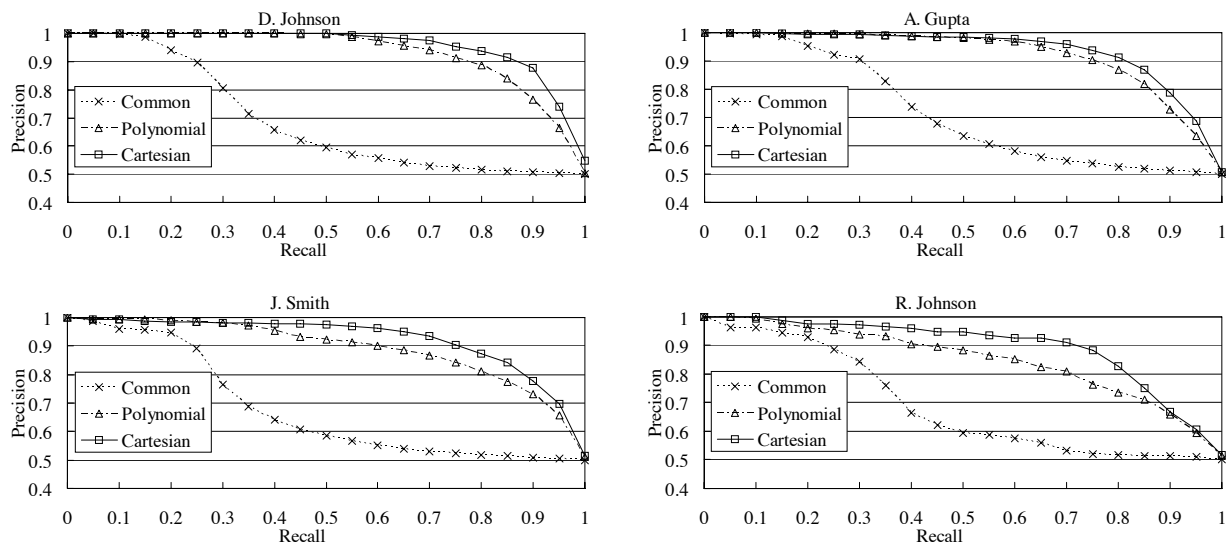


図 3 正例と負例の数が均一なデータセットを用いた場合の著者マッチング問題の適合率 - 再現率曲線

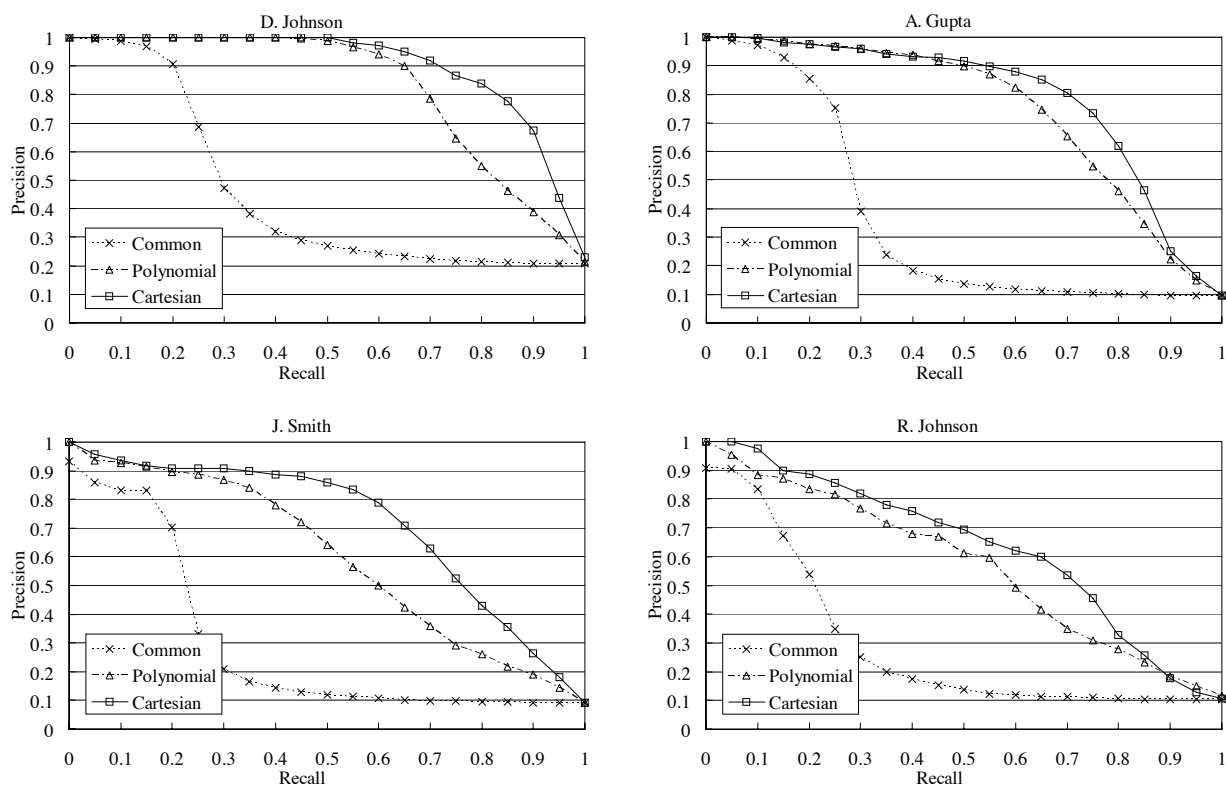


図 4 正例と負例の数が不均一なデータセットを用いた場合の著者マッチング問題の適合率 - 再現率曲線

表 3 正例と負例の数が均一なデータセットを用いた場合の著者マッチング問題の適合率 (P) と再現率 (R) (学習された SVM を二値分類器としてそのまま用いた.)

Abbreviated Name	Common		Polynomial		Cartesian	
	P	R	P	R	P	R
D. Johnson	0.93	0.23	0.99	0.52	0.99	0.60
A. Gupta	0.91	0.28	0.99	0.38	0.98	0.55
J. Smith	0.95	0.20	0.97	0.36	0.98	0.44
R. Johnson	0.93	0.18	0.97	0.20	0.97	0.35

表 4 正例と負例の数が不均一なデータセットを用いた場合の著者マッチング問題の適合率 (P) と再現率 (R) (学習された SVM を二値分類器としてそのまま用いた.)

Abbreviated Name	Common		Polynomial		Cartesian	
	P	R	P	R	P	R
D. Johnson	0.88	0.21	1.00	0.44	1.00	0.53
A. Gupta	0.83	0.23	0.97	0.22	0.94	0.40
J. Smith	0.82	0.18	0.91	0.11	0.89	0.21
R. Johnson	0.67	0.13	1.00	0.02	0.96	0.03

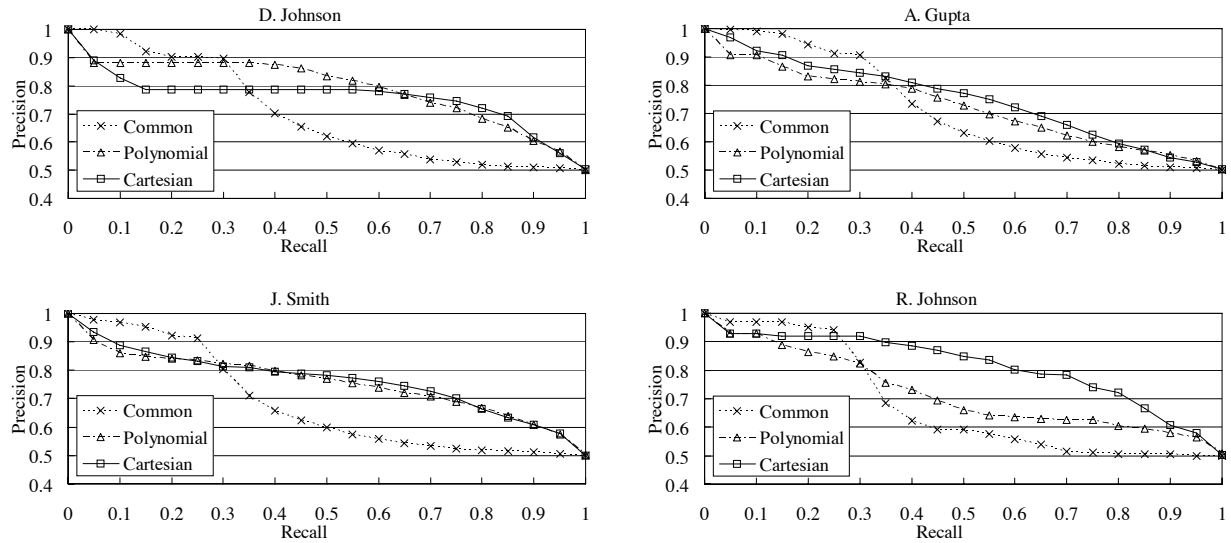


図 5 汎用の分類器を用いた場合の著者マッチング問題の適合率 - 再現率曲線 (均一なテスト集合の場合)

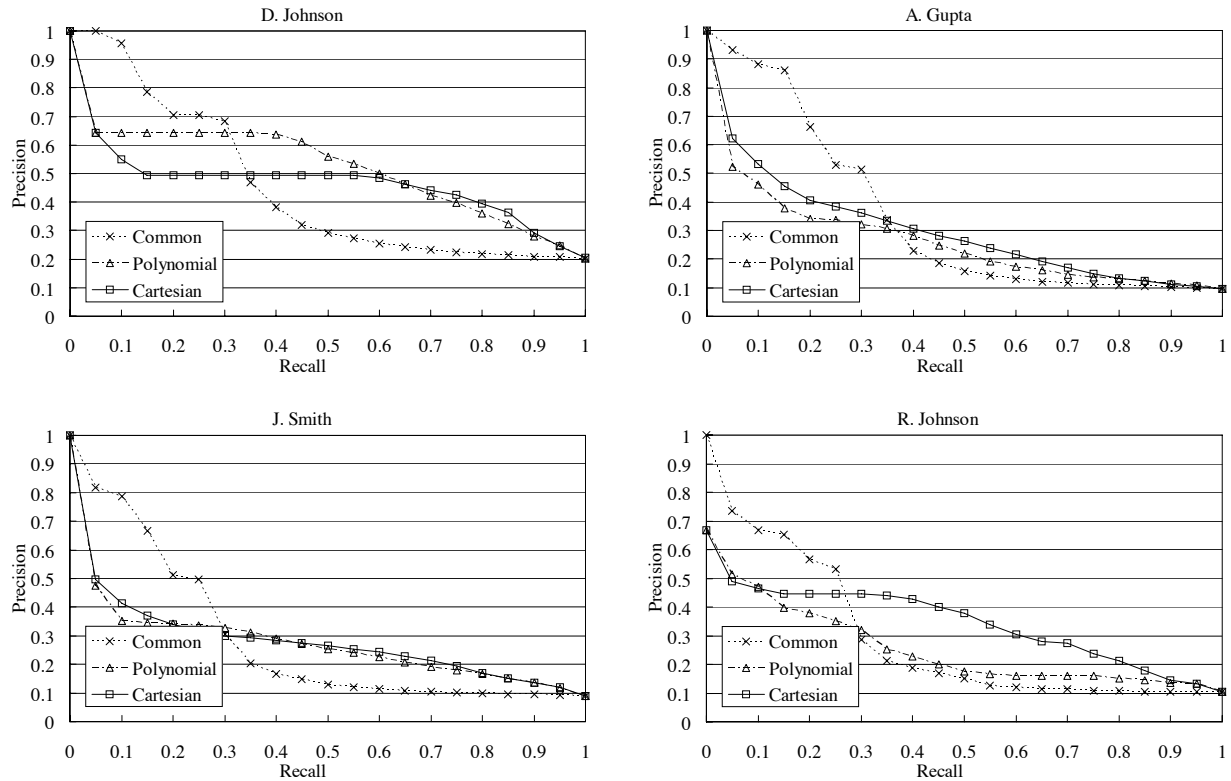


図 6 汎用の分類器を用いた場合の著者マッチング問題の適合率 - 再現率曲線 (不均一なテスト集合の場合)

表 5 汎用の分類器を用いた場合の著者マッチング問題の適合率 (P) と再現率 (R) (学習された SVM を二値分類器としてそのまま用いた.) (均一なテスト集合の場合)

Abbreviated Name	Common		Polynomial		Cartesian	
	P	R	P	R	P	R
D. Johnson	0.90	0.30	0.87	0.40	0.74	0.24
A. Gupta	0.91	0.32	0.81	0.31	0.81	0.39
J. Smith	0.91	0.26	0.84	0.25	0.82	0.28
R. Johnson	0.95	0.25	0.87	0.19	0.91	0.31

表 6 汎用の分類器を用いた場合の著者マッチング問題の適合率 (P) と再現率 (R) (学習された SVM を二値分類器としてそのまま用いた.) (不均一なテスト集合の場合)

Abbreviated Name	Common		Polynomial		Cartesian	
	P	R	P	R	P	R
D. Johnson	0.71	0.30	0.64	0.40	0.44	0.24
A. Gupta	0.51	0.32	0.32	0.31	0.31	0.39
J. Smith	0.49	0.26	0.33	0.25	0.31	0.28
R. Johnson	0.57	0.25	0.37	0.19	0.44	0.31

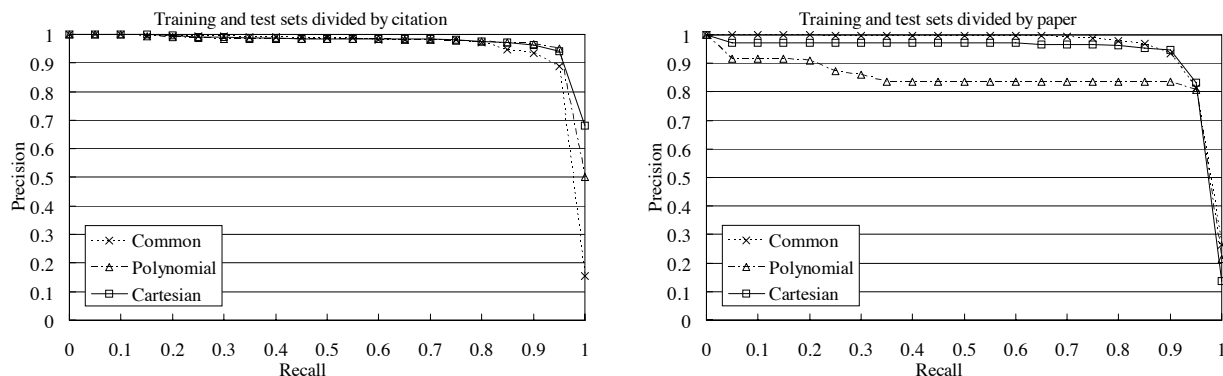


図 7 Cora データセットを用いた場合の引用マッチング問題の適合率 - 再現率曲線

表 7 Cora データセットを用いた場合の引用マッチング問題の適合率 (P) と再現率 (R) (学習された SVM を二値分類器としてそのまま用いた.)

Training and Test Sets	Common		Polynomial		Cartesian	
	P	R	P	R	P	R
Divided by Citation	0.81	0.98	0.86	0.99	0.87	1.00
Divided by Paper	0.86	0.93	0.86	0.29	0.96	0.82

上の実験では、それぞれの省略された著者名ごとに分類器を学習させた。もし、二つの著者が同じ人物を指しているか否かを識別する一般的な分類器が出来れば、多くの分類器を訓練する必要がなくなるため、より有用性が高い。また、新しい著者による論文等、十分な訓練集合が集められない場合に対しても、分類を行うことが可能になる。そこで我々は、計算機科学の分野において二つの著者名が同じ人物を指しているか否かを判別する汎用の分類器を、以下の方法で学習させた。まず、DBLP データセットで 2 件以上の論文を書いている著者の中から、アルファベットの順番に 50,000 人の著者名を選択した。それぞれの著者の論文をランダムに 2 つを選択し、この組を正例とした。また、異なる著者の間の論文の組を作成し、正例と同じ数の負例を用意した。テスト集合は、均一な場合と不均一な場合で、それぞれ図 3 と図 4 の実験と同じものを用いた。表 5、図 5 および、表 6、図 6 に実験結果を示す。再現率が 0.3 より小さな領域においては、**Common** が他よりも高い適合率を示している。しかし、それより高い再現率の領域では、**Common** に対して **Cartesian** と **Polynomial** が高い適合率を示し、両者の中では、**Cartesian** が勝つ場合が多いことが分かる。全体的な適合率は特定の分類器の場合程良くない。これは、特定の省略名に対するテスト集合と、多くの著者による論文から作成された訓練集合における、単語の分布が異なっていることが原因ではないかと考えられる。同様の現象は、文献 [Bilenko 03b] においても報告されている。

表 7 および図 7 は Cora データセットを用いた引用マッ

チング問題の実験結果を示している。ここでは、先行研究 [Bilenko 03c] に従い、訓練集合とテスト集合を分割する 2 通りの手法を試みている。一つは、それぞれの引用が同じ論文を指しているか否かを考慮せずに訓練集合とテスト集合に分割した場合、もう一つは、最初と同じ論文を訓練集合もしくはテスト集合に割り当て、同じ論文を指している引用は訓練集合もしくはテスト集合のどちらか一方にのみ含まれるようにした場合である。Cora データセットに対しては、**Common** も **Cartesian** および **Polynomial** と同等の性能を示している。これは、引用のマッチングにおいては、二つの正例の間に多くの共通の単語があるため、共通の素性だけを用いても分類に十分な手掛かりが得られるためであると考えられる。

5. 関連研究

二つのカーネルの間のテンソル積の一般的な定義は、文献 [Haussler 99] に見られる。入力空間の上でのカーネルとラベル空間の上でのカーネルのテンソル積は、文献 [Zhang 03] において、入力点の間の非類似度が距離に対応する様なユークリッド空間を導出するために用いられている。それに対して我々の研究は、ペアワイズ分類において正例の類似度がゼロになる問題を解決するために、入力空間上のカーネル同士のテンソル積を用いる方法を提案しており、従来の研究とは対象とする問題およびカーネルが定義される空間が異なっている。

エンティティのマッチングや重複発見は、データベースのコミュニティで長い間研究され続けてきた問題であるが、近年機械学習の応用分野として注目を集めている。以下では、最近の研究の中で、我々の研究に最も関係のあると考えられるものについてのみ述べる。伝統的なアプローチを含んだ他の研究については、Bilenko らの論文 [Bilenko 03a] を参照されたい。

機械学習と用いた方法として、Bilenko と Mooney の研究 [Bilenko 03b] が我々の研究に最も関連している。彼らは、重複発見のための文字列の類似度を問題領域に適応した形に学習させることを提案し、固定的な汎用の類

似度を用いた場合に対する優位性を示している。彼らは、それぞれのデータベースのフィールドに関して、異なった類似度を訓練し、複数のフィールドからなるレコードの間の類似度を、フィールドレベルでの類似度を組み合わせることによって計算している。これに対して我々の研究は、例を単一のフィールドからなるとみなし、単一のレベルでの類似度を向上させることに焦点を当てている。彼らの実験では、引用のマッチングや重複したレストランの発見など、同じクラスに属する例が共通の属性を多く持つ場合を扱っており、我々の研究における著者名のマッチングの様な、共通の属性が少ない問題は扱っていない。

Sarawagi と Bhamidipaty の研究 [Sarawagi 02] および Tejada らの研究 [Tejada 02] は、重複している組と重複していない組を分類する分類器を学習する際に、能動学習を用いて訓練データを対話的に発見することで、精度の高い分類器を学習する為に必要な訓練データ数を削減することを可能にしている。Bilenko と Mooney の論文 [Bilenko 03c] は重複発見の評価には適合率-再現率曲線を用いるべきだと主張している。彼らは重複発見に機械学習を用いる際に、訓練集合を収集する幾つかの方法について議論している。

6. 今後の研究課題

本論文では引用データベースにおけるエンティティのマッチングを例として取り上げた。自然言語処理においても、エンティティのマッチングは例えば共参照解決等の形で現れる重要な問題である。共参照解決では、文章において同じエンティティを指している単語やフレーズを発見することが問題となる [Morton 00]。我々はこの共参照解決の問題に本論文で提案した手法を適用することを考えている。

(7) は二つの内積の間のテンソル積になっているが、より一般的に任意のカーネル関数 $K_1(\mathbf{x}, \mathbf{z})$ の間のテンソル積

$$\begin{aligned} K(\hat{\mathbf{x}}, \hat{\mathbf{z}}) &= K((\mathbf{x}^\alpha, \mathbf{x}^\beta), (\mathbf{z}^\alpha, \mathbf{z}^\beta)) \\ &= K_1(\mathbf{x}^\alpha, \mathbf{z}^\alpha) K_1(\mathbf{x}^\beta, \mathbf{z}^\beta) \end{aligned}$$

をペアワイズ分類に用いることも考えられる。例えば、 K_1 として多項式カーネルを用いれば、「片方の例に “VLDB” と “data” という単語が現れ、もう片方に “SIGMOD” と “view” という単語が現れる場合」を指定する様な素性を用いることができる。様々な既知のカーネルのテンソル積を用いた場合のペアワイズ分類の精度の調査は今後の課題の一つである。

また、本論文の手法は、全く異なった種類のデータのマッチングにも適用可能であると考えている。例えば、日本語の文章と英語の文章では、共通の素性はほとんどないが、我々の手法では、日本語の文章同士、英語の文章

同士の内積が定義できれば、辞書のような外部の情報を用いずに、それらの間の対応を学習できる可能性がある。すなわち、我々の手法を拡張することで、異なった種類のデータ間の対応を学習するためのカーネルを、同じ種類のデータ間の類似度を基に定義することができる。テキストと画像といった、全く異なったメディアのデータ同士のマッチングの場合、類似度を直接定義することは困難であるが、テキスト同士、映像同士の類似度は遥かに容易に定義できる。例えば、 K_1 をテキスト同士の内積（類似度）を規定するカーネル、 K_2 を画像同士の内積（類似度）を規定するカーネルとした場合、これらの異なったカーネルのテンソル積

$$K((\mathbf{x}, \mathbf{u}), (\mathbf{z}, \mathbf{v})) = K_1(\mathbf{x}, \mathbf{z}) K_2(\mathbf{u}, \mathbf{v})$$

を用いることで、画像とテキストの対応を判別する分類器を学習できる可能性がある。（ここで、 $\mathbf{x}$ と $\mathbf{z}$ はテキスト、 $\mathbf{u}$ と $\mathbf{v}$ は画像を表す。）このように、我々の提案したカーネルを、異なったメディアのデータ間のマッチングに用いる事も、今後の研究課題として考えている。

7. ま と め

ペアワイズ分類は、エンティティのマッチング等において必要となる基礎技術である。従来手法では、二つの例の間に共通の素性が少ない場合、多くの正例がゼロベクトルに近くなってしまい、精度の良い分類器を学習することが困難であった。我々はこの問題を解決するために、ペアワイズ分類に異なる例からの異なる素性の組合せを用いる方法を提案した。我々が提案したカーネルを用いることで、大きな計算コストをかけずに、実質的に異なる素性の結合を用いた分類を行うことが可能となる。著者名のマッチングと引用のマッチングにおける評価実験の結果により、特に同じクラスの例の間に共通の素性が少ない著者名のマッチングの問題において、我々の手法は従来手法に比べて高い再現率と高い適合率を達成できることが確認された。

謝 辞

有益な助言を頂いた査読者の方々に感謝します。また、本研究の機会を与えて頂いた京都大学大学院情報学研究科田中克己教授に感謝します。本研究の一部は、京都大学 21 世紀 COE プログラム「知識社会基盤構築のための情報学拠点形成」、科学研究費補助金（課題番号 16700097, 16016247）および文部科学省「知的資産の電子的な保存・活用を支援するソフトウェア技術基盤の構築」による。ここに記して謝意を表します。

◇ 参 考 文 献 ◇

- [Bilenko 03a] Bilenko, M., Cohen, W. W., Fienberg, S., Mooney, R. J., and Ravikumar, P.: Adaptive Name-Matching in Information Integration, *IEEE Intelligent Sys-*

tems, Vol. 18, No. 5, pp. 16–23 (2003)

- [Bilenko 03b] Bilenko, M. and Mooney, R. J.: Adaptive Duplicate Detection Using Learnable String Similarity Measures, in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pp. 39–48 (2003)
- [Bilenko 03c] Bilenko, M. and Mooney, R. J.: On Evaluation and Training-Set Construction for Duplicate Detection, in *Proceedings of the KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, pp. 7–12 (2003)
- [Burges 00] Burges, C. J. C. and Crisp, D. J.: Uniqueness of the SVM Solution, in *Advances in Neural Information Processing 12 (NIPS*99)*, pp. 223–229 (2000)
- [Haussler 99] Haussler, D.: Convolution Kernels on Discrete Structures, Technical Report UCSC-CRL-99-10, Baskin School of Engineering, University of California, Santa Cruz (1999)
- [Joachims 99] Joachims, T.: Making Large-Scale SVM Learning Practical, in Schölkopf, B., Burges, C., and Smola, A. eds., *Advances in Kernel Methods - Support Vector Learning*, MIT-Press (1999)
- [Lawrence 99] Lawrence, S., Bollacker, K., and Giles, C. L.: Autonomous Citation Matching, in *Proceedings of the Third International Conference on Autonomous Agents* (1999)
- [Morton 00] Morton, T. S.: Coreference for NLP Applications, in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)* (2000)
- [Sarawagi 02] Sarawagi, S. and Bhamidipaty, A.: Interactive Deduplication using Active Learning, in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, pp. 269–278 (2002)
- [Schölkopf 02] Schölkopf, B. and Smola, A.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press (2002)
- [Shawe-Taylor 04] Shawe-Taylor, J. and Cristianini, N.: *Kernel Methods for Pattern Analysis*, Cambridge University Press (2004)
- [Tejada 02] Tejada, S., Knoblock, C. A., and Minton, S.: Learning Domain-Independent String Transformation Weights for High Accuracy Object Identification, in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, pp. 350–359 (2002)
- [Vapnik 99] Vapnik, V. N.: *The Nature of Statistical Learning Theory*, Springer, 2nd edition (1999)
- [Zhang 03] Zhang, Z.: Learning Metrics via Discriminant Kernels and Multidimensional Scaling: Toward Expected Euclidean Representation, in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, pp. 872–879 (2003)

[担当委員：山川 宏]

2004年6月17日 受理

◇ 付 録 ◇

A. 分類結果がベアインスタンスの構成順序に依存しないことの証明

ここでは、訓練集合の中に元の例の順序を反転したベアインスタンスが両方とも含まれる場合に、式 (7) のカーネルを用いて SVM の学習を行うと、新しい例 $(\mathbf{x}^\alpha, \mathbf{x}^\beta)$ と $(\mathbf{x}^\beta, \mathbf{x}^\alpha)$ は常に同じ分類結果になることを示す。

まず、SVM を求める最適化問題を主問題と双対問題の形で書く、それぞれ以下の形になる*8。

*8 これは、線形分離できない場合も扱うスラック変数を含んだ

主問題

$$\begin{aligned} \min_{\mathbf{w}, \gamma, \xi} \quad & \|\mathbf{w}\|^2 + \frac{C}{m} \sum_{k=1}^m \xi^k \\ \text{s.t.} \quad & y^k (\langle \mathbf{w} \cdot \phi(\mathbf{x}^k) \rangle + \gamma) \geq 1 - \xi^k \\ & \xi^k \geq 0 \quad k = 1, \dots, m. \end{aligned}$$

双対問題

$$\begin{aligned} \max_{\lambda} \quad & \sum_{k=1}^m \lambda^k - \frac{1}{2} \sum_{k=1}^m \sum_{l=1}^m y^k y^l \lambda^k \lambda^l K(\mathbf{x}^k, \mathbf{x}^l) \\ \text{s.t.} \quad & \sum_{k=1}^m y^k \lambda^k = 0 \\ & 0 \leq \lambda^k \leq \frac{C}{m}, \quad k = 1, \dots, m. \end{aligned}$$

ここで ϕ はカーネル K に対応する写像であり、 C は訓練誤差とマージン最大化のトレードオフを指定するパラメータである。この時、主問題と双対問題の最適解は、

$$\mathbf{w} = \sum_{k=1}^m \lambda^k y^k \phi(\mathbf{x}^k)$$

の関係にある。ところで、スラック変数を含まない SVM の場合、主問題は狭義の凸 2 次計画問題なので、最適解 $(\mathbf{w}, \gamma)$ は唯一に定まる。スラック変数を含む 1 ノルムソフトマージン SVM 学習の最適化問題では、主問題の目的関数が狭義の凸関数でなくなるため、主問題の最適解の一意性が保証されない。しかし、その場合でも文献 [Burges 00] により、異なるのは閾値 γ の値だけであり、 $\mathbf{w}$ は一意であることが証明されている。これに対し、双対問題では、目的関数の最適値を実現する解 λ は唯一とは限らない [Schölkopf 02]。即ち、一つの $\mathbf{w}$ に対応する複数の λ が存在する場合がある。(例えば、例題 $\phi(\mathbf{x}^k) (k=1, \dots, m)$ が線型独立でない場合。)

ここで、本文で述べたベアインスタンスの分類問題を考える。 $\mathbf{x}$ と $\bar{\mathbf{x}}$ を、同じ元の例題の順序を逆転したベアインスタンスとする。すなわち、

$$\mathbf{x} = (\mathbf{x}^\alpha, \mathbf{x}^\beta), \quad \bar{\mathbf{x}} = (\mathbf{x}^\beta, \mathbf{x}^\alpha)$$

とする。訓練集合の中に逆転関係にあるベアインスタンスが両方含まれている場合、訓練集合を $\{\mathbf{x}^k | k=1, \dots, \frac{m}{2}\}$ と $\{\bar{\mathbf{x}}^k | k=1, \dots, \frac{m}{2}\}$ に分けることができる。また、訓練集合において、これらのベアインスタンスには同じラベルが与えられているとする。(即ち、 $y = \bar{y}$ 。)

このとき、双対問題は

$$\begin{aligned} \max_{\lambda} \quad & \sum_{k=1}^{\frac{m}{2}} \lambda^k + \sum_{k=1}^{\frac{m}{2}} \bar{\lambda}^k \\ & - \frac{1}{2} \sum_{k=1}^{\frac{m}{2}} \sum_{l=1}^{\frac{m}{2}} \{ y^k y^l \lambda^k \bar{\lambda}^l K(\mathbf{x}^k, \mathbf{x}^l) \\ & + y^k y^l \lambda^k \bar{\lambda}^l K(\mathbf{x}^k, \bar{\mathbf{x}}^l) \\ & + y^k y^l \bar{\lambda}^k \lambda^l K(\bar{\mathbf{x}}^k, \mathbf{x}^l) \\ & + y^k y^l \bar{\lambda}^k \lambda^l K(\bar{\mathbf{x}}^k, \bar{\mathbf{x}}^l) \} \end{aligned} \quad (\text{A.1})$$

$$\text{s.t.} \quad \sum_{k=1}^{\frac{m}{2}} (y^k \lambda^k + y^k \bar{\lambda}^k) = 0 \quad (\text{A.2})$$

$$0 \leq \lambda^k \leq \frac{C}{m}, 0 \leq \bar{\lambda}^k \leq \frac{C}{m}, \quad k = 1, \dots, \frac{m}{2} \quad (\text{A.3})$$

となる。

1 ノルムソフトマージン SVM 学習の最適化問題である。スラック変数を含まない SVM についても同様に証明が可能である。

$(\lambda^1, \dots, \lambda^{\frac{m}{2}}, \bar{\lambda}^1, \dots, \bar{\lambda}^{\frac{m}{2}})$ を上の問題の最適解とした時, 同じ添え字に対応する λ^k と $\bar{\lambda}^k$ 同士を入れ換えた $(\bar{\lambda}^1, \dots, \bar{\lambda}^{\frac{m}{2}}, \lambda^1, \dots, \lambda^{\frac{m}{2}})$ も最適解であることを示す.

まず, 入れ換えを行っても式 (A.2) および式 (A.3) の制約条件は満たされる. 次に目的関数の値であるが, 式 (A.1) の第 1 項と第 2 項の和は入れ換えで不変である. また, 式 (7) のカーネルの定義により, $K(\mathbf{x}^k, \mathbf{x}^l) = K(\bar{\mathbf{x}}^k, \bar{\mathbf{x}}^l)$, $K(\mathbf{x}^k, \bar{\mathbf{x}}^l) = K(\bar{\mathbf{x}}^k, \mathbf{x}^l)$ であることに注意すると, 第 3 項の値も不変であることが分かる. よって, 新しい解は元の解と同じ目的関数値を実現するので, これも最適解である.

以上により, 主問題の最適解は

$$\begin{aligned} \mathbf{w} &= \sum_{k=1}^{\frac{m}{2}} \lambda^k y^k \phi(\mathbf{x}^k) + \sum_{k=1}^{\frac{m}{2}} \bar{\lambda}^k y^k \phi(\bar{\mathbf{x}}^k) \\ &= \sum_{k=1}^{\frac{m}{2}} \bar{\lambda}^k y^k \phi(\mathbf{x}^k) + \sum_{k=1}^{\frac{m}{2}} \lambda^k y^k \phi(\bar{\mathbf{x}}^k) \end{aligned}$$

と 2 通りに表されることが分かる.

最後に, 上の最適化問題を解いて得られた SVM は, ペアインスタンスの例の順序に依存しないことを示す. 学習で得られた解 $(\mathbf{w}, \gamma)$ を用いた SVM の分類器を f とし, テスト例を $\mathbf{x} = (\mathbf{x}^\alpha, \mathbf{x}^\beta)$ とすると,

$$\begin{aligned} f(\mathbf{x}) &= \text{sgn} \left(\sum_{k=1}^{\frac{m}{2}} \lambda^k y^k K(\mathbf{x}^k, \mathbf{x}) + \sum_{k=1}^{\frac{m}{2}} \bar{\lambda}^k y^k K(\bar{\mathbf{x}}^k, \mathbf{x}) + \gamma \right) \\ &= \text{sgn} \left(\left\langle \sum_{k=1}^{\frac{m}{2}} (\lambda^k y^k \phi(\mathbf{x}^k) + \bar{\lambda}^k y^k \phi(\bar{\mathbf{x}}^k)) \cdot \phi(\mathbf{x}) \right\rangle + \gamma \right) \\ &= \text{sgn}(\langle \mathbf{w} \cdot \phi(\mathbf{x}) \rangle + \gamma) \\ &= \text{sgn} \left(\left\langle \sum_{k=1}^{\frac{m}{2}} (\bar{\lambda}^k y^k \phi(\mathbf{x}^k) + \lambda^k y^k \phi(\bar{\mathbf{x}}^k)) \cdot \phi(\mathbf{x}) \right\rangle + \gamma \right) \\ &= \text{sgn} \left(\sum_{k=1}^{\frac{m}{2}} \bar{\lambda}^k y^k K(\mathbf{x}^k, \mathbf{x}) + \sum_{k=1}^{\frac{m}{2}} \lambda^k y^k K(\bar{\mathbf{x}}^k, \mathbf{x}) + \gamma \right) \\ &= \text{sgn} \left(\sum_{k=1}^{\frac{m}{2}} \bar{\lambda}^k y^k K(\bar{\mathbf{x}}^k, \bar{\mathbf{x}}) + \sum_{k=1}^{\frac{m}{2}} \lambda^k y^k K(\mathbf{x}^k, \bar{\mathbf{x}}) + \gamma \right) \\ &= f(\bar{\mathbf{x}}). \end{aligned}$$

よって, 分類結果はペアインスタンスの構成順序に依存しないことが示された*9.

B. 対称な訓練集合から SVM を求める問題と対称化したカーネルを用いて SVM を求める問題が等価であることの証明

まず, 付録 A と同様の議論により, $(\lambda^1, \dots, \lambda^{\frac{m}{2}}, \bar{\lambda}^1, \dots, \bar{\lambda}^{\frac{m}{2}})$ が最適解であるとき, $(\frac{\lambda^1 + \bar{\lambda}^1}{2}, \dots, \frac{\lambda^{\frac{m}{2}} + \bar{\lambda}^{\frac{m}{2}}}{2}, \frac{\lambda^1 + \bar{\lambda}^1}{2}, \dots, \frac{\lambda^{\frac{m}{2}} + \bar{\lambda}^{\frac{m}{2}}}{2})$

*9 ただし, SVM^{light} 等の実装は, 最適性条件がある一定の許容誤差の範囲であれば停止し, 結果として得られる w は真の最適解からの誤差を含んでいる. そのため, ペアインスタンスが決定境界の非常に近くに位置する場合には, w に含まれる誤差のため例の順序によって分類結果が異なる場合がある. しかし, このような例は稀 (SVM^{light} のデフォルトのパラメータで 0.1% 以下) であり分類誤り率に比べれば小さな値であること, また, 決定境界に非常に近い例はそもそも SVM で正確な分類が期待できないことを考えると, 実用上の影響は少ないと考えられる.

も最適解であることを示すことができる. すなわち, 一般には $\lambda^k \neq \bar{\lambda}^k$ であるが,

$$\lambda^k := \frac{\lambda^k + \bar{\lambda}^k}{2}, \bar{\lambda}^k := \frac{\lambda^k + \bar{\lambda}^k}{2} \quad (\text{B.4})$$

と置き換えることによって, いつでも $\lambda^k = \bar{\lambda}^k$ であるような最適解を作ることが可能である.

カーネル関数を

$$\begin{aligned} K'(\mathbf{x}, \mathbf{z}) &= \frac{1}{2} (K(\mathbf{x}, \mathbf{z}) + K(\mathbf{x}, \bar{\mathbf{z}})) \\ &= \frac{1}{4} (K(\mathbf{x}, \mathbf{z}) + K(\mathbf{x}, \bar{\mathbf{z}}) + K(\bar{\mathbf{x}}, \mathbf{z}) + K(\bar{\mathbf{x}}, \bar{\mathbf{z}})) \end{aligned} \quad (\text{B.5})$$

として, 片方の順序のみの $\frac{m}{2}$ 個のペアインスタンス $\{\mathbf{x}^k \mid k = 1, \dots, \frac{m}{2}\}$ からなる訓練集合で SVM の学習を行うことは, 以下の最適化問題に帰着される.

$$\max_{\lambda'} \sum_{k=1}^{\frac{m}{2}} \lambda'^k - \frac{1}{2} \sum_{k=1}^{\frac{m}{2}} \sum_{l=1}^{\frac{m}{2}} y^k y^l \lambda'^k \lambda'^l K'(\mathbf{x}^k, \mathbf{x}^l)$$

$$\text{s.t.} \quad \sum_{k=1}^{\frac{m}{2}} y^k \lambda'^k = 0$$

$$0 \leq \lambda'^k \leq \frac{2C}{m}, \quad k = 1, \dots, \frac{m}{2}.$$

ここで, $\lambda'^k = 2\lambda^k$ と変数を置き換え, 式 (B.5) を用いると,

$$\max_{\lambda} \sum_{k=1}^{\frac{m}{2}} 2\lambda^k - \frac{1}{2} \sum_{k=1}^{\frac{m}{2}} \sum_{l=1}^{\frac{m}{2}} \{y^k y^l \lambda^k \lambda^l (K(\mathbf{x}^k, \mathbf{x}^l) + K(\mathbf{x}^k, \bar{\mathbf{x}}^l) + K(\bar{\mathbf{x}}^k, \mathbf{x}^l) + K(\bar{\mathbf{x}}^k, \bar{\mathbf{x}}^l))\} \quad (\text{B.6})$$

$$\text{s.t.} \quad \sum_{k=1}^{\frac{m}{2}} (y^k \lambda^k) = 0$$

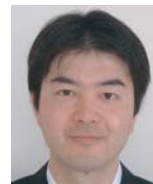
$$0 \leq \lambda^k \leq \frac{C}{m}, \quad k = 1, \dots, \frac{m}{2} \quad (\text{B.8})$$

となるが, これは式 (A.1)~(A.3) において $\lambda^k = \bar{\lambda}^k$ とした問題と等価である. 式 (A.1)~(A.3) の最適解を式 (B.4) によって対称化したものは, 式 (B.6)~(B.8) の最適解であり, 逆も成り立つ.

以上により, 対称な訓練集合から SVM を求める問題と, 対称化したカーネルを用いて SVM を求める問題が, 最適化問題として等価であることが示された.

著者紹介

小山 聡(正会員)



1994 年京都大学工学部数理工学科卒業. 1996 年京都大学大学院工学研究科数理工学専攻修士課程修了. 1996 年-1998 年日本電信電話株式会社. 2001 年-2002 年日本学術振興会特別研究員 (DC2). 2002 年京都大学大学院情報学研究科社会情報学専攻博士後期課程修了. 博士 (情報学). 同年より, 京都大学大学院情報学研究科社会情報学専攻助手. また, 2003 年-2004 年スタンフォード大学 Visiting Assistant Professor. 機械学習, データマイニング, 情報検索等に興味を持つ.

クリストファー D. マニング



1989 年オーストラリア国立大学卒業. 1995 年スタンフォード大学 PhD. 1994 年-1996 年カーネギー・メロン大学 Assistant Professor. 1996 年-1999 年シドニー大学 Lecturer. 1999 年よりスタンフォード大学 Assistant Professor. 2003 年 ACL (計算言語学会) 最優秀論文賞. 自然言語処理, 情報抽出, テキストマイニング等に興味を持つ.