情報処理学会論文誌　数理モデル化と応用　Vol. 4　No. 4　49–58 (Nov. 2011)

文献情報を用いたカーネル法による遺伝子機能アノテーション

ブロンデル マチュー[†1]　関　和広[†1]　上原邦昭[†1]

本稿では，文献の情報を基に遺伝子機能を付与する問題に対して，カーネルを用いた機械学習の手法を提案する．生物医学文献の数は膨大であり，また遺伝子の種類も数多くある．そのため，各文献に記された遺伝子に対して遺伝子機能を手作業で付与（ラベル付け）するには，多大なコストが必要となる．その結果，機械学習を行うために必要な訓練データが，十分に集まらないことが多い．訓練データを必要としない手法として，従来，文字列一致が利用されてきたものの，この手法では，表記の揺れや未知語に対処できないという問題がある．本稿では，付加的な情報を容易かつ効果的に取り込むことができ，計算量的にも優れた性質を持つカーネルを用いることで，これらの問題に対処する．また，マルチラベル分類による遺伝子機能付与を行う際に，各クラスごとに正則化を行うことで，ラベル付きデータの数が特定のクラスに偏っているデータ（不均衡データ）の問題にも対処する．TREC ゲノムトラックのデータを用いた評価実験により，従来手法に対する提案手法の優位性を示す．

Literature-based Gene Function Annotation with Kernels

Mathieu Blondel,[†1] Kazuhiro Seki[†1]
and Kuniaki Uehara[†1]

In recent years, a number of machine learning approaches for literature-based gene function annotation have been proposed. However, due to issues such as lack of labeled data, class imbalance and computational cost, they have usually been unable to surpass simpler approaches based on string-matching. In this paper, we propose a principled machine learning approach focusing on kernel classifiers. We show that kernels are computationally efficient and can address the task's inherent data scarcity by embedding additional knowledge. We also propose a simple and effective solution to deal with the class imbalance problem. From experiments on the TREC Genomics Track data, it is demonstrated that our approach achieves better performance than two existing approaches based on string-matching and cross-species information.

1. はじめに

ヒトゲノムプロジェクトの完了とともに，老化・病気といった人間の身体的機能への理解を目的として，遺伝子の役割を解明する研究がさかんに行われている．これらの研究によって生産される論文の数は膨大であり，その結果，研究者が特定の遺伝子に関する情報を直接論文から網羅的に収集することは，ますます困難になっている．

この問題を解決するため，多くの組織によって，Gene Ontology（GO）と呼ばれるオントロジを用いた遺伝子機能情報の付与（GO アノテーション）が行われている．これにより，モデル生物の多くの遺伝子に関して，生物医学文献の内容に基づく遺伝子機能を容易に把握することが可能になった．現在 GO には，約 30,000 の GO タームが登録されており，これらの GO タームは，分子機能（MF）・細胞成分（CC）・生体内作用（BP）の 3 大分類のもと，有向非巡回グラフ（DAG）の構造で定義されている．一方，人的資源の限界や増加し続ける文献が原因となり，現在の手作業によるアノテーションだけでは，遺伝子機能のデータベースは永久に完了しないという報告がなされている[1)]．

そこで本稿では，機械学習の枠組みによって GO アノテーションの自動化を行う手法を提案する．機械学習を用いることで，文献中から遺伝子の機能情報を抽出し，アノテーションがいまだ行われていない文献に対して，文献の内容に基づいた GO タームの予測を行う．従来用いられてきた機械学習の手法とは異なり，本稿では**カーネル分類器**を用いた方法を提案する．カーネルは，学習を行う際に，分野ごとの知識を付加的に取り込むことができ，さらに効率的な計算ができる性質を有している．また，識別問題においては，従来の機械学習がそのままでは有効に機能しない不均衡データ（各クラスの事例数に大きな偏りがあるデータ）に対しても，簡単なヒューリスティックを用いることで効果的な対処が可能である．

以降，2 章で関連研究について述べ，3 章では提案手法の詳細について述べる．4 章では，提案手法で用いるカーネル分類器の基本的な考え方について紹介する．5 章において，確率的潜在分析モデル（pLSA）に基づく 2 種類のカーネルについて述べる．また，pLSA に基づくカーネルは内部に潜在トピックを持っており，他のカーネルと比較して不均衡データに対して効果的に働くことを実験的に示す．6 章では提案手法と従来手法との比較実験の結果

†1 神戸大学大学院
Graduate School, Kobe University

を示す．7 章で，本研究のまとめと今後の課題について述べる．

2. 関 連 研 究

遺伝子の機能を報告する文献の数は膨大であり，GO アノテーションを手動で行うには，生物医学の専門的な知識と多大な労力を要する．このような背景から，GO ドメインの分類や GO タームのアノテーションを目的として，TREC 2004 ゲノムトラック[7] や BioCreative[2] といった評価型のワークショップが開催された．

ゲノムトラックでは，論文中に記述された遺伝子の機能を GO ドメインと呼ばれる上位のカテゴリ（MF，CC，BP）に分類する共通タスクが設定された．ワークショップの参加者には，マウスの遺伝子とその遺伝子について記述された文献の組が与えられ，文献の内容に基づいて各遺伝子に GO ドメインを付与する．このタスクでは，Seki ら[18] が，同義語辞書と遺伝子名の曖昧一致を用いて，遺伝子について述べた文を同定する方法を提案している．彼らは，特定の遺伝子について言及した文を抽出し，これをベクトル表現に変換，教師情報による語の重み付けを行い，k 近傍法を用いて分類を行った．

このように，ゲノムトラックではオントロジ最上位の GO ドメインだけを対象にしたのに対して，BioCreative では GO ターム（すべて）のアノテーションを目的とした．Ray ら[15] は，GO タームとそれらに関連する単語の共起を用い，ナイーブベイズ分類器を適用することで，GO タームのアノテーションを行った．Chiang ら[5] は文の自動対応付けを行うことで，「*gene product* plays an important role in *function*」といった文の特徴を学習し，文中で GO タームに対応する遺伝子機能が述べられているかの判別を行った．

総じて，ゲノムトラックの参加者は教師付き学習に基づく分類手法の有効性を報告している．一方で，BioCreative の参加者は文字列一致の手法を主に用いている．これらの戦略の違いは，TREC ゲノムトラックが 3 つのクラス，すなわち GO ドメインしか考慮しない一方，BioCreative では約 30,000 種類に及ぶ GO タームを対象としなければならず，訓練データ不足から教師付き学習手法を効果的に適用しにくいことによる．

手法によらず，既存の GO タームのアノテーションの特徴として，精度がきわめて低い点があげられる．この問題が困難である理由の 1 つは，自然言語の取扱いの難しさにある．GO タームには 1 つのタームに対して通常いくつかの同義語が存在し，また，文献中に現れる遺伝子機能を表す文字列に対して複数の GO タームが対応することがある．また，遺伝子機能を表す文字列が文中に現れていたとしても，遺伝子機能の存在を否定する内容であれば，GO タームを付与する必要はない．これらの理由から，既存の文字列一致の方法を用いると，偽陽性（誤った GO タームの付与）が多くなる傾向にあり，適合率が低下する．

もう 1 つの問題として，GO タームの種類の多さがあげられる．そのため，ある GO タームが付与された文献はきわめて少量しか存在せず，また他の GO タームが付与された文献は多数存在するという状況が生じやすい．これは，いわゆる**不均衡データ**の問題であり，このようなデータを用いて信頼性の高い分類器を作成するには特段の注意が必要である．

これらの問題に対処する試みとして，テキスト情報とアノテーションに有益なテキスト以外の情報を組み合わせる手法も提案されている．Stoica ら[20] は与えられた遺伝子の相同分子種（共通祖先の種分化によって生じた機能的に類似の遺伝子）を用いて GO タームのアノテーション候補を制限する方法と，同じ文献に複数アノテーションされた GO タームの共起性を利用することで，不要なアノテーション候補を除去する方法を提案している．Seki ら[17] はこの考えを拡張して，相同分子種を考慮した訓練データを基に，分類器を逐次的に作成する方法を提案している．一方，Si ら[19] は，テキスト，事前知識，生物学的配列の 3 つの情報源から 5 つのスコアを求め，ロジスティック回帰によりこれらのスコアを統合して GO タームの分類を行う方法を提案している．なお，このように複数の異なる知識源を組み合わせることは重要であるものの，相同分子種のような情報はすべての遺伝子や種が持っているわけではない．

これらの研究に対し，本稿では文献情報を用いたカーネル法に基づく学習・予測手法を提案する．カーネルは効率的な計算ができる性質を有しており，また有用な情報を付加的に取り込むことができる．TREC ゲノムトラックのデータを用いた実験から，我々の手法が，文字列一致と相同分子種の情報を用いた従来の手法よりも優れていることを示す．

3. 文献に基づく GO アノテーション

本研究の目的は，文献中の記述と遺伝子機能を表す GO タームとの関係を学習し，新たな文献に対して遺伝機能（GO ターム）のアノテーションを行うことである．これは，アノテーションされていない文献の中で，遺伝子 X に関して GO ターム Y に対応する機能情報が述べられているかを判別することであるといい換えることができる．**図 1** に本研究で提案する GO アノテーションシステムの概要を示す．提案システムは，大別して学習と予測の処理を行う．以下では，特に，前処理として情報抽出と単語集合（Bag-Of-Words）表現への変換について説明する．

情報抽出：文献中から，曖昧文字列一致を用いて遺伝子について言及した部分（本文が取得可能な場合，Seki ら[18] の方法を用いて抽出する），および文献のタイトルとアブストラク

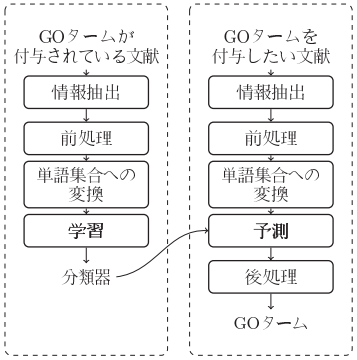

図 1　GO アノテーションシステムの概要：学習（左）と予測（右）
Fig. 1　System overview: learning (left) and prediction (right).

トを抽出する．また，訓練事例に付与されている GO タームの定義文（たとえば，GO ターム「paclitaxel metabolic process（GO:0042616）」の定義文は，「The chemical reactions and pathways involving paclitaxel, an alkaloid compound used as an anticancer treatment.」）を擬似的な正例として（学習時のみ）利用する．これによって，予測の際，もしテストデータの文献に GO ターム定義文の単語が存在すれば，それらの単語も GO タームの予測に考慮される[⋆1]．

前処理と単語集合表現：続いて，抽出したテキスト情報を単語集合表現へと変換する．形式的には，各事例を $\mathbf{x}_i = (x_{i_1}, \ldots, x_{i_D}) \in \mathbf{R}^D$ という形で表現する．ここで，x_{i_j} は，文書 i にある j 番目の語彙の出現回数であり，D は語彙数（単語の数）を表す．

学習：遺伝子機能のアノテーションはマルチラベル分類の問題として考えることができる．つまり，各文献を事例として考えた場合，対応する複数のラベルの組合せを予測することに相当する（対応するラベルがない場合もある）．提案手法では，マルチラベル分類を行うための方法として，各クラス（GO ターム）1 つにつき 1 つの二値分類器を構築する one-vs-all の手法[16] を用いる．この手法は，モデルが簡潔であり，結果の解釈がしやすく（どの特徴が GO タームの分類に役立ったかを確認できる），各分類器は独立であるために並列化が容易である．数式的には，文献 $\mathbf{x}_i$ に対して，$\mathbf{y}_i = (y_{i_1}, \cdots, y_{i_M}) \in \{-1, +1\}^M$ を割り当てることに対応する．ここで，M は GO タームの数であり，$y_{i_c} = +1$ は c 番目の GO タームが文献 $\mathbf{x}_i$ に付与されること，$y_{i_c} = -1$ は付与されないことを表す．

予測と後処理：学習された分類器によって，GO タームの予測を行う．ただし，我々が使用する one-vs-all の手法では，二値分類器を逐次的に適用していくので，GO タームの付与中に矛盾する GO タームの組合せが生じる可能性がある．GO の構造である有向非巡回グラフの利点を生かした後処理を行えば，このような矛盾する GO タームの組合せを除去できるものと考えられる．具体的には，後処理として，GO タームが祖先と子孫の関係にあるとき，それらのうちでもっともらしい方のみを付与する．

次章以降で，カーネル分類器を用いた学習と予測による GO アノテーションの詳細について説明する．

4. カーネル分類器

4.1 線形分類器の正則化

文献 $\mathbf{x}_*$ を所与としたとき，$s_c: \mathbf{R}^D \to \mathbf{R}$ を c 番目の GO タームのスコア関数とする．

$$s_c(\mathbf{x}_*) = \boldsymbol{\beta}_c \cdot \phi(\mathbf{x}_*) \tag{1}$$

$\boldsymbol{\beta}_c = (\beta_{c_1}, \ldots, \beta_{c_D}) \in \mathbf{R}^D$ は c 番目の GO タームの重みベクトルである．直感的な解釈として，強い正の値をとるとき（負の値をとるとき），重み β_{c_j} は j 番目の単語が c 番目の GO タームに強く依存している（依存していない）ことを示し，GO タームを付与するうえで重要な手がかりとなる．ϕ は入力を高次元空間に写像するための関数であり，$\phi(\mathbf{x}) = \mathbf{x}$ のときは写像を行わない．

文献 $\mathbf{x}_*$ が c 番目の GO タームに属しているかどうかを決定するために，c 番目の GO タームの予測関数 $f_c: \mathbf{R}^D \to \{-1, +1\}$ を定義する．

$$f_c(\mathbf{x}_*) = sign(s_c(\mathbf{x}_*)) \tag{2}$$

ここで，$sign(a)$ は $a > 0$ のとき $+1$，それ以外の場合は -1 になる関数である．なお，f_c には閾値が存在しない．

本学習アルゴリズムの目的は，文献の集合 $\mathbf{x}_1, \ldots, \mathbf{x}_N$ と関連するラベルのベクトル $\mathbf{y}_1, \ldots, \mathbf{y}_N$ を与えたときに，以下の目的関数を最小化する重みベクトル $\boldsymbol{\beta}_c$ を学習することである．

$$L_c(\boldsymbol{\beta}_c) = C_c \sum_{i=1}^{N} \ell(y_{i_c}, s_c(\mathbf{x}_i)) + \frac{1}{2}||\boldsymbol{\beta}_c||^2 \tag{3}$$

⋆1 実際には，本研究で評価実験に利用したデータセットにおいては，定義文の利用の有無は精度に影響を与えなかった．

C_c は c 番目の GO タームのハイパーパラメータであり，過学習の問題を防ぐために，モデルの複雑さを制御する役割を果たす．$C_c \to \infty$ は正則化しないことを示し，$C_c = 0$ は，無限の正則化を行うことに対応する．ℓ は誤った予測を行った際の損失関数であり，ヒンジ損失関数，対数損失関数，二乗損失関数がそれぞれサポートベクトルマシン（SVM），ロジスティック回帰（LR），正則化最小二乗分類器（RLSC）に対応する．本稿では，これら 3 つのモデルを正則化線形モデルと見なす．

4.2　カーネル化

リプレゼンタの定理より，訓練データの事例を線形に組み合わせることで式 (3) を表現できる．

$$\boldsymbol{\beta}_c = \sum_{i=1}^{N} \alpha_{c_i} \phi(\mathbf{x}_i) \tag{4}$$

そのため，文献 $\mathbf{x}_*$ を与えた場合，c 番目の GO タームのスコアを $s_c(\mathbf{x}_*)$ と書き直すことができる．

$$s_c(\mathbf{x}_*) = \sum_{i=1}^{N} \alpha_{c_i} \phi(\mathbf{x}_i)\phi(\mathbf{x}_*) = \sum_{i=1}^{N} \alpha_{c_i} \mathcal{K}(\mathbf{x}_i, \mathbf{x}_*) \tag{5}$$

$\mathcal{K}$ はカーネル関数であり，$\mathbf{x}_i$ は $\alpha_{c_i} \neq 0$ のとき，c 番目の GO タームのサポートベクトルに対応する．

4.3　不均衡データの扱い

前述のように，GO タームの総数は約 3 万であり，限られた訓練データ中には，特定の GO タームが付与された論文数が多くなる一方，他の GO タームが付与された文献の数が少ないという状況が生じやすい．さらに，今回使用した one-vs-all の枠組みでは，個々のクラスに関して分類器を学習するため，負例の数が大きくなりやすい．GO アノテーションの従来研究では，不均衡データの問題はほとんど議論されていないものの，GO タームの効果的な予測を行うためには不均衡データへの適切な対処がきわめて重要である．

不均衡データに対処するため，Osuna ら[14] は，正例クラスと負例クラスごとのハイパーパラメータ C^+ と C^- を用いた weighted SVM を提案している．本研究では，Osuna らの手法に着想を得て，次のようなヒューリスティックを提案する．GO タームアノテーションの訓練データに存在する多数の負例の影響を抑えつつ事例を有効活用するため，正例に対して負例よりも強い重み付けを行う．いい換えると，負例のクラスに対して正例のクラスよりも強い正則化を行う．具体的には，重みの学習の際，目的関数である式 (3) の最小化を次式のように行う．

$$L_c(\boldsymbol{\beta}_c) = C_c \sum_{i=1}^{N} \mu_{i_c} \ell(y_{i_c}, s_c(\mathbf{x}_i)) + \frac{1}{2}||\boldsymbol{\beta}_c||^2 \tag{6}$$

$$\mu_{i_c} = \begin{cases} n_{\neg c}/n_c, & \text{if } y_{i_c} = +1 \\ 1, & \text{if } y_{i_c} = -1 \end{cases} \tag{7}$$

ここで μ_{i_c} は，事例 x_i が c 番目の GO を付与された場合，$n_{\neg c}$（c 番目の GO タームをラベルとして付与されていない事例の数）と n_c（c 番目の GO タームをラベルとして付与された事例の数）の比率となり，それ以外のときは 1 となる．式で表現すれば，$n_c = |\{\mathbf{x}_i | y_{i_c} = +1\}|$，$n_{\neg c} = N - n_c$ となる．GO アノテーションの訓練データでは通常 $n_{\neg c} > n_c$ なので，負例よりも正例に強い重みが与えられることになる．Osuna ら[14] の手法と比べると，提案ヒューリスティックには 1 つのハイパーパラメータ（C_c）しかなく，調整が容易である．ただし，両者に本質的な違いはなく，提案ヒューリスティックは，正例クラスと負例クラスのバランスを調整するための weighted SVM の特別な場合と見なすことができる．

4.4　ハイパーパラメータの調整

式 (6) の目的関数には，モデルの複雑さを制御するためのハイパーパラメータ C_c が存在し，より高性能の分類器を得るためには，その値を交差検定などで適切に決定することが重要である．また，前述の不均衡データの問題も考慮する必要がある．本研究では，以下のようにハイパーパラメータを調整する．まず，用意されたデータを検証データと訓練データに分割する．検証用のデータとして，正例 1 つに対して一定数の負例を用意する．負例の数は，ゲノムトラックの全訓練データにおける正負例の割合に従うように設定する．そして，残りを訓練データとして分類器を学習し，用意した検証データを用いて C_c に対する F_1 スコアを算出する．この作業をすべての正例が選択されるまで行い，c 番目の GO タームの F_1 スコア平均を最大化する C_c をハイパーパラメータとして選択する．

5.　潜在トピックカーネル

本研究では，事例を表現するための特徴量として，遺伝子名によって抽出したテキストの断片，アブストラクトおよびタイトル，GO ターム定義文の 3 種類の情報を用いる．しかしながら，これらの情報には平均で 300 種程度の語彙しか含まれず，ベクトルの要素がほぼ 0 の疎なベクトルになるため，GO タームを特徴づける単語が少ない．そこで，pLSA を

用いたカーネルの利用を提案する．pLSA モデルはすべての文献を用いて学習されるため，カーネルの中で単語の平滑化が行われる．少数の文献にしかアノテーションされていない GO タームを予測するうえで，この手法は特に有効であると考えられる．本章では，pLSA の背景とそのカーネル化について紹介する．

5.1　pLSA の説明

確率的潜在モデル（pLSA）[9] は，文書・単語行列の中に隠れた潜在トピックを推定する確率的な枠組みである．このモデルは，たとえ文書中に共通の単語がなかったとしても，単語の組（共起）を共有してさえいれば，それらの文書には，同一のトピックが存在すると考える．pLSA の潜在トピックモデルにおいて，文書と単語は両者に関連する潜在トピックに対して条件付き独立であると仮定する．トピックの数は普通，文書と単語の数に比べて少なく設定され，トピックを中心として，特定の文書の内容に条件付けした単語の予測を効果的に行うことができる．

pLSA は，k 番目の潜在トピックが与えられた場合に，j 番目の単語が生成される確率 $\{P(w_j|z_k)\}_{j,k}$，k 番目の潜在トピックが与えられた場合に i 番目の文書が生成される確率 $\{P(\mathbf{x}_i|z_k)\}_{i,k}$，そして，$k$ 番目の潜在トピックが生じる確率である $\{P(z_k)\}_k$ の 3 つのパラメータからなる．

pLSA は教師なしの学習モデルであり，パラメータを求めるために，通常は期待値最大化アルゴリズム（EM アルゴリズム）を使用する．EM アルゴリズムは初期値に敏感であることから，局所解に陥りやすい．そこで，6.2 節の実験では，異なる初期値を用いた 5 つの結果の平均を最終的な結果とする．なお，焼きなましを用いた EM アルゴリズム[9] についても実験を行ったものの，収束が遅く，かつ一般的な EM アルゴリズムの単純平均と比べて良い結果が得られなかったため，本研究では採用しなかった．

5.2　pLSA カーネル

Jaakkola ら[11] は，識別モデルに生成モデルを組み込むため，一般的な枠組みとしてフィッシャーカーネルを導入した．このカーネルの理論的背景には情報幾何の考え方があり，可逆で微分可能なパラメータの変換に対して不変という性質がある．Hofmann[8] はフィッシャーカーネルを pLSA のためのカーネルとして位置付け，次式のカーネルを用いて，ラベル付けされたデータの量が限られている中で分類精度を向上させた．

$$\mathcal{K}_z(\mathbf{x}_i,\mathbf{x}_n)=\sum_k P(z_k|\mathbf{x}_i)P(z_k|\mathbf{x}_n)/P(z_k) \tag{8}$$

$$\mathcal{K}_w(\mathbf{x}_i,\mathbf{x}_n)=\sum_j \hat{P}(w_j|\mathbf{x}_i)\hat{P}(w_j|\mathbf{x}_n) \left(\sum_k \frac{P(z_k|\mathbf{x}_i,w_j)P(z_k|\mathbf{x}_n,w_j)}{P(w_j|z_k)}\right) \tag{9}$$

ここで，$\hat{P}(w_j|\mathbf{x}_i)=x_{i_j}/\sum_{j'} x_{i_{j'}}$ は単語の経験分布であり，単語の数を数えることで求めることができる．

これら 2 つのカーネルは理論的に導出されているにもかかわらず，直感的な解釈とも一致している．$\mathcal{K}_z$ は文献 $\mathbf{x}_i$ と $\mathbf{x}_n$ のトピックの重なりと解釈でき，$\mathcal{K}_w$ は事後分布の類似度によって重み付けされた文献間の単語の重なりと考えることができる．

$\mathcal{K}_z$ と $\mathcal{K}_w$ は内積形式で書き直せるため，それぞれに対応する ϕ_z と ϕ_w が存在する．

$$\phi_z(\mathbf{x})_k=P(z_k|\mathbf{x})/\sqrt{P(z_k)} \tag{10}$$

$$\phi_w(\mathbf{x})_{j,k}=\hat{P}(w_j|\mathbf{x})P(z_k|\mathbf{x},w_j)/\sqrt{P(w_j|z_k)} \tag{11}$$

ここで，ϕ_z は K 次元の密ベクトルを出力する関数であり，ϕ_w は $D\times K$ 次元の疎ベクトルを出力する関数である．具体的な ϕ 関数が存在しているということは，線形 SVM の実装を用いて学習できることを意味する．6.4 節で，pLSA カーネルを用いた際の線形 SVM とカーネル SVM の効率性について議論する．

本システムでは，Hofmann の手法にならい，2 種類の pLSA カーネルを組み合わせた次式のカーネルを使用する．ここで，表記 K は pLSA モデルが K 個のトピックに従うことを表している．

$$\mathcal{K}^K_{z+w}(\mathbf{x}_i,\mathbf{x}_n)=\mathcal{K}^K_z(\mathbf{x}_i,\mathbf{x}_n)+\mathcal{K}^K_w(\mathbf{x}_i,\mathbf{x}_n) \tag{12}$$

特徴空間では，$\mathcal{K}_z$ と $\mathcal{K}_w$ を足すことは，ϕ_z と ϕ_w の出力を連結することと同じである．

なお，pLSA は教師なしのモデルであり，その副産物として，上記 2 種類のカーネルを統合することで，ラベル付きのデータだけでなく，ラベルなしのデータの情報も利用した半教師付き学習が可能となる．さらに，特定のテスト集合に対して分類器を適応させるトランスダクティブ学習にも利用できる．次章では，これらの実験設定による評価実験の結果についても報告する．

6. 評 価 実 験

提案手法の枠組みを評価するため，TREC 2004 ゲノムトラックで構築されたデータを使

用し，GO タームが付与された文献をテストデータとして実験を行った[⋆1]．このテストデータには 863 個の事例が存在し，各事例は，PubMed ID と文献中で述べられている遺伝子の二つ組で表現されている．これらの事例に計 543 個の GO タームが付与されており，平均 2.14 個/事例の GO タームが存在する．訓練データには，ゲノムトラックの訓練データセット（1,418 件の文献の全文データ）と MGD データベース（6,750 件の文献のアブストラクト）を用いた．前者に付与されている GO ターム数 M は計 825 個であり，平均的 2.52 個/事例の GO タームが存在する．後者の文献には，計 825 個の同じ GO ターム群が関連づけられている．また，テストデータと訓練データに共通して存在している GO ターム数は 226 である（被覆率 = 41.6%）．各事例が異なる遺伝子に関するものであるため，事例数と遺伝子数は同じである．テキストの前処理として，ストップワード・句読点・長い単語（50 文字以上）の除去，語形の変化を取り除き，小文字への変換を行った．これらの前処理後の語彙数 D は 47,859 であった．なお，Gene Ontology に定義されている GO タームの総数は約 30,000，マウスには約 25,000 の遺伝子があると考えられている[12]．すべての遺伝子の機能が解明されているわけではなく，実際に利用される GO タームにも偏りがあるものの，実験データの大きさは比較的小さく，今後，さらに大規模なベンチマークデータの構築が望まれる．

評価方法には，従来研究と直接的な比較を行うため，適合率，再現率，それらの調和平均である F_1 スコアを用いた．適合率は，正しく予測した GO タームの数を予測した GO の数で割った値であり，再現率は正しく予測できた GO タームの総数をテストデータ中にある GO タームの数で割った値である．

6.1　分類器ごとの評価

本実験では，損失関数が異なる 3 つの分類器，すなわちサポートベクタマシン（SVM），ロジスティック回帰（LR），正則化最小二乗分類器（RLSC）を比較した．また，カーネルには式 (13) の線形カーネルを使用した．前述したように，これら 3 つの分類器は同一の予測方法を行い，最適な重みベクトル $\boldsymbol{\beta}_c$ の定義だけが異なる．比較のため，ナイーブベイズ分類器（NB）についても実験を行った．結果を**表 1** に示す．

結果を見ると，SVM が最も良い性能を示していることが分かる．LR と RLSC は SVM にはわずかに劣るものの，ナイーブベイズ分類器よりも高い性能を示した．総じて，正則化線形モデルは GO アノテーションの問題に適しており，4.3 節と 4.4 節で述べたようなハイパーパラメータの調整と不均衡データの扱いを適切に行えば，比較的高い精度が得られることが判明した．

表 1　ナイーブベイズ（NB）と線形カーネルを用いたサポートベクタマシン（SVM），ロジスティック回帰（LR），正則化最小二乗分類（RLSC）の分類性能の比較
Table 1　Comparison of the accuracy of Naive Bayes (NB), Support Vector Machines (SVM), Logistic Regression (LR) and Regularized Least-Squares Classifier (RLSC).

分類器	適合率	再現率	F_1 スコア
NB	0.21	0.14	0.17
SVM	0.36	**0.20**	**0.26**
LR	**0.39**	0.18	0.25
RLSC	0.34	0.18	0.24

表 2　不均衡データの扱いの違いによる精度比較
Table 2　Comparison of class imbalance handling methods.

不均衡データの扱い	適合率	再現率	F_1 スコア
提案ヒューリスティック	**0.36**	**0.20**	**0.26**
ダウンサンプリング	0.26	0.18	0.21
不均衡データ考慮せず	0.14	0.09	0.11

参考まで，不均衡データの扱いに関して，SVM を用いて次の比較実験を行った．比較した方法は，1）提案ヒューリスティック，2）負例のダウンサンプリング（負例数と正例数を合わせる），3）不均衡データを考慮しないの 3 つである．結果を**表 2** に示す．この結果から，提案ヒューリスティックの有効性が確認できる．

6.2　カーネルの効果

本節の実験では，サポートベクタマシンに 3 種類の異なるカーネル，$\mathcal{K}_{linear}$，$\mathcal{K}_{poly}$，$\mathcal{K}_{plsa}$ を適用し，それぞれのカーネルの効果を検証した．$\mathcal{K}_{linear}$ は線形カーネルであり，各事例は自身のノルムによって正規化される（つまりコサイン類似度）．$\mathcal{K}_{poly}$ は d 次の多項式カーネルであり，順序を考慮しない 1 グラム，2 グラム，...，d グラムの全組合せに対応する．$\mathcal{K}_{plsa}$ は pLSA に基づくカーネルであり，Hofmann[8] の方法にならい，トピック数 8，16，32 個の pLSA カーネルの線形結合を用いた．以下，それぞれのカーネルの定義を示す．

$$\mathcal{K}_{linear}(\mathbf{x}_i, \mathbf{x}_n) = \frac{\mathbf{x}_i \cdot \mathbf{x}_n}{||\mathbf{x}_i|| ||\mathbf{x}_n||} \tag{13}$$

⋆1 BioCreative については，GO アノテーションに利用された論文の全文データが参加者以外に提供されておらず，また参加者が予測した GO タームの正否をエキスパートが判断するという評価方法をとっていたことから公平な比較ができない．そこで本研究では，客観的なゴールドスタンダードが存在するゲノムトラックのデータを評価実験に用いた．

表 3　サポートベクタマシン（SVM）のカーネルごとの比較
Table 3　Comparison of kernels for Support Vector Machines (SVM).

カーネル	適合率	再現率	F_1 スコア
$\mathcal{K}_{linear}$	0.36	0.20	0.26
$\mathcal{K}_{poly}$ $(d=2)$	0.35	0.19	0.25
$\mathcal{K}_{plsa}$	0.38	0.20	0.26
$\mathcal{K}_{plsa}$ (+U)	**0.39**	0.22	0.28
$\mathcal{K}_{plsa}$ (+U + T)	0.38	**0.24**	**0.29**

$$\mathcal{K}_{poly}(\mathbf{x}_i,\mathbf{x}_n)=(1+\mathcal{K}_{linear}(\mathbf{x}_i,\mathbf{x}_n))^d \tag{14}$$

$$\mathcal{K}_{plsa}(\mathbf{x}_i,\mathbf{x}_n)=\sum_{K\in\{8,16,32\}}\mathcal{K}^K_{z+w}(\mathbf{x}_i,\mathbf{x}_n) \tag{15}$$

表 3 において，$\mathcal{K}_{plsa}$ (+U) は，MGD データベースから取得したラベルなしのアブストラクト 10,000 件を使用して pLSA モデルを学習した結果である（半教師付き学習）．$\mathcal{K}_{plsa}$ (+U + T) は，テストデータについても pLSA を適用し，より分類対象に注目した学習の結果である（トランスダクティブ学習）．線形カーネルと比べた性能の向上はわずかではあるものの，訓練データ不足が問題である遺伝子アノテーションにおいて，これら生成モデルを用いた手法の結果は興味深い．なお，単語の重要性を定量化する際によく使用される *tf-idf* についても実験を行ったものの，F_1 スコアは 0.21 程度であった．

比較のため，Nigam ら[13] が提案した半教師あり単純ベイズを利用して実験を行ったところ，F_1 スコアは 0.23 であった．単純ベイズの精度が比較的低い理由は，正則化ができないため，限られたデータ（特に正例の少ない GO タームの場合）では過学習しやすいことによる．

さらに，本研究と関係が深い手法として，supervised Latent Dirichlet Allocation (sLDA)[3] を利用して実験を行った．本研究で利用した pLSA モデルは最終的な予測対象である GO タームを考慮しないのに対し，sLDA は分類器とともにトピックモデルを推定するため，GO ターム予測に最適化されたトピックモデルが構築される．ただし，sLDA はラベルなしデータを学習に利用することができないため，GO タームアノテーションにおけるラベルデータ不足に対処することはできない．実験の結果，F_1 スコアは 0.27 であり，表 3 の線形カーネル $\mathcal{K}_{linear}$，あるいはラベルありデータのみを使った pLSA カーネル $\mathcal{K}_{plsa}$ と同程度の精度であった．

6.3　従来手法との比較

GO アノテーションにおいて，アルゴリズムごとの性能差を俯瞰するため，Stoica ら[20]，Seki ら[17] の手法との比較を行った．結果を**表 4** に示す．Stoica らの結果は著者らの実装によるものであり，Seki らの結果は文献 17) に基づく．Proposed (+O) は，後述するように，相同分子種による制約を後処理として提案手法に加えた結果である．

表 4　既存手法との比較
Table 4　Comparison with existing methods.

手法	適合率	再現率	F_1 スコア
Proposed	0.38	0.24	0.29
Proposed (+O)	**0.42**	0.23	**0.30**
Stoica & Hearst	0.19	**0.46**	0.27
Seki, et al.	0.26	0.27	0.26

結果，我々の手法（Proposed）が適合率と F_1 スコアにおいて最も良い結果を示した．一方，再現率については，Stoica らの手法が最良であった．この理由は，後者では，相同分子種の情報を使うことで一貫性のない候補を除去しているためだと考えられる（相同分子種は，共通の祖先の遺伝子を受け継いだ異なる種が持つ遺伝子のことであり，一般的に類似した遺伝的性質を持ちやすい傾向にある）．なお，相同分子種に基づく制約は，後処理として提案手法でも利用可能であり，これを施した結果が Proposed (+O) である．具体的には，相同分子種（ラット）に付与されていない GO タームの予測を抑制することで，適合率の向上を図った．その結果，再現率が微減したものの，適合率が 0.38 から 0.42 に向上し，F1 スコアも 0.30 に向上した．マウスとラットはきわめて関連の高い種どうしであるため，再現率の悪化を最小限にとどめつつ適合率を上げることができたものと考えられる．t 検定を行ったところ，提案手法 Proposed (+O) と Stoica らの手法の差は，有意水準 0.05 で統計的に有意であった（$p=0.03$）．

6.4　高速な学習・予測・交差検定

ここ数年，大規模なデータに対して効率的に線形カーネルを用いた SVM を学習する方法が数多く提案されてきた．それらの中で，双対座標最急降下法[10] は，事例数 N が極端に多い場合，分解法[14] に基づく実装と比較して，劇的に速い学習を行うことができる[10]．しかし，GO タームのようにクラス数 M が多い場合，線形カーネルの場合でも，線形 SVM よりカーネル SVM の学習の方が効率的な場合がある．これは，one-vs-all 形式の場合，負例の多くがすべての二値分類器で共有されるため，計算の多くが無駄に繰り返されているためである．カーネル SVM の計算には，類似度行列の計算時間 $O(N^2D)$ を必要とするものの，類似度行列はすべての二値分類器で共有されているため，GO タームごとにラベル y_{i_c}

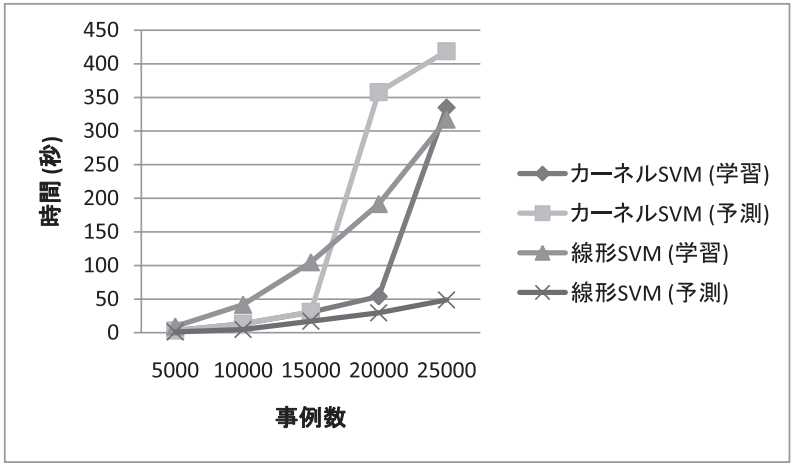

図 2　線形 SVM とカーネル（線形）SVM における学習・予測時間
Fig. 2　Training and Prediction times of Linear SVM and Kernel SVM (linear kernel).

しか変化しない．そのため，類似度行列の計算をいったん済ませれば，双対係数 α_{i_c} を更新するだけで，計算コストをかけずに SVM の学習が行える．

これを例示するため，300 個の非ゼロ要素の特徴を持つ人工データを各 $N = 5{,}000, 10{,}000, \ldots, 25{,}000$ 個用意し，$M = N/50$ 個の二値分類器を one-vs-all 形式で用いた実験を行った．すなわち，各分類器の学習には 50 個の正例が使用されることになる．実験結果を**図 2** に示す．使用した計算機は，Intel(R) Core(TM) i5 CPU 750 @ 2.67 GHz（4 GB RAM）である．

この結果から，事例数 25,000 程度までは，学習時間において線形 SVM よりカーネル SVM の方が優れていることが分かる．なお，4.4 節で述べたような交差検定を行う場合，訓練データを分割して 1 つの GO タームに関して多数の分類器を繰り返し学習する必要がある．しかし，最適化したいパラメータがカーネルに関するものでなければ，類似度行列を再計算する必要はなく，コストをかけずに多くのパラメータを調べることができる．

一方，予測速度については，明らかにカーネル SVM よりも線形 SVM の方が優れている．これは，式 (5) におけるサポートベクトルの数が事例数 N とともに増加するためである．

さらに，ゲノムトラックのデータセットを用いた場合の計算時間を**表 5** に示す．この表は人工データを用いた実験と異なり，4.4 節で述べた交差検定を行った結果である（交差検定の計算時間も含む）．カーネル SVM は libsvm [4] による結果であり，$\mathcal{K}_{linear}$ と $\mathcal{K}_{plsa}$ を直接使用した．線形 SVM は liblinear [6] による結果であり，$\mathcal{K}_{linear}$ と $\mathcal{K}_{plsa}$ に相当する ϕ 関数を使った．また，K_{plsa} の場合，表 5 に示した結果に加えて，pLSA モデルを学習するために 55 分かかった（500 反復）．

表 5　ゲノムトラックのデータを用いたときの計算時間（分）の比較
Table 5　Comparison of the computational time in minutes with Genomics track dataset.

	学習		予測	
	線形 SVM	カーネル SVM	線形 SVM	カーネル SVM
$\mathcal{K}_{linear}$	78	**36**	**3**	8
$\mathcal{K}_{plsa}$	362	**84**	**11**	18

以上の結果からも，GO タームのようにクラス数 M が多い場合，学習にはカーネル SVM を用い，予測には線形 SVM を用いると効率が良いことが分かる．カーネル SVM から線形 SVM への変換は，式 (4) により，双対変数ベクトル $\boldsymbol{\alpha}_c$ を重みベクトル $\boldsymbol{\beta}_c$ へ変換することで行える．この変換を学習後に行えば，予測を高速化することができ，またサポートベクトルを保存する必要もなくなる．

7. おわりに

本研究では，カーネルを用いた GO タームのアノテーション手法を提案した．TREC ゲノムトラックのデータによる実験から，提案手法により，文字列一致および異種間の情報を用いた従来手法よりも高い適合率と F_1 スコアが得られることが分かった．また，GO タームのアノテーションに起こりやすいラベル付きデータが不足するという問題について，潜在トピックをカーネルに取り込むことで効果的に対処することができた．さらに，GO タームの数（クラス数）が膨大であっても，カーネルを用いることで，学習や交差検定を効率的に行えることを示した．加えて，GO タームごと（クラスごと）に正則化を行うことで，貴重な訓練データを活用しつつ，不均衡データに対処する手法を提案した．

今後の発展として，カーネルに基づく 2 つの方法に取り組むことを考えている．1 つは，大規模データに適したカーネルを用いること，もう 1 つは，Gene Ontology の有向非巡回グラフの構造や他の知識源に対してそれぞれカーネルを定義し，マルチカーネルの統合・学習を行うことである．

参 考 文 献

1) Baumgartner, W.A., Cohen, K.B., Fox, L.M., Acquaah-Mensah, G. and Hunter, L.: Manual curation is not sufficient for annotation of genomic databases, *Bioinformatics*, Vol.23, No.13, pp.i41–48 (2007).
2) Blaschke, C., Leon, E., Krallinger, M. and Valencia, A.: Evaluation of BioCreAtIvE assessment of task 2, *BMC Bioinformatics*, Vol.6, No.Suppl 1, p.S16 (2005).
3) Blei, D.M. and Mcauliffe, J.D.: Supervised topic models, *Advances in Neural Information Processing Systems 20* (2007).
4) Chang, C.-C. and Lin, C.-J.: *LIBSVM: A library for support vector machines* (2001), Software available from ⟨http://www.csie.ntu.edu.tw/˜cjlin/libsvm⟩.
5) Chiang, J.-H. and Yu, H.-C.: MeKE: Discovering the functions of gene products from biomedical literature via sentence alignment, *Bioinformatics*, Vol.19, No.11, pp.1417–1422 (2003).
6) Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. and Lin, C.-J.: LIBLINEAR: A Library for Large Linear Classification, *Journal of Machine Learning Research*, Vol.9, pp.1871–1874 (2008).
7) Hersh, W., Bhuptiraju, R.T., Ross, L., Cohen, A.M. and Kraemer, D.F.: TREC 2004 Genomics Track Overview, *Proc. 13th Text REtrieval Conference* (*TREC*) (2004).
8) Hofmann, T.: Learning the Similarity of Documents: An Information-Geometric Approach to Document Retrieval and Categorization, *Advances in Neural Information Processing Systems 12*, Solla, S.A., Leen, T.K. and Müller, K.-R. (Eds.), pp.914–920 (1999).
9) Hofmann, T.: Probabilistic Latent Semantic Analysis, *Proc. Uncertainty in Artificial Intelligence*, *UAI'99*, Stockholm (1999).
10) Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S.S. and Sundararajan, S.: A dual coordinate descent method for large-scale linear SVM, *Proc. 25th International Conference on Machine Learning*, *ICML'08*, New York, NY, USA, pp.408–415, ACM (2008).
11) Jaakkola, T. and Haussler, D.: Exploiting Generative Models in Discriminative Classifiers, *Advances in Neural Information Processing Systems 11*, pp.487–493 (1998).
12) Mouse Genome Sequencing Consortium: Initial sequencing and comparative analysis of the mouse genome, *Nature*, Vol.420, No.6915, pp.520–562 (2002).
13) Nigam, K., Mccallum, A.K., Thrun, S. and Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, Vol.39, No.2, pp.103–134 (2000).
14) Osuna, E.E., Freund, R. and Girosi, F.: Support Vector Machines: Training and Applications, Technical report, Massachusetts Institute of Technology (1997).
15) Ray, S. and Craven, M.: Learning Statistical Models for Annotating Proteins with Function Information using Biomedical Text, *BMC Bioinformatics*, Vol.6, No.Suppl 1, p.S18 (2005).
16) Rifkin, R. and Klautau, A.: In Defense of One-Vs-All Classification, *J. Mach. Learn. Res.*, Vol.5, pp.101–141 (2004).
17) Seki, K., Kino, Y. and Uehara, K.: Gene Functional Annotation with Dynamic Hierarchical Classification Guided by Orthologs, *Proc. Discovery Science*, Vol.5808, pp.425–432 (2009).
18) Seki, K. and Mostafa, J.: Gene Ontology Annotation as Text Categorization: An Empirical Study, *Information Processing & Management*, Vol.44, No.5, pp.1754–1770 (2008).
19) Si, L., Yu, D., Kihara, D. and Fang, Y.: Combining gene sequence similarity and textual information for gene function annotation in the literature, *Information Retrieval*, Vol.11, pp.389–404 (2008).
20) Stoica, E. and Hearst, M.: Predicting Gene Functions from Text Using a Cross-Species Approach, *Proc. Pacific Biocomputing Symposium*, Vol.11, pp.88–99 (2006).

(平成 23 年 2 月 7 日受付)
(平成 23 年 4 月 4 日再受付)
(平成 23 年 5 月 9 日採録)

ブロンデル　マチュー

平成 20 年リール第 1 大学大学院テレコム研究科修士課程修了．現在，神戸大学大学院システム情報学研究科博士課程に在籍．機械学習とその応用の研究に従事．

関　　和広

平成 14 年図書館情報大学大学院情報メディア研究科修士課程修了．平成 18 年インディアナ大学大学院図書館情報学研究科博士課程修了．神戸大学助手，助教を経て，現在，同大学院システム情報学研究科講師．情報検索，自然言語処理，機械学習の研究に従事．Ph.D. 電子情報通信学会，自然言語処理学会，ACM SIGIR 各会員．

上原　邦昭（正会員）

昭和 53 年大阪大学基礎工学部情報工学科卒業．昭和 58 年同大学大学院博士後期課程単位取得退学．同産業科学研究所助手，講師，神戸大学工学部情報知能工学科助教授，同都市安全研究センター教授を経て，現在，同大学院工学研究科教授．工学博士．人工知能，特に機械学習，マルチメディア処理の研究に従事．人工知能学会，電子情報通信学会，計量国語学会，日本ソフトウェア科学会，AAAI 各会員．