情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2010-MPS-77 No.8
2010/3/4

二次錘計画法による多タスク学習算法

加藤 毅[†1] 鹿島久嗣[†3]
杉山 将[†4] 浅井 潔[†1,†2]

関連する複数のタスクがあるとき，これらを別々に解くより同時に解いたほうが効果的なことがある．このようはアプローチは多タスク学習 (MTL) と呼ばれる．この論文では新しい多タスク学習法を提案する．提案法はすべてのタスクの解が必ず近くなることが保証されるように，タスク間の関係を局所的に制御する．この考えをサポートベクトルマシンに適用すると，その最適化問題は二次錘計画問題で表現できることを示す．多クラス識別，順序回帰，リンク予測の問題が多タスク学習問題として扱えることを示し，実験により提案法の有用性を示す．

An SOCP Formulation for Multi-Task Learning

Tsuyoshi Kato,[†1] Hisashi Kashima,[†3]
Masashi Sugiyama[†4] and Kiyoshi Asai[†1,†2]

When we have several related tasks, solving them simultaneously has been shown to be more effective than solving them individually. This approach is called *multi-task learning* (MTL). In this paper, we propose a novel MTL algorithm. Our method controls the relatedness among the tasks *locally*, so all pairs of related tasks are guaranteed to have similar solutions. We apply the above idea to support vector machines and show that the optimization problem can be cast as a *second-order cone program*, which is convex and can be solved efficiently. The usefulness of our approach is demonstrated in ordinal regression, and link prediction, each of which can be formulated as a structured multi-task problem.

1. はじめに

多くの応用において，いくつかの関連する学習タスクがある．関連するタスクは共通する要素を共有していることがしばしばあり，これらを同時に解いたほうが，別々に解くより利点があると期待できる．このようなアプローチは，**多タスク学習 (Multi-Task Learning, MTL)** と呼ばれており，理論的にも実験的にも有用性が証明されている[2),3),5),7),9),10),20),23),26),32)]．

典型的には，タスク間の関係は関連タスク間の解の近さを要求することで実現される．しかし，多タスク学習法はこれまで次のような欠点があった．従来法では，タスク間の関連性を，すべてのタスク間の解の距離の合計の上限を与えることで実現させていた[23)]．このような制約は**大域的制約**(Global Constraint)[27)] と呼ばれる．この制約では，関連するタスクの解は必ずしも近くならず，いずれかはかなり離れた解になりうる．

この論文では，上述の短所を克服する新しい多タスク学習法を提案する．提案法では，関連タスクの各ペアに対して，それぞれ上限距離を設定する．この制約を**局所的制約**(Local Constraint) と呼ぶ．さらに，提案法では，タスク関連ネットワークによってあらわされるタスクの関連性を扱うことができる．我々は，この考えをサポートベクトルマシン (SVM) の枠組みに適用すると，学習算法は**二次錘計画問題**(Second-Order Cone Program)[6)] に帰着できることを見出した．二次錘計画問題は，凸計画問題であり，大域解を効率的に計算できる．実験を通して，提案法は既存の多タスク学習法より予測性能が良いことを示す．

多タスク学習の考え方は様々な応用において有用である：

- **多クラス識別:** 複数のクラスからクラスラベルを予測する[1)]，
- **順序回帰:** ユーザの好み (「好き」「普通」「嫌い」) など順序のあるクラスラベルを予測する[23),25)]，
- **リンク予測:** 部分的にリンクの有無が分かっているネットワークに対して，残りのリンクの有無を予測する[4),17),29),31)]，

†1 東京大学大学院新領域創成科学研究科
Graduate School of Frontier Sciences, University of Tokyo
†2 産総研生命情報工学研究センター
AIST Computational Biology Research Center
†3 東京大学情報理工学系研究科
Department of Mathematical Informatics, University of Tokyo
†4 東京工業大学大学院情報理工学研究科
Graduate School of Information Science and Engineering, Tokyo Institute of Technology

Vol.2010-MPS-77 No.8
2010/3/4

• **協調フィルタリング:** 多くのユーザの好みの情報を使って商品の推薦などを行う[8]. 提案法は多クラス識別問題，順序回帰，リンク予測において従来法を上回る予測性能が有することを実験を通して示す．協調フィルタリングに関しても良好な実験結果は得られているが，紙面の制約により割愛する．定理の証明も割愛せざるを得なかったが，文献[18] には掲載したので，そちらを参照されたい．

問 題 設 定

M 個の 2 クラス識別タスクを考える．それらのタスクは入出力空間 $\mathcal{X} \times \{\pm 1\}$ を共有している[30]．当分の間，簡単のため，$\mathcal{X} \subset \mathbb{R}^d$ を仮定する．第 3 節以降，$\mathbb{R}^d$ を再生カーネルヒルベルト空間に拡張する．M タスクを学習するとし，第 i タスクは n_i 個の学習用例題 $(\boldsymbol{x}_{t,i}, y_{t,i}) \in \mathcal{X} \times \{\pm 1\} \quad (t = 1, \ldots, n_i)$ をもつ．例題の総数を ℓ とする：$\ell \equiv \sum_{i=1}^{M} n_i$. 最終的に得たいものは各識別タスクのスコア関数

$$f_i(\boldsymbol{x}; \boldsymbol{w}_i, b_i) = \boldsymbol{w}_i^\top \boldsymbol{x} + b_i, \quad \text{for } i = 1, \ldots, M,$$

である．ただし，$\boldsymbol{w}_i \in \mathbb{R}^d$ および $b_i \in \mathbb{R}$ は第 i タスクのモデルパラメータである．

2. タスクネットワークを使った局所的多タスク学習：線形版

本節では，新しい多タスク学習法を提案する．

2.1 基本的な考え方

タスク間に関連がない時，M 個の学習タスクを別々に解くことができる：

$$\forall i = 1, \ldots, M : \quad \frac{1}{2}\|\boldsymbol{w}_i\|^2 + C_\alpha \sum_{t=1}^{n_i} \mathrm{Hinge}(f_i(\boldsymbol{x}_{t,i}; \boldsymbol{w}_i, b_i), y_{t,i}). \tag{1}$$

ただし，$C_\alpha \in \mathbb{R}_+$ は正則化定数であり，$\mathrm{Hinge}(\cdot, \cdot)$ はヒンジ損失関数と呼ばれ：

$$\mathrm{Hinge}(f, y) \equiv \max(1 - fy, 0)$$

のように定義されている．式 (1) はサポートベクトルマシン (SVM)[28] として知られている．第 1 項目は 2 つのクラスを分けるマージンの単調減少関数である．SVM は 2 つのクラスを分ける超平面のうちマージンが最大となるものを見つける．

もしそれぞれのタスクの学習用例題が少ないと，この個別に解く方法の性能は悪くなる．性能を良くするにはより多くの例題が必要となる．

この問題を克服するために，我々は関連する解が互いに近くなるように細工を施す．つまり，最適化問題 (1) 上で次の制約を付け加えることとする：

$$\forall i, \forall j \ : \ \frac{1}{2}\|\boldsymbol{w}_i - \boldsymbol{w}_j\|^2 \le \rho. \tag{2}$$

つまり，タスクの解の差に上限 $\rho \in \mathbb{R}_+$ を与えるのである．この制約を**局所的制約**と呼ぶ[27]．しかし，b_i には制約を加えないことにする．なぜなら，これはタスク間で値が大きく異なることがあるからである．本論文では，一つの上限 ρ のみを使うことにするが，タスクのペアごとに異なる上限 $\rho_{i,j}$ を与えるように一般化するのは簡単である．この制約 (2) によって，タスク間で訓練用例題を共有することになり，結果的に訓練用例題が増えるような効果を持つ．

式 (1) と (2) を組み合わせると

$$\frac{1}{2M}\sum_{i=1}^{M}\|\boldsymbol{w}_i\|^2 + C_\alpha \sum_{i=1}^{M}\sum_{t=1}^{n_i} \mathrm{Hinge}(f_i(\boldsymbol{x}_{t,i}; \boldsymbol{\theta}), y_{t,i}) + C_\rho \rho,$$

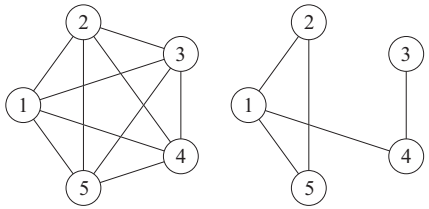
を得る．ただし，$C_\rho \in \mathbb{R}_+$ は非負の定数である．

すると，学習のための最適化問題は次のようになる：

$$\begin{aligned}
\min \quad & \frac{1}{2M}\sum_{i=1}^{M}\|\boldsymbol{w}_i\|^2 + C_\alpha\|\boldsymbol{\xi}\|_1 + C_\rho\rho, \\
\text{wrt} \quad & \boldsymbol{w} \in \mathbb{R}^{Md},\ \boldsymbol{b} \in \mathbb{R}^M,\ \boldsymbol{\xi}_\alpha \in \mathbb{R}_+^\ell,\ \rho \in \mathbb{R}_+, \\
\text{subj. to} \quad & \forall i, \forall j \in \mathbb{N}_M \ : \ \frac{1}{2}\|\boldsymbol{w}_i - \boldsymbol{w}_j\|^2 \le \rho, \\
& \forall i \in \mathbb{N}_M, \forall t \in \mathbb{N}_{n_i} \ : \ y_{t,i}\left(\boldsymbol{w}_i^\top \boldsymbol{x}_{t,i} + b_i\right) \ge 1 - \xi_{t,i}^\alpha, \\
\text{where} \quad & \boldsymbol{w} \equiv \left[\boldsymbol{w}_1^\top, \ldots, \boldsymbol{w}_M^\top\right]^\top, \qquad \boldsymbol{\xi}_\alpha \equiv \left[\xi_{1,1}^\alpha, \ldots, \xi_{n_1,1}^\alpha, \xi_{1,2}^\alpha, \ldots, \xi_{n_M,M}^\alpha\right]^\top.
\end{aligned} \tag{3}$$

一般に，制約が少なければ凸問題は高速に解くことができる．問題 (3) の欠点は $\boldsymbol{w}_i$ のすべてのペアに制約があることである．したがって，この問題は多大な計算コストがかかる．タスクをノードとする全結合のネットワークでこの制約をあらわすとする (図 1(a) 参照)．後の節で，エッジをいくつか削っても性能が下がらないことを実験的に示す．残ったエッジの集合を $\mathcal{E} \equiv \{i_k, j_k\}_{k=1}^K$ とあらわすことにする．すると，最適化問題は次のようになる：

Vol.2010-MPS-77 No.8
2010/3/4



(a) full (b) network

図 1 MTL-SVM の主要な特徴はそれぞれのタスクペアのモデルパラメータの差に上限を設けることである．MTL-SVM (local/full) の上限は (a) に示すようにすべてのタスクペアに与えられる．MTL-SVM (local/network) の上限は (b) に示すように一部のペアだけに設けられる．MTL-SVM (global/full) および MTL-SVM (global/network) の上限は，それぞれ，すべてのタスクペアの差の合計，および，一部のタスクペアの差の合計に与えられる．

$$
\begin{aligned}
\min \quad & \frac{1}{2M}\sum_{i=1}^{M}\|\boldsymbol{w}_i\|^2 + C_\alpha\|\boldsymbol{\xi}\|_1 + C_\rho\rho, && (4)\\
\text{wrt} \quad & \boldsymbol{w}\in\mathbb{R}^{Md},\ \boldsymbol{b}\in\mathbb{R}^{M},\ \boldsymbol{\xi}_\alpha\in\mathbb{R}_+^{\ell},\ \rho\in\mathbb{R}_+,\\
\text{subj. to} \quad & \forall k\in\mathbb{N}_K:\frac{1}{2}\|\boldsymbol{w}_{i_k}-\boldsymbol{w}_{j_k}\|^2\le\rho,\\
& \forall i\in\mathbb{N}_M,\forall t\in\mathbb{N}_{n_i}:y_{t,i}\left(\boldsymbol{w}_i^\top\boldsymbol{x}_{t,i}+b_i\right)\ge 1-\xi^\alpha_{t,i},\\
\text{where} \quad & \boldsymbol{w}\equiv\left[\boldsymbol{w}_1^\top,\ldots,\boldsymbol{w}_M^\top\right]^\top,\qquad \boldsymbol{\xi}_\alpha\equiv\left[\xi^\alpha_{1,1},\ldots,\xi^\alpha_{n_1,1},\xi^\alpha_{1,2},\ldots,\xi^\alpha_{n_M,M}\right]^\top.
\end{aligned}
$$

今後，エッジ集合を $\mathcal{E}$ とするネットワークを**タスクネットワーク**と呼ぶことにする (図 1(b) 参照)．タスクネットワークの自動獲得も挑戦的な課題ではあるが，本研究では，タスクネットワークは事前に与えられていると仮定する．

3. タスクネットワークを使った局所的多タスク学習：カーネル版

本節では，非線形な識別面も学習できるようカーネルトリックを提案法に適用する．

3.1 双対形式

$\boldsymbol{K}_{\rm fea}$ を半正定値行列とし，その (s,t) 要素は特徴ベクトル $\boldsymbol{x}_s$ と $\boldsymbol{x}_t$ の内積とする：

$$K^{\rm fea}_{s,t}\equiv\langle\boldsymbol{x}_s,\boldsymbol{x}_t\rangle.$$

これは特徴ベクトルのカーネル行列である．タスク間のカーネルを

$$\boldsymbol{K}_{\rm net}(\boldsymbol{\lambda})\equiv\left(\frac{1}{M}\boldsymbol{I}_M+\mathcal{U}\boldsymbol{\lambda}\right)^{-1},$$

と定義する．ただし，$\boldsymbol{\lambda}\in\mathbb{R}_+^K$ は K 次元のパラメータベクトルである．ここで，

$$\mathcal{U}\boldsymbol{\lambda}\equiv\sum_{k=1}^{K}\lambda_k\boldsymbol{U}_k,\qquad \boldsymbol{U}_k\equiv\boldsymbol{E}^{i_ki_k}+\boldsymbol{E}^{j_kj_k}-\boldsymbol{E}^{i_kj_k}-\boldsymbol{E}^{j_ki_k}$$

と定義した．$\boldsymbol{E}^{(i,j)}\in\mathbb{R}^{M\times M}$ は，(i,j) 要素のみが 1 で，それ以外は 0 の行列である．これは第 k エッジの重みが λ_k のグラフラプラシアンカーネル[33] として知られている．行列 $\boldsymbol{Z}\in\mathbb{N}^{M\times\ell}$ をそれぞれの例題がどのタスクに対応するか示すように次のように定義する：

$$\boldsymbol{Z}^\top\equiv\begin{bmatrix}\mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \cdots & \mathbf{0}_{n_1}\\ \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \cdots & \mathbf{0}_{n_2}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0}_{n_M} & \mathbf{0}_{n_M} & \cdots & \mathbf{1}_{n_M}\end{bmatrix}.$$

すると，タスクの情報は $\ell\times\ell$ のカーネル行列 $\boldsymbol{Z}^\top\boldsymbol{K}_{\rm net}(\boldsymbol{\lambda})\boldsymbol{Z}$ によってあらわされる．アダマール積 $\circ$ を用いて，この 2 つのカーネル行列 $\boldsymbol{K}_{\rm fea}$ および $\boldsymbol{Z}^\top\boldsymbol{K}_{\rm net}(\boldsymbol{\lambda})\boldsymbol{Z}$ は

$$\boldsymbol{K}_{\rm int}(\boldsymbol{\lambda})\equiv\boldsymbol{K}_{\rm fea}\circ\left(\boldsymbol{Z}^\top\boldsymbol{K}_{\rm net}(\boldsymbol{\lambda})\boldsymbol{Z}\right), \qquad (5)$$

と統合できる．このパラメータつき行列 $\boldsymbol{K}_{\rm int}(\boldsymbol{\lambda})$ は，$\boldsymbol{\lambda}\ge\mathbf{0}_K$ である限り，半正定値であることが保証されている[13]．

次の定理に示すように，上記の記法を用いると，双対形式を簡潔に記述できる：

Theorem 3.1. 主問題 (4) の双対問題はパラメータつき統合カーネル行列 $\boldsymbol{K}_{int}(\boldsymbol{\lambda})$ を使って次のようにあらわされる．

$$
\begin{aligned}
min \quad & \frac{1}{2}\boldsymbol{\alpha}^\top\operatorname{diag}(\boldsymbol{y})\boldsymbol{K}_{int}(\boldsymbol{\lambda})\operatorname{diag}(\boldsymbol{y})\boldsymbol{\alpha}-\|\boldsymbol{\alpha}\|_1, && (6)\\
wrt \quad & \boldsymbol{\alpha}\in\mathbb{R}_+^{\ell},\boldsymbol{\lambda}\in\mathbb{R}_+^{K},\\
subj.\ to \quad & \boldsymbol{\alpha}\le C_\alpha\mathbf{1}_\ell,\quad \boldsymbol{Z}\operatorname{diag}(\boldsymbol{y})\,\boldsymbol{\alpha}=\mathbf{0}_M,\quad \|\boldsymbol{\lambda}\|_1\le C_\rho.
\end{aligned}
$$

□

$\forall i,\mathbb{N}_M\,\forall t\in\mathbb{N}_{n_i}$ に対して，$\alpha_{t,i}$ は

$$\boldsymbol{\alpha}=\left[\alpha_{1,1},\ldots,\alpha_{n_1,1},\alpha_{1,2},\ldots,\alpha_{n_{M-1},M-1},\alpha_{1,M},\ldots,\alpha_{n_M,M}\right].$$

を満たすような双対変数 $\boldsymbol{\alpha} \in \mathbb{R}_+^\ell$ の要素である．

証明は，文献[18] を参照されたい．$\boldsymbol{\alpha}$ および $\boldsymbol{\lambda}$ の最適解は，ℓ_1 ノルムの影響から，疎になる傾向にある．次の定理により，提案法は二次錘計画法の既製のエンジンを使って効率的に学習することができる．

Theorem 3.2. 双対問題 (6) は二次錘計画問題に帰着される． □

カーネル行列 $\boldsymbol{K}_{\mathrm{fea}}$ の定義を線形カーネルから非線形カーネルに変えることによって，提案する多タスク学習法は非線形な識別面も得られるようになる．さらに，文字列カーネルやグラフカーネル[11),12),14)–16),19),21),22)] などを使うことによって，ベクトルではない構造データも扱うことができる．

予測の段階では，第 j タスクに属する未知の例題 $\boldsymbol{x}$ は

$$f_j(\boldsymbol{x}) = \sum_{i=1}^{M} \sum_{t=1}^{n_i} \alpha_{t,i} y_{t,i} k_{\mathrm{fea}}(\boldsymbol{x}_{t,i}, \boldsymbol{x}) k_{\mathrm{net}}(i,j) + b_j,$$

によって識別できる．ただし，$k_{\mathrm{fea}}(\cdot,\cdot)$ および $k_{\mathrm{net}}(\cdot,\cdot)$ は，それぞれ，特徴のカーネル関数とタスクのカーネル関数である．

4. 議　　論

4.1 標準的な SVM との関係

提案する多タスク学習法は標準的な SVM を特殊ケースとして含む．実際に，タスク数が 1 のとき，問題 (6) は標準的な SVM 最適化問題に帰着される．このように，提案法は SVM の自然な拡張とみることができる．

タスクネットワークにエッジが全くなかったとき，提案法は，個別に SVM を訓練する場合と全く等価になる．タスクネットワークのエッジをなくした SVM を **Individually Learned SVM (IL-SVM)** と呼ぶことにする．

4.2 大域的制約・局所的制約

本研究に先んじて，多タスク学習法のための異なる算法がすでに提案されていた[9),10)]．その手法は関連するタスク間のすべてのペアに対して，解の距離の合計に上限 $\frac{1}{2}\sum_{i,j=1}^{M} \|\boldsymbol{w}_i - \boldsymbol{w}_j\|^2 \leq \rho.$ を与えるものであった．合計に上限を与える制約を**大域的制約**[27)] と呼ぶ．タスク間のすべてのペアに対する合計に上限を与えることから，タスクネットワークは完全結合しているとみることができる．このような理由からこのアプローチを **MTL-SVM (global/full)** と呼ぶことにする．大域的制約は距離の合計に上限を与えているだけなので，一部の距離が大きくなることを許してしまう．実際に，これが顕著に性能を低下させてしまうことを第 5 節にて実験的に示す．対照的に，我々は各タスクペアの距離に上限を与えている．これによって，すべてのタスクペアに対して解が近くなることが保証される．各ペアに上限をかけることを**局所的制約**[27)] と呼び，我々のアプローチを **MTL-SVM (local/full)** と呼ぶことにする．

Micchelli & Pontil[23)] は強い関連のあるペアだけに適用するほかの定式化も与えている：$\frac{1}{2}\sum_{k=1}^{K} \|\boldsymbol{w}_{i_k} - \boldsymbol{w}_{j_k}\|^2 \leq \rho.$ ただし，各々の $k = 1, \ldots, K$ に対して，第 i_k タスクと第 j_k タスクは強く関連しているが，それ以外はそうではないとしている．この定式化はタスクネットワークの情報を利用するので，このアプローチを **MTL-SVM (global/network)** と呼ぶ．我々の局所的制約による方法もタスクネットワークを $\forall k \in \mathbb{N}_K : \frac{1}{2}\|\boldsymbol{w}_{i_k} - \boldsymbol{w}_{j_k}\|^2 \leq \rho$ のように利用できる．このアプローチを **MTL-SVM (local/network)** と呼ぶ．この 4 つの方法は図 1 にまとめた．

4.3 リンク予測

多タスク学習はリンク予測にも応用できる．n ノードの無向グラフ $\mathcal{G}$ が与えられた．ノードの集合を $\mathcal{V} = \{1, \ldots, n\}$ であらわし，すべてのノードのペアを $\mathcal{P} \equiv \left\{(i,j) \,\middle|\, 1 \leq i < j \leq n\right\}$ であらわすことにする．$\mathcal{P}$ の要素数は $n(n-1)/2$ である．リンクの集合 $\mathcal{E}$ は $\mathcal{P}$ の部分集合であり，その補集合 $\bar{\mathcal{E}} \equiv \mathcal{P} \setminus \mathcal{E}$ にあるペアはリンクしていない．リンク情報を

$$y_i^j = \begin{cases} +1 & \text{if } (i,j) \in \mathcal{E}, \\ -1 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$$

であらわす．それぞれのノードには特徴ベクトル $\boldsymbol{x} \in \mathbb{R}^d$ が与えられているとする．$\mathcal{P}$ の部分集合 $\mathcal{P}_{\mathrm{tra}}$ にあるノードペアはリンクの有無が既知であるとする．つまり，$\mathcal{P}_{\mathrm{tra}}$ にあるノードペアはクラスラベルが分かっている．リンク予測は，$\mathcal{P}_{\mathrm{tra}}$ から，残りの集合 $\mathcal{P} \setminus \mathcal{P}_{\mathrm{tra}}$ にあるノードペアのリンクの有無を予測する問題ということができる．

一つの標準的な方法は**大域的モデル**[17),29),31)] を使うことである．それは，ネットワーク全体で一つのモデルを構築するものである．しかし，大域的モデルは関連のない情報を取り込みやすい．この問題を解決するために，Bleakley ら[4)] は**局所的モデル**を使うことを提案している．局所的モデルとは一つのノードに対して一つのモデルを作るものである．この方法では，一つの局所的モデルは局所的な情報のみから訓練されるので，無関係な情報の影響を受けにくい．その代わり，訓練に必要な情報は限られるので，この方法から得られる予測値の精度は信頼しがたい．

Vol.2010-MPS-77 No.8
2010/3/4

これに対して，我々は多タスク学習を使うことを提案する．ある指定された局所的モデルを訓練するために，リンクしているほかのノードに対応する局所的モデルを利用する．これは複雑ネットワークの性質[24] を生かしたものになっている．一方，Bleakley らの方法[4] は IL-SVM に対応する．

5. 実　　験

提案法を多クラス識別問題，順序回帰，リンク予測に適用した結果をそれぞれ表 1, 3, 4 に示す．また，計算時間の比較を表 2 に示す．詳細な実験条件は文献[18] を参照されたい．多くの場合，提案法がほかの方法より性能が上回っている．MTL-SVM(local/full) と MTL-SVM(local/network) の間には統計的有意な差はあまり見られなかったが，タスクネットワークを用いることにより計算時間が激減していることが分かる．協調フィルタリングへの応用でも提案法が優れた性能を示したが，その実験結果は紙面の制約から割愛する．

参 考 文 献

1) Amit, Y., Fink, M., Srebro, N. and Ullman, S.: Uncovering shared structures in multiclass classification, *Proceedings of the 24th International Conference on Machine Learning*, pp.17–24 (2007).
2) Bakker, B. and Heskes, T.: Task clustering and gating for Bayesian multitask learning, *Journal of Machine Learning Research*, Vol.4, pp.83–99 (2003).
3) Baxter, J.: A Model of Inductive Bias Learning, *Journal of Artificial Intelligence Research*, Vol.12, pp.149–198 (2000).
4) Bleakley, K., Biau, G. and Vert, J.-P.: Supervised Reconstruction of Biological Networks with Local Models, *Bioinformatics*, Vol.23, No.13, pp.i57–i65 (2007).
5) Bonilla, E.V., Agakov, F.V. and Williams, C.K.I.: Kernel Multi-task Learning using Task-specific Features, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, pp.43–50 (2007).
6) Boyd, S. and Vandenberghe, L.: *Convex Optimization*, Cambridge University Press (2004).
7) Caruana, R.: Multitask Learning, *Machine Learning*, Vol. 28, No. 1, pp. 41–75 (1997).
8) Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R.: Indexing by latent semantic analysis., *Journal of the American Society for Information Science*, Vol.41, No.6, pp.391–407 (1990).
9) Evgeniou, T., Micchelli, C.A. and Pontil, M.: Learning Multiple Tasks with Kernel Methods, *Journal of Machine Learning Research*, Vol.6, pp.615–637 (2005).
10) Evgeniou, T. and Pontil, M.: Regularized Multitask Learning, *Proceedings of the 17th SIGKDD Conference on Knowledge Discovery and Data Mining*, pp.109–117 (2004).
11) Gärtner, T.: A Survey of Kernels for Structured Data, *SIGKDD Explorations*, Vol.5, No.1, pp.S268–S275 (2003).
12) Gärtner, T., Flach, P. and Wrobel, S.: On Graph Kernels: Hardness Results and Efficient Alternatives, *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory*, pp.129–143 (2003).
13) Haussler, D.: Convolution kernels on discrete structures, Technical Report UCSC-CRL-99-10, UC Santa Cruz (1999).
14) Jaakkola, T. and Haussler, D.: Exploiting Generative Models in Discriminative Classifiers, *Advances in Neural Information Processing Systems 11* (Kearns, M.S., Solla, S.A. and Cohn, D.A., eds.), Cambridge, MA., MIT Press, pp.487–493 (1999).
15) Kashima, H. and Koyanagi, T.: Kernels for Semi-Structured Data, *Proceedings of the Nineteenth International Conference on Machine Learning*, pp.291–298 (2002).
16) Kashima, H., Tsuda, K. and Inokuchi, A.: Marginalized Kernels between Labeled Graphs, *Proceedings of the Twentieth International Conference on Machine Learning*, pp.321–328 (2003).
17) Kato, T., Tsuda, K. and Asai, K.: Selective integration of multiple biological data for supervised network inference, *Bioinformatics*, Vol.21, pp.2488–2495 (2005).
18) Kato, T., Kashima, H., Sugiyama, M. and Asai, K.: Conic Programming for Multi-Task Learning, *IEEE Transactions on Knowledge and Data Engineering*.
19) Kondor, R.I. and Lafferty, J.: Diffusion Kernels on Graphs and Other Discrete Input Spaces, *Proceedings of the Nineteenth International Conference on Machine Learning*, pp.315–322 (2002).
20) Lawrence, N.D. and Platt, J.C.: Learning to learn with the informative vector machine, *Proceedings of the Twenty First International Conference on Machine Learning*, pp.512–519 (2004).
21) Leslie, C., Eskin, E. and Noble, W.S.: The Spectrum Kernel: A String Kernel for SVM Protein Classification, *Proceedings of the Pacific Symposium on Biocomputing*, pp.566–575 (2002).
22) Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. and Watkins, C.: Text Classification Using String Kernels, *Journal of Machine Learning Research*, Vol.2, pp.419–444 (2002).
23) Micchelli, C.A. and Pontil, M.: Kernels for Multi-task Learning, *Advances in Neural Information Processing Systems 17*, Cambridge, MA, MIT Press, pp.921–928 (2005).
24) Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D. and Alon, U.:

表 1 タンパク質スーパーファミリー識別．タスクネットワークは無作為に生成した木である．太字は最高性能，下線は最高性能と統計的有意差がないことを示す．

Class	IL-SVM	MTL-SVM (global/full)	MTL-SVM (global/network)	MTL-SVM (local/full)	MTL-SVM (local/network)
1	0.893 (0.033)	**0.896** (0.032)	0.885 (0.053)	0.887 (0.040)	0.884 (0.043)
2	0.803 (0.066)	0.827 (0.040)	0.820 (0.043)	**0.833** (0.035)	0.831 (0.038)
3	0.916 (0.013)	0.919 (0.016)	0.918 (0.018)	**0.926** (0.012)	0.925 (0.012)
4	0.702 (0.103)	0.762 (0.046)	0.769 (0.041)	**0.776** (0.055)	0.771 (0.067)
5	0.938 (0.030)	**0.948** (0.024)	0.947 (0.027)	0.942 (0.023)	0.942 (0.026)
6	0.755 (0.085)	0.763 (0.057)	0.782 (0.041)	**0.786** (0.048)	0.785 (0.061)
7	0.591 (0.061)	0.612 (0.067)	0.616 (0.073)	**0.629** (0.055)	0.627 (0.060)
ave	0.800 (0.024)	0.818 (0.019)	0.819 (0.021)	**0.825** (0.017)	0.823 (0.020)

Network Motifs: Simple Building Blocks of Complex Networks, *Science*, Vol.298, pp.824–827 (2002).
25) Shashua, A. and Levin, A.: Ranking with large margin principle: two approaches, *Advances in Neural Information Processing Systems 15*, Cambridge, MA, MIT Press, pp.937–944 (2003).
26) Thrun, S. and Pratt, L.: *Learning to Learn*, Springer (1997).
27) Tsuda, K. and Noble, W.S.: Learning kernels from biological networks by maximizing entropy, *Bioinformatics*, Vol.20, No.Suppl. 1, pp.i326–i333 (2004).
28) Vapnik, V.N.: *Statistical Learning Theory*, Wiley, New York (1998).
29) Vert, J.-P. and Yamanishi, Y.: Supervised graph inference, *Advances in Neural Information Processing Systems 17*, Cambridge, MA, MIT Press (2005).
30) Xue, Y., Liao, X., Carin, L. and Krishnapuram, B.: Multi-Task Learning for Classification with Dirichlet Process Priors, *Journal of Machine Learning Research*, Vol.8, pp.35–63 (2007).
31) Yamanishi, Y., Vert, J.P. and Kanehisa, M.: Supervised enzyme network inference from the integration of genomic data and chemical information, *Bioinformatics*, Vol.21 Suppl.1, pp.i468–i477 (2005).
32) Yu, K., Tresp, V. and Schwaighofer, A.: Learning Gaussian Processes from Multiple Tasks, *Proceedings of the 22nd International Conference on Machine Learning*, pp.1012–1019 (2005).
33) Zhu, X., Kandola, J., Ghahramani, Z. and Lafferty, J.: Nonparametric transforms of graph kernels for semi-supervised learning, *Advances in Neural Information Processing Systems 17*, Cambridge, MA, MIT Press, pp.1641–1648 (2004).

表 2 計算時間．

Class	IL-SVM	MTL-SVM (global/full)	MTL-SVM (global/network)	MTL-SVM (local/full)	MTL-SVM (local/network)
Time (sec)	0.069 (0.006)	0.269 (0.007)	0.288 (0.016)	**2.339** (0.089)	0.884 (0.057)

表 3 順序回帰における正解率．太字は最高性能，下線は最高性能と統計的有意差がないことを示す．

Dataset	SVOR	IL-SVM	MTL-SVM (global/full)	MTL-SVM (global/network)	MTL-SVM (local/full)	MTL-SVM (local/network)
abalone	0.965 (0.016)	0.965 (0.016)	0.965 (0.016)	0.965 (0.016)	**0.972** (0.012)	0.966 (0.016)
bodyfat	0.957 (0.016)	0.958 (0.013)	0.958 (0.016)	0.958 (0.015)	**0.962** (0.013)	**0.962** (0.013)
cadata	0.974 (0.010)	0.974 (0.010)	0.974 (0.010)	0.974 (0.010)	**0.975** (0.009)	**0.975** (0.010)
housing	0.969 (0.012)	0.970 (0.012)	0.969 (0.012)	0.969 (0.012)	**0.973** (0.011)	0.972 (0.011)
mg	0.969 (0.010)	0.969 (0.010)	0.969 (0.010)	0.969 (0.010)	0.969 (0.009)	**0.970** (0.010)
mpg	0.964 (0.012)	0.966 (0.012)	0.963 (0.011)	0.963 (0.011)	**0.969** (0.011)	**0.969** (0.011)

表 4 酵素ネットワーク予測における ROC カーブの AUC．太字は最高性能，下線は最高性能と統計的有意差がないことを示す．

Dataset	IL-SVM	MTL-SVM (global/full)	MTL-SVM (global/network)	MTL-SVM (local/full)	MTL-SVM (local/network)
ady	0.733 (0.113)	0.744 (0.122)	0.746 (0.120)	n/a	**0.752** (0.118)
blast	0.786 (0.102)	0.792 (0.104)	0.800 (0.097)	n/a	**0.806** (0.098)
diff	0.620 (0.130)	0.630 (0.124)	0.639 (0.122)	n/a	**0.654** (0.109)
expr	0.630 (0.104)	0.635 (0.109)	0.636 (0.107)	n/a	**0.645** (0.100)
fft	0.652 (0.111)	0.663 (0.112)	0.668 (0.109)	n/a	**0.675** (0.103)
lin_int	0.588 (0.111)	0.609 (0.129)	0.611 (0.127)	n/a	**0.614** (0.119)
pfam_hmm	0.740 (0.123)	0.748 (0.123)	0.749 (0.125)	n/a	**0.760** (0.124)
sw	0.732 (0.126)	0.731 (0.135)	0.725 (0.146)	n/a	**0.753** (0.122)