データの転送制御に基づく効率的な分散型 SVM の学習法に関する研究

情報数理応用研究　　5214C043-3　湯川輝一朗
指導教員　後藤正幸

A Study of Learning Method of Distributed Support Vector Machine Based on Transfer Control of Data

YUKAWA Kiichiro

1　研究背景・目的

近年の情報ネットワーク技術の発展に伴い，大量のデータを蓄積することが可能となるとともに多種多様な大量のデータが物理的に離れた複数のデータベースに分散蓄積されるようになった．各データベースは，情報ネットワークを通じ，オンラインで容易に接続することが可能である．蓄積された大量のデータは企業のマーケティング活動や経営戦略への活用が期待されていることから，データ分析の重要性が増加している．分析を行う際に，1 つのデータベースに蓄積されたデータのみを用いて分析を行うよりも，分散蓄積された全てのデータを用いることで多種多様な大量のデータを様々な分析に用いることができる．そのため，分散蓄積されたデータを全て用いて分析を行う必要性が高まっている．分散保持された生データを 1ヶ所に集約し，分析を行うことは可能であるが，データの規模性，通信コスト，プライバシーなどの問題が存在する．そこで，分散保持された生データを共有せずに分析を行う分散データマイニング (DDM) が重要視されている [1].

DDM では，既に様々な分析手法が提案されているが，本研究では水平分割されたデータを対象とした分散型 Support Vector Machine (SVM) に着目する．水平分割されたデータとは，同じ特徴（説明変数）に対して，各データベースが異なる学習データを保持している構造のデータを指す．上記の手法の 1 つに Forrero らによる Consensus-Based Distributed Support Vector Machines (D-SVM) [2] がある．D-SVM は任意のネットワークモデルのもと，分散保持された学習データに対し少ない計算コストで全体で最適な 1 つの SVM のパラメータを学習する手法である．また，与えられたネットワーク構造と各データベースが蓄積するデータの統計的特徴の組み合わせが，学習に必要な計算コストに影響を与えることが知られている．

一方，現在の情報ネットワークでは，インターネットなどに代表されるように，各データベースが各々全て結合されているようなフルコネクト型ネットワークが一般的に普及している．任意のネットワーク構造に対応可能な手法よりも，フルコネクト型ネットワークを仮定することで，大幅に学習の効率化を図ることができるのであれば，実問題への適用においても有効となる可能性が高い．そこで本研究では，フルコネクト型ネットワークモデルを前提とし，パラメータ共有を行うデータベースを選択することで D-SVM の学習に必要な計算回数と通信コストの削減が可能な方法を提案する．人工データと機械学習のベンチマークデータを用いた評価実験を行い，計算量と通信コストの観点から提案手法の有用性を示す．

2　準備

2.1　ネットワークモデル

ネットワークモデルとは，コンピュータやサーバをネットワーク上のノード，ノード間の接続をエッジ，2 つのノードを繋ぐエッジ集合をパスと呼び，その関係性をグラフ構造により表現したものである．本研究では，各データベースをノード，ノード間の接続をエッジとし，ノード間の関係を隣接行列を用いて表現する．いま，ノード数を J，ノード集合を $\mathcal{J} = \{1, \cdots, J\}$, 隣接行列を $\boldsymbol{E} = [e_{ij}] \in \mathbb{R}^{J \times J}$, j 番目のノードとの間にエッジがあるノード（隣接ノード）の集合を $\mathcal{B}_j$, ノード j の基数（隣接ノード数）を $|\mathcal{B}_j|$ とする．e_{ij} はノード i ($i \in \mathcal{J}$) と j ($j \in \mathcal{J}$) 間にエッジが存在するか否かを示しており，以下の式 (1) で定義される．

$$e_{ij} = \begin{cases} 1 & if \ \ i \in \mathcal{B}_j \wedge i \neq j, \\ 0 & otherwise. \end{cases} \tag{1}$$

本研究で扱うネットワークモデルは連結グラフである必要がある．ここで，連結グラフとは任意の 2 つのノード間にパスが存在するネットワークモデルの総称を指す．

2.2　分散データマイニングとデータ構造

分散データマイニング (DDM) とは，異なるデータベースに分散蓄積されたデータ全体を用いて分析し，モデルの学習等を行う手法の総称である．DDM の重要な特徴として，各データベースに蓄積された生データを共有しないこと，生データを 1ヶ所に集約し分析を行った場合と同様の結果を得ることが挙げられる．

一般的に，DDM において各ノードが蓄積するデータ構造には 2 種類存在する．1 つは垂直分割モデル，もう 1 つは水平分割モデルである．本研究では，このうち水平分割モデルを対象とする．水平分割モデルについて，学習データ行列を $\boldsymbol{A} = [a_{ij}] \in \mathbb{R}^{N \times d}$ とし，それを J 個に分割する場合を例に述べる．ただし，N は総学習データ数，d は次元数，a_{ij} は行列 $\boldsymbol{A}$ の (i, j) 番目の要素を示している．水平分割モデルは，同じ特徴に関して各ノードが異なる学習データを蓄積しているデータ構造である．このとき，ノード k は学習データ行列として $\boldsymbol{A}_k = [a^k_{ij}] \in \mathbb{R}^{N_k \times d}$ を保持しているものとする．ただし，N_k はノード k が所持する学習データの数であり，$N = \sum_{k=1}^{J} N_k$ を満たす．

2.3　Support Vector Machine(SVM)

SVM は，優れた識別性能を持つ二値判別器である．SVM では，与えられた学習データに対して識別境界 (マージン) が最大になるように以下の式 (2) で定義される識別関数 $g(\boldsymbol{x})$ のパラメータの学習を行う．

$$g(\boldsymbol{x}) \quad = \quad w_0 + \boldsymbol{w}^T \boldsymbol{x}. \tag{2}$$

ただし，$\boldsymbol{x}$ は d 次元の特徴ベクトル，$(w_0, \boldsymbol{w}) \in \mathbb{R}^{d+1}$ は SVM のパラメータである．

3 問題設定

本研究では，フルコネクト型ネットワークモデルを仮定し，各ノードに蓄積された生データをノード間で授受することなく大域的な SVM のパラメータを推定することを考える．ここで，フルコネクト型ネットワークモデルでは，各ノードが他の全てのノードとエッジで結ばれているネットワーク構造であり，このときノード j の隣接ノード数は $|\mathcal{B}_j| = J-1$ $(\forall j \in \mathcal{J})$ となる．

いま，$\boldsymbol{x}_{ji} \in \mathbb{R}^d$ と $y_{ji} \in \{-1, +1\}$ をそれぞれノード j の i 番目の d 次元特徴ベクトル，$\boldsymbol{x}_{ji}$ のクラスラベルとしたとき，ノード j には N_j 件の学習データセット $\{\boldsymbol{x}_{ji}, y_{ji}\}_{i=1}^{N_j}$ が蓄積されているものとする．対象とする問題は，フルコネクト型ネットワークモデルのもと，各ノードは保持された生データを直接共有することなく大域的最適な SVM のパラメータ $\boldsymbol{w}$ と定数項 w_0 を推定することである．

4 D-SVM [2]

Forrero らは，Alternating Direction Method of Multipliers (ADMM) を用いることで，各ノードに蓄積された生データを直接共有することなく大域的に最適な SVM のパラメータを推定する手法として D-SVM を提案した．D-SVM では，ローカルアップデートとグローバルアップデートを繰り返し行うことでパラメータの推定を行う．ローカルアップデートでは，各ノードにおいて自身の保持する学習データとグローバルアップデートの結果を用いてローカルに SVM（ローカル SVM）のパラメータを推定する．グローバルアップデートでは，隣接ノード間でローカル SVM のパラメータを共有し，コンセンサスパラメータの更新を行う．ここで，コンセンサスパラメータは自身で推定したローカル SVM のパラメータを大域的な最適パラメータに近づけるためのパラメータである．

いま，J 個に水平分割されたデータを用いて SVM のパラメータを推定することを考える．各ノードが保持する特徴行列を $\boldsymbol{X}_j := \left[[\boldsymbol{x}_{j1}, \cdots, \boldsymbol{x}_{jN_j}]^T, \mathbf{1}_j\right] \in \mathbb{R}^{N_j \times (d+1)}$，クラスラベル行列を $\boldsymbol{Y}_j := \mathrm{diag}\left[(y_{j1}, \cdots, y_{jN_j})\right] \in \mathbb{R}^{N_j \times N_j}$ とする．ただし，$\mathbf{1}_j$ は要素が全て 1 で長さ N_j のベクトルである．ここで，$\boldsymbol{v}_j = (\boldsymbol{w}_j^T, w_{0,j})^T$ をノード j が推定するローカル SVM のパラメータとすると，D-SVM は以下の最適化問題を解くことでパラメータを推定する．

$$\underset{\boldsymbol{v}_j, \boldsymbol{\xi}_j}{\text{minimize}} \ \frac{1}{2}\sum_{j=1}^{J} \boldsymbol{v}_j^T \left(\boldsymbol{I}_{d+1} - \boldsymbol{\Pi}_{d+1}\right)\boldsymbol{v}_j + JC\sum_{j=1}^{J}\mathbf{1}_j^T\boldsymbol{\xi}_j, \tag{3}$$

$$\text{subject to } \boldsymbol{Y}_j\boldsymbol{X}_j\boldsymbol{v}_j \succeq \mathbf{1}_j - \boldsymbol{\xi}_j, \ \forall j \in \mathcal{J}, \tag{4}$$

$$\boldsymbol{\xi}_j \succeq \mathbf{0}, \ \forall j \in \mathcal{J}, \tag{5}$$

$$\boldsymbol{v}_j = \boldsymbol{v}_k, \ \forall j \in \mathcal{J}, \ k \in \mathcal{B}_j. \tag{6}$$

ただし，$C > 0$ はペナルティパラメータ，$\succeq$ は左辺のベクトルの各要素が対応する右辺のベクトルの各要素以上の値をとることを示しており，$\boldsymbol{\xi}_j \in \mathbb{R}^{N_j}$ はノード j の学習データに対応するスラック変数ベクトル，$\boldsymbol{I}_{d+1} \in \mathbb{R}^{(d+1)\times(d+1)}$ は $d+1$ 次元の単位行列，$\boldsymbol{\Pi}_{d+1} \in \mathbb{R}^{(d+1)\times(d+1)}$ は $(d+1, d+1)$ 要素が 1 でそれ以外の要素が 0 の行列である．ここで，式 (3) はマージン最大化のための目的関数，式 (4) は学習データがマージン内に存在しないための制約条件，式 (5) はスラック変数が 0 以上になるための制約条件，式 (6) は各ノードで推定したパラメータが同一になるための制約条件である．この最適化問題は，J 個の部分問題に分割して解くことができる．すなわち，ノード j は制約式 (6) のもとで自身に関与する部分のみを解けばよい．

この最適化問題を ADMM を用いて解くことを考える．$\boldsymbol{\lambda}_j$ と $\boldsymbol{\alpha}_j$ をそれぞれ制約式 (4), (6) に対応するラグランジュ乗数ベクトル，$\rho > 0$ をスケールパラメータとし，拡張ラグランジュ関数 $L_\rho(\boldsymbol{v}_j, \boldsymbol{\xi}_j, \boldsymbol{\lambda}_j, \boldsymbol{\alpha}_j)$ を以下の式 (7) で定義する．

$$\begin{aligned} L_\rho(\boldsymbol{v}_j, \boldsymbol{\xi}_j, \boldsymbol{\lambda}_j, \boldsymbol{\alpha}_j) \ &= \ \frac{1}{2}\sum_{j=1}^{J}\boldsymbol{v}_j^T\left(\boldsymbol{I}_{d+1} - \boldsymbol{\Pi}_{d+1}\right)\boldsymbol{v}_j \\ &+ JC\sum_{i=1}^{N_j}\mathbf{1}_j^T\boldsymbol{\xi}_j + \sum_{i=1}^{J}\boldsymbol{\lambda}_j\left(\mathbf{1}_j - \boldsymbol{\xi}_j - \boldsymbol{Y}_j\boldsymbol{X}_j\boldsymbol{v}_j\right) \\ &+ \sum_{j=1}^{J}\sum_{i\in\mathcal{B}_j}\boldsymbol{\alpha}_j^T(\boldsymbol{v}_j - \boldsymbol{v}_i) + \frac{\rho}{2}\sum_{j=1}^{J}\sum_{i\in\mathcal{B}_j}\left\|(\boldsymbol{v}_j - \boldsymbol{v}_i)\right\|_2^2. \end{aligned} \tag{7}$$

ただし，$\|\cdot\|_2$ はベクトルの l_2 ノルムである．式 (7) の拡張ラグランジュ関数を最小化するために，各ノードは繰り返し計算を行う必要がある．t 回目の繰り返しにおいて，ノード j は以下の式 (8)–(10) を計算する．

$$\begin{aligned} \boldsymbol{\lambda}_j^{(t)} \ &:= \ \underset{\mathbf{0}_j \preceq \boldsymbol{\lambda}_j \preceq JC\mathbf{1}_j}{\arg\max} \ -\frac{1}{2}\boldsymbol{\lambda}_j^T\boldsymbol{Y}_j\boldsymbol{X}_j\boldsymbol{U}_j^{-1}\boldsymbol{X}_j^T\boldsymbol{Y}_j\boldsymbol{\lambda}_j \\ &\qquad + \left(\mathbf{1}_j + \boldsymbol{Y}_j\boldsymbol{X}_j\boldsymbol{U}_j^{-1}\boldsymbol{f}_j^{(t-1)}\right)^T\boldsymbol{\lambda}_j, \end{aligned} \tag{8}$$

$$\boldsymbol{v}_j^{(t)} \ := \ \boldsymbol{U}_j^{-1}\left[\boldsymbol{X}_j^T\boldsymbol{Y}_j\boldsymbol{\lambda}_j^{(t)} - \boldsymbol{f}_j^{(t-1)}\right], \tag{9}$$

$$\boldsymbol{\alpha}_j^{(t)} \ := \ \boldsymbol{\alpha}_j^{(t-1)} + \frac{\eta}{2}\sum_{i\in\mathcal{B}_j}\left[\boldsymbol{v}_j^{(t)} - \boldsymbol{v}_i^{(t)}\right]. \tag{10}$$

ただし，$\boldsymbol{U}_j := (1 + 2\eta|\mathcal{B}_j|)\boldsymbol{I}_{d+1} - \boldsymbol{\Pi}_{d+1}$, $\boldsymbol{f}_j^{(t)} := 2\boldsymbol{\alpha}_j^{(t)} - \eta\sum_{i\in\mathcal{B}_j}[\boldsymbol{v}_j^{(t)} + \boldsymbol{v}_i^{(t)}]$ である．D-SVM は，各ノードが自身が保有するデータとグローバルアップデートの結果を用いてローカル SVM を推定するローカルアップデート，隣接ノード間でローカル SVM のパラメータを共有し，隣接ノードのローカル SVM と近づけるグローバルアップデートの 2 つのステップにより成り立つ．ここで，式 (8)–(9) がローカルアップデート，式 (10) がグローバルアップデートを指している．また，式 (8) の $\boldsymbol{\lambda}_j^{(t)}$ の推定には勾配法などを用いる．D-SVM の推定アルゴリズムを以下に示す．

[D-SVM の推定アルゴリズム]

Step0. 適当な初期値 $\boldsymbol{v}_j^{(0)}(\forall j \in \mathcal{J})$ を設定する．
$\boldsymbol{\alpha}_j^{(0)} = \mathbf{0}_j$ として，$t = 1$ とする．

Step1. 式 (8)–(9) を用いて，$\boldsymbol{v}_j^{(t)}$ を推定する．

Step2. 隣接ノード間で $\boldsymbol{v}_j^{(t)}$ を共有する．

Step3. 式 (10) を用いて $\boldsymbol{\alpha}_j^{(t)}$ を計算する．

Step4. 収束条件を満たせば，終了．さもなくば，$t = t+1$ として **Step1** へ戻る．

□

5 提案手法

5.1 準備

従来手法は，与えられた連結グラフを満たすネットワークモデルに対してノード間のパラメータ共有とローカル SVM の繰り返し計算によって大域的最適な SVM のパラメータを推定可能である．ここで，所与のネットワークモデルは与えられた物理的ネットワークの構造によって決まる制約であり，各ノードに蓄積されるデータの統計的特徴とは無関係である．したがって，与えられたネット

ワーク構造と各ノードが蓄積するデータの統計的特徴の組み合わせによってはパラメータを推定する際に多くの計算回数と通信コストを必要とする可能性が考えられる．

一方で，現代の情報ネットワークは一般的にフルコネクト型ネットワークが普及している．フルコネクト型ネットワークが与えられたもとで学習の効率化を図ることができれば，実問題への適用にも有効だと考えられる．

そこで本研究では，フルコネクト型ネットワークモデルが与えられたもとで少ない計算回数と通信コストで D-SVM を学習する方法を提案する．上述した通り，D-SVM では各ノードで推定するローカル SVM のパラメータが一致するまで繰り返し計算を行う．少ない計算回数と通信コストで最適な SVM のパラメータ推定を行うことを考えたとき，ローカル SVM のパラメータを全体最適な SVM のパラメータに効率的に近づけることができるノード間でパラメータ共有をすることが重要であると考えられる．一方で，パラメータ共有が不必要なノード間でパラメータ共有を行うことは不要な計算とネットワーク間の通信コストを増加させると考えられる．そこでフルコネクト型ネットワークモデルが与えられたもとで，各ノードに蓄積されている学習データの統計的特徴を考慮し必要なノード間にのみローカル SVM のパラメータ共有を行うことで計算回数，通信コストの削減を図る．

具体的には，各ノードが蓄積するデータの統計的特徴を用いてノード間の類似度を算出し，類似度の高いノードはパラメータ共有の必要性が低いノード，類似度の低いノードはパラメータ共有の必要性が高いノードと判断する．パラメータ共有の必要性が高いノード間のみでパラメータ共有を行うように隣接行列 $\boldsymbol{E}$ を生成し，これを用いて D-SVM によるパラメータ推定を行う．

5.2 類似度の算出

各ノードが保持するデータの統計的特徴を定量化し，それを用いてノード間の類似度を算出することを考える．

データの統計的特徴は基本統計量や確率分布などで定量化できるが，提案手法では各ノードで最初に推定されるローカル SVM のパラメータを各ノードに蓄積されたデータの統計的特徴として類似度計算に用いる．これを用いる理由として 2 点挙げられる．一つは，一般にサポートベクトル (SV) は与えられた学習データの中でも自身と異なるクラスのデータが近くに存在するようなデータであり，基本統計量や確率分布を用いてこれらを表現するのは困難であるためである．もう一つは，各ノードで最初に推定されたローカル SVM のパラメータは，他のノードの学習データに影響を受けないため自身が保持する学習データの特徴を表すことができることである．D-SVM の推定アルゴリズムでは自身で推定したローカル SVM のパラメータを大域的な最適パラメータに近づけるように更新を行う．そのため，全体で最適な SVM のパラメータを推定することを考えたとき，基本統計量や確率分布より各ノードで推定されるローカル SVM のパラメータを各ノードの特徴と捉える方がより有効であると考えられる．

いま，ノード j で最初に推定されるローカル SVM のパラメータを $\boldsymbol{v}_{L,j} = (\boldsymbol{w}_{L,j}^T, w_{L,0})^T$，ノード i とノード j の類似度を s_{ij}，類似度行列を $\boldsymbol{S} = [s_{ij}] \in \mathbb{R}^{J\times J}$ とする．このとき，s_{ij} を以下の式 (11)–(12) を用いて算出する．

$$s_{ij} = \frac{\boldsymbol{v}_{L,i}^T \boldsymbol{v}_{L,j}}{\|\boldsymbol{v}_{L,i}\|_2 \cdot \|\boldsymbol{v}_{L,j}\|_2} \times d_{ij}, \tag{11}$$

$$d_{ij} = \begin{cases} \frac{\|\boldsymbol{w}_{L,j}\|_2^2}{\|\boldsymbol{w}_{L,i}\|_2^2} & if\ \|\boldsymbol{w}_{L,j}\|_2^2 \le \|\boldsymbol{w}_{L,i}\|_2^2, \\ \frac{\|\boldsymbol{w}_{L,i}\|_2^2}{\|\boldsymbol{w}_{L,j}\|_2^2} & if\ \|\boldsymbol{w}_{L,j}\|_2^2 > \|\boldsymbol{w}_{L,i}\|_2^2. \end{cases} \tag{12}$$

ここで，ノード i と j の類似度 s_{ij} は，式 (11) 右辺第一項のコサイン距離と第二項のマージン比の積で算出されていることがわかる．コサイン距離は $\boldsymbol{v}_{L,i}$ と $\boldsymbol{v}_{L,j}$ の傾きの差異を表し，マージン比は各ノードで選定された SV が特徴空間上のどの位置に存在しているかを表していると解釈できる．各ノードが選定した SV を用いることで適切な類似度を算出する方法も考えられるが，DDM の制約のもとでは用いることができない．そこで，各ローカル SVM のパラメータが各ノードが選定した SV に依存することに着目し，式 (11) を用いることで，各ノードが生データを直接共有することなくそれぞれ蓄積するデータの統計的特徴の類似度を適切に算出できると考えられる．

提案手法では，式 (11) で算出される類似度の値が高いほど蓄積しているデータの統計的特徴が似ているとし，パラメータ共有の必要性が低いと判断する．

5.3 隣接行列の生成

前節で算出した類似度行列 $\boldsymbol{S}$ を用いて，ローカル SVM の共有が必要なエッジを選択することを考える．D-SVM は，ローカルアップデートとグローバルアップデートの 2 つのステップを繰り返すことで最適なパラメータ推定を行っている．この 2 つのステップの繰り返しが終了するときは，各ノードで推定されたローカル SVM のパラメータが一致するときである．少ない計算回数で大域的に最適な SVM のパラメータを推定するためには，各ノードがローカル SVM のパラメータを最適なパラメータに近づけることができるノードのみとパラメータを共有することが重要である．

各ノードが蓄積するデータの統計的特徴を用いて，パラメータの共有が必要なエッジを選択することを考える．このとき，データの統計的特徴に差異があるノード間のみでパラメータ共有をすることで少ない計算回数で最適な SVM のパラメータを学習することができると考えられる．そこで，提案手法では類似度の小さいノード間のみでパラメータを共有するような隣接行列 $\boldsymbol{E}$ を生成し，それを用いて D-SVM によるパラメータ推定を行う．本手法では，閾値を用いてパラメータ共有するエッジを選択する．以下に隣接行列 $\boldsymbol{E}$ の生成アルゴリズムを示す．

[閾値を用いた隣接行列の生成アルゴリズム]

Step0. 閾値の初期値として $\gamma = \varepsilon$ を設定する
(ε は微小な値とする)．

Step1. 全ての s_{ij} において，$s_{ij} \le \gamma$ なら $e_{ij} = 1$，$s_{ij} > \gamma$ なら $e_{ij} = 0$ として隣接行列 $\boldsymbol{E}$ を生成．
ただし，$i = j$ のとき $e_{ij} = 0$ とする．

Step2. 連結グラフならば終了．非連結グラフならば γ の値を微小に大きくして **Step1** へ戻る．

□

6 評価実験

提案手法の有用性を検証するために 2 次元の人工データとベンチマークデータとして UCI 機械学習レポジトリ [3] で提供されているデータを用いて評価実験を行った．

6.1 実験条件

人工データはクラス 1 とクラス 2 の平均ベクトルをそれぞれ $(-1,-1)^T$, $(1,1)^T$，2 変数の分散を 1, 2, 共分散を 0 とする正規分布に従う乱数から生成した．総学習デー

タ数は，100 ∼ 1,000 の 100 件刻みとした．UCI データは，wine, digits, ionosphere, Musk の 4 種類データを用いて実験を行った．ここで，wine と digits のデータは 3 クラス以上のデータが存在するが，本実験では Forrero ら [2] が行った実験と同様に，wine ではクラス 1 とクラス 2，digits ではクラス 2 とクラス 9 のデータを用いた．各データセットの基本情報を表 1 に示す．

表 1: UCI データセット概要

	次元数	総学習データ数
wine	13	130
digits	50	1,750
ionosphere	34	351
Musk	168	476

実験に用いるネットワークのノード数は 10 とし，各ノードに割り当てる学習データ数はノードに関わらず一定とした．比較手法としてフルコネクト型ネットワークモデルを用いた D-SVM(以下，比較手法 1)，ランダムに生成した隣接行列を用いた D-SVM(以下，比較手法 2) を用いた．また，提案手法はコサイン距離のみを用いた手法 (以下，提案手法 1) とコサイン距離とマージン比を用いた手法 (以下，提案手法 2) とする．ただし，提案手法と比較するため，比較手法 2 で用いるネットワーク構造は提案手法で用いた隣接行列と同程度の平均隣接ノード数となるような隣接行列をランダムに生成した．評価指標は，計算回数と式 (13) で定義される通信コストの平均値を用いた．また，D-SVM は用いる隣接行列が連結グラフを満たせば大域的最適な SVM のパラメータが推定可能であり，分類精度は提案手法と比較手法で不変である．

$$\text{通信コスト} \quad = \quad \text{計算回数} \times \frac{\sum_{j=1}^{J} |\mathcal{B}_j|}{J}. \qquad (13)$$

上記の条件で 10 回実験を行い，その平均値を結果として出力する．

6.2 実験結果と考察

以下の図 1, 図 2 に人工データによる実験結果を示す．

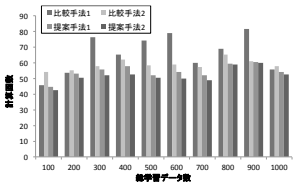

図 1: 計算回数 (人工)

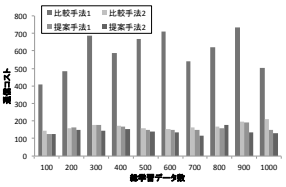

図 2: 通信コスト (人工)

図 1, 図 2 より，全ての学習データ数において計算回数と通信コストの両面で提案手法 2 が優れていることがわかる．この結果から，全ノードでパラメータ共有を行う方法やランダムなネットワークを形成する方法よりも，各ノードに蓄積されたデータの統計的特徴を考慮して隣接行列を構成することで，少ない計算回数と通信コストで学習が行えたと考えられる．また，一部の学習データ数を除いて学習データ数が増加するほど学習に必要な計算回数と通信コストが増加している．一部の学習データ数のときに計算回数と通信コストが比例していないことから，学習データ数のみが学習に必要な計算回数と通信コストに影響を与えていないことが示唆される．D-SVM において，推定に必要な計算回数は各ノードで推定されるローカル SVM のパラメータが直接影響している．そのため，総学習データ数よりも各ノードに振り分けた学習データの統計的特徴の方が，学習に必要な計算回数と通信コストに強い影響を与えていると考えられる．

図 3, 図 4 に UCI データを用いた実験結果を示す．

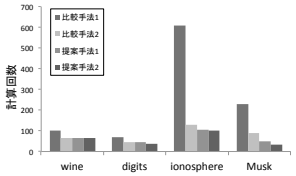

図 3: 計算回数 (UCI)

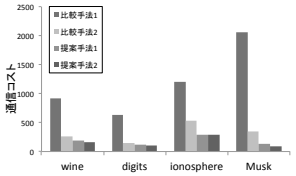

図 4: 通信コスト (UCI)

図 3, 図 4 から人工データでの実験と同様に，計算回数，通信コストともに提案手法 2 が最良となった．人工データの結果 (図 1, 2) と比較すると大幅な改善が見られる．これは人工データと比較し，これらのデータが高次元であることと，各ノードが蓄積した学習データ数に関係していると考えられる．一般に，扱うデータが高次元であるほど，学習に必要なデータ数は増加する．しかしながら，本実験では各ノードに割り振られた学習データ数は扱うデータの次元数に対して少ない場合が多かった．そのため，各ノードが蓄積するデータの統計的特徴に差異が生じたと考えられる．その結果，人工データの場合と比較すると各ノードで最初に推定するローカル SVM のパラメータ $\boldsymbol{v}_{L,j}$ に顕著な差異が見られ，各ノードが蓄積するデータの統計的特徴が似ているノードと似ていないノードを適切に判別することができたと考えられる．これにより，比較手法に比べて提案手法では効率的に大域的最適な SVM のパラメータを推定できたと考えられる．これらの結果から，提案手法は各ノードに蓄積されたデータに偏りが生じる場合，従来手法に比べて少ない計算回数と通信コストでパラメータ推定が行えると考えられる．

7 まとめと今後の課題

本研究では，フルコネクト型ネットワークに対し，各ノードが蓄積するデータの統計的特徴を考慮した隣接行列を構成して D-SVM を適用することにより大域的最適な SVM のパラメータの学習に必要な計算回数と通信コストを削減する手法を提案した．評価実験により，計算回数と通信コストにおいて提案手法の有効性が示された．特に，扱うデータが高次元であるほど，提案手法は従来手法に比べて少ない計算コストでパラメータ推定が可能だと考えらえる．

今後の課題として，隣接行列生成アルゴリズムの定式化，他の分類手法への拡張などが挙げられる．

参考文献

[1] L. Zeng, L. Li, L. Duan, K. Lu, Z. Shi, M. Wang, W. Wu and P. Luo, “Distributed Data Mining: A Survey”, *Information Technology and Management*, Volume 13, Issue 4, pp.403–409, December 2012.

[2] Pedro A. Forrero, Alfonso Cano, and Georgios B. Giannakis, “Consensus-Based Distributed Support Vector Machines,” *The Journal of Machine Learning Research*, pp.1663–1707, 2010.

[3] Bache K. and Lichman M, “UCI Machine Learning Repository [http://archive.ics.uci.edu/ml],” University of California, Irvine, 2013.