F-007

Gilbert Algorithm を用いた分散サポートベクトルマシン
Distributed Support Vector Machine with Gilbert Algorithm

樺島八入 * Yashio Kabashima
菊地賢也 * Kenya Kikuchi
全眞嬉 * Jinhee Chun
徳山豪 * Takeshi Tokuyama

1. はじめに

サポートベクトルマシン (Support Vector Machine: SVM)[1] はしばしば用いられる機械学習の一種であり，最適化問題の二次計画法として定式化される．しかしながら，二次計画法では分散環境で効率良く実装することが困難であることが知られており，無線センサネットワークデータ等に置かれたデータに対して適用することが難しい．また，計算時間についてもデータ数に対して 3 乗の時間が必要となる．そこで，Gilbert Algorithm[3] と呼ばれる凸多面体間の距離を近似的に求めるアルゴリズムを SVM に適用することにより，分散処理が可能かつ，より高速に実行可能な SVM が提案された [4]．本論文では，既存手法を基に実用的な変更を与え，高速に 2 クラス分類を行う分散 SVM を提案し，疑似データを用いて実装および評価を行った．

2. 予備知識

2.1 Gilbert Algorithm を用いた SVM

Gilbert Algorithm を用いた SVM では，線形空間において 2 クラスの線形分離可能なデータについて各クラスの凸多面体を考える．このとき，2 クラス間の凸多面体の最短距離 (多面体距離) を構成するベクトルを重みベクトルに持ち，各クラスのマージンを最大化する決定境界面を定める．

2.2 既存研究

Gilbert Algorithm を用いて，2 クラス間の多面体距離を近似的に求める手法として，B. Gätner らにより，Improved 2 class Gilbert Algorithm と呼ばれるアルゴリズムが提案されており，このアルゴリズムを用いることで，SVM をデータ数に対して線形時間で実行できることが保証されている [2]．また，Gilbert Algorithm の分散化については，Y. Liu らが，Distributed Gilbert Algorithm[4] を提案しており，これらを組み合わせることで，線形時間で計算可能であり，かつ分散化可能な SVM の実現が可能となる．

3. 提案手法

3.0.1 交互 Gilbert Algorithm

既存研究では，アルゴリズムの理論的な保証のみが行われており，実験は行われていない．そこで，既存研究の Imporved 2 class Gilbert Algorithm に関して計算機実験を行い，分散システムの適用を試みたところ通信量が多くなるため，実装が困難であることがわかった．そこで，既存手法を改良し，分散システムへの適用が容易で，かつ計算時間が高速なヒューリスティックアルゴリズムとして，交互 Gilbert Algorithm を提案し，Algorithm 1 に示す．2 クラスにおける Gilbert Algorithm では，暫定解を更新することにより，近似的に多面体距離を構成するベクトルを満たす，各クラスにおける凸多面

Algorithm 1 相互 Gilbert Algorithm

Require: 2 クラスの点集合 $P^{(1)}, P^{(2)} \subseteq \mathbb{R}^d$

Ensure: 多面体距離を構成する点 $\mathbf{x}^{(1)} \in conv(P^{(1)}), \mathbf{x}^{(2)} \in conv(P^{(2)})$

1: 暫定解 $\forall \mathbf{x}_1^{(1)} \in conv(P^{(1)}), \forall \mathbf{x}_1^{(2)} \in conv(P^{(2)})$ を定める

2: $\forall \mathbf{p} \in P^{(1)}$ について，ベクトル $\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}$ に射影した点と $\mathbf{x}_i^{(2)}$ の距離が最短となるときの点 $\mathbf{p}$ を求め，$\mathbf{p}_i^{(1)} = \mathbf{p}$ とする．線分 $(\mathbf{p}_i^{(1)}, \mathbf{x}_i^{(1)})$ 上で $\mathbf{x}_i^{(2)}$ から最短となる点を更新解 $\mathbf{x}_{i+1}^{(1)}$ として定める．

3: 反対側のクラス ($P^{(2)}$) について，同様の処理を行う．

4: ϵ 近似条件 (式 (1)) を満たすまで処理 3, 4 を繰り返す．

* 東北大学大学院情報科学研究科 Tohoku University, Graduate School of Information Science

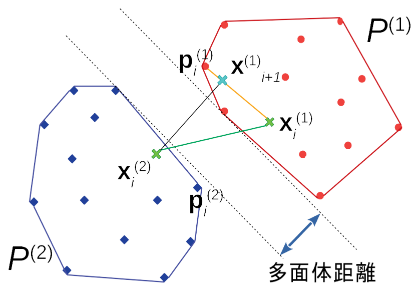

図 1　相互 Gilbert Algorithm による解の更新

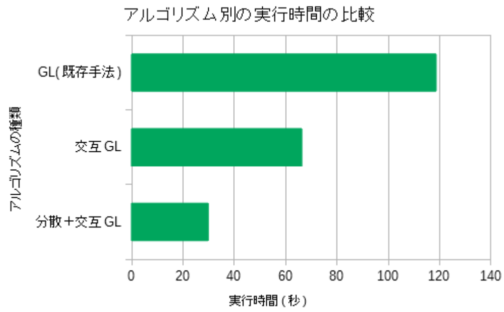

図 2　3 つのアルゴリズムの計算時間

体中の点 $\mathbf{x}^{(1)} \in conv(P^{(1)}), \mathbf{x}^{(1)} \in conv(P^{(1)})$ を求める．この点を求めるために，凸多面体中の任意の点を暫定解として，Algorithm　1 のように暫定解を更新する．最後に，式 1 で定義される近似条件を満たしたときにアルゴリズムを終了する．より理解を深めるために，Algorithm　1 におけるクラス (1) を更新する過程を図 1 に示した．

$$
\begin{aligned}
&(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}) - \frac{(\mathbf{p}_i^{(1)} - \mathbf{p}_i^{(2)})^\top(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)})}{\|\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\|} \\
&\leq \epsilon(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)})
\end{aligned}
\tag{1}
$$

3.0.2　交互 Gilbert Algorithm の分散化

本論文における分散システムは，一つのコーディネータと k 個のノードを考え，コーディネータとノード間の通信のみが許される．各クラスの点集合は k 個のノードに分散され，Algorithm1 中の $\mathbf{p}_i^{(1)}, \mathbf{p}_i^{(2)}$ を選ぶ際に，ノード毎にまず $\mathbf{p}_i^{(1)}, \mathbf{p}_i^{(2)}$ の候補を選び，コーディネータがそれぞれのノードで選ばれた候補の中から最適な $\mathbf{p}_i^{(1)}, \mathbf{p}_i^{(2)}$ を決定する．以上の方法により，相互 Gilbert Algorithm を分散環境に適用することができる．

4.　実装・評価

本研究では，線形空間において線形分離可能な 2 クラスで各クラスのサンプル数が 10000 個の 10 次元擬似データセットを異なる境界面で 5 種類作成した．また，近似条件に用いる ϵ 値は $\epsilon = 0.01$ と定めた．計算機実験では，以上のような擬似データセットに対して，既存手法である Improved 2 class Gilbert Algorithm，提案手法である相互 Gilbert Algorithm，そして相互 Gilbert Algorithm に分散システムを適用したアルゴリズムの計 3 つのアルゴリズムについて実験を行った．このとき，分散システムにおけるノードの数は 5 つに設定した．5 種類のデータセットそれぞれにおける各アルゴリズムの収束時間を計測し，その平均時間 (秒) を測定したところ，図 2 のような結果が得られた．図 2 からわかるとおり，提案手法は既存手法より高速であり，分散化を施すことで，並列計算が可能となり，さらに高速になることが確認できた．

5.　おわりに

本研究では，線形分離可能なデータセットにおいて，データ数に対して線形時間でかつ分散システムが導入可能なアルゴリズムを提案した．しかしながら，線形分離可能でないデータセットやカーネル法は今回提案したアルゴリズムでは適用できない．従って，これらの問題に対応することが今後の課題としてあげられる．

謝辞

本研究は，総合科学技術・イノベーション会議が主導する革新的研究開発推進プログラム (ImPACT) の一環として実施したものです．

参考文献

[1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 1995.

[2] Bernd Gärtner and Martin Jaggi. Coresets for polytope distance. SOCG, 2009.

[3] Elmer G Gilbert. An iterative procedure for computing the minimum of a quadratic form on a convex set. *SIAM Journal on Control*, 1966.

[4] Yangwei Liu, Hu Ding, Ziyun Huang, and Jinhui Xu. Distributed and robust support vector machine. ISAAC, 2016.