制約付き非平滑凸最小化問題を解くための増分および並列型劣勾配法への直線探索法の組み込み

Incorporating Line Search Method into Incremental and Parallel Subgradient Algorithms for Solving Constrained Nonsmooth Convex Optimization Problems

明治大学大学院 理工学研究科 情報科学専攻　菱沼 和弘 *
明治大学 理工学部 情報科学科　飯塚 秀明

Computer Science Program, Graduate School of Science and Technology, Meiji University
Department of Computer Science, School of Science and Technology, Meiji University

Kazuhiro HISHINUMA
Hideaki IIDUKA

概要

射影計算可能閉凸制約上での非平滑凸関数和最小化問題は、サポートベクトルマシンの学習において現れる重要な最適化問題である。射影計算可能閉凸制約上での非平滑凸関数和最小化問題を解くための既存の最適化アルゴリズムは、そのステップ幅を実行前に設定するため、低速な収束を招くことがある。そこで本稿は、実行時に適切なステップ幅を、自動的、アルゴリズム的、かつ臨機応変に選択する、直線探索法を組み込んだ、新しい最適化アルゴリズムを提案する。本稿では、射影計算可能閉凸制約上での非平滑凸関数和最小化問題について考察し、逐次的かつ周期的に各関数の情報を用いて最小化を行う増分劣勾配法に基づくアルゴリズムと、並列かつ独立に各関数の情報を用いて最小化を行う並列型劣勾配法に基づくアルゴリズムの、2 つのアルゴリズムを提案する。これらの提案アルゴリズムは、綿密なステップ幅の調整なしに効率的な収束を実現できる、非常に有用な手法である。本稿で提案する最適化アルゴリズムは、既存手法に用いられるステップ幅の条件を緩和する。すなわち提案アルゴリズムは、非平滑凸最小化に関する既存の増分劣勾配法と並列型劣勾配法の拡張にもなっている。本稿では、上記 2 つのアルゴリズムを提案し、これらがある条件下で射影計算可能閉凸制約上での非平滑凸最小化問題の解に収束することを示す定理を与える。

1　はじめに

本稿では、サポートベクトルマシンの学習において現れる、射影計算可能閉凸制約上での非平滑凸関数和最小化について考察する。サポートベクトルマシンは、効果的な機械学習の手段として広く用いられている [12, 13, 16, 18]。サポートベクトルマシンの学習は、実証に基づいた損失関数と分類器のノルムの最小化として扱うことができる [18, Section 1]。したがっていま、この損失関数が凸関数であるならば、サポートベクトルマシンの学習は非平滑凸関数和最小化問題として扱うことができる。

この最適化問題を解く実用的な手法としては、増分劣勾配法 [14]、および並列型劣勾配法 [8] が挙げられる。増分劣勾配法は、和を構成する各関数の情報を順に用いて最小化し、並列型劣勾配法は各関数の情報を独立にすべて用いて最小化を行う。一方並列型劣勾配法は、各関数を独立に扱うため、このそれぞれの計算を並列化することができるアルゴリズムである。ここで両アルゴリズムは、解の更新の際に**ステップ幅** を用いる。しかしこのステップ幅は、適切に調整しなければ各アルゴリズムを効率的に駆動させることができない。適切なステップ幅は、和を構成する関数の個数や次元、目的関数や制約集合の形状や劣勾配の選び方など、様々な要

* 独立行政法人日本学術振興会 特別研究員 DC.

因に依存する。したがってアルゴリズムの実行前に、この適切なステップ幅を選択することは困難であるといえる。

他方、制約なし最適化アルゴリズムにおいては、適切なステップ幅を選択するために直線探索が行われる [6, 21]。直線探索において、**Wolfe 条件** [20] は、選択されるステップ幅の基準としてよく用いられる。Wolfe 条件は、ステップ幅が Armojio 条件を満たし、かつ曲率条件を満たすことを要求する [15, Chapter 3]。Armijo 条件は、そのステップ幅による関数値の下げ幅が、ある基準となる線形関数よりも上回ることを要求する。この条件により、アルゴリズムの反復が解をより良いものへと更新することを保証する。しかしながらこの条件は、非常に小さなステップ幅がこれを常に満たすため、アルゴリズムが妥当な計算を継続するためには不十分である。したがって、ある程度大きなステップ幅によって解を更新し続けるための曲率条件も共に与える。

直線探索の着想に基づき、本稿では精密なステップ幅の調整なしに効率的に動作する、新たな増分劣勾配法と並列型劣勾配法を提案する。文献 [5] では、直線探索を伴う勾配射影法により、目的関数値の最小化を図っている。しかしながら、この手法は目的関数の微分可能性を仮定している。加えてこのアルゴリズムは集中型であり、計算の並列化のための設計がされていない。文献 [11] では、直線探索を用いて、本稿で扱う最小化問題を含む、不動点問題を解く手法を提案している。この手法は高速な収束性を有するが、その直線探索は計算に現れる凸結合の係数を決めているに過ぎず、また計算の並列化のための設計はされていない。文献 [7, 8, 9, 10, 14] による結果は、効率的な収束のために適切なステップ幅を調整する必要がある。しかしながら、先にも述べた通り、この調整は非常に難しい。

既存の研究に対して本稿では、増分劣勾配法 [14] および並列型劣勾配法 [8] に直線探索を組み込むことで、既存手法よりもより良いステップ幅を用いる手法を提案する。本稿ではまず、アルゴリズムに与える**ステップ幅**を、その候補の集合たる**ステップ幅候補** として拡張する。各提案アルゴリズムにはこのステップ幅候補が与えられ、実行時に候補中から最適なステップ幅を適宜選び出す。ステップ幅候補と直線探索を用いることには、3 つの利点がある。まず第一に、実行時に適切なステップ幅が選択されることにより、より高速な収束と、より良い近似を実現することができる。続いて第二に、ステップ幅に関する事前の精密な調整が不要となる。既存の増分劣勾配法 [14] と並列型劣勾配法 [8] は、適切なステップ幅を事前に与えなければ、効率的な収束を実現できなかった。しかしながら本稿で提案するアルゴリズムは、大雑把なステップ幅候補を与えれば、後は実行時に適切なステップ幅が自動的に選択されるため、この精密な調整が不要となる。したがって提案アルゴリズムは、既存手法と比較して容易に、射影計算可能閉凸制約上での非平滑凸関数和最小化問題に対して適用できる。最後に、適切なステップ幅を事前に決定できないような複雑な問題に対しても、提案アルゴリズムは適用できる。3 章では、解への収束に必要なステップ幅候補の条件を与える。したがってこの条件を満たすステップ幅候補を与えれば、例え最適なステップ幅を具体的に特定できないとしても、効率的な収束を実現することができる。本稿では、ステップ幅候補が漸減するならば、提案アルゴリズムが解に収束することを示す。ここでもし、与えるステップ幅候補が単集合ならば、提案アルゴリズムは既存の増分劣勾配法 [14] や並列型劣勾配法 [8] と一致する。したがって、本稿の解析は既存研究 [8, 14] の拡張にもなっている。

以降の構成は次の通りである。2 章においては、数学的準備を行い、また本稿で扱う主問題を定式化する。3 章においては、提案アルゴリズムとその収束定理を示す。4 章においては、本稿での議論を総括する。

2 数学的準備

$\mathbb{R}^N$ を N 次元 Euclid 空間とし、$\langle \cdot, \cdot \rangle : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ をその標準内積、$\|x\| := \langle x, x \rangle^{\frac{1}{2}}$ を内積から導出されるノルムとする。また自然数全体の集合 $\mathbb{N} := \{1, 2, \ldots\}$ を定義する。点列 $\{x_n\} \subset \mathbb{R}^N$ が点 $x \in \mathbb{R}^N$ に収束することを、$x_n \to x$ により表す。

凸関数 $f : \mathbb{R}^N \to \mathbb{R}$ の点 $x \in \mathbb{R}^N$ における**劣勾配** g を、任意の $y \in \mathbb{R}^N$ に対し $f(x) + \langle y - x, g \rangle \leq f(y)$ が常に成り立つような点 $g \in \mathbb{R}^N$ として定義する。凸関数 f の点 x における劣勾配すべての集合を $\partial f(x)$ として表す [17],[19, Section 7.3]。

空でない閉凸集合 $C \subset \mathbb{R}^N$ に対する**距離射影**を $P_C : \mathbb{R}^N \to C$ によって表し、$\|x - P_C(x)\| = \inf_{y \in C} \|x - y\|$ で定義する [1, Section 4.2]。このとき、P_C は非拡大性を有する [19, Subchapter 5.2]。すなわち、任意の $x, y \in \mathbb{R}^N$ に対し、$\|P_C(x) - P_C(y)\| \leq \|x - y\|$ が常に成り立つ。

2.1 主問題

以降においては、制約付き非平滑凸関数和最小化問題について考察する。いま、$f_i : \mathbb{R}^N \to [0, \infty) \quad (i = 1, 2, \ldots, K)$ を連続な凸関数とし、C を $\mathbb{R}^N$ の空でない閉凸集合とする。このとき、**制約付き非平滑凸関数和最小化問題** を、次式により定める [8, 14]。

$$\begin{aligned} &\text{Minimize } f(x) := \sum_{i=1}^{K} f_i(x), \\ &\text{Subject to } x \in C. \end{aligned} \tag{1}$$

本稿においては、N 次元 Euclid 空間 $\mathbb{R}^N$ の空でない閉凸集合 C が単純であると仮定する。ここで、集合 C が単純であるとは、P_C が有限回の代数的計算によって求めることができることをいう。単純な空でない閉凸集合 C の例としては、閉球、半空間、および 2 つの半空間の共通部分が挙げられる [2, Examples 3.16 and 3.21, and Proposition 28.19]。

f が非平滑で、C が単純なとき、問題 1 は、サポートベクトルマシンの学習のみならず、多くの応用がある。例として、凸集合上での信号の TV 値最小化や L1 ノルムによる Tykhonov-like 問題などの実世界における問題が、これに含まれる [4, I. Introduction]。

本稿においては、以下を仮定する。

仮定 1 (劣勾配有界性 [14, Assumption 2.1]) 各 $i = 1, 2, \ldots, K$ について、それぞれ、

$$\|g\| \leq M_i \quad (x \in C; g \in \partial f_i(x))$$

を満たす正定数 M_i が存在する。加えて、$M := \sum_{i=1}^{K} M_i$ とおく。

仮定 2 (最適解の存在性 [14, Proposition 2.4]) $\operatorname{argmin}_{x \in C} f(x) \neq \emptyset$ が成り立つ。

3 提案手法とその収束定理

3.1 増分劣勾配法

この節では、問題 (1) を解くための増分劣勾配法 (Algorithm 1) を提案する。まず、Algorithm 1 と既存の増分劣勾配法 [14] を比較しよう。両者の違いは、Algorithm 1 に加えられた Step 6 である。既存手法におけるステップ幅 λ_n は、その実行前に与えられる必要がある。しかしながら Algorithm 1 の駆動には、ステップ幅候補 $[\underline{\lambda}_n, \overline{\lambda}_n]$ のみ与えられれば良い。Algorithm 1 は、直線探索を用いて適切なステップ幅をステップ幅候補より自動的に選択し、これを解の更新に用いる。ここで、ステップ幅の選択に用いる直線探索の具体的な方法については、3.3 節で詳しく述べる。なお、Algorithm 1 はステップ幅候補を $\underline{\lambda}_n := \overline{\lambda}_n := \lambda_n$ と設定すれば

Algorithm 1 増分劣勾配法

Require: $\forall n \in \mathbb{N}, [\underline{\lambda}_n, \overline{\lambda}_n] \subset (0, \infty)$.

$n \leftarrow 1$, $x_1 \in C$.
loop
  $y_{n,0} := x_n$.
  for $i = 1, 2, \ldots, K$ **do** ▷ 逐次的に実行
    $g_{n,i} \in \partial f_i(y_{n,i-1})$.
    $\lambda_{n,i} \in [\underline{\lambda}_n, \overline{\lambda}_n]$. ▷ 直線探索により選択
    $y_{n,i} := P_C(y_{n,i-1} - \lambda_{n,i} g_{n,i})$.
  end for
  $x_{n+1} := y_{n,K}$.
  $n \leftarrow n + 1$.
end loop

既存の増分劣勾配法と一致する。なぜならば、このとき Algorithm 1 はステップ幅として、λ_n のみをステップ幅候補 $[\underline{\lambda}_n, \overline{\lambda}_n] = \{\lambda_n\}$ より選択できるからである。したがって Algorithm 1 は、既存の増分劣勾配法に対する拡張となっている。

以下は、射影計算可能な閉凸集合 C 上の非平滑凸関数 $F_i := f_i + g_i \, (i = 1, 2, \ldots, K)$ の和に関する最小化問題に対する、増分近接劣勾配法 [3, (2.1)–(2.2)] である。

$$\begin{cases} z_n := \operatorname{argmin}_{x \in C} \left\{ f_{i_n}(x_n) + \frac{1}{2\lambda_n} \|x - x_n\| \right\}, \\ u_{i_n} \in \partial g_{i_n}(z_n), \\ x_{n+1} := P_C(z_n - \lambda_n u_{i_n}), \end{cases} \tag{2}$$

ただし、$x_0 \in C$ が与えられ、また $\{i_n\} \subset \{1, 2, \ldots, K\}$ はランダムに選択され、$\{\lambda_n\} \subset [0, \infty)$ であるとする。Algorithm (2) は近接点、すなわち $z_{i_n} \in \operatorname{argmin}_{u \in C}\{f_{i_n}(x) + \frac{1}{2\lambda_n}\|u - x_n\|^2\}$ が容易に計算可能であることを仮定している。一方 Algorithm 1 は、その実行に近接点の計算を用いないため、その仮定が必要ない。Algorithm (2) は、一様にランダムな順序で計算を行うことができる。つまり、Algorithm (2) は各反復において、計算に用いる関数対 (f_i, g_i) をランダムに選択できる。他方 Algorithm 1 は、予め定められた順に繰り返し関数対 (f_i, g_i) を選択し、計算を行う。

3.2 並列型劣勾配法

並列型劣勾配法 [8] の拡張を、Algorithm 2 に示す。Algorithm 2 と並列型劣勾配法 [8] の相違は、Algorithm 2 には Step 5 が追加されたことである。この追加により、並列型劣勾配法 [8] では与えられたステップ幅 λ_n を解の更新に用いるが、Algorithm 2 では実行時にステップ幅候補 $[\underline{\lambda}_n, \overline{\lambda}_n]$ より適するステップ幅 λ_n を適宜選び出すことができる。

定数ステップ幅 $\lambda_n := \lambda \in (0, \infty)$ $(n \in \mathbb{N})$ を用いたとき、$F^\star$ を目的関数 $F := \sum_{i=1}^K F_i$ の最小値、M をある定数とすると、Algorithm (2) は $\underline{\lim}_{n \to \infty} F(x_n) \leq F^\star + M\lambda$ の意味で収束する [3, Proposition 3.2]。また、漸減ステップ幅 $\{\lambda_n\}$ を用いたとき、Algorithm (2) は最適解へ収束する [3, Proposition 3.4]。しかしながら、漸減ステップ幅を用いたときの Algorithm (2) の収束率は解析されていない。本稿では、Algorithm 1, 2 が最適解にある収束率をもって収束することを示す。

Algorithm 2 並列型劣勾配法

Require: $\forall n \in \mathbb{N}, [\underline{\lambda}_n, \overline{\lambda}_n] \subset (0, \infty)$.

1: $n \leftarrow 1$, $x_1 \in C$.
2: **loop**
3: **for all** $i \in \{1, 2, \ldots, K\}$ **do** ▷ 各反復は独立に計算可能
4: $g_{n,i} \in \partial f_i(x_n)$.
5: $\lambda_{n,i} \in [\underline{\lambda}_n, \overline{\lambda}_n]$. ▷ 直線探索により選択
6: $y_{n,i} := P_C(x_n - \lambda_{n,i} g_{n,i})$.
7: **end for**
8: $x_{n+1} := \frac{1}{K} \sum_{i=1}^{K} y_{n,i}$.
9: $n \leftarrow n + 1$.
10: **end loop**

Algorithm 2 と並列型劣勾配法 [8] は共に、関数に関する各反復 (Step 3) を独立して計算することができる。すなわち、この反復はその計算順序に依存しない。したがって、各関数についての計算は並列化することが可能である。計算の並列化により、この反復が高速化されることが見込まれる。一般に、Algorithm 2 の各反復は、既存の増分劣勾配法 [14] や既存の並列型劣勾配法 [8] などの劣勾配法と比較して、より多くの計算時間を要する。なぜならば、Algorithm 2 は既存の並列型劣勾配法 [8] に直線探索の処理を加えているからである。しかしながら、計算の並列化を行うことによって、この処理の追加に関する影響を軽減させることができる。

3.3 直線探索法

Algorithm 1 におけるステップ 6 や、Algorithm 2 におけるステップ 5 では、直線探索を行う。直線探索は、Algorithm 1 における $y_{n,i-1}$ や、Algorithm 2 における x_n、そのほか $g_{n,i}$、f_i など、現在の反復において利用可能な各種情報を用いて、更新幅候補 $[\underline{\lambda}_n, \overline{\lambda}_n]$ より最も適切な更新幅 λ_n を 1 つ選び出す。この機構の導入は、本稿において提案するアルゴリズムの主たる特徴である。

直線探索アルゴリズムの簡単な例としては、以下に掲げる Algorithm 3 が挙げられる。まず、比率

Algorithm 3 直線探索アルゴリズムの簡単な例

1: $x_p := \begin{cases} y_{n,i-1} & \text{(Algorithm 1)}, \\ x_n & \text{(Algorithm 2)} \end{cases}$.
2: $\lambda_{n,i} \leftarrow L_1 \overline{\lambda}_n + (1 - L_1) \underline{\lambda}_n$.
3: **for** $L_t \in \{L_2, L_3, \ldots, L_k\}$ **do**
4: $t \leftarrow L_t \overline{\lambda}_n + (1 - L_t) \underline{\lambda}_n$.
5: **if** $f_i(P_C(x_p - t g_{n,i})) < f_i(P_C(x_p - \lambda_{n,i} g_{n,i}))$ **then**
6: $\lambda_{n,i} \leftarrow t$
7: **end if**
8: **end for**

$\{L_1, L_2, \ldots, L_k\} \subset [0, 1]$ をいくつか設定する。続いてこの比率に基づき、最適更新幅に関する探索候補

$\lambda_{n,i} = L_t\overline{\lambda}_n + (1 - L_t)\underline{\lambda}_n \quad (t = 1, 2, \ldots, k)$ を生成し、それらの中で最も関数値を減少させるものを更新幅として採用する。

この Algorithm 3 に限らず、更新幅候補 $[\underline{\lambda}_n, \overline{\lambda}_n]$ よりある指標に基づき最も適切な更新幅 λ_n を 1 つ選び出すアルゴリズムであれば、ここでの直線探索アルゴリズムとして用いることができる。既存手法 [7, 8, 9, 10, 14] は、効率的な収束のためには適切なステップ幅の設定が必要不可欠である。しかしながらこの適切な値は、目的関数の個数や次元、形状および制約集合や劣勾配の選択方法などによって変化する。一方、本稿における提案アルゴリズムは実行時にこれを決定することができる。したがって、既存手法と比較してより柔軟に問題設定へ対応し、効率的な収束を実現することができる。

3.4 収束定理

Algorithm 1 および Algorithm 2 の収束性について議論する。収束定理を与える前に、以下に掲げる命題の成立を仮定する。

仮定 3 (Step-Range Compositions)

$$\sum_{n=1}^{\infty} \overline{\lambda}_n = \infty, \quad \sum_{n=1}^{\infty} \overline{\lambda}_n^2 < \infty, \quad \lim_{n\to\infty} \frac{\underline{\lambda}_n}{\overline{\lambda}_n} = 1, \quad \sum_{n=1}^{\infty} (\overline{\lambda}_n - \underline{\lambda}_n) < \infty.$$

このとき、Algorithm 1 および Algorithm 2 の収束性について、以下の定理が成り立つ。

定理 1 (Main Theorem) 仮定 3 が成り立つとし、$\{x_n\}$ を Algorithm 1 または Algorithm 2 により生成された点列とする。このとき、点列 $\{x_n\}$ は問題 (1) の最適解へ収束する。

4 まとめ

本稿では、実行時に適切なステップ幅を、自動的、アルゴリズム的、かつ臨機応変に選択する直線探索法を組み込んだ、新しい増分劣勾配法と並列型劣勾配法を提案した。提案アルゴリズムについて、それらが制約付き非平滑凸最小化問題の解に収束することを示した。提案アルゴリズムは、既存の増分劣勾配法または並列型劣勾配法の拡張となっている。加えて提案アルゴリズムは直線探索により、大雑把なステップ幅候補さえ与えれば、後は実行時に適切なステップ幅が自動的に選択されるため、その事前の精密な調整が不要となる。したがって提案アルゴリズムは、既存手法と比較してより柔軟にかつ効率的に用いることのできるアルゴリズムであるといえる。

謝辞

本研究は、JSPS 科研費 JP17J09220, JP15K04763 の助成を受けたものです。

参考文献

[1] Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, New York (2011)

[2] Bauschke, H.H., Combettes, P.L.: Convex analysis and monotone operator theory in Hilbert spaces. Springer Science+Business Media (2011)

[3] Bertsekas, D.P.: Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. Lab. for Information and Decision Systems Report **LIDS-P-2848**, 85–119 (2010)

[4] Combettes, P.L., Pesquet, J.C.: A douglas–rachford splitting approach to nonsmooth convex variational signal recovery. IEEE Journal of Selected Topics in Signal Processing **1**(4), 564–574 (2007). DOI 10.1109/JSTSP.2007.910264

[5] Cruz, J.Y.B., Oliveira, W.D.: On weak and strong convergence of the projected gradient method for convex optimization in real Hilbert spaces. Numerical Functional Analysis and Optimization **37**(2), 129–144 (2016). DOI 10.1080/01630563.2015.1080271. URL http://www.tandfonline.com/doi/abs/10.1080/01630563.2015.1080271

[6] Hare, W.L., Lucet, Y.: Derivative-free optimization via proximal point methods. Journal of Optimization Theory and Applications **160**(1), 204–220 (2014). DOI 10.1007/s10957-013-0354-0. URL http://dx.doi.org/10.1007/s10957-013-0354-0

[7] Hayashi, Y., Iiduka, H.: Optimality and convergence for convex ensemble learning with sparsity and diversity based on fixed point optimization. Neurocomputing **273**(Supplement C), 367 – 372 (2018). DOI https://doi.org/10.1016/j.neucom.2017.07.046. URL http://www.sciencedirect.com/science/article/pii/S0925231217313486

[8] Hishinuma, K., Iiduka, H.: Parallel subgradient method for nonsmooth convex optimization with a simple constraint. Linear and Nonlinear Analysis **1**(1), 67–77 (2015)

[9] Iiduka, H.: Parallel computing subgradient method for nonsmooth convex optimization over the intersection of fixed point sets of nonexpansive mappings. Fixed Point Theory and Applications **2015:72** (2015). URL https://doi.org/10.1186/s13663-015-0319-0

[10] Iiduka, H.: Convergence analysis of iterative methods for nonsmooth convex optimization over fixed point sets of quasi-nonexpansive mappings. Mathematical Programming **159**(1), 509–538 (2016)

[11] Iiduka, H.: Line search fixed point algorithms based on nonlinear conjugate gradient directions: application to constrained smooth convex optimization. Fixed Point Theory and Applications **2016:77** (2016). URL https://doi.org/10.1186/s13663-016-0567-7

[12] Leopold, E., Kindermann, J.: Text categorization with support vector machines. how to represent texts in input space? Machine Learning **46**(1), 423–444 (2002). DOI 10.1023/A:1012491419635. URL https://doi.org/10.1023/A:1012491419635

[13] Lin, Y., Lee, Y., Wahba, G.: Support vector machines for classification in nonstandard situations. Machine Learning **46**(1), 191–202 (2002). DOI 10.1023/A:1012406528296. URL https://doi.org/10.1023/A:1012406528296

[14] Nedić, A., Bertsekas, D.P.: Incremental subgradient methods for nondifferentiable optimization. SIAM Journal on Optimization **12**(1), 109–138 (2001). DOI 10.1137/S1052623499362111. URL http://dx.doi.org/10.1137/S1052623499362111

[15] Nocedal, J., Wright, S.: Numerical Optimization, 2 edn. Springer-Verlag, New York (2006)

[16] Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J.H., Jurafsky, D.: Support vector learning for semantic argument classification. Machine Learning **60**(1), 11–39 (2005). DOI 10.1007/s10994-005-0912-2. URL https://doi.org/10.1007/s10994-005-0912-2

[17] Rockafellar, R.T.: Monotone operators associated with saddle-functions and minimax problems. Nonlinear functional analysis **18**(I), 397–407 (1970)

[18] Shalev-Shwartz, S., Singer, Y., Srebro, N., Cotter, A.: Pegasos: primal estimated sub-gradient solver for SVM. Mathematical programming **127**(1), 3–30 (2011)

[19] Takahashi, W.: Introduction to Nonlinear and Convex Analysis. Yokohama Publishers, Inc., Yokohama (2009)

[20] Wolfe, P.: Convergence conditions for ascent methods. SIAM review **11**(2), 226–235 (1969)

[21] Yuan, G., Meng, Z., Li, Y.: A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations. Journal of Optimization Theory and Applications **168**(1), 129–152 (2016). DOI 10.1007/s10957-015-0781-1. URL http://dx.doi.org/10.1007/s10957-015-0781-1