DCASVMを用いた高性能な大規模階層的文書分類

High-Performance Large-Scale Hierarchical Text Classification with DCASVM

佐々木 裕　マハマド・ゴラム・ソフラブ　三輪 誠
Yutaka Sasaki　Mohammad Golam Sohrab　Makoto Miwa
豊田工業大学
Toyota Technological Institute (TTI)

{yutaka.sasaki,sohrab,makoto-miwa}@toyota-ti.ac.jp

1　はじめに

近年の文書分類研究においては、単に学習データやテストデータが大規模なだけではなく、分類対象となるカテゴリー（＝クラス）数が大規模なデータの扱いが課題となっている。我々は、特に、単純な大規模多クラス多ラベル分類問題ではなく、カテゴリーが多層構造を持つ階層的多クラス多ラベル分類問題に興味を持っている。

2012 年開催された Third PASCAL Large-Scale Hierarchical Text Classification (LSHTC3) Challenge[3] の Wikipedia Medium サブタスクのデータの概要を表 1 に示す。さらに、Wikipedia Large サブタスクは，階層化された 478,020 の Wikipedia カテゴリーに関連付けられた 2,365,436 の訓練用 Wikipedia 文書に基づき、452,167 のテスト用 Wikipedia 文書を分類するというさらにチャレンジングなタスクである。Medium サブタスクでは、カテゴリー体系はDAGであるが、Large サブタスクでは一般の有向グラフであり、カテゴリの上位下位関係の循環も許されている。

これまでの研究において、Medium データに関して、既存のシステムを越える予測性能を達成できていたが、 1 回の学習に約 16 時間を要しており [5]、動作の詳細な分析をすることが難しかった。そこで、さらに高速化を進め、予測性能を低下させることなく、Wikipedia Medium データに対するファイル入出力を含むすべての学習を 30 分程度で完了する方法を探る。

本目標を達成するためのポイントは以下の通り。

表 1: Wikipedia Medium データ

#training data	456,866
#distinct features	346,299
max #features per document	1,349
min #features per document	2
average #features per document	48.05
max #documents per leaf category	11,400
min #documents per leaf category	1
average #documents per leaf category	23.09
max #categories per document	50
min #categories per document	1
average #categories per document	1.84
#test data	81,262
#distinct features	132,296
max #features per document	903
min #features per document	2
average #features per document	47.62
#categories in the hierarchy	50,312
#leaf categories	36,504
#edges	65,333
max depth	12

- これまで、学習アルゴリズムとして高速な Pegasos (Primal Estimated sub-GrAdient SOlver for SVM)[6] を用いていたが、さらなる高速化のために Dual Coordinate Ascent SVM (DCASVM) を実装する。
- 予測性能の向上のため、大域的枝刈りの基準として Accumulative Clipped Classification Score (ACCS) を考案し、適用する。

2 手法

2.1 ボトムアップ伝播

学習フェーズにおいてカテゴリー階層構造を利用するためには、訓練データ ID が階層構造に対応づけられていなければならない。LSHTC3 訓練用データには末端のカテゴリーにしか付与されていないため、学習の前処理として、訓練データ ID を階層構造に従って末端ノードからルートに向ってボトムアップに伝播する必要がある。階層構造は複数の親ノードを許すため、複数の親ノードがある場合は、分岐しながらボトムアップに階層に訓練データ ID を割当る。ここで、データ ID を末端からルートに向って伝播するのではなく、末端ノードに割当られているデータを記憶しておき、末端ノードの ID をルートに向って伝播する。

2.2 トップダウン学習

学習はトップダウンに行う。対象のノードに伝播された末端ノード集合を対象に、各エッジについて、その下位（子）カテゴリーに伝播されている末端ノード集合に割当られているデータを正例とし、その他を負例として SVM モデルを学習し、エッジに関連付ける。

学習の高速化のために、DCASVM アルゴリズム（Algorithm 1）を実装した。これは、Dual Coordinate Descent SVM [1] にミニバッチ [7] を導入したものである。ここで $clip_{[a,b]}(\cdot)$ は、引数を [a,b] のレンジに切取る関数である。

Pegasos や SGD SVM は、基本的には DCASVM と同様に高速な SVM 学習アルゴリズムである。しかし、Pegasos や SGD SVM は SVM 最適化問題の主問題を問いているため、KKT 条件を停止条件として利用できず、何回の繰り返しが必要かが明確ではない。そのため、これまで繰り返し回数は、余裕を持って「データ数× 100」を利用していた。一方、DCASVM は双対問題を解いているため、KKT 条件により収束を判定しており無駄な繰り返しを大幅に削減できたことが、学習時間の短縮の主な要因となっている。

Algorithm 1 DCASVM

procedure DCASVM
  Let $t \leftarrow 1$.
  loop
    1. $M \leftarrow -\infty$, $m \leftarrow \infty$.
    2. A be the set of randomly drawn samples.
    3.
    for all $x_i \in A$ **do**
      (a) $G \leftarrow y_i w^T x_i - 1$.
      (b) $PG \leftarrow 0$.
      if $\alpha_i = 0$ **then**
        if $G < 0$ **then**
          $PG \leftarrow G$.
        end if
      else
        if $\alpha_i = C$ **then**
          if $G > 0$ **then**
            $PG \leftarrow G$.
          end if
        else
          $PG \leftarrow G$.
        end if
      end if
      (c) $M \leftarrow max(M, PG)$.
      $m \leftarrow min(m, PG)$.
      (d)
      if $PG \neq 0$ **then**
        $\bar{\alpha_i} \leftarrow \alpha_i$.
        $\alpha_i \leftarrow clip_{[0,C]}(\alpha_i - G/x_i^T x_i)$.
        $w \leftarrow w + (\alpha_i - \bar{\alpha_i}) y_i x_i$.
      end if
    end for
    4.
    if $(M - m < \epsilon) \& (t > L)$ **then** break;
    end if
    5. $t \leftarrow t + 1$.
  end loop
end procedure

表 2: 比較実験の結果

Learning Algorithm	Acc	EBF	LBMaF	LBMiF	HF	Time (hour)
DCASVM	0.4445	0.4961	0.2656	0.4972	0.7083	**0.18**
Pegasos	**0.4459**	**0.4974**	0.2646	**0.4990**	**0.7097**	1.41
SGD-SVM	0.4457	0.4969	0.2646	0.4986	0.7092	1.41
PA	0.4005	0.4486	0.2432	0.4506	0.6667	1.33
ROMMA	0.3814	0.4285	0.2214	0.4337	0.6536	1.17
logreg	0.3515	0.4072	0.1490	0.4171	0.6396	2.61
LSHTC3 System						
(1st) arthur [8]	0.4382	0.4937	0.2674	0.4939	0.7092	-
(2nd) coolvegpuff	0.4291	0.4824	0.2507	0.4779	0.6892	-
(3rd) TTI [4]	0.4200	0.4771	**0.2835**	0.4725	0.6922	-

2.3 トップダウン分類と確信度

テストフェーズにおいて、データ x に関するリンク (n_1, n_2) の SVM の出力値が $g_{(n_1,n_2)}(x)$ であるとすると Clipped Classification Score (CCS) を以下のように定義する。

$$CCS_{(n_1,n_2)}(x) = \frac{clip_{[-1,+1]}(g_{(n_1,n_2)}(x)) + 1}{2}.$$

Accumulative Clipped Classification Score (ACCS) は下記の通り。

$$ACCS(x, m) = \prod_{e \in E} CCS_e(x).$$

ここで E は root からノード m の親ノード至るリンクの集合である。各リンクでの分類を $g_{(n_1,n_2)}(x) + (ACCS(x, n_2) * 2 - 1) > 0$ により判断することで確信度の高い分類パスにおいて再現率を改善する。次節の枝刈りで怪しいカテゴリの割り当てを削除することで適合率を維持する。

2.4 大域的枝刈り

ACCS 値を確信度と看做し、確信度に閾値を設け、大域的に枝刈りする。ただし、確信度が低くとも、少なくとも 1 つのカテゴリーを各データに残す。確信度に対する閾値 θ は、過去の研究によりテストセットへの平均カテゴリー付与数が 1.5 となる値とした。

3 実験結果

3.1 実験の設定

DCASVM を用いて、65,333 のエッジ SVM 分類器を学習した。$\epsilon = 0.01$、バッチサイズは 100 とした。96GB メモリ搭載の Xeon 3.0GHz サーバにより実験を行った。LSHTC 3 Medium データは、標準的なストップワードを除いた単語ユニグラムに基づく素性ベクトルが与えられている。比較対象の学習手法は、sofia-ml の Pegasos、SGD SVM、Passive Aggressive (PA)、ROMMA、Logistic Regression (log-reg) である。SVM の C パラメータは 0.5、その他はデフォルトを用いた。

LSHTC3 においてシステム比較に用いられた評価尺度は下記の 5 種類である、

- Accuracy (Acc):
  $1/\mid D \mid \sum_{i \in D} \mid Y_i \cap Z_i \mid /(\mid Y_i \cup Z_i \mid)$
  ここで、D はデータの集合、Y_i は正解ラベル集合、Z_1 は出力ラベル集合である。
- Example-based F1 measure (EBF):
  $1/\mid D \mid \sum_{i \in D} 2 \mid Y_i \cap Z_i \mid /(\mid Y_i \mid + \mid Z_i \mid)$
- Label-based Macro-average F1 measure (LBMaF): カテゴリ毎の F1 スコアを平均したマクロ F1 スコア
- Label-based Micro-average F1 measure (LBMiF): カテゴリを区別しない F1 スコア
- Hierarchical F1-measure (HF) [2]: 末端カテゴリの上位のカテゴリも考慮した EBF スコア。

3.2 評価結果

各学習手法との比較結果を表 2 に示す。この表の学習時間はファイルの入出力を除き、純粋に機械学習アルゴリズムにより消費された時間である。Pegasos と SGD SVM が最も高い性能を示し

たが、DCASVM は、11 分の学習時間でこれらに近いスコアを残した。他の学習アルゴリズムは、DCASVM より数倍以上の学習を要した。テストフェーズにおける分類時間はどの手法も約 10 分程度である。ファイル入出力やオーバヘッドを含めた DCASVM の学習時間は 31 分であり、ほぼ目標を達成した。LSHTC3 の 1〜2 位システムの学習時間は明らかではないが、十数時間〜数日を要すると推定される。

このように、DCASVM を用いることにより、大規模階層的分類の学習を 30 分程度に抑えることができた。これにより、表 3 に示すような階層の深さごとのスコアを検証することが現実的になる。この結果は、学習フェーズにおいて、各リンクに対応する SVM 分類器を学習する際のデータに対して 2 分割交差確認を行った結果である。ただし、ここでデータは上位からの分類の影響を受けておらず、各リンクでの正解データを用いた交差確認の結果である。なお、交差確認を用いて、上位から下位のノードにデータを分類しながら流していく学習の実験も行ったが、階層が深くなるほど分類誤りが蓄積していき、十分な学習データが確保できなかった。

表 3: DCASVM による学習における階層の深さと性能の関係

Depth	Accuracy	MacroF1	Micro F1
1	0.9535	0.6615	0.9762
2	0.5415	0.4656	0.7025
3	0.6378	0.4606	0.7789
4	0.6722	0.4684	0.8039
5	0.7326	0.4429	0.8457
6	0.6537	0.4443	0.7906
7	0.6912	0.4841	0.8174
8	0.6803	0.4674	0.8097
9	0.7614	0.5049	0.8646
10	0.7572	0.5282	0.8618
11	0.4737	0.5889	0.6429

4 まとめと今後の課題

Wikipedia Medium データを用いて大規模階層的文書分類実験を行い、DCASVM を用いることにより、他の効率的学習アルゴリズムと比較しても十分高い性能（正解率 0.4445）を得ることができた。DCASVM により、従来数時間かかっていた学習時間を 31 分に短縮することができることを明かにした。今後、Wikipedia Large データに本手法を適用していきたい。

謝辞

本研究は、科研費 25330271 の支援をいただきました。ここに深く感謝いたします。

参考文献

[1] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan, A Dual Coordinate Descent Method for Large-Scale Linear SVM, in Proc. of ICML-08, pp. 408–415, 2008.

[2] S. Kiritchenko: Hierarchical Text Categorization, Its Application to Bio-informatics, *Ph.D. thesis*, University of Ottawa Ottawa, Ont., Canada, 2006.

[3] PASCAL LSHTC3 Challenge, http://lshtc.iit.demokritos.gr/.

[4] Y. Sasaki and D. Weissenbacher, TTI'S System for the LSHTC3 Challenge. *ECML/PKDD-2012 Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*, Bristol, 2012.

[5] 佐々木裕, デイヴィー・ヴィッセンバッシャー: LSHTC3 データを対象にした大規模階層的文書分類, 言語処理学会第 19 回年次大会, 2013.

[6] S. Shalev-Shwartz, Y. Singer, and N. Srebro: Pegasos: Primal estimated sub-gradient solver for SVM, *in Proc. of ICML-07*, 2007.

[7] S. Shalev-Shwartz and T. Zhang, Accelerated Mini-Batch Stochastic Dual Coordinate Ascent, *Proc. of NIPS-2013*, 2013.

[8] X.-L. Wang, H. Zhao, and B.-L. Lu. 2012. A Meta-Top-down Method for Large-scale Hierarchical Classification. *in Proc. of ECML/PKDD-2012 Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*, Bristol.