LSHTC3データを対象にした大規模階層的文書分類

Large-Scale Hierarchical Text Classification for LSHTC3 Data

佐々木 裕　デイヴィー ヴィッセンバシャー
Yutaka Sasaki　Davy Weissenbacher
豊田工業大学
Toyota Technological Institute

{yutaka.sasaki,davy.weissenbacher}@toyota-ti.ac.jp

1　はじめに

2012年に開催された Third PASCAL Large-Scale Hierarchical Text Classification (LSHTC3) Challenge [4] では，3種類のトラックが設定された．トラック1は，Wikipediaカテゴリーへの文書分類を対象にしており，データサイズにより Mediumサブタスクと Largeサブタスクの2つのタスクが設定された．

Mediumサブタスクは，階層化された 50,312 の Wikipediaカテゴリー (末端ノードの数は 36,504) に関連付けられた 456,866 の訓練用 Wikipedia 文書に基づいて，81,262 のテスト用 Wikipedia 文書を分類するという非常にチャレンジングなタスクである．文書には末端のカテゴリーのみが付与されており，カテゴリー体系は，ループのない合流を許す階層構造（すなわち DAG）になっている．

Mediumデータは，情報検索のデータとしては膨大とは言えないが，機械学習により扱うデータとしては非常に大規模なデータである．単純に末端カテゴリーへの分類を学習しようとすると，約45万件の訓練データを用いて，各末端カテゴリーに分類するか否かを決定するモデルを3万6千モデル学習することになる．高い分類性能を達成を目指すために，ある程度計算コストのかかる機械学習アルゴリズムを用いる場合，学習および分類にかかる時間を現実的な範囲に抑えることは難しい．実際，このような大規模な問題に対しては，実用時間内に分類処理をするため，情報検索により分類対象文書に近い文書を検索し，関連文書の持つカテゴリを出力するという簡便な手法をとることが多かった．しかしながら，このような方法では，機械学習を用いる方法と比べると，高い分類性能を実現することはできない．

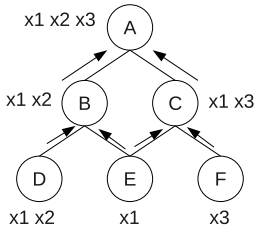

図 1: Bottom-up propagation of training data

そこで，本報告では，カテゴリーの階層構造を積極的に活用することにより，SVMを用いた高精度な大規模文書分類を現実的な時間内で実現することを目指した TTI システムについて述べる．

2　階層的分類手法：学習フェーズ

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，訓練データが階層構造に対応づけられていなければならない．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（図 1）．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

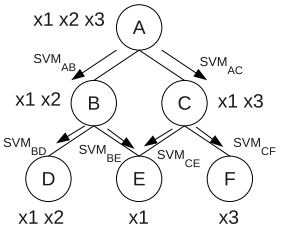

図 2: Top-down training

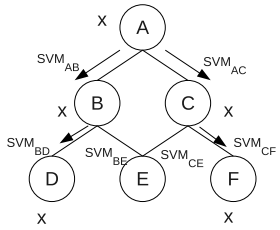

図 3: Top-down classification

データを割当る．この過程は単純であるが，本タスクのように階層構造が非常に大規模な場合は，実装レベルでメモリの消費量と処理時間のバランスに注意する必要がある．実際，Large データでは，カテゴリーサイズが Medium の数倍あるため，現在の実装ではデータを階層構造に実用時間内に割り当てることができなかった．

2.2 トップダウン学習

LSHTC3 チャレンジはコンペティションであり，より高い分類性能を達成するために SVM を機械学習アルゴリズムとして採用した．具体的には，階層構造の各エッジに対して SVM 分類器を割り当て，総計 65,333 の SVM モデルを学習した．[1]

学習はトップダウンに行う．対象のノードに割り当てられているデータを対象に，各エッジについて，その下位カテゴリーに割り当てられている文書を正例とし，その他を負例として SVM モデルを学習し，関連付ける．以下同様に上位のカテゴリーに割り当てられた文書データを対象に下位のカテゴリへ分類を判定する SVM モデルを末端に至るまで学習していく．

図 2 にトップダウン学習の例を示す．カテゴリ B に注目すると，x1 と x2 が割り当てられているので，x1 と x2 のみが学習の対象となる．ノード B から D への分類器 SVM_{BD} にとって x1 と x2 はともに正例であり，ノード B から E への分類器 SVM_{BE} にとって x1 が正例，x2 が負例となる．

階層構造を利用するメリットは，学習が階層の下位レベルに進むに従って学習対象データ数が減少していき，学習時間が大幅に短縮されることにある．

[1] 複数の親ノードを持つ場合があるため，エッジの数はカテゴリー総数より大きくなる．

3 階層的分類手法：分類フェーズ

テストデータは，階層構造に従って，ルートカテゴリーから末端のカテゴリに向かって，先に学習した SVM 分類器により振り分けられる．階層的な分類アプローチをとると，テストデータをルートから末端に向かって分類するにつれ，分類誤りが蓄積していき，トータルでの性能が悪くなると言われていた．[3] ここでは，SVM を用い，かつ分類エラーを修正するための枝刈りを後処理として導入することでこのような問題に対処する．

3.1 トップダウン分類と確信度

図 3 にテストデータ x の分類の例を示す．最初に x は，$\mathrm{SVM}_{AB}(x)$ と $\mathrm{SVM}_{AC}(x)$ により，カテゴリー B と C に分類される．

ここで，正例と負例のバランスを修正するためのバイアス β を導入する．つまり，親ノード p から子ノード c への分類を決定する場合，$\mathrm{SVM}_{pc}(x) > \beta$ であれば，x は子ノードに分類される．$\mathrm{SVM}_{AB}(x) > \beta$ と $\mathrm{SVM}_{AC}(x) > \beta$ の両方が満たされた場合は，x は A と B の両方に分類される．これにより複数カテゴリーへの分類を自然に実現する．また，どの下位ノードへの分類も閾値を満たさない場合でも，必ず 1 つの下位ノードを選択する．選択する基準は，SVM の出力値が最大のものを選ぶ．

また，後段の枝刈りに備えて，階層的分類に合わせて分類の確信度も逐次的に計算していく．$\mathrm{SVM}(x)$ の値を，下記のシグモイド関数により [0,1] の実数値に正規化する．

$$\sigma(x) = \frac{1}{1+\exp(-x)}.$$

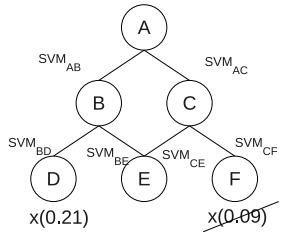

図 4: Global pruning

x が末端ノード n に至ると，確信度 $c(x,n)$ は以下の式で計算される．

$$c(x,n) = \prod_{(n_1,n_2)\in E} \sigma(SVM_{n_1n_2}(x)),$$

ここで，E はルートから n に至る経路で辿ってきたエッジの集合である．実装上は，分岐の時点でそのノードに至る確信度を計算しておき，下位の分岐の際に利用する．

3.2 大域的枝刈り

分類誤りのため各分岐のローカルな分類の時点で判断された分岐が，階層の下位において良くない分岐であったとわかることがある．これは，確信度に閾値を設け，大域的に枝刈りすることで実現できる．この確信度による枝刈りは性能向上に非常に効果的があり，LSHTC3 において TTI システムが上位にランクされた要因となっている．

データ x が階層全体において分類された後，確信度の低い分類を削除する（図 4）．閾値 θ により $c(x,n) < \theta$ となる x の ノード n に対する割り当てを削除する．ただし，階層全体において x に対するカテゴリーの割り当てが全くなくなる場合は，最も確信度の高いノードに対する x の割り当てを残す．

4 実験結果

4.1 実験の設定

SVM^{perf} [1] の線型カーネルを用いて，65,333 のエッジ SVM 分類器を学習した．C パラメータは 1.0 とした．[2] 8GB メモリ搭載の Core i7 3.0GHz マシンにより実験を行った．LSHTC3 では，オーガナイザにより文書データから数値ベクトルに変換されたデータセットが配布されているので，そのデータを利用した．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特徴ベクトルである．

LSHTC3 においてシステム比較に用いられた評価尺度は，Accuracy (Acc)，Example-based F1 measure (EBF)，Label-based Macro-average F1 measure (LBMaF)，Label-based Micro-average F1 measure (LBMiF)，Hierarchical F1-measure (HF) [2] の 5 種類である．

4.2 評価結果

456,886 データによる階層構造中のすべてのカテゴリに対する学習時間は 58,389 秒（約 16 時間）であった．テストフェーズにおける 81,262 データの分類時間は 9,864 秒（約 2.7 時間) であった．本報告の手法により，大規模な階層的分類を現実的な処理時間内で実現することができた．

表 1 にスコアを示す．TTI システムは，Macro F1 スコアでは一位，階層的 F スコアでは二位，その他の尺度では 3 位にランクされた．Medium サブタスクで用いられたデータは，2011 年の LSHTC2 で用いられたデータと同じであったが，2011 年の最高スコアは Acc で 37 %であり，約 5 %の向上がみられた．このことは TTI システムが，これまでに用いられていなかった手法を提案したことを裏付けている．

表 2 は，後処理における枝刈りパラメータ θ を変えたときのスコアの変化を表している．θ=0 のスコアは，2011 年の LSHTC2 において同じデータを用いて評価したときの最上位システムのスコアを同等である．これに対して，θ=0.12 の閾値により枝刈りを行うことにより，5 %程度のスコアの改善が得られた．これにより，TTI システムのスコアが参加システムの上位にランクされた．

4.3 まとめと今後の課題

LSHTC3 Wikipedia Medium データを用いて大規模階層的文書分類実験を行い，参加 17 システム中で上位 3 位以内にランクされる TTI システムを構築した．大規模階層に含まれる約 6 万のエッ

[2] 実際には，SVM^{perf} の C パラメータ C^{perf} は，データ数を N とすると $N * C/100$ である．

表 1: Evaluation Results in the Wikipedia Medium subtask

Name	Acc	EBF	LBMaF	LBMiF	HF
arthur	**0.4382**	**0.4937**	0.2674	**0.4939**	**0.7092**
coolvegpuff	0.4291	0.4824	0.2507	0.4779	0.6892
TTI	0.4200	0.4771	**0.2835**	0.4725	0.6922
chrishan	0.4117	0.4768	0.2454	0.4187	0.6766
anttip	0.4077	0.4460	0.2385	0.4309	0.6803
dhlee	0.3848	0.4352	0.2824	0.4209	0.6662
szarak	0.3711	0.4366	0.2195	0.3769	0.6447
brouardc	0.3536	0.4182	0.2398	0.3739	0.6428
daq	0.3526	0.3896	0.1929	0.3653	0.6330
SSir	0.3270	0.3873	0.1681	0.3879	0.6394
hautcs	0.3204	0.3473	0.1037	0.3274	0.6030
KULeuven	0.2976	0.3408	0.1825	0.3045	0.5494
Peaceguard	0.2499	0.2917	0.0555	0.2754	0.5916
Knn Baseline	0.2491	0.3176	0.1758	0.2979	0.5609
TUD_KE	0.2448	0.2839	0.1303	0.2762	0.5368
dicaro	0.0626	0.0805	0.0211	0.0716	0.3448
glouppe	0.0473	0.0846	0.1758	0.0971	0.6676

表 2: Results

θ	Acc	EBF	LBMaF	LBMiF	HF
0.00	0.3603	0.4316	0.2763	0.3833	0.6197
0.10	0.4140	0.4742	0.2889	0.4652	0.6875
0.11	0.4172	0.4758	0.2860	0.4692	0.6901
0.12	0.4200	0.4771	0.2835	0.4725	0.6922
0.13	0.4220	0.4778	0.2803	0.4751	0.6939
0.14	0.4236	0.4781	0.2767	0.4767	0.6951
0.15	0.4244	0.4778	0.2730	0.4775	0.6956
0.16	0.4245	0.4770	0.2689	0.4776	0.6959
0.17	0.4240	0.4759	0.2645	0.4769	0.6957

ジすべてに対応する SVM 分類器を構築し，後処理において分類確信度を用いた枝刈りにより，高い分類性能を実現した点が本研究の特徴である．

今後の課題として，Medium データに対する性能の向上が上げられる．多くの指標で一位となったシステムは，階層全体に対するメタな分類学習を行っている．さらなる枝刈り手法の検討を行っていきたい．もうひとつの課題は，Wikipedia Large データを対象にした階層的分類を実現することにある．現状の手法および実装は高性能ではあるが，Large データに対しては，そのデータおよび階層構造のサイズが巨大すぎで学習・分類が実用的時間内には行えない．現在，SGD SVM を改良するなど，より高速な SVM アルゴリズムを考案するとともに，GPGPU 等による超並列処理により高速化する手法の検討を進めている．

参考文献

[1] Joachims, T.: A Support Vector Method for Multivariate Performance Measures, *Proc. of the International Conference on Machine Learning (ICML-05)*, pp. 377–384 (2005)

[2] Kiritchenko, S.: Hierarchical text categorization, its application to bio-informatics. *Ph.D. thesis*, University of Ottawa Ottawa, Ont., Canada (2006)

[3] Malik, H.: Improving hierarchical SVMs by hierarchy flattening and lazy classification. *In Proc. of the ECIR 2010 Large Scale Hierarchical Classification Workshop* (2010)

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