

動画画像自動分類への大規模分散オンライン機械学習フレームワーク Jubatus の適用

Application of Jubatus, Distributed Online Machine Learning Framework, to Automatic Motion-Picture Classification

辻 雄介[†]
Yusuke Tsuji

黄 宏軒[†]
Hung-Hsuan Huang

川越 恭二[†]
Kyoji Kawagoe

1. はじめに

近年、インターネット網普及とセンサ技術の発展に伴い、計算機で扱うデータの種類と量が急激に増加している。離れた場所に設置された多数のカメラやセンサー機器から、文字列だけでなく、大量の画像や音声や動画などの多様なマルチメディアデータを効率よく扱う必要性が高まっている。特に、監視用ビデオカメラの多数設置やインターネット動画配信サービスの普及から、ビデオからの不正行動監視やビデオの著作権保護などへのニーズも高まりつつある。

一方、データマイニングやビジネスインテリジェンスのように、大規模で複雑に関連したデータに対して高速に知識獲得・分析を行う技術も研究されている。特に、高速に知識獲得・分析を行う大規模分散オンライン学習フレームワークが開発されている。

しかし、先に述べたように、登録された動画画像の著作権物チェックや、監視カメラからのストリーム動画画像による不正行動監視等を行うには、従来の機械学習による知識獲得・分析方法では、効率的ではない。例えば、大量データを対象として、分散処理フレームワークである MapReduce 上で動作する機械学習ライブラリとして Mahout[1] がある。Mahout を用いてあらかじめ蓄積した大量の数値および文字列を対象とした知識学習を行うことは可能であるが、頻繁に投稿される動画画像データに対しては適用可能ではない。例えば、1 分間あたり 72 時間分の動画画像がアップロードされるインターネット動画配信サービスで動画画像を対象とした知識獲得・分析は困難である。

そこで、本研究では、大規模分散オンライン学習フレームワークとして既に一般に提供され利用可能な Jubatus を用いて、その動画画像分類処理への適用を試みる。適用する際に、Jubatus 利用のための新たな現実的構成法を提案する。なお、Jubatus のテキストマイニング分野での適用例は報告されているが、著者らが調べたところ、動画画像への適用例は報告されていない。

2. Jubatus [2]

Jubatus は、大量のデータを高速に深い分析をすることを旨とした大規模分散機械学習フレームワークである [2]。ここで、深い分析とは高度な統計処理や機械学習を意味する。この深い分析を分散環境で実施するには複数の計算機が処理結果を必要に応じて共有することが必要となる。この共有化のための通信がボトルネックとなり性能が出ないことがある。そこで、Jubatus では、共有化のための通信ボトルネックを避けるため、学

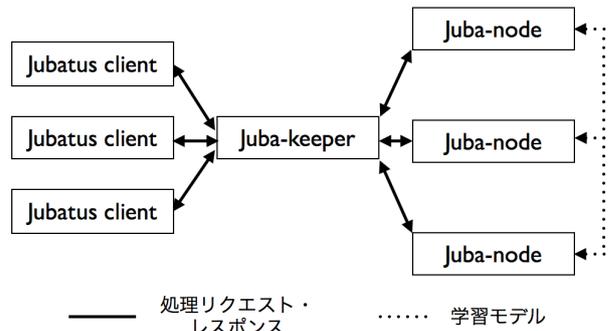


図 1: Jubatus の基本構成

習がある程度進むか一定時間が経過したタイミングで他のサーバの学習結果との差分をとり、その差分のみを共有することによって克服している。Jubatus は、現在、多クラス分類・レコメンド・線形識別器・統計処理の各機能が実装されている。

Jubatus の基本構成を図 1 に示す。

図 1 において、Jubatus client は処理のリクエストと処理結果の受信を行う。Juba-node は他クラス分類・レコメンド・統計処理などを行い、その結果を Jubatus クライアントに返す。Juba-keeper は Juba-node を隠蔽している。これにより Jubatus client は起動しているノードの数や場所を知る必要がなくなり、Juba-node をスケールアウトさせることができる。

3. 動画画像自動分類への適用

3.1. Jubatus 適用における基本的考え方

Jubatus を動画画像分類に適用するには、次の点を決定する必要がある。1) 知識獲得・分析のための動画画像処理をどこで行うか、2) 大量の動画画像をどのように Jubatus client や Juba-keeper や Juba-node が参照できるようにするか。

1) に関して、多クラス分類・レコメンド・線形識別器・統計処理を行う Juba ノードと同様に、動画画像分類を行うための専用ノードを開発するアプローチがある。しかし、動画画像分類では動画画像データから特徴量を抽出することで、従来の多クラス分類機能が活用できることから、「動画画像プラグイン」の方法を採用する。Jubatus には、分散処理ノードで処理を実行する前後にアプリケーションに特化した独自の処理をプラグインとして記述し実行できる仕組みが提供されている。このプラグインを分類を行うための動画画像データへの前処理として実現することで、既存の機能を活用することが可能となる。

2) に関しては、文字列や数値と同様に、Jubatus client から Juba-keeper、さらに各 Juba-node に対し

[†]立命館大学

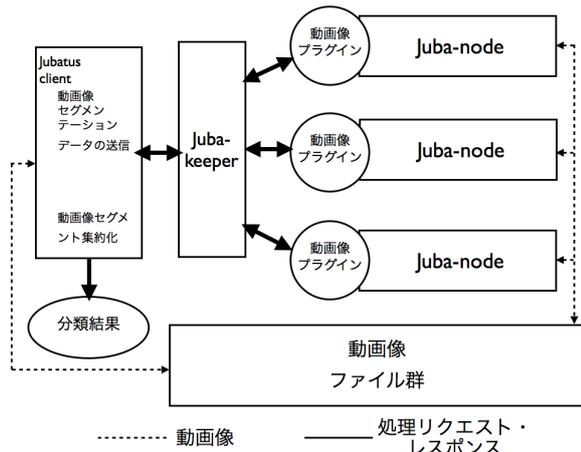


図 2: 動画画像への Jubatus の適用

て動画画像を転送するアプローチがある。しかし、通信ボトルネックを解消することができた Jubatus ではこの直接転送がさらに通信ボトルネックを再発させる可能性がある。そこで、動画画像データは随時、登録されるものであり、一旦、登録されれば更新や削除は行われない特性を考える。動画画像データを転送するのではなく必要時に動画画像データを格納したファイルにアクセスすることで、各コンポーネントから動画画像を参照することができる。したがって、動画画像データが格納されている場所情報だけを転送すればよい。

3.2. Jubatus 利用の動画画像分類システムの構成

Jubatus 利用の動画画像分類システムの構成を図 2 に示す。図 2 において Jubatus client では、動画画像の分類処理要求と分類結果を受け取る、一方、Juba-node では、動画画像プラグインで、分類のための特徴量を動画画像から抽出する。実際の動画画像データはファイルとして格納し共有化を図る。

3.3. 動画画像分類システムのコンポーネント

Jubatus client, Juba-keeper, Juba-node, 動画画像プラグインの 4 つのコンポーネントから構成される。それぞれの役割について説明する。

Jubatus client: Jubatus client は対象となる動画画像を適当な長さの動画に分割し、Juba-node からアクセス可能にする。Juba-keeper へ学習もしくは分類データとして対象動画画像へのパスを送信する。また、分類した動画画像ごとに分類結果が得られるため、それらを統合し入力動画画像としての分類結果を生成する。

Juba-keeper: Juba-keeper は Jubatus client から Juba-node 群へのリクエストを仲介するプロセスである。Jubatus では、Juba-keeper を介することによって起動している Juba-node を意識する必要がなくなり、スケールアウトが容易になる。本システムでは Jubatus の Juba-node を変更なしに用いる。

Juba-node: Jubatus では、Jubatus client からのリクエストを受け、学習や統計処理を行い、リクエストに応じて分類結果や学習結果を返す。本システムでは Jubatus の Juba-node を変更なしに用いる。

動画画像処理プラグイン: Juba-node 上で動作し、動画画像に関する知識獲得・分析のために必要な特徴量の算

出を行う。

3.4. Jubatus 利用の動画画像分類システムの流れ

1. 入力: 利用者が本システムを使用するには、Jubatus client への要求によって行う。対象動画画像を直接伝えるのではなく、対象動画画像へのパスを指定する。Jubatus client は対象動画画像を数秒毎のデータに分割する。分割したデータは Jubatus client および Juba-node 群の双方から参照できる領域に保存される。その後、分割したそれぞれのデータについて、保存した領域へのパスを含めたリクエストを送信する。学習のリクエストの場合は正しいクラス名を併せて送信する。

2. 動画画像プラグインによる特徴抽出: リクエストを受け取った Juba-node は、プラグインによって動画画像を読み込み、特徴を抽出する。

3. 分類結果の統合: Juba-node での動画画像分類処理が終わると Jubatus client は各動画画像セグメントについて、どのクラスに近いかという情報を得る。クライアントは各セグメントについての情報を統合し、動画画像全体での判定結果を出力する。

4. Jubatus 利用の動画画像分類システムの実装と評価

Jubatus のバージョン 0.2.3 を使用する。また、動画画像分類のための特徴量として、OpticalFlow [3] ベースの特徴量を用いる。本システムの実行環境は、PC を 2 台 (OS:Windows 7 Professional, CPU: Intel Core i7-3930K, メモリ:24GB) と使用するとともに、1 台あたり 3 台の仮想マシン (OS:Ubuntu 12.04 LTS, CPU: 仮想 2 コア, メモリ:4GB) である。この環境で本システムの性能評価を行う。性能評価によって、通信オーバーヘッドを克服していること、さらに Jubatus と同様のスケールアウト性能を持つことを確認する予定である。

5. おわりに

本研究で大規模分散オンライン機械学習フレームワーク Jubatus を動画画像分類へ適用するための新たな現実的構成法を提案した。Jubatus では、分散処理ノードへの入力文字列と数値に限定されているため、動画画像プラグインおよび動画画像データ共有を用いた構成を提案した。

今後はクライアントと分散処理ノード間での動画画像の直接交換およびストリーミング的に入力される動画画像データへの対応を行う予定である。将来的には、本構成法およびその改良によって、動画画像データに関する知識獲得・分析機能を追加していきたい。

参考文献

- [1] Chu, C.T. et al., (2007), Map-reduce for machine learning on multicore, Adv. in Neural Inf. Proc. Sys., vol. 19, pp. 281
- [2] Jubatus : Distributed Online Machine Learning Framework, Jun 2012. <http://jubat.us/>.
- [3] Brezeale, D. et al., (2008), Automatic video classification: A survey of the literature, IEEE Trans. on SMC(C), vol.38, No. 3, pp. 416-430