インテル® DAAL を使用した主成分分析パフォーマンスの向上

この記事は、インテル® デベロッパー・ゾーンに公開されている「Improving the Performance of Principal Component Analysis with Intel® Data Analytics Acceleration Library」の日本語参考訳です。

ウェブサイトにアクセスしようとして、長時間待たされたり、アクセスできなかったことはありませんか? もしそうであれば、そのウェブサイトはサービス拒否 [1] (DoS) 攻撃の犠牲になっていたかもしれません。DoS 攻撃は、攻撃者がスパムメールのような情報をネットワークに氾濫させ、ネットワークがその情報の処理に忙しくなり、ほかのユーザーからのリクエストを処理できなくなることで起こります。

スパムメールの DoS 攻撃を防ぐため、ネットワークは「ガベージ」/スパムメールを識別してフィルタリングする必要があります。これを行う方法の 1 つとして、メールパターンをメールスパム署名ライブラリーにあるものと比較します。ライブラリーのものと一致する受信パターンは、攻撃としてラベル付けされます。スパムメールは多種多様であるため、すべてのパターンを網羅したライブラリーを構築することは不可能です。スパムメールの検出率を高めるには、分析しやすくするためデータを再構成する方法が必要です。

この記事では、データの簡素化に使用可能な主成分分析 [4] (PCA) と呼ばれる、教師なし [2] マシンラーニング [3]・アルゴリズムについて説明し、インテル® データ・アナリティクス・アクセラレーション・ライブラリー (インテル® DAAL)[5] を使用してこのアルゴリズムをインテル® Xeon® プロセッサー・ベースのシステム向けに最適化する方法を紹介します。

主成分分析とは?

PCA[7,8] は一般的な分析手法です。データの特性を失わずに複雑さを軽減して、視覚化や分析を容易にするため使用されます。データの複雑さを軽減するということは、元のデータセットの重要な特徴は維持しつつ、元の次元よりも小さい次元にすることを意味します。一般に、K 平均法 [6] などのマシンラーニング・アルゴリズムの前処理段階として使用され、モデルを簡素化して、パフォーマンスを向上します。

図 1 – 3 は、PCA アルゴリズムの仕組みを示しています。問題を簡素化するため、ここでは範囲を 2 次元空間に限定します。

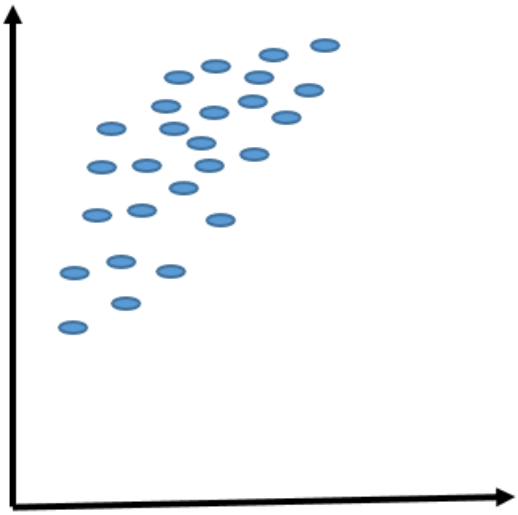

図 1. 元のデータセットのレイアウト

図 1 は、データセットのオブジェクトを示しています。分散が最大になる方向を見つけます。

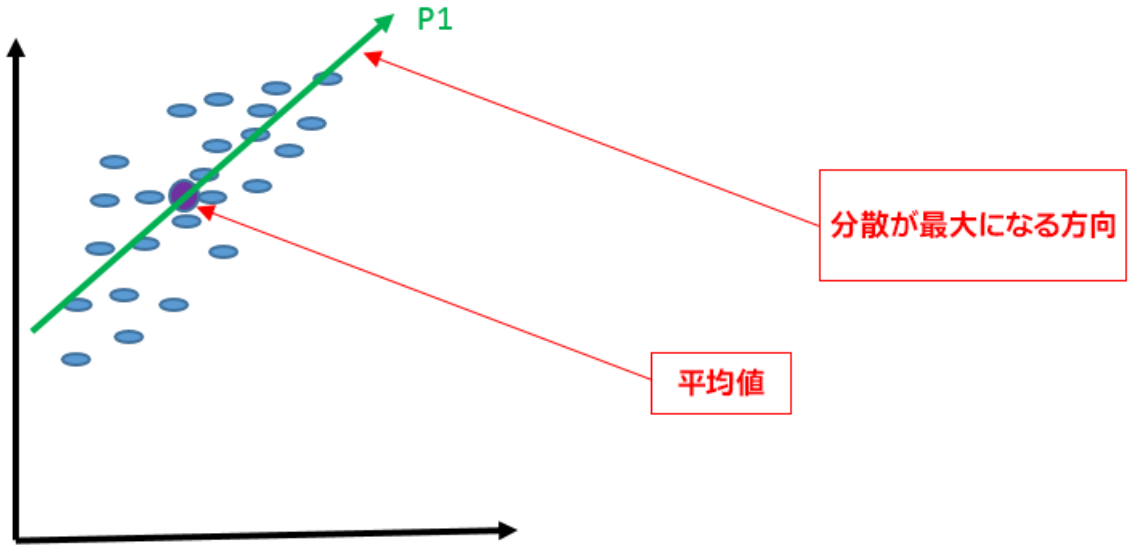

図 2. 平均値と分散が最大になる方向

図 2 は、データセットの平均値と、分散が最大になる方向を示しています。分散が最大となる方向を第 1 主成分と呼びます。

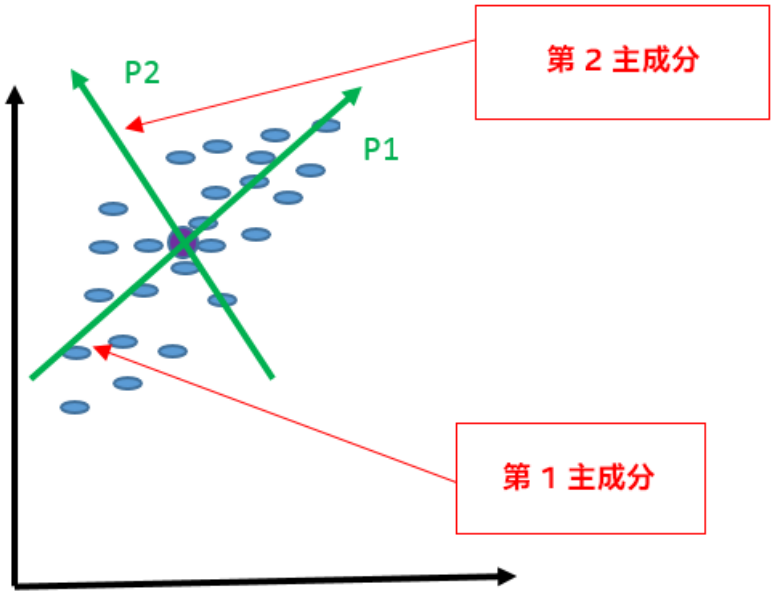

図 3. 次の主成分を見つける

図 3 は、次の主成分を示しています。次の主成分は、分散が 2 番目に大きくなる方向です。第 2 主成分は、第 1 主成分と直行しています。

図 4 – 6 は、PCA アルゴリズムを使用して次元を縮小する方法を示しています。

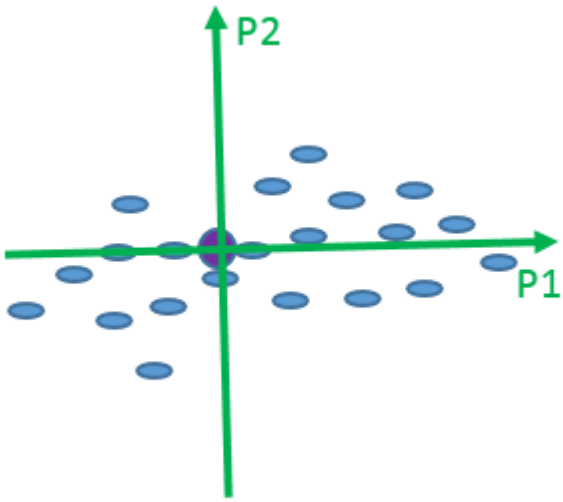

図 4. 回転したグラフ

図 4 は、第 1 主成分に対応する軸 (P1) が横軸になるように回転したグラフです。

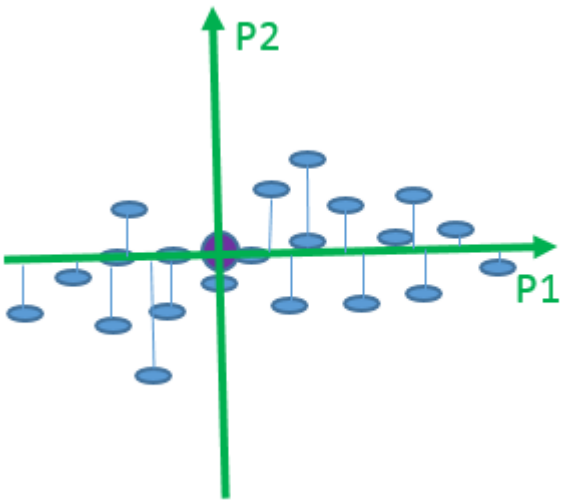

図 5. オブジェクトを P1 軸に投影

図 5 では、第 1 主成分に対応する軸 (P1) が横軸になるようにグラフ全体が回転されています。

図 6. 2 次元から 1 次元に縮小

図 6 は、PCA を使用して、最大分散をベースに 2次元 (P1 と P2) から 1 次元 (P1) に縮小した結果を示しています。同様に、多次元データセットで同じ概念を使用して、分散の低い次元を削除することで、特性を維持しつつ、次元を縮小します。

PCA の使用例

以下は、PCA の使用例です。

- DoS 攻撃とネットワーク・プローブ攻撃の検出
- 画像圧縮
- パターン認識
- 医用画像解析

PCA のメリットとデメリット

以下は、PCA のメリットとデメリットです。

- メリット
  - 高速なアルゴリズム
  - データの最大分散を示すことができる
  - 元のデータの次元を縮小できる
  - ノイズを排除できる

- デメリット
    - 非線形構造は PCA ではモデル化が困難

インテル® DAAL

インテル® DAAL は、データ解析とマシンラーニング向けに最適化された多くの基本ビルディング・ブロックからなるライブラリーです。これらの基本ビルディング・ブロックは、最新のインテル® プロセッサーの機能向けに高度に最適化されています。

次のセクションでは、インテル® DAAL の Python* API である PyDAAL[9] を使用して PCA アルゴリズムを呼び出す方法を示します。

インテル® DAAL の Python* PCA アルゴリズムを使用する

次のステップに従って、インテル® DAAL の Python*[10] PCA アルゴリズムを呼び出します。

1. `import` コマンドと `from` コマンドを使用して、必要なパッケージをインポートします。
    1. 次のコマンドを実行して、データのロードに必要な関数をインポートします。

       ```
       from daal.data_management import HomogenNumericTable
       ```

    2. 次のコマンドを実行して、PCA アルゴリズムをインポートします。

       ```
       import daal.algorithms.pca as pca
       ```

    3. 次のコマンドを実行して NumPy* をインポートします。

       ```
       import numpy as np
       ```

2. `createSparseTable` 関数をインポートして、ファイルから読み取った入力データを格納する数値テーブルを作成します。

   ```
   from utils import createSparseTable
   ```

3. 上記で宣言したデータセット・オブジェクトにデータをロードします。

   ```
   dataTable = createSparseTable(dataFileName)
   ```

   ここで、`dataFileName` は入力 .csv データファイルの名前です。

4. 相関メソッドを使用して、PCA のアルゴリズム・オブジェクトを作成します。

   ```
   pca_alg = pca.Batch_Float64CorrelationDense ()
   ```

   注: svd (単値分解) メソッドを使用する場合は、次のコマンドを使用します。

   ```
   pca = pca.Batch_Float64SvdDense()
   ```

5. アルゴリズムの入力を設定します。

```
pca_alg.input.setDataset(pca.data, dataTable)
```

6. 結果を計算します。

```
result = pca_alg.compute()
```

次のコマンドを実行して、結果を取得します。

```
result.get(pca.eigenvalues)
result.get(pca.eigenvectors)
```

まとめ

PCA は、データセットの次元の縮小に使用される最も単純な教師なしマシンラーニング・アルゴリズムの 1 つです。インテル® DAAL の PCA アルゴリズムは最適化されています。インテル® DAAL を使用することで、アプリケーションを変更せずに、インテル® DAAL の最新バージョンにリンクするだけで、将来の世代のインテル® Xeon® プロセッサーの新機能を利用できます。

関連情報

1. https://ja.wikipedia.org/wiki/DoS 攻撃
2. https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ (英語)
3. https://ja.wikipedia.org/wiki/機械学習
4. https://ja.wikipedia.org/wiki/主成分分析
5. インテル® DAAL の概要 (英語)
6. K 平均法アルゴリズム (英語)
7. http://blog.translucentcomputing.com/2014/03/principal-component-analysis-for.html (英語)
8. www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf (英語)
9. Python* 向けインテル® ディストリビューションのインストール方法 (英語)
10. https://www.python.org/ (英語)

製品とパフォーマンス情報

[1] インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。

注意事項の改訂 #20110804