データマイニング 機械学習機の実装

S202148 柳澤快

課題内容

ナイーブベイズに基づく機械学習器の実装を行い、それに基づき文書の自動分類を行う。

1. 学習用データ（trainディレクトリの記事を形態素解析した train.list ）からパラメータ（事前確率 P(c)、条件付き確率 P(e|c)）を計算し、それらを記録しておくmodelファイルを生成する学習用プログラムを作成。
2. 作成されたmodelファイルからパラメータを読み込み、そのパラメータを使用してテスト用データ（testディレクトリの記事を形態素解析した test.list）の記事を分類する分類用プログラムを作成。

処理手順

- パラメータ推定
  1. ファイルデータを変数へ格納
  2. 各クラスの事前確率P(c) を計算。
  3. 各クラスにおける名詞の条件付き確率P(e|c) を計算
  4. modelファイルへの書き出し
- 文書分類
  1. modelファイルのデータを変数へ格納
  2. test.listのデータを格納
  3. log(P(c)) 計算
  4. ナイーブベイズ学習式 log(P(c))+Σ[i=1..n]log(P(ei|c)) 計算
  5. 学習式計算結果の最大値をクラス名に選定

ソースコード

bayes_learn.pl

```
#!/usr/bin/perl

use strict;
use Encode;
use utf8;


main();

sub main
```

```
{
    my $train_file =
"/Users/yanagisawakai/college2023_1/data_mining/college_perl_project/implementationNaiv
eBayes/train.list";
    createModelFile($train_file);
}

sub createModelFile
{
    my $train_file = $_[0];
    open(my $IN, $train_file) or die("error :$!");

    # ファイルデータを変数へ格納
    my @ClassList;
    my %Occur;
    my $tag_sum;
    while (my $data_utf8 = <$IN>)
    {
        chomp($data_utf8);

        my @FileInfo = split(/ /, decode_utf8($data_utf8));
        my @Tag = split(/,/, $FileInfo[2]);

        my $file_name = $FileInfo[0];
        my $class_name = $FileInfo[1];

        push(@ClassList, $class_name);

        $tag_sum += $#Tag + 1;
        for (my $i = 0; $i <= $#Tag; $i++)
        {
            $Occur{$class_name}{$Tag[$i]}++;
        }
    }

    my %ClassProbability;
    # 各クラスの事前確率計算 P(c)
    my $class_sum = $#ClassList + 1;
    for (my $i = 0; $i <= $#ClassList; $i++)
    {
        $ClassProbability{$ClassList[$i]}++;
    }
    foreach my $class_pro (keys %ClassProbability)
    {
        $ClassProbability{$class_pro} /= $class_sum;
    }

    # 各クラスにおける名詞の条件付き確率 P(e|c)
```

```
    foreach my $class_name (keys %Occur)
    {
        my $tag_sum;
        foreach my $tag (keys %{$Occur{$class_name}})
        {
            $tag_sum += $Occur{$class_name}{$tag};
        }
        foreach my $tag (keys %{$Occur{$class_name}})
        {
            $Occur{$class_name}{$tag} /= $tag_sum;
        }
    }
    close($IN);

    # ファイルへの書き出し
    open(my $OUT, ">model");
    print $OUT encode_utf8("<feature> $tag_sum\n");
    foreach my $class_name (keys %ClassProbability)
    {
        print $OUT encode_utf8("<prior> $class_name $ClassProbability{$class_name}\n");
    }
    foreach my $class_name (keys %Occur)
    {
        foreach my $tag (keys %{$Occur{$class_name}})
        {
            print $OUT encode_utf8("<conditional> $class_name $tag
$Occur{$class_name}{$tag}\n");
        }
    }
}
```

bayes_classify.pl

```
use strict;
use Encode;
use utf8;

main();

sub main
{
    open(my $IN, "model");

    my %ConditionalHash;
    my %PriorHash;
```

```
    my $H = 1;

    # データを変数へ格納
    while (my $data_utf8 = <$IN>)
    {
            chomp($data_utf8);

        my @Data = split(/ /, decode_utf8($data_utf8));

        if ($Data[0] eq "<conditional>")
        {
            $ConditionalHash{$Data[1]}{$Data[2]} = $Data[3];
        }
        elsif ($Data[0] eq "<prior>")
        {
            $PriorHash{$Data[1]} = $Data[2];
        }
        elsif ($Data[0] eq "<feature>")
        {
            $H = $Data[1];
        }
    }

    close($IN);


open($IN,"/Users/yanagisawakai/college2023_1/data_mining/college_perl_project/implemen
tationNaiveBayes/test.list") or die("error :$!");
    while (my $data_utf8 = <$IN>)
    {
        my %NewHash;

            chomp($data_utf8);

        # test.listのデータ格納
        my($input_file, $class, $word) = split(/ /, decode_utf8($data_utf8));
        my @WordList = split(/,/, $word);

        my $nearest = 0;
        my $nearest_key;

        foreach my $class_key (keys %ConditionalHash)
        {
            # log(P(c)) 計算
            my $c_probability = $PriorHash{$class_key};
            $NewHash{$class_key} = log($c_probability);

            # 学習式 log(P(c))+Σ[i=1..n]log(P(ei|c)) 計算
```

```
        foreach my $word_key (@WordList)
        {
            my $e_c_probability = $ConditionalHash{$class_key}{$word_key};
            if($e_c_probability == 0)
            {
                $e_c_probability = 1 / $H;
            }
            $e_c_probability = log($e_c_probability);
            $NewHash{$class_key} += $e_c_probability;
        }

        # 上の学習式計算結果の最大値をクラス名に選定
        if ($nearest == 0 || $NewHash{$class_key} > $nearest)
        {
            $nearest = $NewHash{$class_key};
            $nearest_key = $class_key;
        }
            }

      print encode_utf8("$input_file -> $nearest_key\t$nearest\n");
   }

   close($IN);
}
```

実行結果

課題1

```
yanagisawakai at yanagisawakainoMacBook-Air in
~/college2023_1/data_mining/college_perl_project/implementationNaiveBayes (main●●)
$ /usr/bin/perl
"/Users/yanagisawakai/college2023_1/data_mining/college_perl_project/implementationNaiv
eBayes/bayes_learn.pl"


model

<feature> 43081
<prior> コンピューター 0.0977777777777778
<prior> 周辺機器 0.164444444444444
<prior> 半導体 0.186666666666667
<prior> 情報・コンテンツ 0.0622222222222222
<prior> システム・ソフト開発 0.36
<prior> ゲーム・娯楽 0.0266666666666667
<prior> 通信・インターネット 0.102222222222222
```

<conditional> 半導体 双方 0.00011953143676787
<conditional> 半導体 分散 0.00203203442505379
<conditional> 半導体 データ通信 0.00011953143676787

課題2

yanagisawakai at yanagisawakainoMacBook-Air in ~/college2023_1/data_mining/college_perl_project/implementationNaiveBayes (main●●)
$ perl "/Users/yanagisawakai/college2023_1/data_mining/college_perl_project/implementationNaiveBayes/bayes_classify.pl"
2012-04-18_1.txt -> 周辺機器　-649.101140337079
2012-04-18_3.txt -> 周辺機器　-466.87990365628
2012-04-20_0.txt -> 半導体　-970.489736481923
2012-04-20_1.txt -> システム・ソフト開発　-147.677876398387
2012-04-24_0.txt -> 半導体　-909.909287260704
2012-04-24_1.txt -> コンピューター　-982.200510560697
2012-04-25_0.txt -> システム・ソフト開発　-1690.75046140876
2012-04-25_1.txt -> コンピューター　-935.250243455582
2012-04-26_0.txt -> システム・ソフト開発　-1917.1878678912
2012-04-26_1.txt -> 半導体　-1460.43764090083
2012-04-26_2.txt -> コンピューター　-1019.08921249161
2012-04-27_0.txt -> システム・ソフト開発　-1217.25731739616

考察

実行例のように期待するmodelファイルを作成することができた。
input_fileにファイルパスを格納する際に相対パスでは読み込みすることができずエラーを出してしまった。絶対パスで記述すれば問題なく動作した。原因は不明だが実行環境は次の通りであった。m1 macOS Ventura13.4.1。
コードについてはブロックごとにコメントでどんなブロックなのかの説明をすることでわかりやすいコードを意識した。さらに変数名は小文字、リストやハッシュなどはforeach文の局所変数に変換する際に扱いやすいよう大文字で記述することで変数名に一貫性を持たせた。
ハッシュを用いて多重ループをできる限りなくしているがO(n^2)になってしまっている箇所があるため学習データが膨大になった際の実行速度が懸念点。
また、コース実験でC++での文書分類を実装したがその時はハッシュを用いなかったため非常に実行速度が遅かったが、今回の課題ではハッシュで多重ループをなるべく減らしたためコース実験と比較してかなり速度が向上した。