MACHINE LEARNING Course Report

Name: Zahra Bagherzadeh Khalkhali

Student ID: 610393020

Abstract

We are currently living in the big data era. That term "big data" was first coined around the time the big data era began. The large data volume does not solely classify this as the big data era, because there have always been data volumes lager than ability to effectively work with the data have existed. I will look for ideas and information to turn data into information and knowledge.

Big data is a term for massive data sets having large, more varied and complex structure with the difficulties of storing, analyzing and visualizing for further processes or results. The process of research into massive amounts of data to reveal hidden patterns and secret correlations named as big data analytics. This useful information for companies or organizations with the help of gaining richer and deeper insights and getting an advantage over the competition. For this reason, big data implementations need to be analyzed and executed as accurately as possible. This paper presents an overview of big data's content, scope, samples, methods, advantages and challenges and discusses privacy concern on it.

I work on the prediction of a protein's contact map dataset in this paper.

Introduction

In recent years, the goal of many researchers' in different fields has been to build systems that can learn from experiences and adapt to their environments. This evolution has resulted into an establishment of various algorithms such as decision trees, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), baysian and Random Forest (RF), … that are transforming problems rising from industrial and scientific fields. Based on the nature of the dataset, either balanced or unbalanced, different performance measures and estimation methods tend to perform differently when applied to different machine learning algorithms. The available performance measures, such as accuracy, error rate, precision, recall and ROC analysis, are used while assessing and comparing one machine learning algorithm from the other. In addition to machine learning performance measures, there are various statistical tests, such as McNemar's test and a test of the difference of two proportions, also used to assess and compare classification algorithms. I describe 2 machine learning performance estimation methods these are, k-fold cross validation and test the model on test set.

In this paper we present average the results of the experiments performed using balanced and select feature 4 (for some classifications method) different datasets to be taken from a big data. The accuracy

of the unbalanced big dataset with all instances will be regarded as the threshold, that is, the minimum value for the two performance estimators. (Binary classification)

Methodology

Tree-based methods have long been popular in predictive modeling. Their popularity is owed, in no small part, to their simplicity and predictive power along with the small number of assumptions.

Tree-based methods of modeling have been around since the 1960s, with popular methods including CART, C4.5, and CHAID as common implementations.

- Decision Trees are hierarchical models for supervised learning whereby the local region is identified in a sequence of recursive "splits".
- Decision Trees are composed of internal decision nodes and terminal leaves.
- Each Decision node m implements a test function $f_m(x)$ with discrete outcomes labeling the "branches"

ID3: using a greedy strategy to choose, based on the instances corresponding to the sub-tree in construction, the root of this sub-tree.

Most researches in this area concentrate on the search of new methods to compare attributes and to determine the point where the top-down construction must stop (the pruning problem).

The majority of the algorithms aiming to solve this problem, like **ID3** and **C4.5** work in two different phases:

- Training phase (where the decision tree is built from the available instances)
- Testing or performing phase (where new instances may be classified using the just constructed model)

Random forest: random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest.

Kernel Method

- **Transform $x \rightarrow \phi(x)$**
- **The linear algorithm depends only on xx_i, hence transformed algorithm depends only on $\phi(x)\phi(x_i)$**
- **Use kernel function $K(x_i,x_j)$ such that $K(x_i,x_j)= \phi(x)\phi(x_i)$**

Support Vector Machine

- Support Vector Machines are a system for efficiently training linear learning machines in kernel-induced feature spaces, while respecting the insights of generalisation theory and exploiting optimisation theory.

- با فرض اینکه دسته ها بصورت خطی جداپذیر باشند، ابرصفحه هائی با حداکثر حاشیه (maximum margin) را بدست می آورد که دسته ها را جدا کنند.
- در مسایلی که داده ها بصورت خطی جداپذیر نباشند داده ها به فضای با ابعاد بیشتر نگاشت پیدا میکنند تا بتوان آنها را در این فضای جدید بصورت خطی جدا نمود.
- جدی ترین مسئله در روش SVM انتخاب تابع کرنل است.
- روشها و اصول متعددی برای این کار معرفی شده است:

- diffusion kernel, Fisher kernel, string kernel, …

- در عمل و تحقیقاتی نیز برای بدست آوردن ماتریس کرنل از روی داده های موجود در حال انجام است.

- In practice, a low degree polynomial kernel or RBF kernel with a reasonable width is a good initial try

- Note that SVM with RBF kernel is closely related to RBF neural networks, with the centers of the radial basis functions automatically chosen for SVM

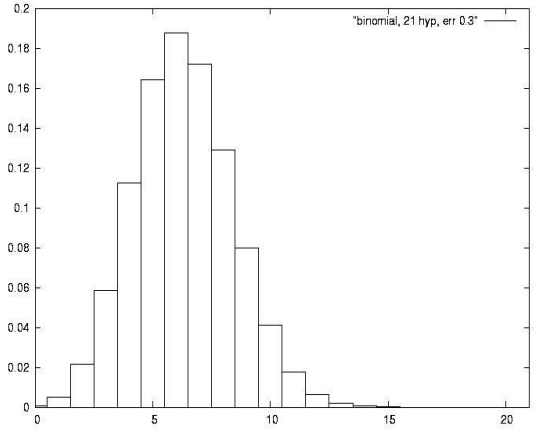
Bayesian Method

- استدلال بیزی روشی بر پایه احتمالات برای استنتاج کردن است
- اساس این روش بر این اصل استوار است که برای هر کمیتی یک توزیع احتمال وجود دارد که با مشاهده یک داده جدید و استدلال در مورد توزیع احتمال آن میتوان تصمیمات بهینه ای اتخاذ کرد.
- در برخی کاربردها(نظیر دسته بندی متن) استفاده از روشهای یادگیری بیزی(نظیر دسته بندی کننده بیزی ساده)توانسته است راه حلهای عملی مفیدی را ارائه کند .نشان داده شده است که کارائی این روش قابل مقایسه با درخت تصمیم و شبکه عصبی بوده است.
- مطالعه یادگیری بیزی به فهم سایر روشهای یادگیری که بطور مستقیم از احتمالات استفاده نمیکنند کمک میکند.
- نگرش بیزی به یادگیری ماشین (و یا هر فرایند دیگر)بصورت زیر است:
- دانش موجود در باره موضوع را بصورت احتمالاتی فرموله میکنیم

- برای اینکار مقادیر کیفی دانش را بصورت توزیع احتمال، فرضیات استقلال و غیره مدل مینمائیم .این مدل دارای پارامترهای ناشناخته ای خواهد بود.
- برای هر یک از مقادیر ناشناخته، توزیع احتمال اولیه ای در نظر گرفته میشود که بازگو کننده باور ما به محتمل بودن هر یک ازاین مقادیر بدون دیدن داده است.
- داده را جمع آوری مینمائیم
- با مشاهده داده ها مقدار توزیع احتمال ثانویه را محاسبه میکنیم
- با استفاده از این احتمال ثانویه:
- به یک نتیجه گیری در مورد عدم قطعیت میرسیم
- با میانگین گیری روی مقادیر احتمال ثانویه پیش بینی انجام میدهیم
- برای کاهش خطای ثانویه مورد انتظار تصمیم گیری میکنیم

Naive Bayes Classifier

Boosting and bagging method



- اگر از نتیجه چند دسته بندی کننده بصورت زیر استفاده شود:

$$f_{com} = vote(f_i, f_j, f_k, f_l, f_m)$$

- به شرط مستقل بودن توابع با استفاده از روابط توزیع دو جمله ای داریم:

$$P(error) = \sum_{k=\frac{N}{2}+1}^{N} \binom{N}{k} p^k (1-p)^{N-k}$$

- برای اینکه بتوان نتیجه مناسبی از ترکیب دسته بندی کننده ها گرفت، این دسته بندی کننده ها باید شرایط زیر را داشته باشند:
- هر یک به تنهائی در حد قابل قبولی دقیق باشند. البته نیازی به بسیار دقیق بودن آنها نیست.
- هر کدام مکمل دیگری عمل کنند. به این معنا که همگی نباید مشابه هم بوده و نتیجه یکسانی تولید کنند.

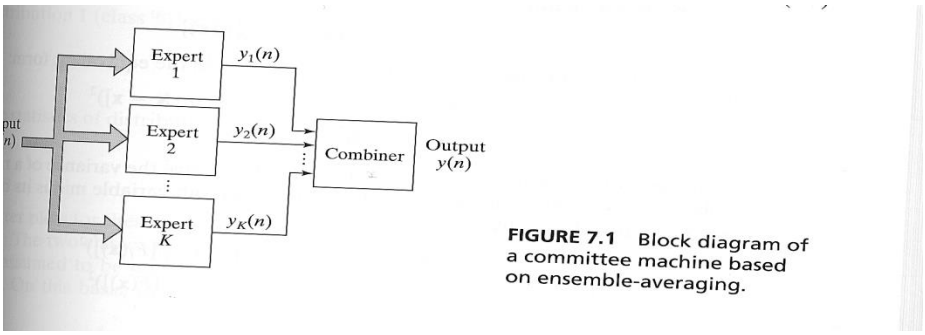


FIGURE 7.1 Block diagram of a committee machine based on ensemble-averaging.

Bagging

این روش نیز مبتنی بر رای گیری است با این تفاوت که یادگیرهای پایه با داده های آموزشی متفاوتی آموزش داده میشوند تا اندکی با هم تفاوت داشته باشند. در نتیجه در حالی که این یادگیرها بدلیل آموزش از مجموعه اصلی مشابه هم خواهند بود بدلیل انتخاب تصادفی نمونه های آموزشی اندکی با هم اختلاف نیز خواهند داشت.

Bagging (Bootstrap Aggregating) - Breiman, 1996

take a training set D, of size N

for each network / tree / k-nn / etc…

- build a new training set by sampling N examples, randomly with replacement, from D
- train your machine with the new dataset

end for

output is average/vote from all machines trained

Boosting

- ایده اصلی:
  - اگر یادگیرهای پایه مشابه هم باشند ترکیب آنها نتیجه متفاوت محسوسی نخواهد داشت. بهتر است که یادگیرها تصمیم گیری متفاوتی داشته و مکمل یکدیگر باشند.
  - در Bagging تفاوت بین یادگیرها از روی شانس و ناپایداری یادگیرهاست.
- در Boosting سعی میشود تا تعدادی یادگیر پایه ضعیف که مکمل هم باشند تولید شده و آنها را با اشتباه یادگیر قبلی آموزش داد.
  - منظور از یادگیر ضعیف این است که یادگیر فقط کافی است که یک کمی از حالت تصادفی بهتر عمل کند. ($e < ½$)
- در مقابل به یادگیری که با احتمال بالائی به دقت دلخواه برسد یادگیر قوی گفته میشود.

- منظور از Boosting این است که یک یادگیر ضعیف را به یک یادگیر قوی تبدیل کنیم.
- به هر یک از دسته بندی کننده های مورد استفاده یک خبره (expert) گفته میشود. هر خبره با مجموعه داده ای با توزیع متفاوت آموزش داده میشود.
- برای پیاده سازی Boosting سه روش مختلف وجود دارد:
  - Filtering
    - در این روش فرض میشود مجموعه داده خیلی بزرگ است و مثالهائی که از آن انتخاب میشوند، یا حذف شده و یا به مجموعه داده برگردانده میشوند.
  - Subsampling
    - این روش با مجموعه داده های با اندازه ثابت بکار برده میشود. داده ها با استفاده از یک توزیع احتمال مشخص مجدا نمونه برداری میشوند.
  - Reweighting
    - این روش نیز با مجموعه داده های با اندازه ثابت بکار برده میشود. ولی داده ها توسط یک یادگیر ضعیف ارزش گذاری شده و به آنها وزن داده میشود

Boosting

take a training set D, of size N

do M times

train a network on D

find all examples in D that the network gets wrong

emphasize those patterns, de-emphasize the others, in a new dataset D2

set D=D2

loop

output is average/vote from all machines trained

- هر خبره بر روی قسمتی از مسئله که یادگیری آن سخت است تمرکز مینماید.
- داده آموزشی هر یادگیر از توزیع متفاوتی بدست می آید.

- نیاز به مجموعه داده آموزشی زیادی دارد
  - برای رفع این مشکل از Adaboost استفاده میشود.

AdaBoost

- در این روش احتمال انتخاب یک نمونه x^t برای قرار گرفتن در مجموعه داده های آموزشی دسته بندی کننده j+1 بر مبنای احتمال خطای دسته بندی کننده c_j تعیین میشود:
  - اگر نمونه x^t بدرستی دسته بندی شده باشد، احتمال انتخاب شدن آن برای دسته بندی کننده بعدی افزایش داده میشود.
  - اگر نمونه x^t بدرستی دسته بندی نشود، احتمال انتخاب شدن آن برای دسته بندی کننده بعدی کاهش داده میشود.
  - تمامی یادگیرها ضعیف و ساده بوده و باید خطائی کمتر از ½ داشته باشند در غیر اینصورت آموزش متوقف میشود زیرا ادامه آن باعث خواهد شد تا یادگیری برای دسته بندی کننده بعدی مشکلتر شود.
- احتمال اولیه انتخاب نمونه های آموزشی (وزن آنها) یکنواخت در نظر گرفته میشود. در واقع وزن هر مثال نشان دهنده اهمیت آن مثال خواهد بود.
- اگر نمونه آموزشی i بدرستی توسط دسته بندی کننده ضعیف فعلی ارزیابی شود توزیع احتمال بعدی آن با ضرب کردن وزن مثال i در عددی مثل $b \in (0,1]$ تعیین میشود (کاهش داده میشود). در غیر اینصورت وزن آن ثابت باقی می ماند.
- فرضیه نهائی از طریق رای گیری وزن دار تعداد T فرضیه ضعیف بدست می آید.

Training:

For all $\{x^t, r^t\}_{t=1}^N \in \mathcal{X}$, initialize $p_1^t = 1/N$

For all base-learners $j = 1, \ldots, L$

Randomly draw $\mathcal{X}_j$ from $\mathcal{X}$ with probabilities p_j^t

Train d_j using $\mathcal{X}_j$

For each (x^t, r^t), calculate $y_j^t \leftarrow d_j(x^t)$

Calculate error rate: $\epsilon_j \leftarrow \sum_t p_j^t \cdot 1(y_j^t \neq r^t)$

If $\epsilon_j > 1/2$, then $L \leftarrow j - 1$; stop

$\beta_j \leftarrow \epsilon_j/(1 - \epsilon_j)$

For each (x^t, r^t), decrease probabilities if correct:

If $y_j^t = r^t$ $p_{j+1}^t \leftarrow \beta_j p_j^t$ Else $p_{j+1}^t \leftarrow p_j^t$

Normalize probabilities:

$Z_j \leftarrow \sum_t p_{j+1}^t$; $p_{j+1}^t \leftarrow p_{j+1}^t / Z_j$

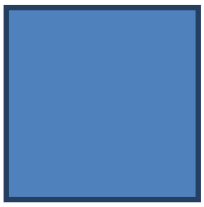
Testing:

Given x, calculate $d_j(x), j = 1, \ldots, L$

Calculate class outputs, $i = 1, \ldots, K$:

$$y_i = \sum_{j=1}^{L} \left(\log \frac{1}{\beta_j} \right) d_{ji}(x)$$

1 page



Block diagram of the proposed method

Advantages and disadvantages of the proposed method

–فضای فرضیهء ID3، یک فضای کامل از توابع مقدار گسستهء متناهی، مربوط به صفات موجود است. چون هر تابع مقدار گسسته، می تواند با نوعی از درخت تصمیم نشان داده شود؛ این الگوریتم مانع از خطرات روشهایی که فضاهای فرضیهء ناقص را جستجو می کنند می شود (مانند روش هایی که فقط فرضیات فصلی را در نظر می گیرند): که در آنها، فضای فرضیه ممکن است حاوی تابع هدف نباشد

– برخلاف برخی روش ها همچون روش حذف–کاندید، فضای نسخه که مجموعهء تمام فرضیات سازگار با مثال های جاری را نگهداری می کرد، در طی عمل تصمیم گیری برای ساخت درخت، فقط یک فرضیهء جاری را حفظ می کند. البته، این الگوریتم توانایی هایی که از بازنمایی صریح تمام فرضیات سازگار ناشی می شود را از دست می دهد. (مثلاً توانایی تعیین اینکه چند درخت تصمیم دیگر با داده های آموزشی سازگار هستند یا ساخت پرسش های نمونهء جدید که به شکل بهینه باعث انتخاب یک فرضیه از بین این فرضیات رقیب می شوند.)

– این الگوریتم هیچ بازگشت به عقبی را انجام نمی دهد. هرگاه صفتی را برای آزمایش در سطح خاصی از درخت انتخاب کرد هیچگاه دوباره آن را بررسی نمی کند. بنابراین مستعد خطرات معمول جستجوی hill-climbing بدون بازگشت به عقب می باشد: همگرایی به راه حل های بهینهء محلی که بهینهء سراسری نمی باشند.

–برخلاف روش هایی که تصمیمات تدریجی خود را بر پایهء مثال های آموزشی می گیرند، الگوریتم ID3، برای اتخاذ تصمیمات آماری راجع به چگونگی بهبود فرضیهء فعلی خود، تمام مثال ها را در هر گام از جستجو بکار می برد. مزیت این عمل این است که جستجوی نهایی، کمتر

به خطاهای موجود در مثال های آموزشی حساس خواهد بود. (می توان این الگوریتم را با تغییر شرط پایانی آن، برای پذیرفتن فرضیاتی که کاملاً با داده های آموزشی جفت نمی شوند توسعه داد.)

Overfitting

الگوریتم ID3 هر شاخهء درخت را تا عمقی رشد می دهد که درخت بتواند مثال های آموزشی را کاملاً دسته بندی کند. هرچند این استراتژی مناسب می باشد، زمانی که در داده ها نویز وجود داشته باشد و یا تعداد مثال های آموزشی برای تولید نمونه ای از تابع هدف صحیح بسیار کم است می تواند برای باعث مشکلاتی شود. در هر یک از این حالات، الگوریتم ID3 می تواند درختانی را تولید کند که مثال های آموزشی را اورفیت می کنند. وقتی درخت با نویز در داده ها سازگار شود اورفیتینگ رخ داده است و در این هنگام ممکن است روی داده های مجموعهء تست بدتر عمل کند.

اورفیتینگ یک مسئلهء عملی مهم برای یادگیری درختان تصمیم و بسیاری متدهای یادگیری دیگر می باشد. برای مثال با مطالعات آزمایشگاهی روی پنج وظیفهء یادگیری با داده های حاوی نویز غیرقطعی، نشان داده شد که اورفیتینگ دقت درخت تصمیم یادگیری شده را 10 تا 25 درصد کاهش می دهد.

خطای مجموعهء آموزشی – تعداد نمونه هایی که مقدار پیش بینی شدهء آنها توسط درخت یادگیری شده با مقدار واقعی آنها متفاوت است. هرچه این مقدار کوچکتر باشد بهتر است.

لازم به ذکر است که random forest برای داده هایی که متوازن نشده اند جواب بهتری میدهد و با دقت 90٪

Support Vector Machine

- Strengths
    - Training is relatively easy
    - Good generalization in theory and practice
    - Work well with few training instances
    - Find globally best model, No local optimal, unlike in neural networks
    - It scales relatively well to high dimensional data
    - Tradeoff between classifier complexity and error can be controlled explicitly
    - Non-traditional data like strings and trees can be used as input to SVM, instead of feature vectors
- Weaknesses

- Need to choose a “good” kernel function.

- SVMs find optimal linear separator
  - They pick the hyperplane that maximises the margin
  - The optimal hyperplane turns out to be a linear combination of support vectors
- The kernel trick makes SVMs non-linear learning algorithms
  - Transform nonlinear problems to higher dimensional space using kernel functions; then there is more chance that in the transformed space the classes will be linearly separable.

Bayesian Method

Half page

Experiments

Description about the details of experiments for example, division of data to test and train set, number of test and train samples, parameter settings etc..

Method name	Parameter name	value	Unit
SVM	gamma	.2	-
	…	5	J
	…	7	Byte
	…	..	..

In each experiment and exercise, complete the tables

روی داده های آموزشی این دقت ها را دارم

Method	Accuracy	Precision	Recall	F-measure	TP rate	FP rate	….
Decision tree	0.628	0.646	0.629	0.616	0.629	0.376	
Random forest	0.609	0.754	0.609	0.540	0.609	0.404	
C4.5	0.647	0.710	0.647	0.615	0.647	0.363	

…							
…							

روی داده های تست دقت به صورت زیر است

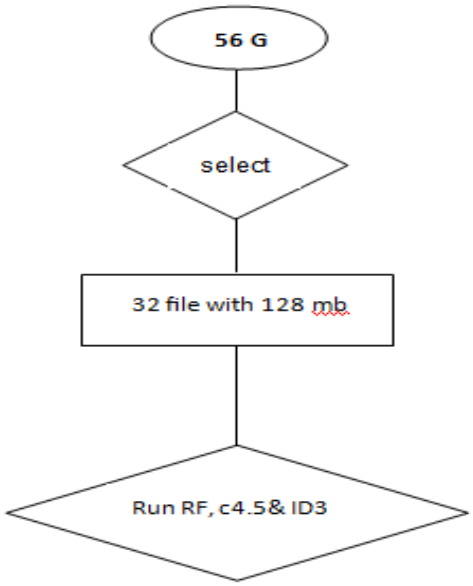
Method	Accuracy	Precision	Recall	F-measure	TP rate	FP rate	….
Decision tree	0.813	0.931	0.813	0.863	0.833	0.624	
Random forest	0.958	0.935	0.958	0.938	0.958	0.953	
C4.5	0.904	0.923	0.904	0.913	0.904	0.865	
…							
…							

Discussion about the results

به تعداد 32 قطعه تصادفی 128 مگا بایتی با به طور میانگین 67000 نمونه از فایل 57 گیگا بایتی ایجاد کردم و روی 4 تا از آنها به طوری که در بالا توضیح داده شده مدل را ساخته و به صورتی که 0.8 داده ها برای آموزش و بقیه برای تست استفاده شده است قبل از آن تنها روی داده های آموزش کار کردم که دقت خوبی داشت زیرا تنها روی داده های آموزش کار میکرد و داده تستی نداشت به صورتی که در بالا گفته شد 10-fold نیز روی آن گرفتم در وکا و میانگین نتیجه های 4 دیتا ست بدست آمده از دیتا ست اصلی را بدست آوردم (همانطور که میدانیم داده های اصلی خود balance نبوده اند و به همین ترتیب قطعه های گرفته شده از آن نیز balance نبوده پس ابتدا balance کرده ام و سپس آموزش انجام داده ام و دقت گرفتم و سپس feature selection انجام داده ام و بعد تست گرفتم در هر دو حالت دقت های بدست آمده درحد چند ده هزارم فرق داشت که من دقت داده ها بعد از انتخاب ویژگی را قرار داده ام) به دلیل اینکه برای گزارش این هفته حاضر نمیشود روی 32 قطعه کار انجام نداده ام و نیز تست نگرفته ام روی روی داده های تست و بر روی هر چهار مدل که در صورت مطرح شد. ولی همانطور که مشاهده میکنید سی 4.5 بهتر عمل کرده اما در کل دقت برای همه داده ها یکسان است.

اگر داده های بدست آمده را متوازن نمیکردم دقت 98٪ و 99٪ بدست می آمد اما این مد نظر نبود و طبقه بندی خوبی ارائه نمی شد اگر و حتی اگر از تمام داده های آموزش استفاده میکردم و دقت درخت را برای داده های آموزش بالا میبردم نیز پسندیده نبود همانطور که در بالا توضیح داده شد پدیده overfitting اتفاق می افتاد.

در اینجا بهتر است بررسی شود که درختان باید تا چه عمقی رشد داده شوند. که در اینجا لازم است از نرم افزارهای دیگری استفاده کرد که بتوان روی طول درخت و نحوه هرس درخت ملاحظه داشت.



یادگیری بیزی:

در این بخش تصمیم براین شد که به تعداد 104 فایل از مجموعه داده های آموزشی انتخاب شود و تعداد داده های کلاس 0 آن از داده های کلاس 1 آن جدا شود و سپس بر روی داده های کلاس 0 عملیات خوشه بندی انجام شود(خوشه بندی به روش k-means با k=۱۰۰ و داده های تصادفی انجام شد) سپس فایل خوشه های بدست آمده برای کلاس های متفاوت نرمال شد و بصورت تقریبی فایلی 25000 تایی از رکورد ها بدست آمد که تعداد اعضای کلاس 0 و کلاس 1 در آن به یک نسبت است سپس روش بیز ساده را روی آن اجرا کردم (در ابتدا قصد داشتم تاایده استفاده از چند کلاسیفایر را اجرا کنم به این معنا که داده های تست را روی تمام کلاسیفایرها اجرا کنم و بین آنها بیشترین دقت بدست آمده را معیار قرار داده و تصمیم بگیرم که آیا متعلق به کلاس است یا خیر در اواسط امر تصمیم گرفتم که تعداد مناسبی از یک ها را انتخاب کرده و ایده کلاسیفایر 1 کلاس را پیاده کنم اما به دلیل زمانی که کلاسترینگ از بین برد به این امر نایل نشدم) و ابتدا برروی داده های آموزش بدست آمده به توضیح بالا اجرا کردم و دقت با جزئیات زیر را بدست آوردم(براساس10-fold validation):

Method	Accuracy	Precision	Recall	F-measure	TP rate	FP rate	ROC Area
Naïve_bayes	0.659	0.659	0.659	0.659	0.659	0.341	0.717
	0.904	0.987	0.904	0.942	0.904	0.442	0.843
	.641	1	.641	.781	.641	.000	1
	.293	0.929	0.293	0.394	0.293	0.141	0.658
	.697	.976	.697	.808	0.697	.428	.679
	.916	.988	.916	.950	.916	.626	.741

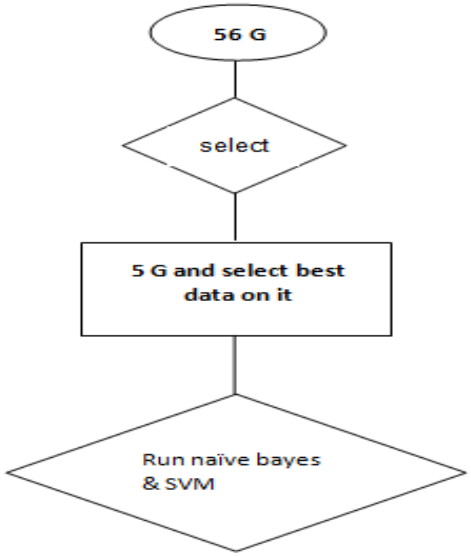
در سطر دوم این جدول جزئیات برای تنها یک مجموعه داده تست با 66000 رکورد است. در سطر سوم این جدول جزئیات برای یک مجموعه داده تست با 111000 رکورد است. در سطر چهارم جدول جزئیات برای یک مجموعه داده تست با90000 رکورد است.

سطر 5 مربوط به یک مجموعه داده تست با 140000 رکورد است. و سطر 6 مربوط به یک مجموعه داده تست با 145000 رکورد است.

تحلیل جدول فوق :

حال به تحلیل جدول بالا میپردازیم

توجه شود که روش بیز زمانی جواب خوبی میدهد که تعداد داده های آموزش به اندازه خوبی باشد (نمایی از تعداد ویژگی ها) به همین جهت در اینجا نتایج خوبی مشاهده نمیکنیم زیرا تعداد داده های آموزش پوشش خوبی را ایجاد نکرده اند. یک نکته که در جدول قابل توجه است دقت است که روی داده آموزش 0.659 است و در سطر دوم و ششم دقت قابل توجهی را مشاهده میکنیم این نیز به این خاطر اتفاق می افتد که در زمان کلاستر کردن مجموعه های جدا شده از داده آموزش ممکن است از هر فایل داده هایی انتخاب شده باشند که شباهت زیادی به هم دارند و یا داده هایی که پوشش زیادی ندارند انتخاب شده باشند زیرا خوشه بندی بر روی کل داده های آموزش انجام نشده است. (در این بین یک آزمایش دیگر نیز انجام داده ام که داده هارا اضافه کردم اما تعدادی از فایل های تست که در مرحله قبل دقت خوبی داشتند در این مرحله دقت بسیار پایین داشتند و این نیز نشان میدهد که در این حالت توزیع داده ها عوض شده و داده هایی که در مرحله قبل برای این دسته بندی تمایز زیادی داشتند در این مرحله داده هایی به مرز نزدیک تر بودند) نکته دیگری که در این بین قابل توجه است میانگین ROC Area در 5 تست انجام شده است که 0.784 است. مجددا عنوان میکنم این روش انتظار میرود با افزایش داده جواب های با دقت بالایی تولید کند اما اگر بتواند توزیع را به درستی مشخص کند.



ماشین بردار پشتیبان(کرنل RBF و normalized poly kernel):

به این دلیل که ماشین بردار پشتیبان با داده های کم جواب های خوبی میدهد و به داده های مرزی حساس است و به این دلیل که تعداد داده های آموزش بسیار زیاد است و هر بار توانایی کار بر روی یک تعدادی از آن را داریم ترجیح دادم که از این کلاسیفایر نیز استفاده شود و نیز چون محاسبات این روش بسیار بیشتر از روش بیز ساده است فایلی را که انتخاب کردم با 1600 رکورد و نتایج آن روی داده های آموزش به صورت زیر در سطر اول جدول زیر است :

Method	Accuracy	Precision	Recall	F-measure	TP rate	FP rate	ROC Area
SVM	0.964	0.965	0.964	0.964	0.964	0.036	0.964
Test1(RBF)	.3	1	.3	.460	.3	0	?
Test1(nor_Poly)	.335	1	.335	.502	0.335	0	?

Method	Accuracy	Precision	Recall	F-measure	TP rate	FP rate	ROC Area
SVM	0.82	0.82	0.82	0.82	0.82	0.18	0.82
Test1(RBF)	.3	1	.3	.460	.3	0	?
Test1(nor_Poly)	.335	1	.335	.502	0.335	0	?

در اینجا بر روی یک فایل تست با 111000 رکورد ماشین بردار پشتیبان را با دو کرنل متفاوت اجرا کرده ام و جزئیات در بالا آمده است.

تحلیلی بر جدول بالا:

در اینجا مشاهده میشود که با وجود داده های کمی که به ماشین بردار پشتیبان داده ایم اما false positive در حد 0 و 0 است و این خود نکته قابل توجهی است. و با کرنل های مختلف درست است که true positive تغییر زیادی نکرده اما این نشان میدهد با کرنل های مختلف و داده های بیشتر که مرز را به خوبی مشخص کنند نتایج بهتری خواهیم گرفت.

AdaBoost with Random forest and id3 classifiers:

در این قسمت روی یک فایل با 88505 رکورد که به تعداد برابر رکوردهای کلاس 0 و یک دارد الگوریتمهای مذکور مورد استفاده قرار گرفت. (بدست آوردن داده ها به این صورت بود که در قسمت قبل نزدیک به 40000 رکورد کلاس 1 بدست آمد و سپس یک کلاسترینگ

روی کلاس 0 انجام شد از کلاستر های مربوط به رکورد های 0 به تعداد برابر با رکورد های کلاس 1 موجود انتخاب شد و فایلی با 88505 رکورد بدست آمد)

Method	Accuracy	Precision	Recall	F-measure	TP rate	FP rate	ROC Area
AdaBoost(RF)	0.976	0.976	0.976	0.97	0.976	0.024	0.992
AdaBoost(id3)	0.790	0.790	0.790	0.790	0.790	0.210	0.868
Bagging(id3)	0.757	0.758	0.757	0.757	0.757	0.243	0.833
Bagging(j48)							
Random Forest	0.975	0.975	0.975	0.975	0.975	0.025	0.997
J48							
Id3							

همانطور که در جدول بالا مشخص است با کلاسیفایر های یکسان عملکرد AdaBoost بهتر است و دقت و نیز ROC بهتری را بدست میدهد. (AdaBoost بهینه تر از Baggingاست) روش bagging در نرم افزار وکا با کلاسیفایر RF جواب نداد. در جدول بالا ملاحظه میکنیم که random forestو adaboost با کلاسیفایرهای random forest عملکرد یکسانی داشته است.

Half page

Time evaluation

Details of your system (RAM, CPU speed), implementation language, and anything affects the consumed time.

RAM = 4G and CPU = 1.6 GHz

Method	train	test	unit
Decision tree	15	1	secs
…			
…..			
…			
…			

Discussion about the results

Half page

نتیجه بر روی کل داده تست برای هریک از مدل ها به صورت زیر است:

Naïve Bayes

===Confusion Matrix===

a	b	<-- classified as	
1564867	1384413	a = 0	0.469
37987	10650	b = 1	0.781

TN = 0.469

TP = 0.781

TN*TP = 0.366

SVM

=== Confusion Matrix ===

a	b	<-- classified as	
1198986	1332343	a = 0	
16319	29344	b = 1	

TN = 0.473

TP = 0.642

TN * TP = 0.303

Random Forest

=== Confusion Matrix ===

a	b	<-- classified as	
212183	1364626	a = 0	

1415 31142 | b = 1

TP = 0.956

TN = 0.134

TP*TN = 0.128

Conclusion

Half page

References

1. BIG DATA, DATA MINING AND MACHINE LEARNING, Jared Dean
2. http://www.stat.berkeley.edu/~breiman/Random forests
3. IMPROVING ADAPTABILITY OF DECISION TREE FOR MINING BIG DATA, HANG YANG and SIMON FONG
4. MICHINE LEARNING, TOM M. MITCHELL
5. PATTERN RECOGNITION AND MACHINE LEARNING, CHRISTOPHER M. BISHOP
6. Introduction to Machine Learning, Ethem Alpaydin
7. Wikipedia
8. Boosting – Schapire & Freund 1990