What I have done last 2 days:

1. I took Mushrooms data from Kaggle, because column names are added.
2. Followed the code on [**https://www.kaggle.com/mokosan/mushroom-classification**](https://www.kaggle.com/mokosan/mushroom-classification) and repeated it:
   1. Use the same chunk for splitting the data to train and test parts
   2. The same chunk for choosing the variables x1=odor and x2=spore.print.color for logistic regression classifier
   3. Same schema and functions about computing confusion matrix and plotting ROC curves.
   4. Same code for the 5 classificators
   5. Similar code for presenting the results in table format
   6. Same code for plotting all ROC curves together.

File: **golden\_truth\_classifiers.R**

**Classification results:**

Accuracy Sensitivity Specifity

Logistic Regression 0.9941545 0.9888535 1.000000

Naive Bayes 0.9782881 0.9692429 0.988465

Neural Nets 0.9941545 0.9888535 1.000000

Random Forest 0.9941545 0.9888535 1.000000

Decision Tree 0.9941545 0.9888535 1.000000

1. Next I repeated exactly the same code, but instead the original data **data\_mushrooms\_KAGGLE** I have used **dataConsensus**, obtained from simulated 10 workers noisy data (30% noise for every worker).

File: **consensus\_classifiers.R**

**Classification results:**

Accuracy Sensitivity Specifity

Logistic Regression 0.8981211 0.9076433 0.8876207

Naive Bayes 0.8926931 0.9066774 0.8776042

Neural Nets 0.8981211 0.9076433 0.8876207

Random Forest 0.8981211 0.9076433 0.8876207

Decision Tree 0.8981211 0.9076433 0.8876207

Here the results show that some poison mushrooms are recognized as eatable (shown in confusion matrices), which could be dangerous.

Note:

Till now I have not used workers ID’s at all. It seems that we do not need them when we use majority voting. It may be that we will need these ID’s if we want to use weighted majority voting, where the particular weights are related to the corresponding ID’s. Then during shuffling for splitting train and test data sets we will need to preserve these ID’s as well.

**What was doen today:**

We chose more realistic mixed populations with quite small sizes (as it was described in most examples in the papers) and small amount of experts. For example in Mechanical Turk most workers are not qualified.

1. Mixed population of 10 workers: true experts: 5% noise – 2, experts 20% noise – 2 amateurs, 40% noise – 3, and 3 adversaries.

File: **consensus\_mixed\_worker\_population\_N10.R**

1. Mixed population of 50 workers: true experts: 5% noise – 5, experts 20% noise – 5, amateurs, 40% noise – 15, and 25 adversaries.

File **consensus\_mixed\_worker\_population\_N50.R**

1. Mixed population of 50 workers: true experts: 5% noise – 10, experts 20% noise – 10, amateurs, 40% noise – 35, and 45 adversaries.

File **consensus\_mixed\_worker\_population\_N100.R**

1. Mixed population of 500 workers: true experts: 5% noise – 50, experts 20% noise – 50, amateurs, 40% noise – 150, and 250 adversaries.

File **consensus\_mixed\_worker\_population\_N500.R**

**REPEATED DATA:**

We assume that we have only N/2 data points, and simulate the workers population as before, but the sizes of different populations are doubled.

1. Mixed population of 20 workers: true experts: 5% noise – 4, experts 20% noise – 4 amateurs, 40% noise – 6, and 6 – adversaries

File: **consensus\_repeated\_data\_N20.R**

1. Mixed population of 100 workers: true experts: 5% noise – 10, experts 20% noise – 10, amateurs, 40% noise – 35, and 45 – adversaries.

File **consensus\_repeated\_data\_N100.R**

1. Mixed population of 200 workers: true experts: 5% noise – 40, experts 20% noise – 40, amateurs, 50% noise – 50, and 70 – adversaries.

File **consensus\_mixed\_worker\_population\_N1000.R**

1. Mixed population of 1000 workers: true experts: 5% noise – 100, experts 20% noise – 100, amateurs, 40% noise – 350, and 450 – adversaries.

File **consensus\_mixed\_worker\_population\_N1000.R**

**Results:**

All for **majority voting** as consensus algorithm.

Mixed population of:

* **10 workers**: true experts: 5% noise – 2, experts 20% noise – 2 amateurs, 40% noise – 3, and 3 adversaries
* **20 workers**: true experts: 5% noise – 4, experts 20% noise – 4 amateurs, 40% noise – 6, and 6 – adversaries
* **50 workers**: true experts: 5% noise – 5, experts 20% noise – 5, amateurs, 40% noise – 15, and 25 adversaries
* **100 workers**: true experts: 5% noise – 10, experts 20% noise – 10, amateurs, 40% noise – 35, and 45 – adversaries
* **200 worker**: 5% noise – 40, experts 20% noise – 40, amateurs, 50% noise – 50, and 70 – adversaries
* **500 workers**: 5% noise – 50, experts 20% noise – 50, amateurs, 40% noise – 150, and 250 adversaries
* **1000 workers**: true experts: 5% noise – 100, experts 20% noise – 100, amateurs, 40% noise – 350, and 450 – adversaries

Repeated data: since the number of data points is 50% less, we have to double the original population size.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **10 workers** | | **50 workers** | | 100 workers | | 500 workers | |
| **Confusion matrix** | Mixed population | Repeated data | Mixed population | Repeated data | Mixed population | Repeated data | Mixed population | Repeated data |
| Log istic Regresion | Ref  Pred. e p  e 1102 **134**  p 154 1005 | algorithm did  not  converge | Ref.  Pred.e p  e 1238 **0**p 18 1139 | algorithm did  not  converge | Ref.  Pred.e p  e 1230 **10**  p 26 1129 | algorithm did  not  converge | Ref.  Pred. e p  e 1242 **0**p 14 1139 | Ref.  Pred. e p  e 1242 **0**  p 14 1139 |
| Naive Bayes | Ref.  Pred.e p  e 1090 **146**  p 153 1006 | Ref.  Pred.e pe 942 **6**p 37 208 | Ref.  Pred.e p  e 1225 **13**p 43 1114 | Ref.  Pred.e pe 967 **3**p 12 211 | Ref.  Pred.e p  e 1217 **23**  p 26 1129 | Ref.  Pred.e pe 968 **3**  p 11 211 | *Ref.*  *Pred. e p*  *e 1229* ***13****p 39 1114* | *Ref.*  *Pred. e p*  *e 1229* ***13****p 39 1114* |
| Neural Nets | Ref.  Pred.e pe 1102 **13** p 154 1005 | Ref.  Pred.e pe 942 **6**p 37 208 | Ref.  Pred.e p  e 1238 **0**p 18 1139 | Ref.  Pred.e pe 967 **3**p 12 211 | Ref.  Pred.e p  e 1217 **23**  p 26 1129 | Ref.  Pred.e pe 968 3  p 11 211 | Ref.  Pred. e p  e 1242 **0**p 14 1139 | Ref.  Pred. e p  e 1242 **0**  p 14 1139 |
| Random Forest | Ref.  Pred.e pe 1102 **13** p 154 1005 | Ref.  Pred.e pe 942 **6**p 36 209 | Ref.  Pred.e p  e 1238 **0**p 18 1139 | Ref.  Pred.e pe 967 **3**p 11 212 | Ref.  Pred.e p  e 1217 **23**  p 26 1129 | Ref.  Pred.e pe 968 **3**  p 10 212 | Ref.  Pred. e p  e 1242 **0**p 14 1139 | Ref.  Pred. e p  e 1242 **0**  p 14 1139 |
| Decision Tree | Ref.  Pred.e p  e 1102 **134**  p 154 1005 | Ref.  Pred.e pe 942 **6**p 37 208 | Ref.  Pred.e p  e 1238 **0**p 18 1139 | Ref.  Pred.e pe 967 **3**p 12 211 | Ref.  Pred.e p  e 1217 **23**  p 26 1129 | Ref.  Pred.e pe 968 **3**  p 11 211 | Ref.  Pred. e p  e 1242 **0**p 14 1139 | Ref.  Pred. e p  e 1242 **0**  p 14 1139 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **10 workers** | | **50 workers** | | 100 workers | | 500 workers | |
| **Accuracy** | Mixed population | **Repeated data** | **Mixed population** | **Repeated data** | Mixed population | Repeated data | Mixed population | Repeated data |
| Log istic Regresion | 0.8797 | - | 0.9925 | - | 0.985 | - | 0.9942 | 0.9942 |
| Naive Bayes | 0.8752 | 0.8751 | 0.9766 | 0.9766 | 0.9795 | 0.9795 | 0.9783 | 0.9783 |
| Neural Nets | 0.8797 | 0.8797 | 0.9925 | 0.9924 | 0.985 | 0.9849 | 0.9942 | 0.9942 |
| Random Forest | 0.8797 | 0.8797 | 0.9925 | 0.9924 | 0.985 | 0.9849 | 0.9942 | 0.9942 |
| Decision Tree | 0.8797 | 0.8797 | 0.9925 | 0.9924 | 0.985 | 0.9849 | 0.9942 | 0.9942 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **10 workers** | | **50 workers** | | 100 workers | | 500 workers | |
| **Sensitivity** | Mixed population | **Repeated data** | **Mixed population** | **Repeated data** | Mixed population | Repeated data | Mixed population | Repeated data |
| Log istic Regresion | 0.8774 | 0.8773 | 0.9857 | 0.9856 | 0.9793 | - | 0.9889 | 0.9889 |
| Naive Bayes | 0.8769 | 0.8769 | 0.9661 | 0.9660 | 0.9791 | 0.9790 | 0.9692 | 0.9692 |
| Neural Nets | 0.8774 | 0.8773 | 0.9857 | 0.9856 | 0.9793 | 0.9792 | 0.9889 | 0.9889 |
| Random Forest | 0.8774 | 0.8773 | 0.9857 | 0.9856 | 0.9793 | 0.9792 | 0.9889 | 0.9889 |
| Decision Tree | 0.8774 | 0.8773 | 0.9857 | 0.9856 | 0.9793 | 0.9792 | 0.9889 | 0.9889 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **10 workers** | | **50 workers** | | 100 workers | | 500 workers | |
| **Specifity** | Mixed population | **Repeated data** | **Mixed population** | **Repeated data** | Mixed population | Repeated data | Mixed population | Repeated data |
| Log istic Regresion | 0.8824 | - | 1.0000 | **-** | 0.9912 | - | 1.0000 | 1.0000 |
| Naive Bayes | 0.8733 | 0.8732 | 0.9885 | 0.9884 | 0.9800 | 0.9800 | 0.9885 | 0.9885 |
| Neural Nets | 0.8824 | 0.8823 | 1.0000 | 1.0000 | 0.9912 | 0.9912 | 1.0000 | 1.0000 |
| Random Forest | 0.8824 | 0.8823 | 1.0000 | 1.0000 | 0.9912 | 0.9912 | 1.0000 | 1.0000 |
| Decision Tree | 0.8824 | 0.8823 | 1.0000 | 1.0000 | 0.9912 | 0.9912 | 1.0000 | 1.0000 |

**Conclusions:**

1. Last 3 classification algorithms are the best.
2. Results for repeated data are better for all classification algorithms.

**What else has to be done:**

1. To repeat the same procedure using **two other data sets**. We can choose something similar to mushrooms from KAGGLE – no later than the middle of this week (9-15.04; **11-12.04**). .
2. Try to find **other working consensus algorithm** and apply it to the data. Weighted majority voting should be easiest to realize, but I am afraid we do not have enough time. (deadline also middle of this week, **11-12.04**).
3. Describe the results and prepare **slides**. Easiest way: to use the slides from previous presentation and modify them. I suppose we can start to do it at the end of this week, **14-15.04.18**.

Add Read.me file in GitHub?