1 Scenario

Let us assume that we have a **small** labeled dataset from one domain (we are going to call this dataset the $TargetDataset$) and [usually] it is costly to obtain new labelled data from the same domain (so the quantity is limited).

Also, let us also assume that we have a **large** labeled dataset from a ***related*** domain (we are going to call this dataset the $SourceDataset$) and [usually] it is affordable to obtain new labelled data from this domain.

If we would like to build a model for the $TargetDataset$ using it a lone, then the model will likely perform undesirably as the dataset is quite small!

The idea is to make maximum use of the $SourceDataset$ to build a model for the $TargetDataset$ and classify data from its domain.

In this experiment, I am using the technique explained in a paper called *"Boosting for Transfer Learning"* to gain knowledge from the *Source* data and use it in training a classifier for the $Target$ data. The proposed algorithm is called $TrAdaBoost$ and it is an extension of the classical *AdaBoost* Algorithm!

The technique works by voting on the usefulness of each instance of the *Source* data (i.e. giving each instance a weight according to how close/useful it is to the $Targert$ data)

I believe this learning process corresponds to transferring knowledge learned from the *source* data to a new situation (the *target* data) – and this is Instance-based Transfer Learning.

2 Experiment I

2.1 The Data

- By looking at the table provided with the Eve data, I have extracted and merged labeled data from the Sapphire Channel (Sapphire Active/Inactive) for assays TS3 and TS6. This is for strain *Plasmodium vivax*. This is going to be our $SourceDataset$
- I have also extracted labeled data from the Venus Channel (Venus Active/Inactive) for assay TS6. This is for strain *Plasmodium falciparum*. This is going to be our $TargetDataset$
- I have split the $TargetDataset$ into two subdatasets. One for training and one for testing. Notice that this training subdataset is going to be our actual $TargetDataset$
- Now our datasets look like:
  - $SourceDataset$ has 2781 instances (for Pv from TS3 and TS6 – file TS3-TS6-Pv.arff)
  - $TargetDataset$ has 46 (for Pf from TS6 – file TS6-Pf.arff)
  - $TestDataset$ has 1388 (for Pf from TS6 – file TS6-Test-Pf.arff)

2.2 Experimental Setup

The authors of the paper have provided the java source code of their implementation so I have downloaded it, plugged it into WEKA's source code and recompiled WEKA.
Remember that our $TargetDataset$ is quite small and our $SourceDataset$ is large and from a related domain!
We will use our $TestDataset$ for evaluation (remember it is from the same domain as $TargetDataset$)

Here is what I have done:

- I have built a classification model with the TrAdaBoost Algorithm using both the $Target$ and $Source$ Datasets to carry out Transfer Learning.
- I have built classification models with WEKA's NaiveBayes, SVM, KNN and J48 Decision Trees using the $TargetDataset$ only. This is because usually we build models using data from the same domain!

2.3 Experimental Results

After building the models as explained above, I have evaluated them using the $TestDataset$ which is of the same domain as the $TargetDataset$. I have counted Actual vs Predicted results! The following table shows how many miss-classifications each model makes:

TrAdaBoost	NaiveBayes	SVM	KNN	J48 Decision Trees
6	115	10	9	25

Observe that the TrAdaBoost model (the one that does Transfer Learning) makes less classification errors meaning it outperforms models built using the $TargetDataset$ alone!

3 Experiment II

In this experiment, I am going to try and do TL at assay level. Meaning I will use Active/Inactive labelled datasets from the eve data (assays TS3,4,5,6,7)

3.1 The Data

- For the $SourceDataset$, I have used TS4 (1284 instances – file TS4-Labeled.arff)
- For the $TargetDataset$, I have used TS3, I have randomly removed many instances to reduce its size (434 instances – file TS3-Labeled.arff)
- For the $TestDataset$, I have used TS5 (1394 instances – file TS5-Labeled.arff)

3.2 Experimental Setup

Exactly the same as Experiment I

3.3 Experimental Results

After building the models as explained above, I have evaluated them using the $TestDataset$ which is of the same domain as the $TargetDataset$. I have counted Actual vs Predicted results! The following table shows how many miss-classifications each model makes:

TrAdaBoost	NaiveBayes	SVM	KNN	J48 Decision Trees
5	50	7	6	6

Observe that the TrAdaBoost model (the one that does Transfer Learning) makes less classification errors meaning it outperforms models built using the $TargetDataset$ alone!