Main technical details of the recent changes in the Web Classification module

Initial tasks

1. Accomplish and validate new implementation of the classification scheme: get rid of dependencies on RapidMiner – based modules and Java wrappers, move to Sklearn – based implementation and Python wrappers;
2. Check a number of recently suggested improvements for the current implementation scheme that can positively affect accuracy of the approach (during the experiments it is necessary to update the set of important categories: some categories do not seem to be important nowadays, update training and testing sets with new instances, also remove found duplicates and mislabeled objects):
    a. Update input feature space: the idea to use numeric information about terms (words) appearance looks promising and correct, especially for big texts classification;
    b. Check if it is possible to get rid of initial threshold-based logic (confidence – level based filtering of questionable decisions);
    c. Check alternatives for 2nd and 3rd level classifiers, decision trees cause high variance.

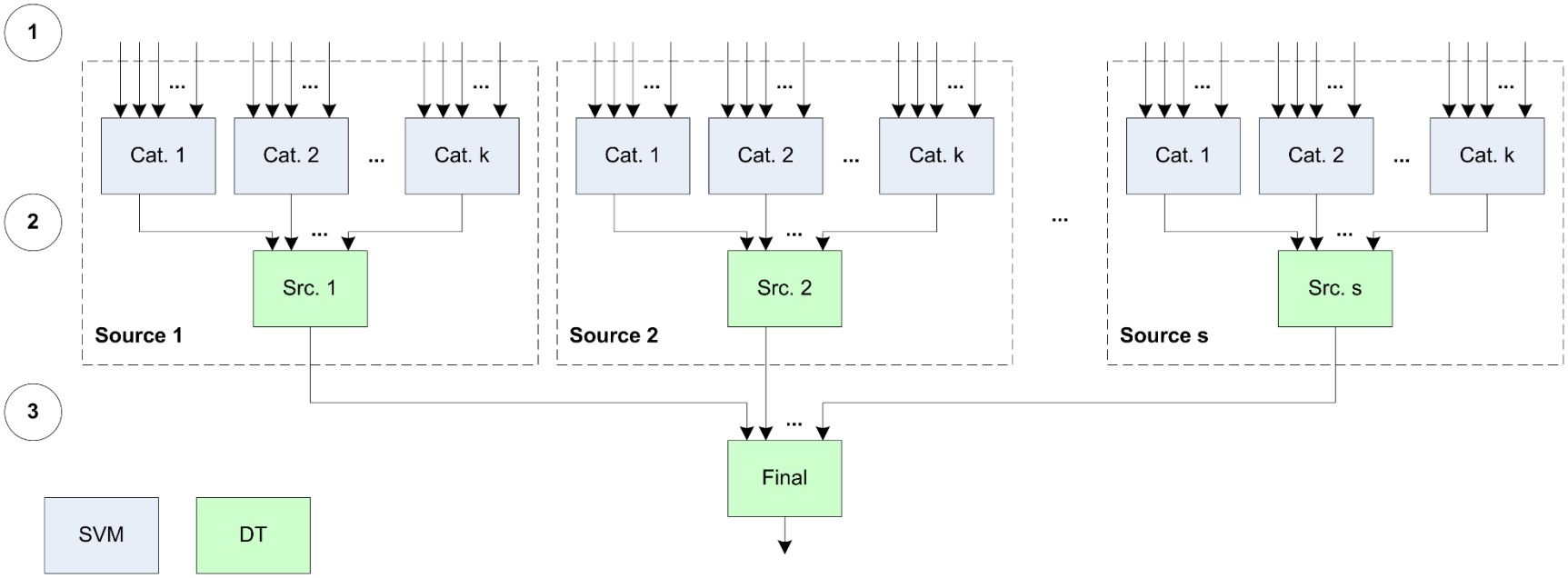

Figure 1. Initial configuration of the classification scheme

Description of the initial scheme (figure 1)

It is a 3 – Layer decision model that includes:

1. SVM: A single SVM classifier for each source-category pair:
    a. Input: Vector of source-category specific data features;
    b. Features: Binary features corresponding to the presence of specific terms in web page components;
    c. Output: A single Boolean value which specified the assignment of the source the category (threshold – based logic is enabled).
2. Decision tree: A single DT (C4.5) for each source:
    a. Input: A binary vector containing all the decisions of the first-layer classifiers for a single source;
    b. Output: Category of the specified source.
3. Decision tree: A single DT (C4.5):
    a. Input: A vector containing the categories of all the sources in a web page;
    b. Output: Category of the specified web page.

Task 1: Accomplish and validate new implementation of the classification scheme

1. No major difference in the obtained results in terms of precision and recall;
2. The results indicated 80 % overlap.

Task 2.a: New input feature space

1. Initial feature space:
    a. Boolean features:
        i. Corresponds the presence of a specific term in the web page;
        ii. Drawback: Does not incorporate the significance of a term in a single classifier.
2. New feature space:
    a. Float features:
        i. Corresponds the number of occurrences of a specific term in the web page;
        ii. Standardized into floats with zero mean and unit standard deviation.
    b. Results indicated minor improvement to precision.

Task 2.b: Update decision scheme (get rid of thresholds)

1. Initial decision scheme:
    a. Threshold – based logic applied in each of the 3 layers;
        i. Part of the decision was sacrificed because of small likelihood.

2. New decision scheme:
    a. Send probabilities to the layer 2 instead of Boolean;
    b. Obtained results were not promising:
        i. 2nd layer DT does not fit into the new float inputs.

Task 2.c: Update decision scheme (alternative classifiers), figure 2:

1. 2nd layer DT was replaced by NN (Extreme Learning Machines):
    a. Obtained results were not promising:
        i. Errors propagated into lower layers;
        ii. 3rd layer DT generated random predictions.
2. Made decision: remove 3rd layer completely:
    a. Use a single ELM for layer 2:
        i. Good and stable results for 'Adult', 'Occults' and 'Marijuana' categories, visible improvements in terms of precision for other target categories.

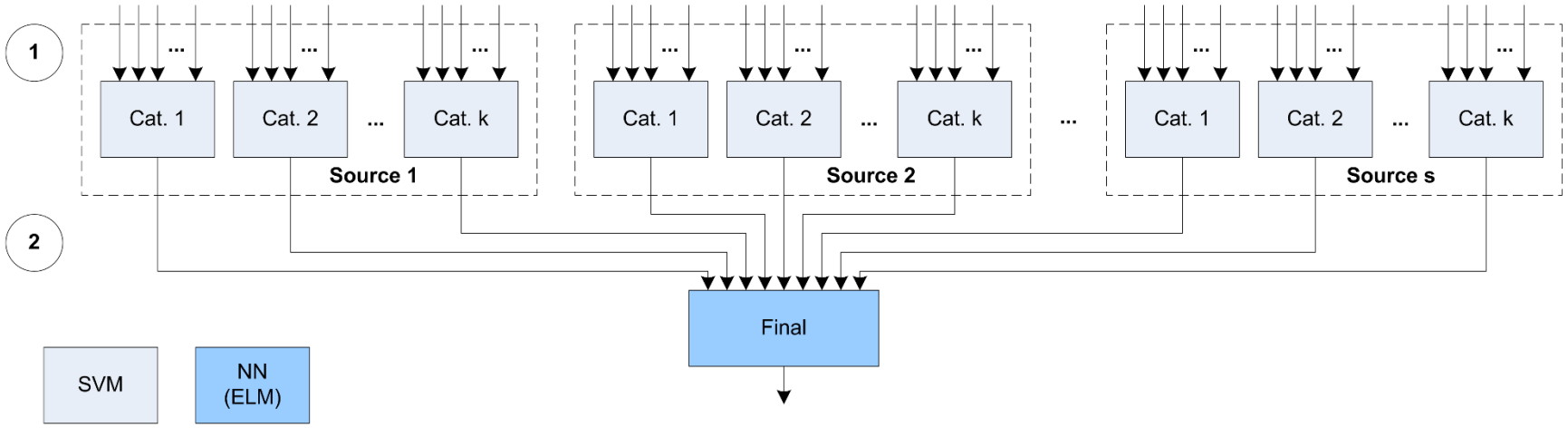

Figure 2. Updated configuration of the classification scheme

Conclusions:

1. It is necessary to continue amendments of feature space: the labels for SVM classifiers are related to web pages rather than sources:
    a. Samples with identical features are mapped into different labels in the training phase.

2. Using a single classifier for the whole feature set would make sense, this assumption requires new cycle of experiments;
3. ELM classifier is taken into use instead of Decision Trees (for low level decisions). This can cause extra difficulties if feature space grows further:
    a. Heavy memory consumption;
    b. Issues with matrix singularity that are caused by the sparse training set. As an idea, it makes sense to check feature compression approaches like sparse autoencoders (aka pre-learning phase).