CS583 – Project Type I

In this project, you will implement the iterative classification algorithm and compare it to content-only classification on various networked datasets. For iterative classification algorithm, please see the class notes and following papers:

- https://kdl.cs.umass.edu/papers/neville-jensen-srl2000.pdf
- http://www.cs.umd.edu/~getoor/Publications/icml03.pdf
- http://www.cs.iit.edu/~ml/pdfs/sen-aimag08.pdf

You are required to code in python 2.7. The allowed packages are:

- scikit-learn, version 0.14
- numpy, version ??
- scipy, version ??

If you'd like to use additional packages, you need to get permission from me.

You are required to implement:

1. **Load a networked dataset:** Your internal representation (matrices, lists, etc.) is up to you. You should be able to represent directed networks. Each node will have attributes of their own.
2. **5-fold cross-validation:** To evaluate your model, you need to perform cross-validation. You can use sckit-learn's existing functionality to split the data into train and test splits if you like.
3. **Content-only classification:** you should be able to perform content-only classification, where you train a model and test it using only the node attributes.
4. **ICA:** Implement ICA as discussed in class. Use two classifiers: one that uses only content information, and one that uses both content and relational information. For the relational attributes, you are required to experiment with the following aggregates (each one separately).
    a. Count
    b. Proportion
    c. Exist

d. Mode
e. Directed and undirected versions of these. In the undirected version, you use all of your neighbors for creating the feature values. For the directed version, you create two sets of relational features: one uses only the incoming links, and one uses only the outgoing links.

For your ICA implementation, make sure that

- When you are creating relational features during training, you can use only the labels of the training instances; you cannot look at the labels of the test instances.
- When you are creating the relational features during testing, you use the observed labels of the training instances, and predicted labels of the test instances.

For underlying classifiers, experiment with

- Logistic regression with L2 norm
- Logistic regression with L1 norm
- Support vector machines with linear kernels
- Support vector machines with RBF kernels
- Multinomial Naïve Bayes
- C4.5

All of these classifiers are already implemented in scikit-learn. For evaluation, use accuracy.

Here are the datasets that you will be experimenting with. The first four datasets are available at http://linqs.cs.umd.edu/projects/projects/lbc/index.html. I'll be releasing the fifth dataset soon.

1. Cora
2. CiteSeer
3. PubMed Diabetes
4. Wikipedia
5. Twitter data

Like Type II and Type III projects, Type I also requires a progress report and a final report.