AlvsPK Challenge: FACT SHEET

Title:
High-Throughput Screening with Two-Dimensional Kernels

Name, address, email:
Chloe-Agathe Azencott and Pierre Baldi

Institute for Genomics and Bioinformatics
236 Info & Computer Science Bldg. 2
University of California, Irvine
Irvine, California 92697-3445

cazencot@ics.uci.edu

Acronym of your best entry: SVM

Reference:
None

Method:

- Preprocessing
  The molecules of the HIVA dataset are represented as two-dimensional graphs where nodes are atoms and edges are bounds. The nodes are labelled by different schemes, including atomic number or element-connectivity label (atomic number together with number of connected atoms); the edges are labelled by bond type.
  Molecular graphs are then translated into fingerprints, vectors for which each component accounts for a given two-dimensional feature. More specifically, circular fingerprints [1] are used.
- Feature selection
  No feature selection is applied.
- Classification
  - Specific kernels, namely Tanimoto and MinMax, are used on top of the fingerprints to derive support vector machines. Generally speaking, these kernels allow for comparing fingerprints by comparing the number of features they share.
  - SVMTorch implementation [2] is used
  - Over-sampling of the active class is used to compensate for the unbalanceness of the dataset
- Model selection/hyperparameter selection
  - Optimal labeling schemes, depth of fingerprints and kernels are chosen by 10-fold cross-validation on the training set
  - Optimal SVM parameters C and epsilon are chosen among grid values by 10-fold cross-validation on the training set

Results:

Table 1: Our methods best results

Dataset	Entry name	Entry ID	Test BER	Test AUC	Score	Track
ADA	final svm # 1	936	0.2751	0.8084	0.9448	Agnos
GINA	final svm # 1	936	0.1984	0.8915	0.9872	Agnos
HIVA	SVM	992	0.2693	0.7643	0.008	Prior
NOVA	final svm # 1	936	0.2005	0.9574	0.9423	Agnos
SYLVA	final svm # 1	936	0.0434	0.9912	0.9246	Agnos
Overall	SVM	992	0.1973	0.8826	0.7614	Prior

Table 2: Winning entries of the AlvsPK challenge

Best results agnostic learning track						
Dataset	**Entrant name**	**Entry name**	**Entry ID**	**Test BER**	**Test AUC**	**Score**
ADA	Roman Lutz	LogitBoost with trees	13, 18	0.166	0.9168	0.002
GINA	Roman Lutz	LogitBoost/Doubleboost	892, 893	0.0339	0.9668	0.2308
HIVA	Vojtech Franc	RBF SVM	734, 933, 934	0.2827	0.7707	0.0763
NOVA	Mehreen Saeed	Submit E final	1038	0.0456	0.9552	0.0385
SYLVA	Roman Lutz	LogitBoost with trees	892	0.0062	0.9938	0.0302
Overall	Roman Lutz	LogitBoost with trees	892	0.1117	0.8892	0.1431
Best results prior knowledge track						
Dataset	**Entrant name**	**Entry name**	**Entry ID**	**Test BER**	**Test AUC**	**Score**
ADA	Marc Boulle	Data Grid	920, 921, 1047	0.1756	0.8464	0.0245
GINA	Vladimir Nikulin	vn2	1023	0.0226	0.9777	0.0385
HIVA	Chloe Azencott	SVM	992	0.2693	0.7643	0.008
NOVA	Jorge Sueiras	Boost mix	915	0.0659	0.9712	0.3974
SYLVA	Roman Lutz	Doubleboost	893	0.0043	0.9957	0.005
Overall	Vladimir Nikulin	vn3	1024	0.1095	0.8949	0.095967

- quantitative advantages
  - computation of features is relatively fast
  - no feature selection is needed

- qualitative advantages
  - cross-validation results are in favor of low over-fitting
  - generic enough to be applied to other problems of the chemistry domain
  - the method can easily be adapted to other problems where the data can be represented as graphs

Code:

- Computation of the molecular graph (Python Module)
  With the help of the OpenBabel library [3], we create a graph where the nodes are the atoms and the edges the bonds between them
- Computation of the fingerprints (Python Module)

    - Assign an initial label to each heavy (i.e. non hydrogen) atom. The label can be, for instance, the atomic number of the atom (atomic number scheme), or its atomic number paired with its number of connections to other heavy atoms (element-connectivity scheme)
    - For each iteration (typically: 2 to 4), update each atom label in the following way: the new label is a hash of the old label with the labels of all the connected atoms
    - Eventually, each atom label (across the whole dataset) is a possible feature and feature vectors (that are very sparse) are created
- SVM implementation
    - The SVMTorch module [2] is used (the MinMax kernel, that can be used on binary vectors as the Tanimoto kernel, is added to the kernels)
    - A Python wrapper is used to perform grid-search hyper-parameters optimization

Keywords:
- Preprocessing or feature construction: molecular graph, circular fingerprints
- Feature selection approach: None
- Feature selection engine: None
- Feature selection search: None
- Feature selection criterion: None
- Classifier: SVM, active class over-sampling
- Hyper-parameter selection: grid-search,10-fold cross-validation.
- Other: 2D kernels for small molecules

[1] Dubois, J. E. **Chemical Applications of Graph Theory**; Academic Press, London: **1976** and Hassan, M., Brown, R. D., Varma-O'Brien, S., Rogers, D. **Cheminformatics analysis and learning in a data pipelining environment**. *Molecular Diversity* **2006**, *10*, pp 283-299.

[2] http://www.idiap.ch/machine-learning.php

[3] http://openbabel.sourceforge.net