IMPROVED SMALL MOLECULE ACTIVITY DETERMINATION VIA CENTROID NEAREST NEIGHBORS CLASSIFICATION

Phuong Dao*, Farhad Hormozdiari*, Hossein Jowhari and S.Cenk Sahinalp †
School of Computing Science, SFU, Burnaby, BC, Canada
Email: {pdao,fha1,hjowhari,cenk}@cs.sfu.ca

Kendall Byler, Artem Cherkasov
UBC Division of Infectious, Diseases, Vancouver, BC, Canada
Email: {artc, kbyler}@interchange.ubc.ca

Zehra Cataltepe
Istanbul Technical University, Computer Engineering Department, Maslak, Istanbul, Turkey
Email: zehra@cataltepe.com

Small molecules which alter biological processes or disease states are of significant interest. In-silico drug discovery commonly uses measures of structural similarity for identifying the "right" small molecule for a given task. Because explicit structure similarity determination is a very difficult task, modern chemoinformatics solutions typically use "quantitative structure-activity relationships" (QSAR), in the context of which small molecules are described with real valued descriptor arrays. In this paper we show how to identify the bioactivity exhibited by compounds of interest through Centroid based Nearest Neighbor (CBNN) classifiers, in which, on a given training set, the "best" representative compounds of each specific bioactivity need to be selected. For that purpose we introduce the Combinatorial Centroid Nearest Neighbor (CCNN) method which determines the representative compounds in a way that would yield no classification errors on the training set. On a number of data sets CCNN method was applied, we observed that CCNN provides the highest accuracy over the data sets of three different bioactivities among all classifiers we tested.

1. INTRODUCTION

Small molecules are chemical compounds with molecular weight less than 500; they are commonly used as drugs - in fact most drugs are small molecules - and also used for purposes of better understanding molecular and cellular functions.

Although small molecules which alter biological processes or disease states are of significant interest, finding the right molecule for a given task is very challenging. For purposes of in-silico drug discovery, the most common computational tools used are based on structural similarity determination. Chemical compounds are usually structurally similar if and only if their physiochemical properties and/or biological activities are similar [18]. As a result, it is many times possible to predict the specific bioactivity exhibited by compounds via structural similarity search among biomolecules whose bioactivities are known.

In the field of cheminformatics, "quantitative structure-activity relationships" (QSAR) provide various means for enumerating chemical similarity. In fact modern structure-activity cheminformatics relies on a variety of parameters called descriptors reflecting different aspects of possible intra- and intermolecular interactions a molecule may be engaged in. Interestingly, while the number of available QSAR descriptors is very large and is constantly expanding, the set of available algorithms, used to relate the descriptors to bioactivity is very limited. In general, finding novel analytical approaches for the field of QSAR (including novel ways to enumerate chemical similarity) is an important and challenging task.

Descriptor-based structural similarity. Given two descriptor arrays X and Y representing two small molecules, each with n real dimensions, it is com-

*Joint first authors
†Corresponding author

mon to use Minkowski distances to measure structure similarity. Minkowski distance of order p, denoted by L_p is defined as $L_p(X,Y) = (\sum_{i=1}^{n} |X[i] - Y[i]|^p)^{1/p}$. Clearly, for $p = 2$, the Minkowski distance is the Euclidean distance. For $p = 1$, the Minkowski distance becomes the Hamming distance, when X and Y are binary descriptor arrays [36].

Small molecule classification methods. A variety of statistical learning methods have been successfully applied to structural classification.

In many of these methods, chemical structures of molecules are modeled as labeled graphs. Micheli *et al.* [15], for example, first convert chemical structures into trees and then use a standard recursive neural network for relating the structure and activity of molecules.

There are also a number of kernel methods which are designed for small molecule classification [32, 38, 21, 20, 25, 3, 28, 27]. Kashima *et al.* [21] introduce Marginalized Kernel (MK) which is based on counting the number of labeled paths in random walks, and consider those as feature vectors. The similarity measure they use between two graphs is proportional to the number of labeled random walks they share. Mahe *et al.* [27] improve the computational cost of MK feature extraction. In addition, they modify random walks in MK to avoid 'totters' (walks that visit a node which was visited in two previous stages), using a second-order Markov chain. Swamidass *et al.* [32] introduce a kernel that takes 3D Euclidean coordinates of atoms in small molecules into account; for each pair of atoms they consider the "histogram" distance. Similarity between two molecules can then be computed based on the similarities in the histograms. Decomposition of kernels have been used by Ceroni *et al.* [3] to combine 2D and 3D kernels in order to reach better accuracy in classification.

Recently, Cao *et al.*[38] consider to measure the similarity between two molecules via the maximum common substructure (MCS) of two graphs representing these molecules. Although computing MCS in general is intractable, a backtracking algorithm and various heuristics are applied to compute the MCS between small molecules. Unfortunately, computing MCS between a compound to all other compounds is still time consuming: if there are n compounds in the data set we need to perform $O(n^2)$ MCS comparisons. This method thus computes a subset of randomly picked "basis" compounds. For each compound, only MCSes between the compound and all basis compounds are computed and used as features for the purposes of classification.

One of the most common tools for identifying bioactivity levels of small molecules is k-Nearest Neighbor (kNN) classification method. Given a small molecule with an unknown bioactivity, the Nearest Neighbor classifier estimates its bioactivity as that of its "nearest neighbor", or the structurally most similar compound with known activity. Similarly, the k-Nearest Neighbor(kNN) classifier would estimate the bioactivity of a compound as the majority activity of the k compounds which are structurally most similar to the query compound [11].

It is well known that the (asymptotic) expected error returned by the NN classifier is bounded above by twice the "Bayes error" (the minimum error achieved by having complete knowledge of underlying probability density functions of the classes) [11]. This and other desirable properties of the NN classifier have turned it into a powerful method with many applications in bioinformatics. Unfortunately, NN also has a number of well known drawbacks especially in high dimensional vector spaces - typically referred to as the curse of dimensionality:

1) Speed/memory: among a set of points in a high-dimensional vector space, finding the nearest neighbor of a query point is costly in terms of either time and/or space.

2) Accuracy: in high-dimensional vector spaces, NN method has a tendency to do overfitting.

The above problems become particularly acute if the number of points in the vector space of interest is "high". As a result, the bulk of the research for improving NN classification focuses on sparsifying (obtaining a small subset of) the input set of points (named centroids) on which NN classification is then performed much faster. Overfitting in NN is due to the "non-smooth" decision boundary in a high-dimensional vector space [29] and sparsification of the input data set can make the decision boundary "smoother".

Centroid based NN method. Centroid based NN (CBNN) classification is a modified version of NN in which the bioactivity of an unknown compound is determined according to its nearest centroid. Although it is desirable to have as few centroids as possible, for improving the speed, the "optimal" trade-off between speed and accuracy may be obtained when a non-trivial fraction of the points in the training set are maintained as centroids.

The CBNN problem has been considered by both the machine learning and computational geometry communities. On the machine learning side, Hart, for example [26] introduced CNN (Condensed Nearest-Neighbor) method, which suggests to iteratively build the "centroid set" S as follows. Initially S includes exactly one randomly picked point from the training set. At each iteration, the points whose nearest neighbor in S has the opposite class is added to S - until no such point remains in the training set. It is shown experimentally [37] that the CNN method typically has a worse accuracy in comparison to the standard NN classifier employing the complete training set. Furthermore, the initial set S has a significant role in determining how CNN method performs. One recent method in this direction, FCNN (Fast Condensed Nearest-Neighbor) [16], initially sets S to the collection of medoids of each class[a]. In each iteration, for every $p \in S$, FCNN adds to S one point q in p's Voronoi cell which belongs to a different class. The method terminates when no such q remains in the training set.

On the computational geometry side, arguably two most popular approaches are the "Relative Neighborhood" and "Gabriel" graphs. The relative neighborhood graph(RNG) [34] is a graph constructed from the points in the training set. Each point is represented by a node, and an edge exists between two nodes i and j if for any other node k: $distance(i, j) < max\{distance(i, k), distance(j, k)\}$. After constructing the graph, each node with an edge to another node from a different class is chosen as a centroid. The Gabriel graph(GG) [9, 23], on the other hand, uses the same general idea with RNG with the exception that two nodes have an edge between them, if there is no other point in their diametrical sphere.

Interestingly, Support Vector Machine (SVM) another popular classification method commonly used in QSAR analysis, when used with a linear kernel, is equivalent to a centroid based NN classifier with two centroids: here the two centroids are on the opposite sides of a normal line to the separating plane, and are equidistant to the plane.

Our contributions. The main contribution of this paper is on the Combinatorial Centroid Selection (CCS) problem which we define as follows. Given a training set of points (compounds) in a metric space, CCS problem asks to find the minimum number of centroids (selected compounds) such that for every point (compound) p in the training set its nearest centroid (one of the selected compounds) is in the same class. CCS can be applied after initial screening or High-Throughput screening phase to obtain a small subset of compounds. NN based or other QSAR models can then be built on this set of compounds to filter the library of huge number of compounds for experimental screening.

This paper focuses on classification in general Minkowski distances and binary classification problems - although our methods can be generalized to cases with more than two classes. Note that for 2D Euclidean space, Wilfong [19] showed that the Minimum Consistent Subset problem with three "labels" (a problem very similar to the CCS problem with three classes) is NP-hard. Here we prove that the CCS problem for general metric spaces is NP-hard even when there are two classes to consider - our proof also indicates that no algorithm is likely to approximate the CCS problem with an approximation factor better than $\Omega(\log n)$ where n is the number of points. We then describe an integer programming formulation to the CCS problem for (weighted) Minkowski distances and provide a number of LP relaxations to solve it in practice.

In many in-silico drug discovery applications, there are typically very few compounds that have the particular bioactivity of interest; the majority of the data set does not exhibit the bioactivity. Thus a simpler variant of the CCS problem in which one

[a] a medoid in a set T is a point $p \in T$ whose distance to all other points is smallest possible

may want to maintain all members of a particular class as centroids and aim to minimize the number of centroids in the opposite class is of significant interest. Although for the general CCS problem no approximation algorithm is available, for this simpler version we provide an $O(\log n)$-approximation algorithm in this paper.

We have tested our methods on publicly available data sets of three different bioactivities for small molecules: mutagenicity, toxicity and (being a) drug. Details on these data sets are provided in Section 3. A set 30 inductive 3D QSAR descriptors and 32 conventional QSAR descriptors are employed in all three data sets. These inductive descriptors have been used in several studies [4, 6, 7, 5, 14] and have been calculated by a SVL script that can be freely downloaded through the SVL exchange [40]. The conventional QSAR descriptors are implemented within the MOE (Molecular Operating Environment) [22]. More details about these descriptors are presented in Section 3.

There are three main methods we have compared against NN classification: SVM, MK by Kashima *et al.* [21] and MCS by Cao *et al.* [38]. MK has been commonly used as a benchmark to compare with other graph kernel methods [27, 32] while MCS is one of the most recent methods for classification of small molecules. Experimental results show that our CCS based classification methods outperform all the above classifiers (including the standard NN classifier) in all data sets - maintaining only small fraction of chemical compounds in the training set.

2. Methods

For simplicity, we focus on the binary classification problem; our methods can easily be generalized for multi-class classification. Consider a training set of points T in a metric space V with distance d. Suppose each point $p \in V$ belongs to one of the two classes C_1 and C_2. Let C_{1_i} and C_{2_j} each denote a point in class C_1 and class C_2 respectively. Given an integer k, the k-Combinatorial Centroid Selection (k-CCS) problem asks whether there is a set of k points in T, namely $S \subset T$ such that for each $p \in T$, its nearest neighbor in S is in the same class with p. Below, we show that the k-CCS problem is NP-Complete in a general metric space even when there are two classes through a slight generalization of the proof by Wilfong [19] for three classes.

Lemma 2.1. *The k-CCS problem is NP-Complete in a general metric space with two classes.*

Proof. The k-CCS problem is obviously in NP. Let S_1 and S_2 respectively denote the set of centroids from the classes C_1 and C_2; it is trivial to check whether $|S_1 \cup S_2| \leq k$. It is also trivial to check for each point $p \in T$ whether its nearest centroid is in the same class with p. We now show that k-Dominating Set $\leq_P$ k-CCS. Given an undirected graph $G = (V, E)$ and an integer k, the k-Dominating Set problem asks whether there is a subset $V' \in V$ such that $|V'| \leq k$ and for each vertex v, either $v \in V'$ or there is an edge $(u, v) \in E$ and $u \in V'$. From an instance of k-Dominating Set problem, we construct an equivalent k-CCS problem with two sets of points $C_1 = \{C_{1_1}, C_{1_2}, \ldots, C_{1_{|V|}}\}$, $C_2 = \{C_{2_1}\}$ and a distance function $d(.,.)$ as follows. Let c be an arbitrary constant:

- $d(C_{2_1}, C_{2_1}) = d(C_{1_i}, C_{1_i}) = 0$
- $d(C_{1_i}, C_{1_j}) = c$ if $(v_i, v_j) \in E$
- $d(C_{1_i}, C_{1_j}) = 2c$ if $(v_i, v_j) \notin E$
- $d(C_{2_1}, C_{1_i}) = \frac{3}{2}c$

It is not hard to see that d is a metric. If G has a dominating set V', $S_1 = \{C_{1_i} | v_i \in V'\}$ and $S_2 = \{C_{2_1}\}$ is a solution to k-CCS problem of size $|V'| + 1$. Otherwise, there is some $C_{1_l} \in C_1$ such that for every $C_{1_i} \in S_1$, $d(C_{1_l}, C_{2_1}) < d(C_{1_l}, C_{1_i})$ i.e. the closest centroid of C_{1_l} is C_{2_1}. This is not the case since there must be some $v_j \in V'$ or $C_{1_j} \in S_1$ such that $(v_j, v_l) \in E$, thus, $d(C_{1_l}, C_{2_1}) = 2c > d(C_{1_l}, C_{1_j}) = c$. If we pick k points as centroids, there are $k-1$ centroids from C_1 due to the fact that we always need to pick the one centroid C_{2_1} from C_2. Let $V' = \{v_i | C_{1_i} \in S_1\}$ of size $k-1$, for every vertex $v_t \notin V'$ or $C_{1_t} \notin S_1$ there is a vertex $v_l \in V'$ or $C_{1_l} \in S_1$ such that $d(C_{1_t}, C_{1_l}) = c < d(C_{1_t}, C_{2_1}) = \frac{3}{2}c$ or $(v_t, v_l) \in E$. Thus, V' is a dominating set of G of size $k-1$. In conclusion, G has a dominating set of size $k-1$ if and only if we can pick k centroids for the equivalent k-CCS problem. □

$$
\begin{aligned}
\text{minimize} \quad & \sum_{C_{1_i} \in C_1} \delta(C_{1_i}) + \sum_{C_{2_j} \in C_2} \delta(C_{2_j}) \\
\text{s.t.} \quad & \\
& \sum_{C_{1_k} \in R(C_{1_i}, d(C_{1_i}, C_{2_j}))} \delta(C_{1_k}) \geq \delta(C_{2_j}) \qquad C_{1_i} \in C_1, C_{2_j} \in C_2 \\
& \sum_{C_{2_l} \in R(C_{2_p}, d(C_{2_p}, C_{1_q}))} \delta(C_{2_l}) \geq \delta(C_{1_q}) \qquad C_{2_p} \in C_2,\, C_{1_q} \in C_1 \\
& \sum_{C_{1_i} \in C_1} \delta(C_{1_i}) \geq 1 \\
& \sum_{C_{2_j} \in C_2} \delta(C_{2_j}) \geq 1 \\
& \delta(X) \in \{0, 1\} \qquad X \in C_1 \cup C_2
\end{aligned}
\tag{1}
$$

The minimum number of centroids is one more than the size of minimum dominating set. The Minimum Dominating Set problem is equivalent to the Minimum Set Cover under an L-reduction [31]. There is no approximation algorithm for the Minimum Set Cover problem with an approximation factor better than $(1-\epsilon)\ln|V|$ unless NP $\in$ DTIME$[|V|^{\log\log|V|}]$ [35]. Thus, it is unlikely that one can obtain an approximation algorithm for the CCS problem with an approximation factor better than $(1-\epsilon)\ln(|C_1|+|C_2|)$ in a general metric space.

An Integer Linear Program Based Solution. In this section, we present an integer linear program (IP) formulation Combinatorial Centroid Selection (CCS) problem for two classes. The generalization of this formulation for three classes or more is quite easy, but is not presented here for brevity.

The objective of the IP formulation is to minimize the total number of centroids in the two classes. We will denote by $\delta(X)$ the indicator variable for the point X i.e. $\delta(X) = 1$ if X is chosen as a centroid, otherwise $\delta(X) = 0$. For each point C_{k_i} of each class k, we will denote by $R(C_{k_i}, r)$ the set of all points in class k which are within distance r from C_{k_i}; more formally $R(C_{k_i}, r) = \{C_{k_j} | C_{k_j} \in C_k \textit{ and } d(C_{k_i}, C_{k_j}) \leq r\}$. In the IP formulation below, the first and second set of constraints ensure that (one of) the nearest centroid(s) of a point is a point from the same class - this constraint ensures that each "ball" centered at a point from one of the classes which includes a centroid from an opposite class should also include a centroid from the class of the center point itself. The third and fourth set of constraints ensure that for each class, at least one centroid is picked. The final constraint simply defines the indicator variable as one with values 0 or 1.

It is not difficult to see that a solution to the IP formulation (1) above corresponds to a solution to the centroid problem and vice versa. Hence, the minimum number of centroids picked by the above formulation is equal to the optimal solution of (1). Unfortunately solving the IP problem (1) is quite costly due to the fact that the number of binary variables are linear with the number of points and the number of constraints are quadratic with the number of points. Although for smaller data sets the above formulation works quite well in practice, for large data sets of interest the standard way to address the problem is to solve it as a standard Linear Program (LP).

LP formulation for the CCS problem differs from the IP formulation in the way the indicator variable $\delta(X)$ is treated: in the LP formulation it becomes a real valued variable in the range $[0, 1]$. Once the LP solver determines real values for $\delta(X)$ for all points X, one can simply pick X as a centroid if $\delta(X) > t$ for some threshold value $0 < t \leq 1$. Note that because the CCS problem is hard to approximate, this LP formulation can not provide a guaranteed approximation to the CCS problem. Furthermore, because

$$
\begin{array}{lccr}
\text{minimize} & \displaystyle\sum_{C_{1_i} \in C_1} \delta(C_{1_i}) + \sum_{C_{2_{i'}} \in C_2} \delta(C_{2_{i'}}) & & \\
\text{s.t.} & & & \\
& \displaystyle\sum_{d(C_{1_i}, C_{1_k}) \leq d(C_{1_i}, f(C_{1_i}, t))} \delta(C_{1_k}) \geq \delta(f(C_{1_i}, t)) & t \leq T,\, C_{1_i} \in C_1 & \\
& \displaystyle\sum_{d(C_{2_p}, C_{2_l}) \leq d(C_{2_p}, f(C_{2_p}, t))} \delta(C_{2_l}) \geq \delta(f(C_{2_p}, t)) & t \leq T,\, C_{2_p} \in C_2 & (2) \\
& \displaystyle\sum_{d(C_{1_i}, C_{1_j}) \leq d(C_{1_i}, f(C_{1_i}, T))} \delta(C_{1_j}) \geq 1 & C_{1_i} \in C_1 & \\
& \displaystyle\sum_{d(C_{2_p}, C_{2_q}) \leq d(C_{1_p}, f(C_{1_p}, T))} \delta(C_{2_q}) \geq 1 & C_{2_{p^{‘}}} \in C_2 & \\
& 0 \leq \delta(X) \leq 1 & X \in C_1 \cup C_2 &
\end{array}
$$

the number of constraints are still quadratic with the number of points, the LP formulation, even when tackled with some of the best known LP solvers, takes considerable amount of time and the classification results obtained are not very accurate. Finally it is quite likely that an LP solver may return the trivial solution to the above problem, i.e. if $|C_1| = |C_2|$, $\delta(X) = 1/|C_1|$ for all X. Note that this solution not only satisfies all constraints but also ensures that the sum of the indicator variables of points in each class is exactly 1, which is as small as it can get due to the last constraint. Obviously this is not a solution of interest.

In the remainder of the paper we describe the relaxation to the above IP formulation which is not only faster to solve, but also have no trivial solutions.

Fixed Size Neighborhood Based Solution. In the LP formulation (2) below, we fix the number of constraints per point X to a user defined value T as follows. Let $f(X, t)$ be t-th nearest neighbor of point X from the opposite class. By the first and second set of constraints we ensure that, given a point X, for each centroid $f(X, t)$ such that $t \leq T$, there is a centroid from the same class of X which is closer to X than $f(X, t)$. By the third and fourth set of constraints, we ensure that for each point X, there is at least one centroid from the same class which is closer to X than $f(X, T)$. As a result, for each point X there are $T + 1$ constraints in the LP formulation (2).

The solution of the linear programs (1) and (2) can be used for choosing centroids from the training data set and with the aim of classifying points of unknown classes. This is achieved by picking each point X whose $\delta(X) > 0$ as a centroid, which ensures, for both formulations, that not only the nearest centroid for each point Y is in its own class but there is at least one centroid within the desired distance to Y - we prove this below.

Lemma 2.2. *Picking each point X for which $\delta(X) > 0$ as a centroid satisfies all constraints in the linear program (2).*

Proof. We prove the above lemma for LP formulation (2). Consider for each point Y, all $X = f(Y, t)$ from the opposite class with $\delta(X) > 0$ such that $d(X, Y) = d(Y, f(Y, t)) \leq d(Y, f(Y, T))$. The solution of LP formulation (2) ensures that there is at least one point Z in the same class with Y which is closer to Y than X and $\delta(Z) > 0$. This implies that Z will be picked as a centroid guaranteeing that Y will have a centroid in its own class which is at least as close to Y as each centroid in the opposite class. Furthermore, the solution to the LP formulation (2) ensures that there will be at least one point Z with $\delta(Z) > 0$ in the same class with Y whose distance to Y is at most $d(Y, f(Y, T)$. Thus all constraints for LP formulation (2) will be satisfied after rounding.□

The above LP formulation and its corresponding solution typically give more accurate solutions

Table 1. Distribution of positive and negative examples in the PTC, Mutag, and Drug data sets

	MM	FM	MR	FR	Mutag	Drug
No. of Pos.	129(38.4%)	143(41.0%)	152(44.2%)	121(34.5%)	125(66.5%)	958(12.8%)
No. of Neg.	207(61.6%)	206(59.0%)	192(55.8%)	230(65.5%)	63(33.5%)	6550(87.2%)
Total	336	349	344	351	188	7508

to the small molecule classification problems we experimented with.

This particular LP formulation will represent the CCNN method in Section 4 where we discuss our experimental results.

We finally note that although the general CCS problem is difficult to approximate, for the special case that one of the classes is much smaller than the other, it is possible to obtain a guaranteed approximation. We show how to approximate this special version of the CCS problem below and report results obtained by this algorithm in Section 4.

When one of the classes is small: a logarithmic approximation to CCS problem. In many in-silico drug discovery applications, there are typically very few compounds that have the particular bioactivity of interest; the majority of the data set will be "negative examples", i.e. those which do not exhibit the bioactivity. Our approach is particularly useful for these applications as it enables to eliminate a vast majority of the negative examples which may have no particular effect in the accuracy obtained, yet which would slow down the classification task. Here we focus on the CCS problem in which we may want to maintain all members of a particular class as centroids but aim to minimize the number of centroids in the opposite class so as to achieve perfect accuracy in the classification of the training set. Interestingly this seemingly simpler version of the problem yields a simple $O(\log n)$-approximation algorithm as follows.

Given that all points $C_{1_i} \in C_1$ will be picked as centroids, we need to pick the minimum number of points $C_{2_j} \in C_2$ such that for any point X, the closest centroid to X should be from the same class of X. Let $S(C_{2_j}) = \{C_{2_k} | \forall C_{1_i}, d(C_{2_j}, C_{2_k}) < d(C_{2_k}, C_{1_i})\}$ i.e. $S(C_{2_j})$ is the set of points in C_2 that are correctly classified by choosing C_{2_j} as a centroid. Now the problem of picking minimum number of points $C_{2_j} \in C_2$ reduces to choosing the minimum number of sets $S(C_{2_j})$ that cover C_2 - with the exception of the sets $S(C_{2_\ell})$ such that picking C_{2_ℓ} as a centroid missclassifies one or more of the points in class C_1. A simple greedy algorithm, which, in each step picks the set that covers the maximum number of uncovered points gives an approximation factor of $O(\log n)$ [12].

The above greedy approach will be denoted as CCNN2 in Section 4.

3. Data Sets

To test the performance of our proposed methods, we performed experiments on three well known, publicly available data sets on mutagenicity, toxicity and drug bioactivities of chemical compounds. The mutagenicity and toxicity data sets have been commonly used as benchmarks to access the performance of new classification methods for small molecules [32, 27, 28, 39]. There are also a number of studies on the whole or part of the dataset of drug activity [13, 4, 14].

Mutagenitiicy data set. There are originally 230 chemical compounds in this data set including aromatic and hetero-aromatic nitro compounds that are tested for mutagenicity on Salmonella typhimurium [8]. Small molecules with positive levels of log mutagenicity are considered as positive examples and negative ones do not have mutagenic activity or in low level. However, the first analysis [8] showed that 188 compounds (125 positive and 63 negative examples as shown Table1) could be fitted by regression. Only these compounds are considered in our experiments.

Predictive Toxicology Challenge data set. This Predictive Toxicology Challenge (PTC) data set [10] reports experimental results of US National Toxicology Program on the carcinogenicity of chemical compounds for male mice (MM), female mice (FM) male rats (MR) and female rats (FR)(Table1). MM has 129 positive and 207 negative examples; FM has 143 positive and 206 negative examples; MR has 152 positive and 192 negative examples, and FR has 121 positive and 230 negative examples.

Table 2. Comparison between our methods and other centroid based nearest neighbor classifiers as applied on Drug data set. The best value in each category of comparison is highlighted.

Method	# Centroids	% Training Set	Accuracy	Precision	Recall
RNG	1705	28.39	89.00	72.00	58.00
CG	4804	79.99	92.00	68.5	68.8
CCNN	1489	24.79	89.89	56.36	61.18
CCNN2	**1052**	**17.51**	**92.17**	69.12	69.70
NN(L_1)	6006	100.0	91.02	64.70	65.30

Drug data set. The last data set (denoted as "Drug") is the complete small molecule collection from [4, 14], which includes 958 drug compounds and 6550 non-drug compounds including antibiotics, human, bacterial, plant, fungal metabolites and drug-like compounds (Table1). The redundancy in the data set has been eliminated through the SMILES records as follows. All duplicate entries have been removed and all organometallic structures as well as inorganic components have also been eliminated. All molecules containing basic and/or acidic groups have been converted into un-ionized form. All molecular structures have been further optimized with the MMFF94 force-field as it is implemented within the MOE modeling package [22].

Descriptors. We used the same set of descriptors for all three data sets. The optimized structures of the compounds have been used for calculating 30 3D inductive QSAR descriptors also used in our previous studies [4, 6, 7, 5, 14] and about 32 conventional QSAR descriptors implemented within the MOE (Molecular Operating Environment) [22]. These 3D inductive' QSAR descriptors include various local parameters calculated for certain kinds of bound atoms (for instance most positively/negatively charges, etc), groups of atoms (for substituent with the largest/smallest inductive or steric effect within a molecule, etc) or for the entire molecule. One common feature for all of the introduced inductive descriptors is in their relation to atomic electronegativity, covalent radii and interatomic distances. It should also be noted, that all descriptors (except the total formal charge) depend on the actual spatial structure of molecules. All the inductive QSAR descriptors have been calculated by the custom SVL scripts that can be freely downloaded through the SVL exchange [40]. The readers can contact us for the list of these inductive parameters and conventional QSAR descriptors. Also on each of the three data sets, each descriptor is normalized according the observed maximum and minimum value of that descriptor in order to remove the bias towards descriptors with larger values.

4. Experimental Results

In this section we aim to assess the performance of our methods in terms of the number of centroids they pick and their accuracy in comparison to state of art methods for classification of small molecules.

All the experiments for each data set are reported based on 20 independently run experiments each of which splits the data into training and test sets consisting of 80% and 20% of the data set respectively. The reported values are the averages over these 20 runs. For solving the linear program in CCNN, we used Coin-OR linear programming solver version 1.5 [24].

We note that $T = 19$ was the setting used for CCNN: for larger values of T, the number of centroids returned do not change substantially. All the experiments with our classifiers as well as other methods were performed on a Pentium IV with 3.2 Ghz speed and 2 Gb of RAM.

Centroid selection comparison. In this experiment we compare our classifiers with alternative, state of art algorithms for centroid selection, namely, Relative Neighborhood Graph (RNG), Gabriel Graph (GG). Since the number of compounds in Mutag and PTC data sets is small (in the order of hundreds) we only compare the number of centroids of all the methods picked from Drug data set. For each centroid nearest neighbor classifier, Table2 reports the number of centroids it picks from the training set on the average, percentage of the number centroids, accuracy (TP+TN)/(TP+TN+FP+FN),

Table 3. Comparison between our methods and other popular classifiers as applied on Mutag, PTC and Drug data sets. The highest accuracy for each data set is highlighted.

Data set	Method	Precision	Recall	Accuracy	Running time (mins)
Mutag	NN(L_1)	87.80	92.00	86.17	6
	CCNN	92.00	92.74	89.94	6
	CCNN2	92.13	94.35	**90.91**[*]	6
	SVM-Linear	92.00	92.00	89.36	6
	SVM-Poly (degree 2)	91.30	92.00	88.83	6
	SVM-Radial ($\gamma = 1.0$)	86.60	92.80	85.63	6
	Cao *et al.*	88.2	77.8	82.35	20
	MK Kashima *et al.*	94.4	88.7	89.10	6
FM	NN(L_1)	55.00	49.70	62.64	26
	CCNN	56.13	60.84	64.37	26
	CCNN2	58.11	60.14	**65.80**	26
	SVM-Linear	59.10	38.50	63.79	26
	SVM-Poly (degree 2)	49.00	51.00	58.05	26
	SVM-Radial ($\gamma = 1.0$)	65.20	31.50	64.94	26
	Cao *et al.*	60.00	38.00	64.6	30
	MK Kashima *et al.*	14.00	80.00	63.30	7
MM	NN(L_1)	45.30	45.30	58.21	26
	CCNN	50.35	55.47	62.09	26
	CCNN2	52.76	52.34	**63.88**	26
	SVM-Linear	53.70	34.40	63.58	26
	SVM-Poly (degree 2)	47.50	51.60	59.70	26
	SVM-Radial ($\gamma = 1.0$)	48.10	19.50	61.19	26
	Cao *et al.*	54.20	25.00	63.28	30
	MK Kashima *et al.*	27.1	50.1	61.9	7
FR	NN(L_1)	47.00	45.80	63.71	26
	CCNN	46.97	51.67	63.43	26
	CCNN2	51.24	51.67	**66.57**	26
	SVM-Linear	45.60	21.70	63.58	26
	SVM-Poly (degree 2)	44.40	45.80	61.71	26
	SVM-Radial ($\gamma = 1.0$)	42.30	9.20	64.57	26
	Cao *et al.*	49.2	26.2	65.52	30
	MK Kashima *et al.*	51.24	50.12	66.10	7
MR	NN(L_1)	55.20	59.90	60.64	26
	CCNN	56.36	61.18	61.81	26
	CCNN2	57.32	61.84	**62.68**	26
	SVM-Linear	57.80	58.60	62.68	26
	SVM-Poly (degree 2)	52.60	53.30	58.02	26
	SVM-Radial ($\gamma = 1.0$)	59.10	36.20	60.64	26
	Cao *et al.*	53.80	55.60	59.47	30
	MK Kashima *et al.*	56.00	46.00	59.01	7
Drug	NN(L_1)	64.70	65.30	91.02	121
	CCNN	56.36	61.18	89.89	181
	CCNN2	69.12	69.70	**92.17**	150
	SVM-Linear	76.10	8.70	87.89	121
	SVM-Poly (degree 2)	77.10	38.30	90.17	180
	SVM-Radial ($\gamma = 1.0$)	80.10	35.00	90.60	121
	Cao *et al.*	81.20	56.20	92.00	5760($\approx$ 5days)
	MK Kashima *et al.*	53.70	57.00	89.10	1080($\approx$ 1day)

[*] Although CCNN2 is always better than CCNN, but unfortunately CCNN2 is applicable only to cases we have binary classification; for classification problem which has more than two classes we have to use the CCNN.

precision TP/(TP+FP) and recall TP/(TP+FN). Observe that CCNN and CCNN2 pick much fewer centroids compared to CG; the accuracy is almost the same in the case of CCNN and better in the case of CCNN2. Although the number of centroids RNG picks is close to ours, the accuracy of CCNN2 is much better.

Comparison with small molecule classification methods. Here we compare the performance of our methods with a popular classifier used in QSAR modelling such as SVM, Marginalized Kernel (MK by Kashima *et al.* [21]) which is one of kernel machines that we could obtain the software package and Maximum Common Substructure (MCS by Cao *et al.* [38]) which is one of the most recent methods for classification of small molecules. MK has been usually used to compare with other graph kernel methods [27, 32]. The performance of the methods are assessed through their accuracies and running times as per Table3.

We use the SVM implementation from the machine learning software suite WEKA, version 3.5.8 [33]. We use three types of kernels: linear kernel (SVM-Linear), polynomial kernel of degree 2 (SVM-Poly) and Radial basis kernel (SVM-Radial) with $\gamma = 1.0$ and all kernels with the complexity parameter $c = 1.0$.(value of c and γ was chosen such that the best accuracy is achieved) For using MCS Cao *et al.*, we choose 10, 20, and 100 basis compounds for Mutag, PTC and Drug data sets respectively. Note that 30 inductive and 32 conventional QSAR descriptors used by our methods and by different kernels of SVM are different from MK and MCS. Also note that the running time reported on the table includes the time for computing descriptors as well. It takes 5, 25, 120 mins to compute 62 descriptors for Mutag, PTC and Drug data sets respectively.

In general, CCNN2 performs the best in term of accuracy compared to all other methods. It is always better than the NN method. Although we can only apply CCNN2 on data sets of two kinds of bioactivities, CCNN can be applied to multiclass data sets. Even though CCNN falls behind NN classification in Drug and FR data sets, it is the second best after CCNN2 on the average.

We note that for Mutag and PTC data sets, the accuracy of CCNN2 may be slightly lower than the best method reported in the literature: Swamidass *et al.* [32] report an accuracy of 91.5 on Mutag, 66.4 on MM, 65.7 on MR. However because this method is not publicly available we could not verify these figures and hence do not report them in Table3. Finally, we note that the running time of our methods, especially in bigger data sets such as Drug, is many times better than both MCS and MK.

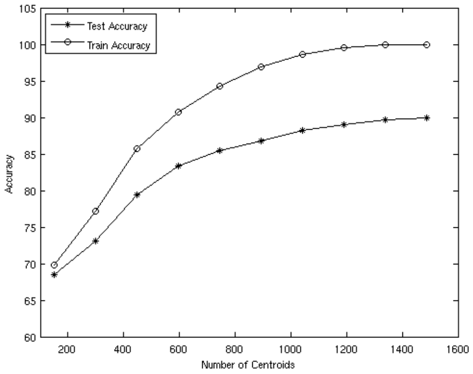

Fig. 1. Accuracy vs. Number of Centroids

The trade-off between the accuracy and the number of centroids. We explore the trade-off between the number chosen centroids and accuracy obtained from test and training sets in Figure 1. Note that the accuracy on the test and training set both decrease as we decrease the number of centroids picked from the training set. In order to pick t centroids, first we solve the fixed size neighborhood version of CCNN - then we pick the top t compounds as centroids.

5. Conclusion

Nearest Neighbor based classification is one of the most commonly used methods for QSAR modelling. However, the standard NN methods suffer from several drawbacks such as being slow and over-fitting in high dimensional data. These two issues are present in all available NN based classification methods for small molecules. In order to address these issues, we introduce Combinatorial Centroid Nearest Neighbor method which determines a few representative compounds for each bioactivity class in a way that yields no classification errors on the training set. Experimental results on three public data sets on mutagenicity, toxicity and drug activities show that CCNN classifiers outperform other kernel methods and SVM in terms of accuracy and retain only a small fraction of the training sets. Moreover, our methods run several times faster than other methods in bigger datasets.

6. Acknowledgement

We thank Yiqun Cao for his help in running MCS program and Hisashi Kashima for providing his Marginalized Kernel software package.

References

1. Machines using Sequential Minimal Optimization, chapter 12. MIT Press, 1999.
2. NCI Open Database Compounds. http://cactus.nci.nih.gov/ncidb2/download.html, 2003.
3. F.Costa A.Ceroni and P.Frasconi. Classification of small molecules by two- and three-dimensional decomposition kernels. *Bioinformatics*, 23:2038–2045, 2007.
4. A.Cherkasov. Inductive descriptors: 10 successful years in QSAR.*Current Comp-Aided Drug Design*, 1(1):563–569, 2005.
5. A.Cherkasov. Can "bacterial-metabolite-likeness" model improve odds of in silico antibiotic discovery?*J. Chem. Inf. Model.*, 46:1214–1222, 2006.
6. A.Cherkasov and B.Jankovic. Application of "inductive" qsar descriptors for quantification of antibacterial activity of cationic polypeptides. *Molecules*, 9:1034–1052, 2004.
7. M.Fallahi A.Cherkasov, Z.Shi and G.L. Hammond. Successful insilico discovery of novel non-steroidal ligands for human sex hormone binding globulin. *J. Med. Chem.*, 48:3203–3213, 1995.
8. G.Debnath A.J.Shusterman A.K.Debnath, R.L.Lopez de Compadre and C.Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *J. Med. Chem.*, 34:786-797, 1991.
9. R.S.Poulsen B.K.Bhattacharya and G.T.Toussaint. Application of proximity graph to editing nearest neighbor decision rule. In *International Symposium on Information Theory*, 1981.
10. S.Kramer C.Helma, R.D.King and A.Srinivasan.The predictive toxicology challenge 2000-2001. *Bioinformatics*, 17(1):107-108, 2001.
11. T.M. Cover and P.E.Hart. Nearest neighbor pattern classification.*IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
12. D.S.Johnson. Approximation algorithms for combinatorial problems.*Proceedings of the fifth annual ACM symposium on Theory of computing*, 9:256–278, 1972.
13. E.Karakoc A.Cherkasov and S.C.Sahinalp.Distance Based Algorithms for Small Biomolecule Classification and Structural Similarity Search. In *ISMB*, pages 243–251, 2006.
14. E. Karakoc A. Cherkasov and S.C. Sahinalp. Comparative qsar analysis of bacterial-, fungalplant- and human metabolites. In *Pacific Symposium on Biocomputing*, pages 133–144, 2007.
15. Micheli et al. Analysis of the internal representations developed by neural networks for structures applied to quantitative structure-activity relationship studies of benzodiazepines. *J. Chem. Inf. Comput. Sci.*, 41:202-218, 2001.
16. F.Angiulli. Fast condensed nearest neighbor rule. In *ICML*, pages 25–32, 2005.
17. National Center for Biotechnology Information (NCBI) Pub Chem. http://pubchem.ncbi.nlm.nih.gov/ 2006.
18. G.M.Maggiora and M.A. Johnson.*Concepts and Applicatons of Molecular Similarity.* Wily, 1990.
19. G.Wilfong. Nearest Neighbor Problems. In *Proceedings of the seventh annual symposium on Computational Geometry*, pages 383–416, 1991.
20. H.Kashima and A.Inokuchi. Kernels for graph classification. In *IEEE ICDM*, 2002.
21. K.Tsuda H.Kashima and A.Inokuchi. Marginalized kernels between labled graphs. In *ICML*, 2003.
22. Chemical Computing Group Inc. MOE: Molecular Orbital Environment, 2006.
23. K.R.Gabriel and R.R.Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259-270, 1969.
24. R.Lougee-Heimer. The Common Optimization Interface for Operations Research. *IBM Journal of Research and Development*, 47:57–66, 2003.
25. H.Saigo L.Ralavola, S.J.Swamidass and P.Baldi. Graph kernels for chemical informatics. *Neural Networks*, 18:1093–1110, 2005.
26. P.E.Hart.The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3):515–516, 1968.
27. T.Akutsa J.Perret P.Mahe, N.Ueda and J.Vert. Extensions of marginalized graph kernels. In *ICML*, 2004.
28. T.Akutsa J.Perret P.Mahe, N.Ueda and J.Vert. Graph kernel for molecular structure-activity relationship analysis with support vector machines. *J.Chem. Inf. Model*, 45:939–951, 2005.
29. P.Vincent and Y.Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. In *NIPS*, pages 985–992, 2001.
30. J.M.Banard P.Willett and G.M. Downs. Chemical Similarity Searching. *J.Chem. Inf. Comp. Sci*, 38(6):983–996, 1998.
31. R.Bar-Yehuda and S.Moran. On approximation problems related to the independent set and vertex cover problems.*Journal of Discreet Applied Mathematics*, 9(5):1–10, 1984.
32. J.Bruand P.Phung L.Ralavola S.J.Swamidass J.Chen and P.Baldi. Kernels for small molecules and the prediction of mutagenicity, toxicity and anticancer activity.*Bioinformatics*, 21:359–368, 2005.
33. C.Bhattacharyya S.S.Keerthi S.K.Shevade and K.R.K. Murthy. Improvements to Platt's SMO

Algorithm for SVM Classifier Design.*Neural Computation*, 13(3):637–649, 2001.

34. G.T. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.
35. U.Feige. A threshold of ln(n) for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.
36. X.Chen and C.H.Reynolds. Performance of Similarity Measures in 2D fragment-Based Similarity Searching: Comparison of Structural Descriptors and Similarity Coefficients. *J.Chem. Inf. Comp. Sci*, 42(6):1407–1414, 2002.
37. Y.Baram. A geometric approach to consistent classification. *Pattern Recognition*, 13:177–184, 2000.
38. T.Jiang Y.Cao and T.Grike. A maximum common substructure-based algorithm for searching and predicting drug-like compounds. In *Bioinformatics*, pages 366–374, 2008.
39. A.Smalter, J.Huan,and G.Lushington. Graph Wavelet Alignment Kernels for Drug Virtual Screening. In *CSB*, 2008.
40. MOE SVL exchange community: *http://svl.chem comp.com/index* May 2006