INHIBITOR PREDICTION BY MACHINE LEARNING APPROACHES

YAO LIXIA

(B.Eng., Dalian University of Technology)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTATIONAL SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgments

I would very much like to thank my supervisor, Dr. Chen Yuzong, who had given me the opportunity to work on such an interesting research project, paid patient guidance to me, and given me much invaluable help and constructive suggestion on it.

I would also like to express my gratitude to Dr. Cai Congzhong and Dr. Xue Ying for their helpful advice and cooperation. Thanks are also owed to my colleagues and friends: ChenXin, Chunwei, Zhiliang, Zhiwei, Jifeng, Lianyi, Chanjuan and Honghuang for being ever so willing to share with me their ideas.

Lastly but not least, I would like to dedicate this work to my parents, for their unconditional love and support throughout my life.

Yao Lixia

June 2004

Contents

Synopsis

Discovery of a new drug often starts with screening large collection of compounds against a certain biological target. These chemical libraries are typically very large, on the order of hundreds of thousands, or even millions of compounds. Although new technologies, such as High Throughput Screening (HTS) or Ultra High Throughput Screening (UHTS), allow an assay of hundreds of thousand compounds per day, it is still very expensive to screen the entire chemical library.

Therefore scientists have been trying to predict the activity of a compound by using molecular properties as input variables. Based on the activity response and the molecular properties on a sample, a quantitative structure activity relationship (QSAR) model can be built and used to predict the activity of the remaining compounds in the library. The most commonly used tools in QSAR are Linear Regression and Partial Least Square. But these methods are simplification of the real-world problems and do not perform well on heterogeneous datasets.

In this thesis, we propose several state-of-the-art algorithms from the machine learning community to facilitate building QSAR models. These techniques have good capability for generating complicated mathematical rules for classification,

prediction and recognition tasks without requiring much domain knowledge in many fields. Our aim here is to evaluate the feasibility of introducing these machine learning approaches to lead identification and its ADME/toxicity properties analysis.

Specifically, three machine learning methods, namely decision tree, k-nearest neighbor and support vector machine , as well as preprocessing techniques such as normalization and principal component analysis were explored. These methods were tested on the inhibitor/antagonist prediction for the therapeutic targets— 5-HT_2 receptor and cholinesterase, and an ADME related target—CYP3A4.

The flow of work includes four steps. First of all, the examples of the inhibitors/antagonists of the three protein targets are collected manually from the available references. Then the 3D structures of the compounds are encoded into numerical vectors by QSAR molecular descriptors, which are analyzable by the machine learning techniques. After that, different machine learning models are derived from the data sets. At the end, the prediction capacities of these models are evaluated and analyzed.

The experimental results observed in all three data sets demonstrate that support vector machine beats decision tree and k-nearest neighbor and gives very good results. It suggests that support vector machine may be a promising approach to analyze the pharmaceutical properties of chemical compounds, which may lead to a practical tool for drug design in the near future. Besides, principal component analysis also shows usefulness in dimensionality reduction, which, in general, improve the prediction capacity, when used with support vector machine.

List of Figures

Chapter 1

Introduction

1.1 Modern drug discovery and development

The drug discovery and development process is a sophisticated process (Figure 1.1). It typically begins with target identification and validation. Target is usually an enzyme, a receptor or an antibody associated with disease. Then different technologies such as chemical database screening and combinatorial chemistry are applied to identify lead compounds (usually enzyme inhibitor or receptor agonist/antagonist), which bind to the drug target to stop or alleviate disease. Preclinical tests on lead compounds follow after, in order to select a few compounds with the best overall profile on efficacy, absorption, distribution, metabolism, excretion, and toxicology. The resulting compounds are filed as investigational new drugs (IND) with the FDA (U.S. Food and Drug Administration), and are then carried into Phase I, Phase II, and Phase III clinical trials. At the conclusion of successful preclinical and clinical testing, a new drug application (NDA) is filed with the FDA. Nevertheless getting the FDA approval is still not the end of the story. After that, pharmaceutical companies may still need to conduct post-marketing studies.

On average, it requires an investment of US$880 million and as long as $10 \sim 15$

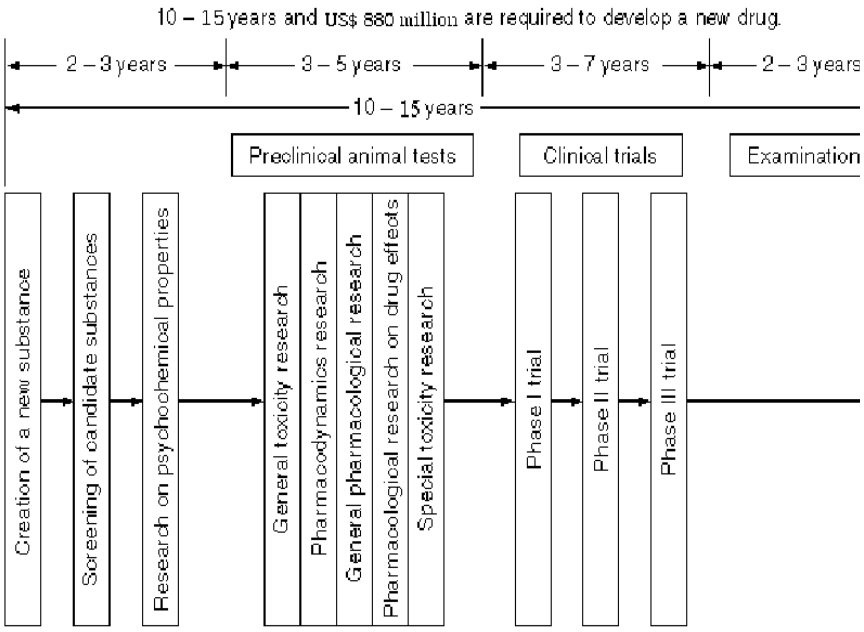

Figure 1.1: Drug discovery and development procedure

years for an experimental drug to be approved by the FDA for marketing in the United States[1]. About 75% of this cost (∼US$660 million) is attributable to failure along the pharmaceutical value chain[1, 2]. Most compounds that are discovered and undergone preclinical testing fail to meet the high safety and efficacy requirements before testing in humans can proceed. Statistics shows that only five leads in 5,000 are able to enter human clinical testing. Furthermore, only one out of five drug candidates approved for human clinical testing is ultimately approved for marketing.

To ensure that only those pharmaceutical products that are both safe and efficacious are brought to market, FDA and other relevant regulatory agencies have imposed a very strict procedure for those pharmaceutical companies who apply for permission to begin clinical testing on humans. Due to the high cost incurred from these complicated and expensive protocols for the clinical and post-clinical processes, pharmaceutical companies are paying more and more efforts on new technologies

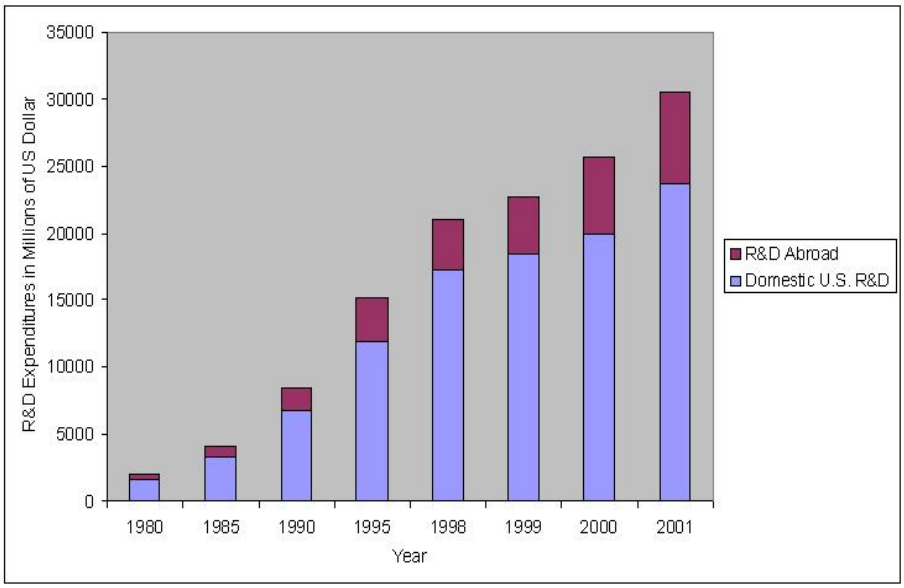

Figure 1.2: R&D Expenditures in Pharmaceutical Industry in U.S. and Abroad , 1980–2001 (*Data Source:PhRMA annual survey 2001*)

pertaining to drug leads discovery and preclinical testing in order to improve the quality of drug lead and reduce the failure rates throughout the pharmaceutical value chain. Figure 1.2 shows the increasing trend of global R&D (research and development) expenditures from 1980 to 2001 in pharmaceutical industry. Currently nearly one-third of company financed R&D is devoted to the drug discovery phase and the preclinical development. To innovate or to die has become the rule of thumb of the game.

1.2 Current lead identification and optimization techniques

After studying the pathology of the diseased state and determining the target where intervention is most likely to be effective, scientists can start looking for the possible lead compounds. Lead compound identification and optimization is considered a bottle-neck in the whole drug discovery process. How to select lead candidates which can be proven successful in preclinical and clinical trials is its challenging objective. In the following subsections, the most widely used lead identification and optimization methods were reviewed. However, the linear process of lead identification, evaluation and refinement activities is moving towards a more integrated parallel process.

1.2.1 Lead identification

In the lead generation phase, compounds which can interact with the target protein and modulate its activity are identified. Such compounds are mainly identified by screening a chemical library. The prevailing approaches include high-throughput screening, virtual screening and NMR screening as introduced below.

1.2.1.1 High-throughput screening

In the past five years, the technique of high-throughput screening (HTS), has gone through a rapid development. Today, HTS for drug discovery has been established in most pharmaceutical companies, and conducted in more than 500 laboratories worldwide. An entire in–house compound libraries with millions of compounds can be screened at a speed of 10,000 (HTS) to 100,000 compounds per day (uHTS, ultra high–throughput screening) via very robust test assays[3, 4]. Combinatorial

chemistry and parallel synthesis are employed to generate such huge numbers of compounds.

However, there are some concerns regarding the "numbers' game" . On one hand, the largest chemical libraries available today include up to 10 million compounds (10^7), while the chemical space consisted of carbon, oxygen, nitrogen, hydrogen, sulfur, phosphorous, fluorine, chlorine and bromine, having a molecular weight of less than 500 Da is estimated to cover 10^{62}–10^{63} compounds[5]. Many potential drug leads might be excluded from our investigation. On the other hand, as current compound libraries are guided by simple chemistries, they are composed of compounds that do not necessarily own "drug–like" attributes, such as target specificity, solubility and oral bioavailability. Consequently even a compound is identified by HTS, it might not always be suitable for initiation of further medicinal chemistry analysis.

1.2.1.2 Virtual screening

A different approach is *in silico* or virtual screening[6, 7]. With this computer method, three–dimensional (3D) structures of compounds from virtual or physically existing libraries are docked into binding sites of target proteins with known or predicted structures. Empirical scoring functions are used to evaluate the steric and electrostatic complementarity (the fit) between the compounds and the target protein. The highest ranked compounds are then suggested for biological tests. Since they do not require the compound entities and thus avoid expensive synthesis, these software tools have drawn much attention after their debut in early 1980's[8]. Furthermore, they allow rapid and thorough understanding of the relationship between chemical structure and biological function. It usually takes less than a minute to screen a chemical structure when using the most elegant algorithms available today. The throughput for a computer with 100 parallel CPUs is even higher compared to

current uHTS technologies[6].

The main advantage of virtual screening is that it does not depend on the availability of compounds, meaning that not only in-house libraries, but also external or virtual libraries can be searched. The application of scoring functions on the resulting data sets facilitate smart decision-making as to which chemical structures bear the potential to exhibit the desired biological activity. However, one important prerequisite for these technologies is the availability of structure data of the target, and if possible, in complex with the biological ligand or an effector molecule. Presently, only about 1% of all highly annotated protein sequences have the experimental high resolution structure information. Comparative models for more than 40% of these proteins are available[9]. So, *in silico* screening can be applied on average to one third and probably in the future to half of drug discovery projects.

1.2.1.3 NMR for lead identification

Traditionally nuclear magnetic resonance (NMR) was used only as an analytical tool to aid chemists in characterizing small molecule compounds. Today, due to the rapid development and wide application of X-ray crystallography and NMR to determine atomic-resolution structures of proteins, nucleic acids, and their complexes, NMR has been established as a key method in structure-based drug design. It has the unique advantage of being able to detect and quantify interactions with high sensitivity without requiring prior knowledge of protein function. Moreover, NMR can provide structural information on both the target and the ligand to aid subsequent optimization of weak–binding hits into high–affinity leads. More about its application in drug discovery can be found in *NMR in Drug Discovery*[10].

Once hits (compounds that elicit a positive response in a particular biological assay) have been identified by applying the different screening approaches, they

will be validated by re–testing them and checking the purity and structure of the compounds. This is to avoid wasting time with further characterizations of false positive compounds from the screening. Only when the hits fulfill certain criteria will these be regarded as leads. The criteria are developed from different aspects such as:

- pharmacodynamic properties: efficacy, potency and selectivity *in vitro* and *in vivo*;
- physicochemical properties: e.g., Lipinski's "rule–of–five"[1], water–solubility and chemical stability;
- pharmacokinetic properties: e.g., permeability in the Caco–2 assay;
- chemical optimization potential: the difficulty of chemical synthesis can be crucial, "dead–end–leads" which are synthetically not easily amendable to many variations should be avoided;
- patentability: compounds that are to some extent protected by competitor's patents are certainly less interesting than entirely new lead structures.

1.2.2 Lead optimization

During the lead optimization phase, small organic molecules are chemically modified and pharmacologically characterized in order to obtain compounds with suitable

[1]Christopher Lipinski, established a set of rules, known throughout the drug discovery world simply as the Rule of Five. Although there are only four rules, each rule involves a multiple of five as a parameter. Lipinski's Rule of Five has become a touchstone for drug researchers to predict which compounds are most drug-like and thus the best potential candidates. According to these rules, poor absorption is likely when several of the following occur: (1)There are more than 5 hydrogen bond donors; (2)There are more than 10 hydrogen bond acceptors; (3)The molecular weight is over 500; (4)The $ClogP$ is over 5.

pharmacodynamic and pharmacokinetic properties to become a drug. This process ideally requires the simultaneous optimization of multiple parameters and thus is a time-consuming and costly step, which probably constitutes the "tightest" bottleneck in drug discovery. However, by turning a biologically active chemical into an effective and safe drug, lead optimization contributes essentially towards added value in the drug discovery chain.

Lead optimization generally involves iterative rounds of synthetic organic chemistry and compound evaluation that can take from months to years of time and involve the efforts of a team of synthetic organic chemists. The starting point is the collection and analysis of structure-activity relationship (SAR) data. Initial SAR data can be obtained from commercial sources. The first synthetic efforts will generally focus on systematic exploration of tolerated candidate molecule. Later rounds will build the initial analysis and are often targeted at solving particular problems. One important goal of compound optimization is to improve potency, since high potency minimizes dose and hence improves specificity and reduces toxicity. Potencies (expressed as EC50) in the range of 100 nM or higher are desired, though much lower potency can be tolerated given high specificity, low toxicity, and the administration mode that can accommodate the required dose.

Several approaches are available to maximize the utility of SAR information for directed acquisition and synthesis of structural analogs to improve compound potency. Advice from experienced medicinal chemists is of considerable practical value to identify undesirable structural features that may result in failure because of toxicity, poor biological stability, immunogenicity, or mutagenic potential[11]. Consideration of empirical information about drug successes and failures can be quite helpful in directing synthesis efforts.

At present, three major computational methods have been used in lead optimization: rational structural design, pharmacophore analysis, and quantitative structure-activity relationship (QSAR) analysis. These three may be used individually or in combination.

Rational design methods generally use a high-resolution structure of the target to direct the synthesis of new analogs. The process often involves generating of a very large *in silico* library of potential derivatives and use of computational docking methods to select derivatives that may interact with the target on the basis of shape complementarities and charge placement. While intellectually appealing, there have been few successful examples from application of this strategy alone.

Pharmacophore methods involve definition of the minimal unit that leads to activity (usually a combination of hydrogen bond donor/acceptors, hydrophobic groups, and other functional groups) in a three-dimensional space[12, 13, 14, 15]. The consensus pharmacophore is then used to examine the allowed placement of groups in a set of candidate compounds. Pharmacophore analysis can be carried out without structural information and is most useful in identifying new compounds with a desired activity based on a three-dimensional similarity to early leads. Interestingly, the structures of many of the novel cystic fibrosis transmembrane conductance regulator (CFTR) activators identified by HTS fit well in the flavone-based pharmacophore model, suggesting a common binding site[16].

The last approach is to establish QSAR models that relate calculated physicochemical properties of molecules, rather than strictly structural characteristics, to activity[17, 18]. This type of modelling is particularly important in directing modifications to pharmacokinetic and pharmacodynamic parameters. QSAR modelling requires a set of structurally related compounds with a wide range of activities, ideally 1,000-fold variation in activity, which is often a difficult requirement to meet.

1.2.3 ADME and toxicity properties of lead compounds

The three major reasons a drug fails during clinical trials are lack of efficacy, unacceptable adverse effects, and unfavorable absorption, distribution, metabolism, and excretion (ADME) properties. Therefore, the ultimate success of a compound is not only defined by its biological activity and potency, but also its ADME/toxicity properties. Although high-throughput screening has substantially increased the number of lead compounds, most of these compounds are eliminated during additional screening and testing. Of the drug candidates that enter the clinical development phase, more than 40% fail to reach the market because of unfavorable drug metabolism and pharmacokinetic properties, with an additional 11% eliminated because of toxicity[19, 20, 21]. Therefore, it would be desirable to predict ADME properties of the drug lead in early stage of drug development. Based on a "fail fast, fail cheap" philosophy, ADME and toxicity tests of lead compounds have been moved forward from preclinical trial to lead optimization phase, or even lead identification phase.

High-throughput screening technology has allowed a dramatic increase in the number of lead compounds, whereas technologies used in developing screens for pharmacokinetic properties have lagged behind. Nowadays, the fastest methods in use for ADME screening are several orders of magnitude slower than those for lead identification. Sometimes, when the data size is not that big, medicinal chemists may use liquid chromatography–mass spectrometry (LC–MS) techniques. From the long time experience, many medicinal chemists have also come up with some "common sense" to facilitate the judgement of the ADME/toxicity properties of some compounds.

1.3 Inhibitor/antagonist — a plentiful source of drug leads

Previous study shows that drugs act via the interaction with mainly two type of proteins—enzymes and receptors[22]. According to TTD–the therapeutic target database(TTD)[2][23], enzymes contribute to 44% of the total drug targets and receptors covers 33%.

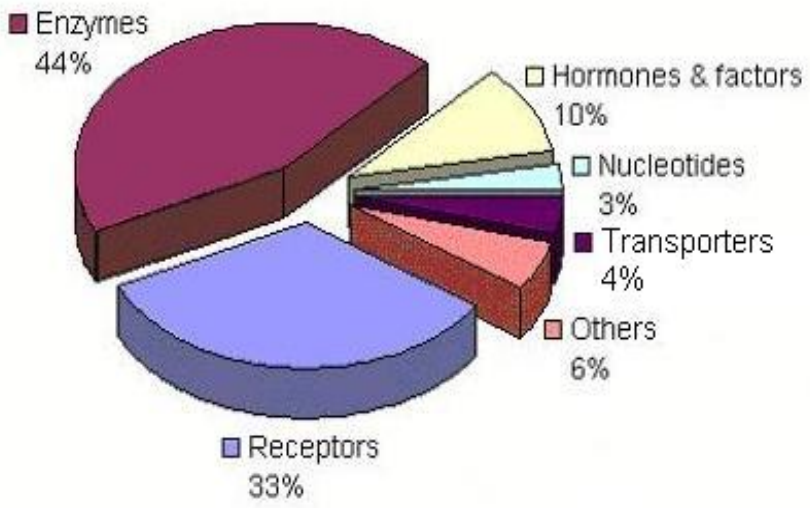

Figure 1.3: Biochemical classes of drug targets in TTD

Enzyme inhibition represents a major strategy in drug design and almost one-third of the current top fifty drugs by sales are enzyme inhibitors. The inhibition of an enzyme-catalyzed reaction will enable the selective modulation of a variety of

[2]It is a comprehensive database providing information about the known and explored therapeutic protein and nucleic acid targets, the targeted disease, pathway information and the corresponding drugs/ligands directed at each of these targets. This database currently contains information about 125 different diseases/condistions, 1174 targets and 1251 drugs/ligands. It is available at http://xin.cz3.nus.edu.sg for free non-commercial use.

biochemical processes such as making cell growth, division and viability untenable, or interrupting major metabolic pathways by blocking the formation of an essential or undesirable metabolite.

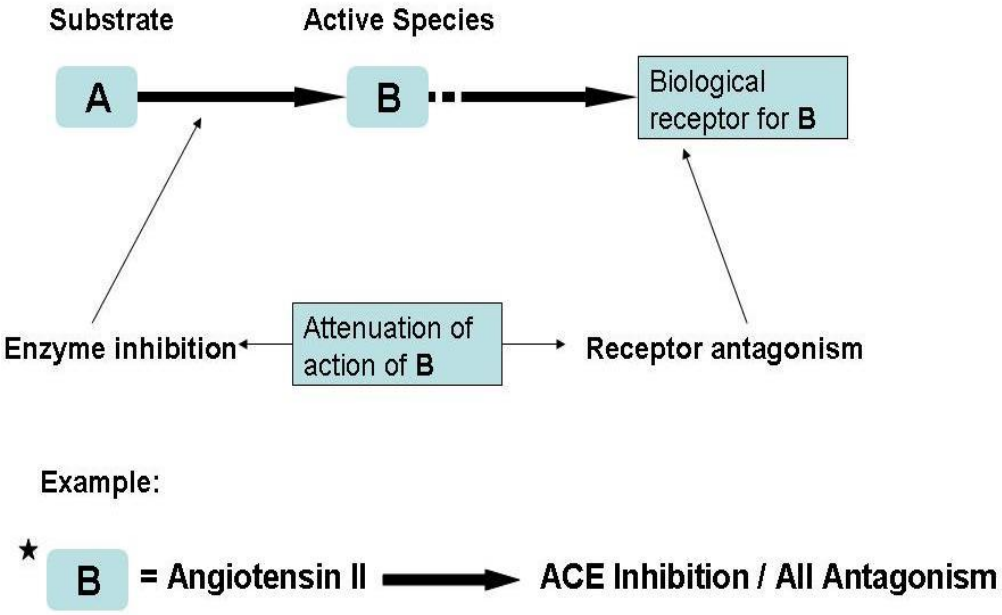

Figure 1.4: Complementarity between enzyme inhibition and receptor antagonism

Receptor modulation via antagonist is complementary to enzyme inhibition. As shown in Figure 1.4, the biological activity of species "B" can be attenuated via inhibition of the enzyme involved in its biosynthesis. The same overall effect can also be achieved via antagonism of the receptor(s) for "B". A good example of this is the attenuation of the action of the vasoconstrictor peptide angiotensin II (AII), which can be achieved via inhibition of its biosynthesis by the angiotensin converting enzyme (ACE) inhibitor or via AII receptor antagonism.

There are a number of successful drugs that were developed from inhibitors/antagonists. For example, angiotensin II receptor antagonists are currently approved for the treatment of hypertension. Various antipsychotic agents and antidepressants bind with

relatively high affinity at 5-HT_{2A} receptors as antagonists. And most new anti-AIDS agents are HIV protease inhibitors.

Another point that should not be ignored is in many cases, few drugs interact only with their intended targets. Most drug molecules can combine with not only their main therapeutic target, but also other enzymes, receptors or other biological entities. This is also a major cause of adverse drug reaction.

Therefore inhibitor identification for many targets are of great significance. Not only may it lead to potential novel drug leads, but also it may provide insights to the mechanism of toxicology/ADME profile of these molecules. Scientists are paying increasingly more efforts to find new inhibitors which may have less side effects and no or little resistance for many diseases.

1.4 Thesis outlines

New drugs are constantly required to combat drug resistance, even though it can be minimized by the correct use of medicines by patients. They are also required to improve the treatment of existing diseases, to treat newly identified diseases and to produce safer drugs with no or minimal adverse side effects. To this end, new efficient techniques for drug design which are able to help us develop high-quality drugs at lower cost and in shorter development period are urgently needed.

In this thesis, I have tried to establish a work flow to predict the possibility of a compound to become a potential drug based on its potential to inhibit (or antagonize) the target protein, as well as its potential to inhibit ADME/toxicity related proteins using a series of widely used machine learning techniques.

This is an *in silico* method which is based on previous pharmaceutical knowledge and information, and thus a large number of expensive and labor-concentrated assays can be saved. More importantly, compared with other modelling methods, this

method is independent of the structure of target protein. So it is particularly useful when the high-resolution 3D structure of the target protein is not available or when the target undergoes big conformational changes during target-ligand interaction.

Specifically, I have worked on the inhibitor prediction of the receptor target—5-HT_2, the enzyme target—cholinesterase and the ADME related target—CYP3A4 using different machine learning approaches. Results suggest that support vector machine (SVM) outperforms other methods and shows promising application potential to drug discovery.

This thesis consists of six chapters:

Chapter 2 illustrates the framework of how we apply state-of-art machine learning techniques to the field of drug discovery. Many technical details are introduced, such as feature vector construction, pharmaceutical data preprocessing, machine learning algorithms, and performance analysis and evaluation. In Chapter 3, Chapter 4 and Chapter 5, these techniques are tested on three typical inhibitor/antagonist prediction scenarios. Finally Chapter 6 concludes the thesis.

Chapter 2

Practical implementation of machine learning screening

2.1 Motivation

As discussed in Chapter 1, two most important goals in drug design are to find active compounds in large databases quickly and to obtain an interpretable model for what properties make a specific subset of compounds active. Taking the economics of pharmaceutical industry into consideration, drug discovery is really an exciting field full of uncertainty and challenges. It is drawing more and more attention of scientists and researchers in both academia and industry.

Recently machine learning approaches have been introduced to address drug design problems, which have shown intriguing successes. For example, some machine learning techniques have been successfully applied to the problem of SAR analysis. Thomas M. Frimurer and his co-workers used artificial neural network to discriminate potential drug-like molecules from large compound databases in 2000[24]. They found that this method gives much better prediction than the widely used "Rule of Five". Markus Wagener and his co-workers developed a model of decision tree to

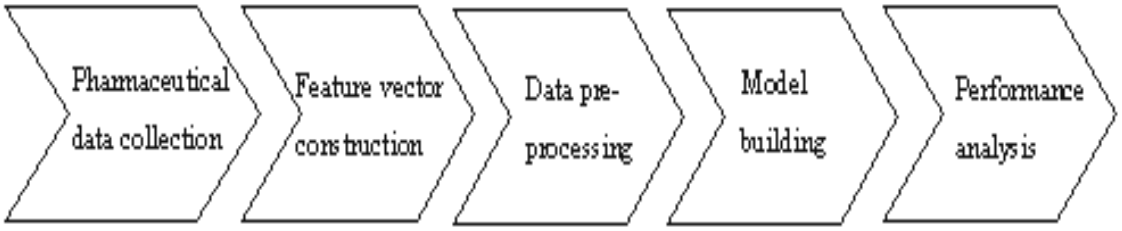

Figure 2.1: Flow chart of machine learning screening

discriminate between potential drugs and nondrugs[25]. Besides acceptable prediction accuracy, they have delivered some comprehensible rules to explain the most significant differences between drugs and nondrugs.

In this work, I furthered their study of identifying the drug-like chemical compounds to the identification of potential drug leads for a particular disease. Specifically I have worked on the inhibitor/antagonist prediction for three particular protein targets of pharmacological significance using three representatives of different machine learning algorithms. The aim is to evaluate the feasibility of introducing machine learning approaches to lead identification and toxicity/ADME properties analysis.

The flow chart of my work is shown in Figure 2.1. First the examples of the inhibitors/antagonist of a certain protein target need to be collected from different reference sources. Then, these data of 3D structures of compounds need to be transformed into a numerical vectors recognizable by machine leaning techniques. After that a machine learning model is derived from the data sets. Finally, the prediction capacities of these models are evaluated. In the remaining of this chapter, three main technical problems regarding the implementation of the machine learning screening I have used are addressed in detail.

2.2 Feature vector construction

The data we have collected are the 3D structures of those inhibitor and non-inhibitor compounds. But most machine learning algorithms cannot accept the chemical structures directly. In order to use different machine learning algorithms to build classifiers, we have to transform the structure information of these compounds into feature vector information of homogeneous length. In SAR or QSAR analysis, molecular descriptors are widely used to represent a wealth of information implicitly encoded in the 2D- and 3D- structures of chemical compounds, such as molecular weight, atom type distribution, polarizability index etc. In this study, a total of 159 molecular descriptors are calculated for all positive and negative examples, which includes 14 simple descriptors, 116 topological descriptors, 13 quantum chemical descriptors and 16 geometrical descriptors. The full list of the 159 descriptors could be found in Table 2.1.

Table 2.1: The full list of 159 descriptors

No.	Name	Description	Reference
Simple descriptors (14 parameters)			
1	W_{mol}	Molecular weight	
2	N_{hyd}	Count of hydrogen atoms	
3	N_{hal}	Count of halogen atoms	
4	N_{het}	Count of hetero atoms	
5	N_{hea}	Count of heavy atoms	
6	N_F	Count of F atoms	
7	N_{Cl}	Count of Cl atoms	
8	N_{Br}	Count of Br atoms	
9	N_I	Count of I atoms	
10	N_C	Count of C atoms	
11	N_P	Count of P atoms	
12	N_S	Count of S atoms	
13	N_O	Count of O atoms	
14	N_N	Count of N atoms	
Topological descriptors (116 parameters)			
15	N_{ring}	Numbers of rings	
16	N_{rot}	Number of rotatable bonds	

No.	Name	Description	Reference
17	N_{donr}	Number of H-bond donors	
18	N_{accr}	Number of H-bond acceptors	
19–21	${}^{0}\chi, {}^{1}\chi, {}^{2}\chi$	Simple molecular connectivity Chi indices for path order 0-2	[26, 27]
22	${}^{3}\chi_p$	Simple molecular connectivity Chi indices for path order 3	
23	${}^{3}\chi_c$	Simple molecular connectivity Chi indices for cluster	
24	${}^{4}\chi_{pc}$	Simple molecular connectivity Chi indices for path/cluster	
25 – 28	${}^{3}\chi_{CH}$ ${}^{4}\chi_{CH}$ ${}^{5}\chi_{CH}$ ${}^{6}\chi_{CH}$	Simple molecular connectivity Chi indices for cycles of 3, 4, 5, and 6 atoms	
29-31	${}^{0}\chi^{\pi}, {}^{1}\chi^{\pi}, {}^{2}\chi^{\pi}$	Valence molecular connectivity Chi indices for path order 0-2	
32	${}^{3}\chi_p^{\pi}$	Valence molecular connectivity Chi indices for path order 3	
33	${}^{3}\chi_c^{\pi}$	Valence molecular connectivity Chi indices for cluster	
34	${}^{3}\chi_{pc}^{\pi}$	Valence molecular connectivity Chi indices for path/cluster	
35 – 38	${}^{3}\chi_{CH}^{\pi}$ ${}^{4}\chi_{CH}^{\pi}$ ${}^{5}\chi_{CH}^{\pi}$ ${}^{6}\chi_{CH}^{\pi}$	valence molecular connectivity Chi indices for cycles of 3, 4, 5, and 6 atoms	
39 – 41	${}^{1}k$ ${}^{2}k$ ${}^{3}k$	Molecular shape Kappa indices for one-three boned fragments	[26]
42 – 44	${}^{1}k_{\alpha}$ ${}^{2}k_{\alpha}$ ${}^{3}k_{\alpha}$	Kappa alpha indices for one-three boned fragments	
45	phi	Kier molecular flexibility index	
46	0^k	Zero order Kappa index	
47	S_{hev}	Sum of electrotopological state (Estate) indices of heavy atoms	[28, 29]
48	S_{car}	Sum of Estate indices of carbon atoms	[30, 31]
49	S_{het}	Sum of Estate indices of hetero atoms	
50	S_{hal}	Sum of Estate indices of halogen atoms	
51	$S(1)$	Atom-type H Estate sum for -OH	
52	$S(2)$	Atom-type H Estate sum for =NH	
53	$S(3)$	Atom-type H Estate sum for -SH	
54	$S(4)$	Atom-type H Estate sum for $-NH_2$	
55	$S(5)$	Atom-type H Estate sum for >NH	
56	$S(6)$	Atom-type H Estate sum for : NH:	
57	$S(7)$	Atom-type H Estate sum for ♯CH(sp)	
58	$S(8)$	Atom-type H Estate sum for $= CH_2(sp^2)$	
59	$S(9)$	Atom-type H Estate sum for =CH–(sp^2)	
60	$S(10)$	Atom-type H Estate sum for :CH:(sp^2, aromatic)	

No.	Name	Description	Reference
61	$S(11)$	Atom-type H Estate sum for $CH_nX(sp^3, X = F, Cl, Br, I)$	
62	$S(12)$	Atom-type H Estate sum for CH_n (Saturated)	[27-30]
63	$S(13)$	Atom-type H Estate sum for CH_n (unsaturated)	
64	$S(14)$	Atom-type H Estate sum for CH_n (aromatic)	
65	$S(15)$	Atom-type H Estate sum for AH_n (not C, N, O, S)	
66	$S(16)$	Atom-type Estate sum for $-CH_3$	
67	$S(17)$	Atom-type Estate sum for $=CH_2$	
68	$S(18)$	Atom-type Estate sum for $>CH_2$	
69	$S(19)$	Atom-type Estate sum for ≡CH	
70	$S(20)$	Atom-type Estate sum for =CH-	
71	$S(21)$	Atom-type Estate sum for : CH : (aromatic)	
72	$S(22)$	Atom-type Estate sum for >CH-	
73	$S(23)$	Atom-type Estate sum for =C=	
74	$S(24)$	Atom-type Estate sum for ≡C-	
75	$S(25)$	Atom-type Estate sum for =C<	
76	$S(26)$	Atom-type Estate sum for : C:-	
77	$S(27)$	Atom-type Estate sum for : C ::	
78	$S(28)$	Atom-type Estate sum for >C<	
79	$S(29)$	Atom-type Estate sum for -NH_2	
80	$S(30)$	Atom-type Estate sum for =NH	
81	$S(31)$	Atom-type Estate sum for >NH	
82	$S(32)$	Atom-type Estate sum for :NH:	
83	$S(33)$	Atom-type Estate sum for ≡N	
84	$S(34)$	Atom-type Estate sum for =N-	
85	$S(35)$	Atom-type Estate sum for :N:	
86	$S(36)$	Atom-type Estate sum for >N-	
87	$S(37)$	Atom-type Estate sum for -N<< (NO_2)	
88	$S(38)$	Atom-type Estate sum for :N:-	
89	$S(39)$	Atom-type Estate sum for -OH	
90	$S(40)$	Atom-type Estate sum for =O	
91	$S(41)$	Atom-type Estate sum for -O-	
92	$S(42)$	Atom-type Estate sum for :O:	
93	$S(43)$	Atom-type Estate sum for -F	
94	$S(44)$	Atom-type Estate sum for -SiH_3	
95	$S(45)$	Atom-type Estate sum for -SiH_2-	
96	$S(46)$	Atom-type Estate sum for >SiH-	
97	$S(47)$	Atom-type Estate sum for >Si<	
98	$S(48)$	Atom-type Estate sum for -PH_2	
99	$S(49)$	Atom-type Estate sum for -PH-	
100	$S(50)$	Atom-type Estate sum for >P-	
101	$S(51)$	Atom-type Estate sum for →P=(P.O)	
102	$S(52)$	Atom-type Estate sum for -=P=($P.O_2$)	

No.	Name	Description	Reference
103	$S(53)$	Atom-type Estate sum for $=PH_3$	
104	$S(54)$	Atom-type Estate sum for -SH	[27-30]
105	$S(55)$	Atom-type Estate sum for =S	
106	$S(56)$	Atom-type Estate sum for -S-	
107	$S(57)$	Atom-type Estate sum for :S:	
108	$S(58)$	Atom-type Estate sum for >S=O	
109	$S(59)$	Atom-type Estate sum for >S<<	
110	$S(60)$	Atom-type Estate sum for -Cl	
111	$S(61)$	Atom-type Estate sum for GeH_3	
112	$S(62)$	Atom-type Estate sum for $-GeH_2-$	
113	$S(63)$	Atom-type Estate sum for >GeH-	
114	$S(64)$	Atom-type Estate sum for >Ge<	
115	$S(65)$	Atom-type Estate sum for $-AsH_2$	
116	$S(66)$	Atom-type Estate sum for -AsH-	
117	$S(67)$	Atom-type Estate sum for >As-	
118	$S(68)$	Atom-type Estate sum for →As=	
119	$S(69)$	Atom-type Estate sum for -SeH	
120	$S(70)$	Atom-type Estate sum for =Se	
121	$S(71)$	Atom-type Estate sum for -Se-	
122	$S(72)$	Atom-type Estate sum for :Se:	
123	$S(73)$	Atom-type Estate sum for >Se=	
124	$S(74)$	Atom-type Estate sum for -=Se=-	
125	$S(75)$	Atom-type Estate sum for -Br	
126	$S(76)$	Atom-type Estate sum for $-SnH_3$	
127	$S(77)$	Atom-type Estate sum for $-SnH_2-$	
128	$S(78)$	Atom-type Estate sum for >SnH-	
129	$S(79)$	Atom-type Estate sum for >Sn<	
130	$S(80)$	Atom-type Estate sum for -I	
Quantum chemical descriptors (13 parameters)			
131	π_i	Polarizability index	[32, 33]
132	ε_a	Hydrogen bond donor acidity (covalent HBDA)	
133	ε_b	Hydrogen bond acceptor basicity (covalent HBAB)	
134	q^+	Atomic charge on the most positively charged H atom	
135	q^-	Largest negative charge on an non-H atom	
136	μ	Molecular dipole moment	
137	η	Absolute hardness	[34, 35]
138	SN	Softness	
139	IP	Ionization potential	
140	A	Electron affinity	
141	μ_{cp}	Chemical potential	
142	χ_{en}	Electronegativity index	
143	ω	Electrophilicity index	

No.	Name	Description	Reference
Geometric descriptors (16 parameters)			
144 – 146	dis1 dis2 dis3	Length vectors (longest distance, longest third atom, 4th atom)	
147	V_{mc}	Van der Waals molecular volume	
148	AS	Solvent accessible surface area	[36]
149	VS	van der Waals surface area	
150	MS	Molecular surface area	
151	PSA	Polar molecular surface area	
152	Sapc	Sum of solvent accessible surface areas of positively charged atoms	
153	Sanc	Sum of solvent accessible surface areas of negatively charged atoms	
154	Sapcw	Sum of charge weighted solvent accessible surface areas of positively charged atoms	
155	Sancw	Sum of charge weighted solvent accessible surface areas of negatively charged atoms	
156	Svpc	Sum of van der Waals surface areas of positively charged atoms	
157	Svnc	Sum of van der Waals surface areas of negatively charged atoms	
158	Svpcw	Sum of charge weighted van der Waals surface areas of positively charged atoms	
159	Svncw	Sum of charge weighted van der Waals surface areas of negatively charged atoms	

2.3 Data preprocessing

While neglected by many machine learning researchers in the area of bioinformatics, preprocessing is regarded as a crucial step for serious real world data mining by the machine learning community. Normally the preprocessing steps include data cleaning (such as management of missing values and replicate handling), normalization, feature selection and et al. After these steps, the processed data sets can be sent to various machine learning tools. The data cleaning will be addressed case by case in following chapters, together with data collection. Here, in this section, normalization and feature selection are discussed.

2.3.1 Normalization

In the application of drug design, the range of features values differ a lot, such as the value of "molecular weight" (the first descriptor in Table 3.1) is hundreds or even thousands greater than "Valence molecular connectivity χ indices for cluster" (the 33th descriptor whose range is in $[0, 5]$ in Table 3.1). Thus "molecular weight" might overpower the "Valence molecular connectivity χ indices for cluster" when the data sets are sent to principal component analysis and distance-based classifier like $k$nn though the facts should not be that. Normalization or scaling is a widely used preprocessing technique for this dilemma. Scaling data values in a range such as $[0, 1]$ or $[-1, 1]$ prevents features with large range outweighing over features with smaller range. Besides scaling may improve the accuracy and efficiency of computation involving distance multiplication or division operations. In this work, the same normalization scheme is adopted as by LIBSVM[37], a famous support vector machine classification toolbox.

$$A_i^{new} = \frac{2\{A_i - [\frac{\max(A_i)-\min(A_i)}{2}]\}}{\max(A_i) - \min(A_i)}; \tag{2.1}$$

where A_i is the i-th feature. After this process, all the features will be in the region of $[-1, 1]$.

2.3.2 Principle Component Analysis

Another very important issue of preprocessing is dimensionality reduction. Bellman first proposed the term "curse of dimensionality" in 1961, which refers to the exponential growth of hyper-volume as a function of dimensionality[95]. Most statistical learning models can be thought of as mappings from an input space to an output space. Thus, loosely speaking, a statistical learning model needs to somehow "monitor", cover or represent every part of its input space in order to know how

that part of the space should be mapped. Covering the input space takes resources, and, in the most general case, the amount of resources needed is proportional to the hyper-volume of the input space. The exact formulation of "resources" and "part of the input space" depends on the type of the model and should probably be based on the concepts of information theory and differential geometry. The curse of dimensionality causes a model with lots of irrelevant inputs to behave relatively badly. When the dimension of the input space is too high, the model uses almost all its resources to represent irrelevant portions of the space. Even if a statistical learning algorithm is able to focus on important portions of the input space, the higher the dimensionality of the input space is, the more examples are needed to make a reasonable sampling.

Dimensionality reduction has been the focus of preprocessing research for quite some time. Principal Component Analysis (PCA) is the widely recognized representative. It constructs a new set of features from the original features to minimize the information loss when discarding any of the original features.

The basic idea in PCA is to construct new components $s_1, s_2, \ldots, s_n$ so that they can explain the maximum amount of variances in the input space by linear transformation. PCA can be defined in an intuitive way using a recursive formulation. Define the direction of the first principal component, say w_1, by

$$w_i = arg \max_{\|w\|=1} E[(w^T \cdot x)^2]; \tag{2.2}$$

where w_1 is of the same dimension as the example vector x. Thus the first principal component is the projection on the direction in which the variance of the projection is maximized. Having determined the first $k-1$ principal components, the k-th principal component is determined as the principal component of the residual:

$$w_k = arg \max_{\|w\|=1} E\{w^T[x - \sum_{i=1}^{k-1} w_i w_i^T x]^2\}; \tag{2.3}$$

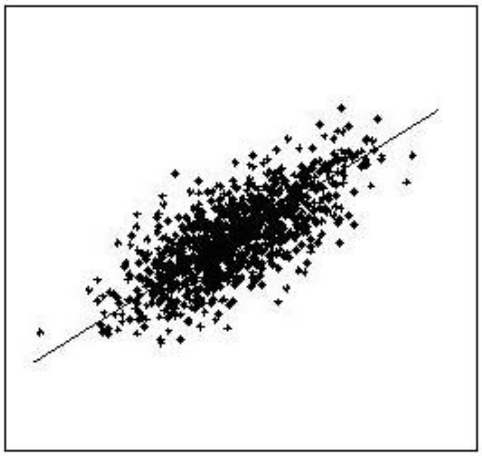

Figure 2.2: Principle component analysis of a 2D data set. The line shown is the direction of the first principal component, which gives an optimal (in the mean-square sense) linear reduction of dimension from 2 to 1.

The principal components are then given by,

$$s_i = w_i^T x; \tag{2.4}$$

In practice, the computation of the w_i can be simply accomplished by using the (example) covariance matrix $E\{x^T x\} = C$. The w_i are the eigenvectors of C that correspond to the k largest eigenvalues of C.

By choosing the first n components, PCA is used to reduce the dimensionality of the input data. One usually chooses $n < N$ (N is the dimension of the original feature vector). It can be proven that the representation given by PCA is an optimal linear dimension reduction technique in the mean-square sense[39]. By this means, noise may be reduced, as the data not contained in the n first components may be mostly due to noise. A simple illustration of PCA is found in Figure 2.2, in which the first principal component of a two-dimensional data set is shown.

2.4 Machine learning theory and algorithms

2.4.1 Philosophy

A little kid can easily tell the apples from a basket of fruit. More than that, he can exclude the rotten pieces. This problem is not very difficult since we human beings (but not restricted to human beings) have the ability of thinking. However automating this process turns out to be fairly complicated since at the current stage we still have not fully understood the mechanism of the brain, and consequently we can not build a mathematical model to simulate the thinking process of the brain on a computer. An alternative strategy for solving this type of problem is to make the computer learn the input/output functionality from examples, in the similar way that children learn which are apples simply by being told which of a large number of fruits are apples rather than by being given the precise definition of apple from encyclopedia. The approach of using examples to train a computer model to learn a specific concept is known as the machine learning or pattern recognition methodology. And in particular, when the examples are input/output pairs it is called supervised learning.

A learning problem with simple yes/no tag like outputs is defined as a binary classification or concept learning problem. There are also multi-class classification problems whose output will be one of a finite number of categories. A case in point is predicting protein families from amino acid sequence. In this work, we mainly focus on the binary classification problems since most multi-class classification problems can be decomposed to several binary classification problems.

Mathematically, when learning the target concept, the learner is given a set of training data X, each consisting of an instance x, along with its target concept value $f(x)$. Instances for whom $f(x) = 1$ are called positive examples, or members

of the target concept. Instances for whom $f(x) = -1$ are called negative examples, or nonmembers of the target concept. Thus the problem faced by the learner is to hypothesize, or estimate the target function f. Let H denote the set of all possible hypotheses that the learner may consider regarding the identity of the target concept. Each hypothesis h in H represents a Boolean-valued function defined over X; that is, $h : X \rightarrow \{-1, 1\}$. The goal of the learner is to find a hypothesis h, such that $h(x) = f(x)$ for all x in X. In short, concept learning can be thought as the searching through a large space of hypotheses implicitly defined by the hypothesis representation. The goal of this search is to find the hypothesis that best fits the training examples.

It is theoretically proved that most machine learning algorithms are capable of representing any function[40]. But for thorny training sets they might give a hypothesis that behaves like a rote learner. That is, the hypothesis correctly classifies the data in the training set, but makes essentially uncorrelated predictions on unseen data. Here comes, however, a more fundamental concern with machine learning algorithms. That is how we can find a hypothesis consistent with not only the training data, but also unseen data. The ability of a hypothesis to correctly classify unseen data is known as the generalization ability. Ockham's razor[40] is a philosophical principle that can give us helpful hint on how to improve the generalization ability of a machine learning model. It suggests that unnecessary complications are not helpful, or perhaps more accurately, complications must pay for themselves by giving significant improvements in the classification rate on the training data. Thus we expect that by minimizing the classification error on training data plus a complexity measure of the hypothesis, the optimal hypothesis could be found.

Over the years, various classification algorithms have been developed by the machine learning community. Representatives of these algorithms are decision tree,

artificial neural network, genetic algorithm, k-nearest neighbor learning, Bayesian learning , support vector machine and etc. Depending on the characteristics of the data sets being classified, certain algorithms tend to perform better than others. In recent years, algorithms based on the support vector machine and the k–nearest neighbors have been shown to produce reasonably good results for problems whose features are continuous. For this reason, we are mainly interested in these two algorithms. We also include the decision tree algorithm, which is the classical benchmark for classification algorithms and can be applied universally. These algorithms are described briefly in the following sections.

2.4.2 Decision tree

Decision tree is a learning method for approximating discrete-valued target functions, in which the learned function is represented by a tree with branches. Learned trees can also be re-represented as sets of if-then rules to improve human readability. These learning methods are among the most popular inductive inference algorithms and have been successfully applied to a broad range of tasks from diagnosing medical cases to assessing credit risk of loan applicants.

Decision trees classify instances by sorting them down the tree from the root to some leaf node. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch according to the value of the attribute of the given example. This process is then repeated for subtree rooted at the new node. Figure 2.3 gives an illustration of a typical learned decision tree.

Currently many softwares have been developed for learning decision trees, like

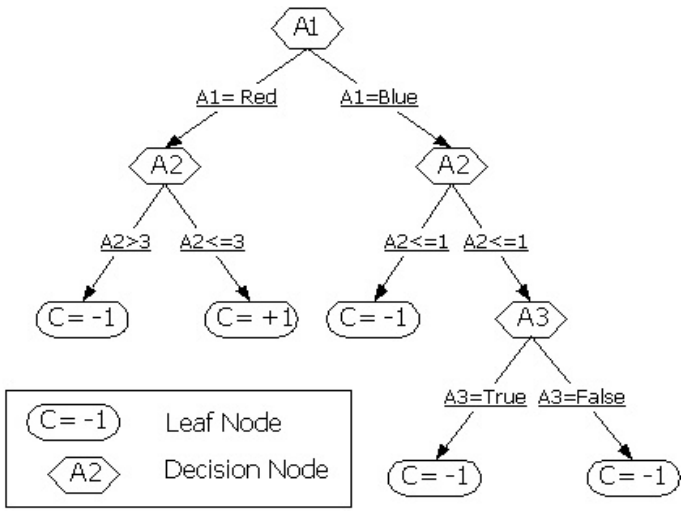

Figure 2.3: Illustration of the decision tree

ID3[41], CART[42] and C4.5[43]. Most of them are variations based on a core algorithm that employs a top-down, greedy search through the space of possible decision trees. Table 2.1 gives the basic algorithm for decision tree learning.

Table 2.2: Decision Tree Algorithm Chart

To construct a decision tree T from learning data set D:

If D contains one or more examples, all of which belongs to a single class C, stop.

If D contains no example, the most frequent class at the parent of this node is chosen as the class, stop.

Otherwise begin

1. Select the "most informative" attribute A
2. Partition D according to $A's$ value
3. Recursively construct subtrees T_1, T_2,..., for the subset of D.

The primary focus of the decision tree growing algorithm is to select which attribute to test at which decision node in the tree. Here some technical terms from

information theory, namely "information gain", "gain ratio" are borrowed as the quantitative measure of how well a given attribute separates the training examples.

In order to define information gain precisely, we need to first introduce another measure parameter—entropy, which characterizes the impurity of an arbitrary collection of examples. Given a training data set D, in the binary classification setting, the entropy of set D is defined as:

$$Entropy(D) = -p_{\oplus}\log_2(p_{\oplus}) - p_{\ominus}\log_2(p_{\ominus}); \tag{2.5}$$

where $p_{\oplus}$ is the proportion of positive examples in D and $p_{\ominus}$ is the proportion of negative examples in D. In all calculations involving entropy we define $0\log_2(0)$ to be 0. Thus the entropy is 1 (at its maximum value) when the collection contains equal numbers of positive and negative examples, and the entropy is 0 if all members of D belong to only one class, which is the stop criteria of tree splitting. More frequently, the entropy is between 0 and 1, since the collection contains unequal numbers of positive and negative examples.

After that, we are able to define the "information gain", a measure of the effectiveness of an attribute in classifying the training examples. It is simply the expected reduction in entropy caused by partitioning the examples according to this attribute. Mathematically, the information gain $Gain(D, A)$ of an attribute A, relative to a collection of examples D, is defined as

$$Gain(D, A) = Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} Entropy(D_v); \tag{2.6}$$

where $Values(A)$ is the set of all possible discrete values for attribute A, and D_v is the subset of D in which attribute A has the value v. Note the first term in the equation (2.2) is just the entropy of the original collection D and the second term is the expected value of the entropy after D is partitioned using attribute A. The expected entropy described by this second term is simply the sum of the entropies

of each subset D_v, weighted by the fraction of examples $|D_v|/|D|$ that belongs to D_v. $Gain(D, A)$ is therefore the expected reduction in entropy caused by knowing the value of attribute A.

The measure of information gain tends to favor those attributes with more possible discrete values. For example, a decision tree can be established to predict the disease of a patient using only one attribute: the case serial number. However, such a decision tree would probably fail when a new patient with a new case serial number comes. Another measurement, Gain Ratio is defined to avoid this bias, which can be calculated as follows.

$$GainRatio(D, A) = Gain(D, A)/SplitInformation(D, A); \tag{2.7}$$

and

$$SplitInformation(D, A) = -\sum_{v \in Values(A)} \frac{|D_v|}{|D|} \log_2 \frac{|D_v|}{|D|} \tag{2.8}$$

$SplitInformation(D, A)$ discourages the selection of attributes with many values. Therefore, Gain Ratio is the ratio of information gained that is pertinent to classification by branching on A.

The initial definition of decision tree is restricted to attributes that take on discrete values. To incorporate continuous-valued attributes into the learned tree, we can partition the continuous attribute values into a discrete set of intervals. For instance, for an attribute like temperature that has continuous values, the ID3 algorithm can dynamically create a new Boolean attribute A' whose value is low if the temperature is below 20^oC and high otherwise. The only question is how to select the best value for the threshold. Intuitively, we would like to pick a threshold that produces the greatest information gain. By sorting the examples according to the continuous attribute A, then identifying adjacent examples that differ in their class labels, we can generate a set of candidate thresholds midway between the

corresponding values of A. It can be proven that the value of the threshold that maximizes information gain always lie at such a boundary. These candidate thresholds can then be evaluated by computing the information gain associated. This dynamically created Boolean attribute can then compete with the other discrete–valued candidate attributes available for growing the decision tree.

In principle, the above decision tree algorithm can be used to grow a tree with as many branches as to perfectly classify all the training examples. While this is sometimes a reasonable strategy, in most cases it leads to difficulties when there is noise in the data, or when the size of the training data set is too small to give a representative sample of the real-world problem. In either of these cases, this simple algorithm will produce trees that over-fit the training examples. Therefore, after the tree is constructed, a pruning process is applied to gradually remove decision nodes that give the least improvements on accuracy and assign to these nodes the class label of the majority of remaining examples. In this case, the prune level will be a free parameter to be optimized in the decision tree induction, which controls the complexity of the tree.

2.4.3 K-Nearest Neighbor

K-Nearest Neighbor ($k$nn) is a well known and widely used instance-based classification algorithm due to its conceptual simplicity, general applicability and efficiency. Learning in $k$nn consists of simply storing the presented training data. When a new query instance is encountered, a set of similar related instances is retrieved from memory and used to classify the new query.

The basic idea behind this classification paradigm is first to compute the similarity between a query instance and all the examples in the training data set, then select the k most similar training examples, and finally to determine the class label

of the query instance based on the class labels of its k nearest neighbors.

Two steps are critical to the performance of the k-Nearest Neighbor. The first is how to measure the similarity between a query instance and a training example. For data sets in which the examples are represented by multi-dimensional vectors, like our application, the measurement commonly used to compute the similarity is the Euclidean distance. More precisely, suppose an arbitrary instance x can be described by the feature vector $\{a_1(x), a_2(x), \ldots, a_n(x)\}$, where $a_r(x)$ denotes the value of the rth attribute of instance $x, r = 1, 2, \ldots, n$. Then the distance between two instances x_i and x_j is defined as $d(x_i, x_j)$, where

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} [a_r(x_i) - a_r(x_j)]^2} \tag{2.9}$$

The second problem is how to determine the class label of the query instance based on the class labels of its nearest neighbors. The most straightforward strategy is that the minority should be subordinate to the majority. In other words, the query instance will be classified to the class to which most of the k-nearest neighbors belong. In the case of two-class problems, this decision function can be expressed as:

$$C = sign\{\sum_{i=1}^{k} f(x_i)\}; \tag{2.10}$$

where $f(x_i)$ is the class label of the ith nearest neighbor x_i, either $+1$ or -1.

In the k-Nearest Neighbor learning, the value of k is required. It has been found that $k < \sqrt{N}$ is a general criterion that should be met for good results, where N is the total number of training examples. Therefore, the number of effective nearest neighbors k will be a free parameter in the k-nearest neighbor algorithm to be optimized according to test results.

However, this $k$nn algorithm is incompetent for the problems showed in Figure 2.4. On the left, the 1-nearest neighbor algorithm classifies x_q positive, whereas 5-nearest neighbor classifies it as negative. On the right, the 4-nearest neighbor

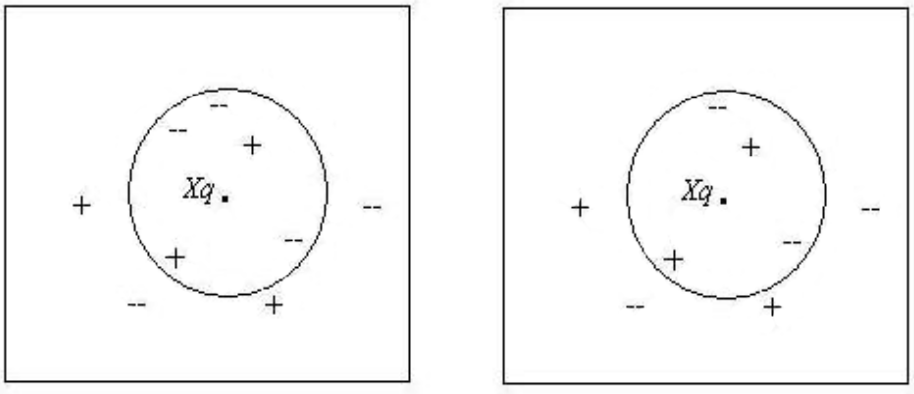

Figure 2.4: Illustration of k-nearest neighbor

algorithm will not be able to assign the query instance a label based on its 4-nearest neighbors which are two positive and two negative.

One feasible refinement is to weight the contribution of each of the k neighbors according to their distance to the query point x_q, giving greater weight to closer neighbors. Thus, we define

$$w_i = \frac{1}{d(x_q, x_i)^2}; \tag{2.11}$$

Now we can re-write equation (2.10) as below:

$$C = sign\{\frac{\sum_{i=1}^{k}[w_i f(x_i)]}{\sum_{i=1}^{k} w_i}\}; \tag{2.12}$$

The distance-weighted k-Nearest Neighbor algorithm is a highly effective inductive inference method for many practical problems. It is robust to noisy training data and quite effective when it is provided a sufficiently large set of training data. Note that by taking the weighted average of the k neighbors nearest to the query point, it can smooth out the impact of isolated noisy training examples.

2.4.4 Support vector machine

Support vector machine was first introduced in 1995 by Vapnik and his co-workers[44]. Then the research on both SVM theory and application has exploded in the last decade. Now it has become one of the most powerful statistical learning methods and has outperformed many other machine learning methods in a wide variety of applications, such as text categorization[45, 46], hand-written digit recognition[47], image classification and object detection[48, 49], flood stage forecasting[50], micro-array gene data analysis[51], protein folding recognition[52], protein secondary structure prediction[53], protein-protein interaction prediction[54] and etc.

The main idea of support vector machine is to construct a hyperplane in a high-dimensional space as the decision surface in such a way that the margin of separation between positive and negative examples is maximized. In order to find such an optimal hyperplane, special "kernels" are introduced to help automatically conduct nonlinear mapping from the input space onto a feature space in which the training examples can be linearly separated.

Let us consider a training data set consisting of n examples. Each example is denoted as $(\mathbf{x_i}, y_i), i = 1, 2, \ldots, n$. where $y_i \in \{-1, +1\}$ is the corresponding target output. Given a weight vector $\mathbf{w}$ and a bias b (See Figure 2.5), it is assumed that these examples can be separated by a hyperplane with a margin of γ:

$$\mathbf{w}^T \cdot \mathbf{x_i} + b \geq +1, \qquad \text{for} \qquad y_i = +1; \tag{2.13}$$

$$\mathbf{w}^T \cdot \mathbf{x_i} + b \leq -1, \qquad \text{for} \qquad y_i = -1; \tag{2.14}$$

Equation (2.13) and (2.14) can be combined into a single inequality:

$$y_i(\mathbf{w}^T \cdot \mathbf{x_i} + b) \geq 1, \qquad \text{for} \qquad i = 1, 2, \ldots, n; \tag{2.15}$$

The objective of SVM is to determine the optimal weight $\mathbf{w_o}$ and optimal bias b_o

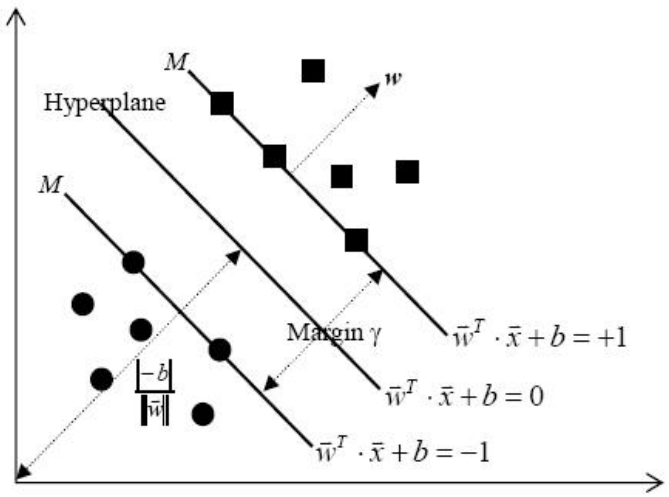

Figure 2.5: Definition of hyperplane and margin (*The square dots and circular dots represents samples of class +1 and class -1, respectively*)

so that the corresponding hyperplane, namely the Optimal Separating Hyperplane (OSH), separates the positive and negative training data with maximum margin, which is expected to produce the best generalization performance (See Figure 2.6).

The width of the two corresponding margins is

$$\gamma(\mathbf{w}, b) = \min_{\{\mathbf{x}|y=+1\}} \frac{\mathbf{w}^T \cdot \mathbf{x}}{\| \mathbf{w} \|} - \max_{\{\mathbf{x}|y=-1\}} \frac{\mathbf{w}^T \cdot \mathbf{x}}{\| \mathbf{w} \|} \tag{2.16}$$

Given the constraint of equation (2.15), one obtains

$$\gamma_{max} = \gamma(\mathbf{w_o}, b_o) = \frac{2}{\| \mathbf{w_o} \|} \tag{2.17}$$

This is equal to minimize

$$\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}, \qquad \text{for} \qquad i = 1, 2, \ldots, n; \tag{2.18}$$

subject to $y_i(\mathbf{w}^T \cdot \mathbf{x_i} + b) \geq 1$.

We may solve the constrained optimization problem using the method of Lagrange multiplier:

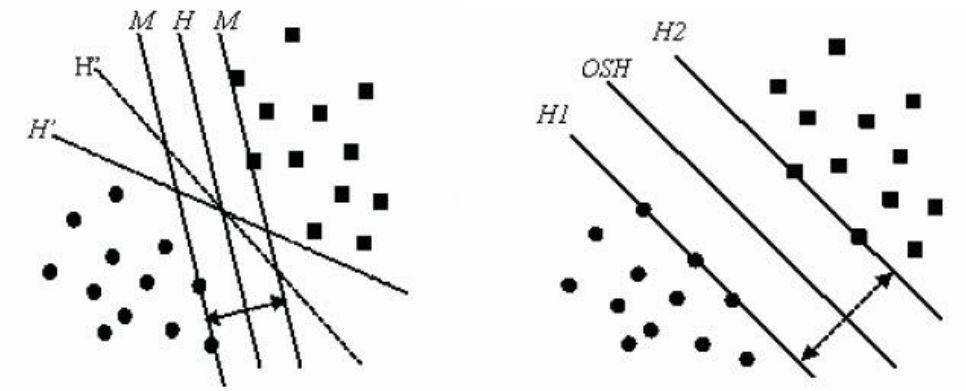

Figure 2.6: Available separating hyperplanes and Optimal Separating Hyperplane

First, we construct the primal Lagrangian function:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \alpha_i[y_i(\mathbf{w}^T \cdot \mathbf{x_i} + b) - 1]; \tag{2.19}$$

where the auxiliary nonnegative optimization problem is determined by the saddle point of the function $L(\mathbf{w}, b, \alpha)$, which has to be minimized with respect to $\mathbf{w}$ and b. By setting $\frac{\partial L}{\partial \mathbf{w}}|_{\mathbf{w}=\mathbf{w_o}} = 0$ and $\frac{\partial L}{\partial b}|_{b=b_o} = 0$, we get the following conditions:

$$\mathbf{w_o} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x_i}; \tag{2.20}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0; \tag{2.21}$$

By taking equation (2.20) and (2.21) into equation (2.19), we will maximize the following formula;

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j (\mathbf{x_i} \cdot \mathbf{x_j}) \tag{2.22}$$

subject to the constraints $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $\alpha_i \geq 0$ for $i = 1, 2, \ldots, n$.

There are standard algorithms like sequential minimization optimization[55] and decomposition algorithm[56] to solve this optimization problem.

The examples that have positive coefficients α_i satisfy the condition of $y_i(\mathbf{w}\cdot\mathbf{x_i}+b)=1$, and thus are located on the margin hyperplanes. They form the boundary of the margin and are called Support Vectors (SVs).

The bias b_o can be calculated as follows:

$$b_o = -\frac{1}{2}[\min_{\{\mathbf{x_i}|y=+1\}}(\mathbf{w_o}^T\cdot\mathbf{x_i}) + \max_{\{\mathbf{x_i}|y=-1\}}(\mathbf{w_o}^T\cdot\mathbf{x_i})]; \tag{2.23}$$

After determination of Support Vectors and bias, the decision function that separates the two classes can be written as:

$$f(x) = sign[\sum_{i=1}^{n}\alpha_i y_i(\mathbf{x_i}^T\cdot\mathbf{x}) + b_o] = sign[\sum_{SV}\alpha_i y_i(\mathbf{x_i}^T\cdot\mathbf{x}) + b_o]; \tag{2.24}$$

The linear classifier based on the above scheme can be easy to handle, but they pose severe restrictions on the learning task. The target concept may be too complex to be expressed as a linear combination of the given attributes. This problem can be overcome by an approach called kernel technique[1]. Its general idea is to map the training data set X from the input space into a high-dimensional feature space F via a Mercer kernel[2] operator K and separate it there by a linear classifier. This will result in a nonlinear classifier in input space.

Mathematically, let Φ denote an implicit mapping function from the input space to the feature space F. Then all the above equations are transformed into the following form when we substitute $\mathbf{x_i}$ and the inner product in input space $(\mathbf{x_i}\cdot\mathbf{x})$ by $\Phi(\mathbf{x_i})$ and inner-product kernel $K(\mathbf{x_i},\mathbf{x})$ respectively.

Here the Kernel function can be expressed as a legitimate inner product in a feature space:

$$K(\mathbf{x_i},\mathbf{x}) = [\Phi(\mathbf{x_i})\cdot\Phi(\mathbf{x})]; \tag{2.25}$$

[1]Soft margin is another possible approach which will be found in many references.

[2]Mercer's theorem states that any positive definite kernel $K(x,y)$ can be expressed as a dot product in a high-dimensional space.

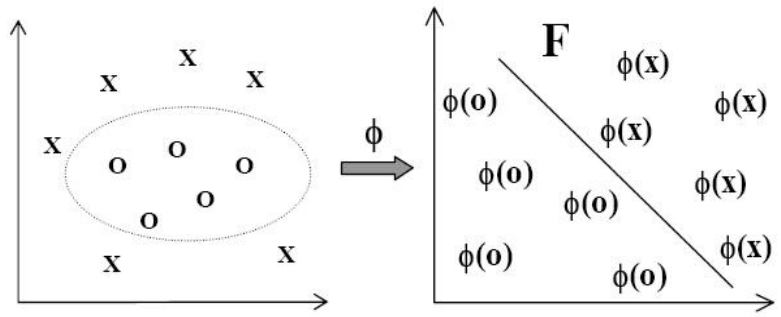

Figure 2.7: Projecting the training data nonlinearly into a higher-dimensional feature space and constructing a hyperplane to separate positive and negative data with maximum margin there.

With this expansion at hand, we may now rewrite equation (2.22) as following,

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}); \tag{2.26}$$

subjects to the constraints: $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $\alpha_i \geq 0$, for $i = 1, 2, \ldots, n$.

Consequently the decision function changes to be

$$f(x) = sign[\sum_{SV} \alpha_i y_i K(\mathbf{x_i}, \mathbf{x}) + b_o]; \tag{2.27}$$

and

$$b_o = -\frac{1}{2}\{\min_{\{\mathbf{x_i}|y=+1\}}[\sum_{SV} \alpha_i y_i K(\mathbf{x_i}, \mathbf{x})] + \max_{\{\mathbf{x_i}|y=-1\}}[\sum_{SV} \alpha_i y_i K(\mathbf{x_i}, \mathbf{x})]\}; \tag{2.28}$$

In Table 2.3 we summarize the most prevalent kernel functions.

2.5 Performance analysis and evaluation

In addition to preprocessing, another important issue of machine learning is the performance analysis and evaluation. We might describe the step as an evaluation

Table 2.3: Summary of Inner-Product Kernels

Kernel Name	Kernel Function $K(\mathbf{x_i}, \mathbf{x}), i = 1, 2, \ldots, n$	Comments
Polynomial Kernel	$(\mathbf{x}^T\mathbf{x_i} + 1)^p$	Power p is specified a priori by the user.
Radial-basis Kernel (Gaussian Kernel)	$\exp\left(-\frac{\|\mathbf{x_i}-\mathbf{x}\|^2}{2\sigma^2}\right)$	The width σ^2, common to all the kernels, is specified a priori by the user
Two-layer perceptron	$\tanh(\beta_0\mathbf{x}^T\mathbf{x_i} + \beta_1)$	Mercer's theorem is satisfied only for some values of β_0 and β_1

of the performance of a proposed solution to the concept learning task. Most commonly used error measurement indicators and error estimating strategies (mainly independent testing data set) are introduced in this section.

2.5.1 Training versus Testing

In Figure 2.8 (the left two rectangles), we see two views of collected data. The first one has one data set that contains all available data. The second view has two samples, the larger one for training and the smaller one for evaluation.

In most applications, we prefer the second strategy. This is because even though it is very easy to get exceptional results when we solely rely on training data, these results are likely not generalize to new examples. Researchers have developed many training techniques that reduce the likelihood of "overfitting" to the training data. An effective way is to hide some data and then do a fair comparison of training results to unseen test results. Of course, one could wait until new data arrives

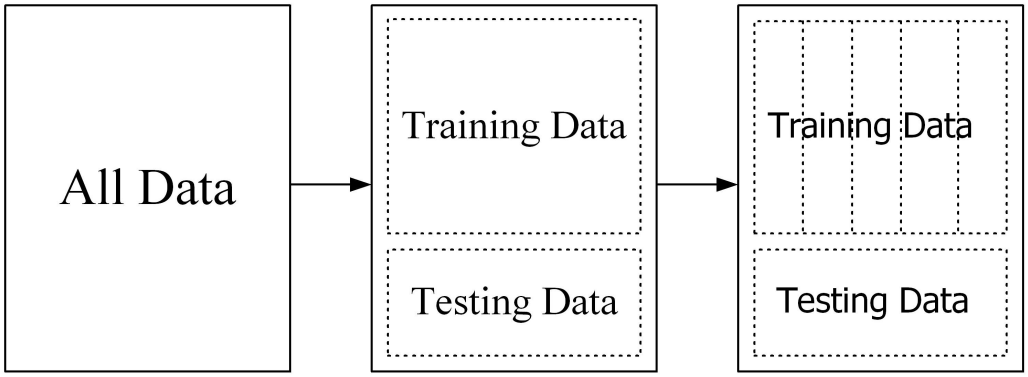

Figure 2.8: Data organization for empirical evaluation

during application of the solution, but it is wise to test performance prior to actual application. It prevents unexpected poor results and gives the developers time to extract the best performance from the application system.

Usually in machine learning algorithms, free parameters need to be tuned to obtain a good classification model, such as the k in $k$nn and the Gaussian kernel width σ used in SVM. In this work, in order to find the optimal parameters for each algorithm, I apply 10-fold cross validation to training sets. Then a model is trained using all the training data with the parameters selected. This procedure ensures that the testing data are never seen during the model training stage.

K-fold cross validation is a model evaluation method that is widely believed to be better than residuals. The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the testing set and the other $k-1$ subsets are put together to form a training set. Then the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a testing set exactly once, and gets to be in a training set $k-1$ times. The disadvantage of this method is that the training algorithm has to be rerun from scratch k times,

which means it takes k times as much computation to make an evaluation. The variance of the resulting estimate is reduced as k is increased. Ron Kohavi suggests that for many real-world datasets, the best method to use is 10-fold stratified cross validation, even if the computation power allow using more folds[58].

2.5.2 Measuring error

For classification problem, we say the measurement of performance is the measure of error. If we confine our goal to measuring an overall rate of error, we can readily reverse the computation and speak in terms of accuracy. However, many times our interest is not just in overall performance. Instead, as we know that a predictor yields different types of errors, our attention will frequently focus on the specific breakdown of error, where not all errors are treated equally important. Different types of errors and how they influence our interpretation of performance are discussed as follows.

2.5.2.1 Confusion metrics

When talking about performance evaluation, people firstly think of Confusion Matrix, which is the most simple and informative way to analyze the behavior of a classifier. It contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix. The following table shows the confusion matrix for a 2-class classifier. TP is the number of correctly classified positive examples. TN is the number of correctly classified negative examples. FP is the number of negative examples which are predicted positive. FN is the number of positive examples which are predicted negative.

Table 2.4: A confusion matrix for binary classification

		Actual Labels	
		Positive	Negative
Predicted Labels	Positive	TP	FP
	Negative	FN	TN

2.5.2.2 Precision, Recall, and the F Measure

Although confusion matrix gives a very informative method to evaluate the classification performance, it is not a convenient measurement that can be used to compare two or more models and tell which one is better than the others. For this purpose, single-value measurement need to be defined.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}; \tag{2.29}$$

For drug discovery applications, there is usually a large number of negative data. A classifier can achieve a very high accuracy by simply saying that all data are negative. It is thus useful to measure the classification performance by ignoring correctly predicted negative data. Three ratios have achieved particular prominence: precision, recall and F measure.

$$Precision = \frac{TP}{TP + FP}; \tag{2.30}$$

$$Recall = \frac{TP}{TP + FN}; \tag{2.31}$$

$$Fmeasure = \frac{2 \times Precision \times Recall}{Precision + Recall}; \tag{2.32}$$

An alternative way of evaluating a model's performance is sensitivity [3] and specificity. I also include them as a set of measurement in this work since they are widely used in clinical research.

$$Sensitivity = \frac{TP}{TP + FN}; \tag{2.33}$$

$$Specificity = \frac{TN}{TN + FP}; \tag{2.34}$$

2.6 Summary

This chapter presents some issues regarding the practical implementation that machine learning approaches face. They are very important to the success of a machine learning model. Key points to remember include:

- Data preparation is the art of wringing the most valuable information out of the available data whereas data mining is the art of discovering meaningful patterns in data.
- Molecular descriptors are borrowed from SAR and QSAR to encode compound structure information numerically.
- Preprocessing techniques, such as normalization and PCA, can influence the performance of many data mining algorithms.
- Machine learning addresses the question of how to build computer models that improve their performance through experience for tasks in different fields. Algorithms like decision tree, k-nearest neighbor and support vector machine draw on ideas from a diverse set of discipline, including artificial intelligence, probability and statistics, computational complexity, information theory, control theory and philosophy.

[3] Sensitivity is exactly the same as Recall, defined by Equation 2.31.

- To evaluate the performance of different machine learning algorithms, two data sets are needed: training set and testing set. And 10-fold cross-validation is recommended in training stage.

In order to evaluate different classification and preprocessing techniques, an efficient tool to implement different algorithms is needed. Here in this work, I choose MatLab as the programming environment since the matrix operation provided by MatLab makes the representation of numerical data and implementation of the different algorithms much easier. Most programs including normalization, k-nearest neighbor and SVM are implemented with MatLab R13 licensed from NUS. Whilst decision tree program I use is C4.5 by Quinlan[43], which is downloadable from http://www.cse.unsw.edu.au/∼quinlan/.

The support vector machine algorithm was implemented with a Gaussian kernel. This is because the Gaussian kernel always performs better than others in our previous studies of protein function classification[59, 60].

All the experiments were carried on a Dell Optiplex GX240 computer with one 2.4GHz Intel Pentium IV CPU and 512M memory.

Chapter 3

Antagonist prediction of the therapeutic target–5-HT_2 receptor

In this chapter, the three machine learning methods described in Chapter 2, namely decision tree, kNN and SVM are applied to the antagonist prediction of the therapeutic target, 5-hydroxytryptamine 2 (5-HT_2) receptor. The most potent approach is identified based on the the experimental results of the 5-HT_2 receptor antagonist data.

In section 3.1 the biology and pharmacology of 5-HT_2 receptor and 5-HT_2 receptor antagonism are introduced. It is followed by the detailed description of data preparation procedure and experimental design. The results obtained are then presented and discussed in section 3.3. In section 3.4, a summary is given.

3.1 Introduction

3.1.1 5-HT and 5-HT receptor subtypes

Serotonin (5-hydroxytryptamine, 5-HT) is a major neurotransmitter in the brain. It mediates a wide range of physiological functions, such as anxiety, depression,

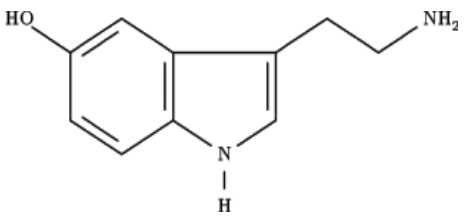

Figure 3.1: The chemical structure of serotonin

schizophrenia, drug abuse, sleep, dreaming, hallucinogenic activity, headache, cardiovascular disorders and appetite control, by interacting with multiple receptors. Seven distinct families of 5-HT receptors have been identified (5-HT_1—5-HT_7), and subpopulation have been described for several of these families[64] (See Table 3.1).

The profusion of 5-HT receptors should eventually allow a better understanding of the different and complex processes in which 5-HT is involved. In this thesis, I am primarily interested in 5-HT_2 receptors. A brief introduction of 5-HT_2 receptors, their antagonists[1] and clinical implication is given below.

3.1.2 5-HT_2 receptors and their antagonism

Serotonin receptors were firstly divided into 5-HT_1 and 5-HT_2 receptor families in 1979[62] and the latter were subsequently divided into subfamilies 5-HT_{2A}, 5-HT_{2B}, and 5-HT_{2C} (formerly 5-HT_{1C}) receptors. There is approximately 70-80% sequence homology among the three receptor subtypes[63]. Many of the original agents thought to be 5-HT_2 selective, including standard antagonists such as ketanserin and the agonists DOB and DOI, were later shown to bind nonselectively to

[1]Agonist is a chemical that binds to a receptor and activates it, producing a pharmacological response (e.g. contraction, relaxation, secretion, enzyme activation, etc.). Antagonist is a chemical that binds to a receptor and blocks it, producing no response but preventing agonists from binding, or attaching, to the receptor. It is similar to inhibitor functionally. Inhibitor is a chemical compound that has the effect of blocking or slowing an enzyme.

Table 3.1: Overview of 5-HT receptor subtypes

G protein-coupled receptors	
5-HT_1 "Family":	5-HT_{1A}, 5-HT_{1B}, 5-HT_{1D}, 5-HT_{1E}, 5-HT_{1F}
	5-HT_{dro2A}, 5-HT_{dro2B}, 5-HT_{snail}
5-HT_7 "Family":	5-HT_7, 5-HT_{drol}
5-HT_5 "Family":	5-HT_{5A}, 5-HT_{5B}
5-HT_2 "Family":	5-HT_{2A}, 5-HT_{2B}, 5-HT_{2C}
5-HT_6 "Family":	5-HT_6
5-HT_4 "Family":	5-HT_{4S}, 5-HT_{4L}
Ligand-gated ion channels	
5-HT_3	
Transporters	
5-HT uptake site	

both 5-HT_{2A} and 5-HT_{2C} receptors.

It is still not known with confidence specifically what pharmacological effects are related to what 5-HT_2 subpopulation. But results with the newer agents indicate that 5-HT_{2A} receptors might be involved in psychosis, depression, and hallucinogenic activity[64, 65, 70], and that 5-HT_{2C} receptors may play a role in obsessive-compulsive disorders, panic, anxiety, and depression[66]. In the periphery, 5-HT_{2B} receptors seem to be involved in muscle contraction[67, 68]; however their function in the CNS is still a matter of speculation. Based on some preliminary studies, and on their central distribution in brain, it was suggested that 5-HT_{2B} receptors might be involved in anxiety, cognition, food intake, neuroendocrine regulation, locomotor coordination, and balance[69]. Several novel approaches may assist further elucidating the roles of these subpopulations and the developing of site selective agents. For example, site directed mutagenesis and synthesis of chimeric receptors[70, 71], coupled with the use of molecular graphics modeling studies[70, 72], are beginning to identify what portions of the receptors are important for ligand binding.

In this work, I do not distinguish between the subpopulations of the 5-HT_2

family and do antagonist prediction for the whole 5-HT_2 receptor family since 5-HT_{2A}, 5-HT_{2B} and 5-HT_{2C} receptors are so closely related, making it rather difficult to design agents selective for any one of the subpopulations[73]. This is also due to the fact that the nomenclature scattered in different references is quite confusing and not unified. In addition, from the perspective of machine learning algorithms, the number of selective antagonists for a particular subpopulation may not be enough to train an accurate model; while combining three subpopulations together may be better.

Various compounds, which have antagonistic properties on 5-HT_2 receptor, are currently available in markets. For instance, clozapine is a prevalent drug in treatment of schizophrenia. Besides, a lot of other $5HT_2$ antagonists are currently under development and some of them have already entered the clinical trial stage, such as zipraisdone (by *Pfizer*), Eplivanserin (by *Sanofi-Synthélabo*) and Org-5222 (by *Organon Laboratories*). New $5HT_2$ antagonists are expected to help in developing drugs of high efficacy, low toxicity and personalized treatments for physiologic disorders.

3.2 Data preparation

In order to train a model that can predict whether a molecular structure represents a potential 5-HT_2 receptor antagonist or not, examples of both classes, compounds known to be 5-HT_2 receptor antagonists (positive examples) and compounds not known to be 5-HT_2 receptor antagonists (negative examples) are needed. While the former is readily available in many references, it is much harder to come up with negative examples. My approach is to take the well-established inhibitors/antagonists of other proteins which are dissimilar to 5-HT_2 receptor in both structure and function as representatives of negative examples. Although some of them might also

have potential inhibitory effect on 5-HT_2 receptor, it is reasonable to assume that the chances are very slim for these well-studied inhibitors of other irrelevant proteins to be 5-HT_2 receptor antagonists. Therefore it is safe to say that such a data collection method is able to represent the real situation.

In this work, 106 different 5-HT_2 receptor antagonists were collected from various references in PubMed. 1366 negative examples were collected from the inhibitors/antagonists of a group of "negative proteins", which are proteins that are not functionally and structurally related to 5-HT_2 receptor. The approach I used to obtain the group of "negative proteins" is the same as the approaches described by Cai et al[75] described for generating negative protein data sets for protein function classification.

Specifically, a total of 1097 proteins from TTD database[23] are selected as "negative proteins", expect for dopaminergic, histaminergic and adrenergic neurotransmitter receptors and other serotonin receptors. This is because these proteins are very similar to 5-HT_2 receptor. The ligands binding to these receptors and those binding to 5-HT_2 overlap to a certain extent. For example, spiperone (See Figure 3.2) which has been employed as a 5-HT_2 antagonist, is also a dopamine antagonist, a 5-HT_{1A} antagonist and a 5-HT_7 antagonist[74]. These proteins (like the dopamine receptor, 5-HT_{1A} and 5-HT_7 in this example) must be excluded from the "negative proteins", otherwise it will lead to confusion or conflict during the machine learning model training time.

The 2D structures of the 5-HT_2 antagonists and "non-" 5-HT_2 antagonists are found in ChemIDPlus database online. To calculate the 159 descriptors for each structure, the 2D structures are converted into 3D structures using Concord 4.0.1[2]

[2]CONCORD^{TM} is a product of Tripos. It sets the industry standard for extremely rapid conversion of 2D (or even crude 3D) input to accurate, geometry-optimized 3D structures. Currently it is most often used for the conversion of large corporate and commercial databases worldwide.

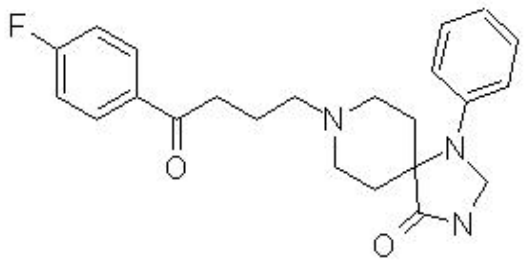

Figure 3.2: The chemical structure of spiperone, an antagonist for multiple receptors.

licensed from NUS.

The structures of the compounds of the two classes were cleaned in a way similar to the one described by Sadowski J. et al[76].

- Unsuitable compounds were removed (e.g. acid halides, anhydrides, metal containing compounds, compounds with a molecular weight below 150 or above 1000, etc.);
- Counterions and solvent molecules were removed;
- Duplicates within each class were removed;
- Compounds shared by both class were removed from negative examples.

Final data include 106 antagonists and 1272 "non-" antagonists of 5-HT_2 receptor.

These compounds are further separated into two sets: training and testing sets. The training set is used by different machine learning methods to develop a statistical model and the testing set is used to evaluate the classification performance of the model. The split method used here was described by R. W. Kennard and L. A. Stone[77]. The ratio is set to roughly 8 : 2. For algorithms like SVM and k-NN,

free parameters need to be tuned. Therefore 10-fold cross validation is performed during training to help find the optimal free parameters. For decision tree, since the software I used is C4.5 package developed by Quinlan et al[43], the parameter tuning is implemented inside. Hence I used all the data in the training set without cross-validation to train a model. This is to ensure fair comparison of the performance which is based on independent testing set.

Table 3.2: The training and testing data sets

Data Set	No. of positive examples	No. of negative examples	Total No.
Training set	85	1018	1103
Testing set	21	254	275
Total No.	106	1272	1378

Now that the data have been collected and split into training and testing set, these data are further processed by scaling and PCA techniques as discussed in Chapter 2. Figure 3.3 shows the PCA result of the training data. From it, we can see that without loss of any variances encoded in the data set, we can reduce 46 redundant dimensions out of 159 dimensions. Keeping 99% of the variances, 101 dimensions can be removed. Keeping 90% of the variances, 136 dimensions can be removed. It is generally believed that the low variances dimensions are highly possible to be dimensions of noises. Therefore the PCA dimensionality reduction are expected to shorten the calculation time and improve the signal-to-noise ratio in data. Taking the noise of data into consideration, I choose to construct new training data sets with the first n principal components representing 90%, 99% and 100% of variances which correspond to $n = 23$, $n = 58$, and $n = 113$ respectively.

To assess the prediction capability of different algorithms, the testing data shall

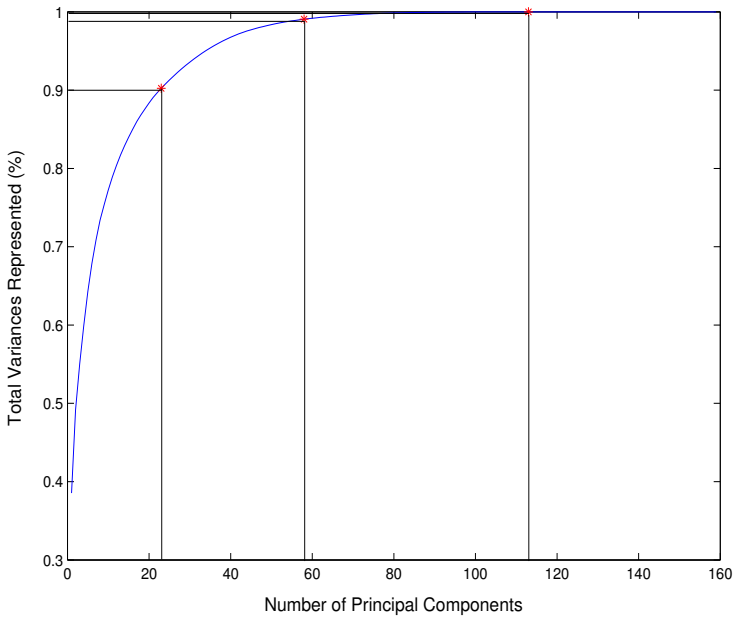

Figure 3.3: The number of principal components VS the percentage of total variance they represent.

undergo exactly the same transformation as training data.

3.3 Prediction results and analysis

The preprocessed data are then subjected to different machine learning analysis. The results are discussed in the following subsections.

3.3.1 Decision tree prediction

The unpruned decision tree C4.5 generated for original data is shown in Figure 3.4. The feature $x86$, $x134, \ldots$, correspond to the $86th$ descriptor, $134th$ descriptor,..., in Table 2.1. A branch node is like $\{x134 <= -0.43652 :\}$. It means the data falling into this category are still messy and need to be further branched. A decision leaf is like $\{x93 > -0.057129 : \quad 1 \quad (2)\}$. This particular example means that

there are two examples falling into this leaf $\{x93 > -0.057129\}$ under all other parent branches and they are predicted to be positive, or 5-HT_2 receptor antagonists. And two compounds are true positive while no compound is negative but predicted positive. In some cases, the number in brackets may be followed by a second number (e.g., $\{x106 > -0.52536 : \quad -1 \quad (3/1)\}$). Here, the second value (1) equals the number of classification errors encountered out of the total number of classifications made from the training data in that particular path of the decision tree (3)[3].

Table 3.3: Summary of the decision tree's performance on original and PCA preprocessed data sets

Measurement	On original data	On the data sets after PCA		
		113-dimension	58-dimension	23-dimension
Precision	88.23%	100.00%	76.92%	100.00%
Recall/Sensitivity	71.43%	71.43%	47.62%	57.14%
F measure	78.95%	83.33%	58.82%	72.72%
Specificity	99.21%	100.00%	98.82%	100.00%

Table 3.3 gives the performance evaluation on the four testing data sets. The precision, recall, F measure, sensitivity and specificity are compared on the four data sets. It is found that in general the prediction accuracy for negative examples are quite high. This is expected as in training set, the number of negative examples are more than 10 times greater than that of positive examples, which may give more information on the characteristics of negative examples. As for positive examples,

[3]The threshold values used in this decision tree are those after scaling. To test unseen instances, they have to be scaled to $[-1, 1]$ using the strategy described in Section 2.3.1. On original data, no PCA preprocessing is done. The next few chapters will also follow this convention.

```
x86 <= -1:
|   x134 <= -0.43652:
|   |   x93 > -0.057129: 1 (2)
|   |   x93 <= -0.057129:
|   |   |   x7 > -1: 1 (3/1)
|   |   |   x7 <= -1:
|   |   |   |   x135 <= -0.61371: -1 (24)
|   |   |   |   x135 > -0.61371:
|   |   |   |   |   x4 <= -0.6: 1 (2)
|   |   |   |   |   x4 > -0.6: -1 (2)
|   x134 > -0.43652:
|   |   x64 <= -0.53751:
|   |   |   x4 > -0.33333: -1 (415)
|   |   |   x4 <= -0.33333:
|   |   |   |   x12 > -1: -1 (69)
|   |   |   |   x12 <= -1:
|   |   |   |   |   x138 <= -0.85516: -1 (204)
|   |   |   |   |   x138 > -0.85516:
|   |   |   |   |   |   x17 <= -0.5:
|   |   |   |   |   |   |   x136 <= -0.399: -1 (59)
|   |   |   |   |   |   |   x136 > -0.399:
|   |   |   |   |   |   |   |   x47 <= -0.42841: 1 (2)
|   |   |   |   |   |   |   |   x47 > -0.42841: -1 (5)
|   |   |   |   |   |   x17 > -0.5:
|   |   |   |   |   |   |   x91 <= -0.79849: -1 (8)
|   |   |   |   |   |   |   x91 > -0.79849: 1 (3)
|   |   x64 > -0.53751:
|   |   |   x62 <= 0.47967: 1 (2)
|   |   |   x62 > 0.47967: -1 (23)
x86 > -1:
|   x6 <= -1:
|   |   x147 <= -0.11423:
|   |   |   x48 <= 0.19397:
|   |   |   |   x14 <= -0.11111:
|   |   |   |   |   x106 > -0.52536: -1 (3/1)
|   |   |   |   |   x106 <= -0.52536:
|   |   |   |   |   |   x17 <= -0.75: -1 (33)
|   |   |   |   |   |   x17 > -0.75:
|   |   |   |   |   |   |   x27 <= -0.59518: -1 (20/1)
|   |   |   |   |   |   |   x27 > -0.59518: 1 (4/1)
|   |   |   |   x14 > -0.11111:
|   |   |   |   |   x2 <= -0.076923: -1 (2)
|   |   |   |   |   x2 > -0.076923: 1 (2)
|   |   |   x48 > 0.19397:
|   |   |   |   x2 <= -0.11538: 1 (22)
|   |   |   |   x2 > -0.11538:
|   |   |   |   |   x148 > -0.02887: 1 (5)
|   |   |   |   |   x148 <= -0.02887:
|   |   |   |   |   |   x133 <= 0.47562: -1 (12)
|   |   |   |   |   |   x133 > 0.47562: 1 (2)
|   |   x147 > -0.11423:
|   |   |   x153 <= 0.12445: -1 (74)
|   |   |   x153 > 0.12445:
|   |   |   |   x68 <= -0.215:
|   |   |   |   |   x134 <= 0.25141: -1 (36)
|   |   |   |   |   x134 > 0.25141:
|   |   |   |   |   |   x132 <= 0.13716: -1 (4)
|   |   |   |   |   |   x132 > 0.13716: 1 (2)
|   |   |   |   x68 > -0.215:
|   |   |   |   |   x49 <= -0.33753: 1 (6)
|   |   |   |   |   x49 > -0.33753: -1 (3)
|   x6 > -1:
|   |   x89 > -0.71974: -1 (8)
|   |   x89 <= -0.71974:
|   |   |   x2 > 0.11538: -1 (5)
|   |   |   x2 <= 0.11538:
|   |   |   |   x142 <= -0.30769: -1 (4)
|   |   |   |   x142 > -0.30769:
|   |   |   |   |   x132 <= 0.13716: 1 (24)
|   |   |   |   |   x132 > 0.13716:
|   |   |   |   |   |   x75 > 0.027532: 1 (2)
|   |   |   |   |   |   x75 <= 0.027532:
|   |   |   |   |   |   |   x29 <= 0.25771: -1 (5)
|   |   |   |   |   |   |   x29 > 0.25771: 1 (2)
```

Figure 3.4: The decision tree generated for 5-HT_2 receptor antagonists

the original data set and 113 principal components(PCs) data set give the highest recall (sensitivity) value of 71.43%, which means 15 positive examples out of 21 are predicted correctly. On a whole, the algorithm performs best on the 113 PCs data set, whose *Fmeasure* is as high as 83.33%, then are original data set, 23 PCs data set, and 58 PCs data set, in the order of performance decrease. Last but not least, it is interesting to see that when discarding 1% variances, the 58 PCs data set gives lower accuracy compared to the 23 PCs data set discarding 10% variances. It indicates that the noise of the data are not concentrated in the smallest variance region, but in the dimensions where the variances are bigger.

3.3.2 K-Nearest Neighbor prediction

The parameter k used in $k$nn is scanned in the range of $1, 2, \ldots, 33$. The number "33" is calculated as the square root of the number of training examples. Such a large scanning range normally ensures that the optimal performance can be found. In the application of drug design, k is tuned in the direction of improving *Fmeasure*. This is because that in the training data, the number of negative examples are 10 times greater that of positive examples, and the prediction accuracy for negative examples are all above 98%. So discovering positive examples or improving prediction accuracy on positive examples becomes the priority. *Fmeasure* is an ideal measurement because it balances between how many positive examples are recognized and how many examples are true positive among those predicted positive ones.

In order to find the optimal k, the *Fmeasure* is plotted against k. Figure 3.5 illustrates the parameter tuning process. The best *Fmeasure*s on the original data set, 113 PCs data set, 58 PCs data set and 23 PCs data set are 66.15%, 66.15%, 67.18% and 68.00% respectively. And the maximum *Fmeasure* are reached when k is set to 7, 7, 7 and 5 respectively. Finally the classification results are measured on

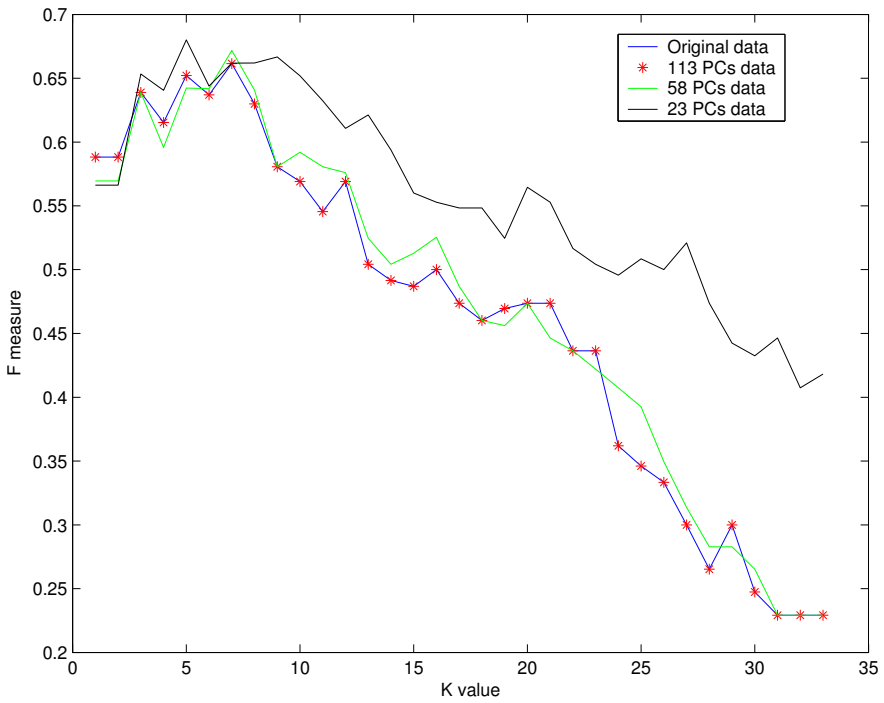

Figure 3.5: The k-NN parameter tuning process based on 10-fold cross validation

testing data sets using the models trained with these optimal k values (See Table 3.4).

Table 3.4: Summary of the $k$nn's performance on original and PCA preprocessed testing sets

Measurement	On original data	On the data sets after PCA		
		113-dimension	58-dimension	23-dimension
Precision	100.00%	100.00%	100.00%	100.00%
Recall/Sensitivity	71.43%	71.43%	66.67%	66.67%
F measure	83.33%	83.33%	80.00%	80.00%
Specificity	99.29%	99.29%	98.57%	98.57%

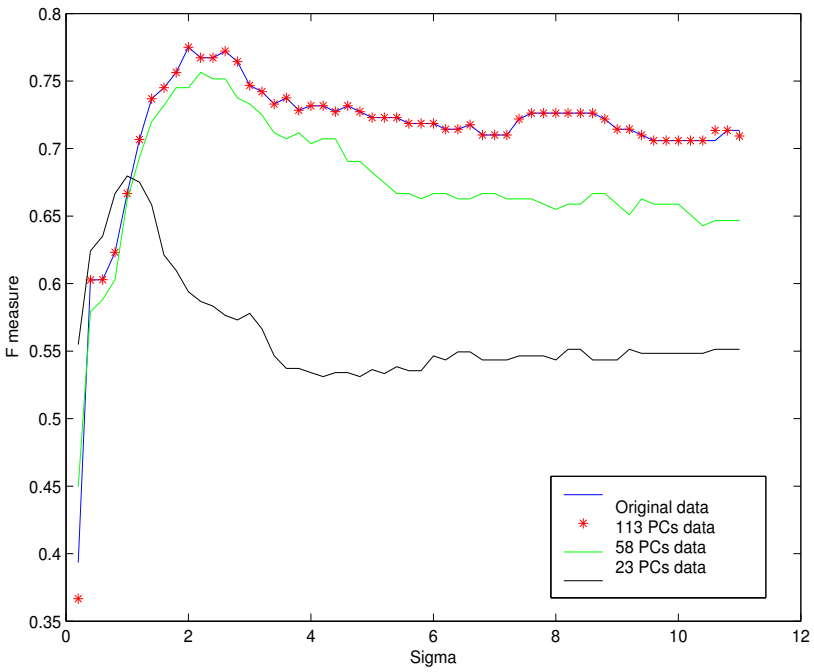

Figure 3.6: The SVM parameter tuning process based on 10-fold cross validation

Generally speaking, as a very simple approach, the $k$nn gives a really good performance, compared to decision tree. However, this superiority is only obvious for 58 principal components consisting data set and 23 principal components consisting data set. Decision tree and $k$nn give exactly the same and best results on 113 principal components consisting data set. The best $Fmeasure$ is 83.33% and the best recall value is 71.43%.

3.3.3 SVM prediction

SVMs are trained with the free parameter of Gaussian kernel width (σ), scanned in the range of $[0, \ldots, 11]$ with an interval of 0.2, which is the empirical range that gives optimal classification results on chemical compounds classification.

For every σ, the training data set is divided into 10 subsets, and the holdout method is repeated 10 times. Each time, one of the 10 subsets is used as the small

testing set and the other 9 subsets are put together to form a small training set. Then the average *Fmeasure* across all 10 trials is computed. In the end, *Fmeasure* are plotted against σ, on four training sets (See Figure 3.6). By optimizing *Fmeasure*, the best σ found are 2.0, 2.0, 2.2 and 1.0. Finally the classification performance are measured on four testing sets with their optimal σ values. The summary of the results is shown in Table 3.5. The *Fmeasure* calculated on the four testing data sets are 92.31%, 92.31%, 87.47%, and 77.78% for the original data set and the data sets consisting of first 113, 58, and 23 principal components respectively.

Table 3.5: Summary of the SVM's performance on original and PCA preprocessed testing sets

Measurement	On original data	On the data sets after PCA		
		113-dimension	58-dimension	23-dimension
Precision	100.00%	100.00%	100.00%	93.33%
Recall/Sensitivity	85.71%	85.71%	80.95%	66.67%
F measure	92.31%	92.31%	87.47%	77.78%
Specificity	100.00%	100.00%	100.00%	99.61%

Such results are much better than those of decision tree and $k$nn. The best recall (sensitivity) we get by SVM is more than 14% higher than that of decision tree and $k$nn. And the *Fmeasure*, precision and specificity are also higher than those of decision tree and $k$nn to different extent. This indicates the 5-HT_2 inhibitor data in the input space are, from a machine learning perspective, quite complex and require more powerful classification algorithm. Consistent with the previous two classification algorithms, the best result with SVM is also obtained from the data set that consists of the 113 principal components representing all the variances.

Therefore, PCA preprocessing does help to significantly reduce the computation load required for large data set, as a result of reduced number of features, from 159 to 113 in this case.

3.4 Summary

The neurotransmitter serotonin mediates a wide range of physiological functions including a number of normal human functions (e.g. sleep, sexual activity and appetite) as well as human disorders (e.g. migraine, depression and anxiety), by interacting with multiple receptors. Among these receptors, 5-HT_2 receptors have been reported to play important roles in these pathological and psychopathological conditions. New $5HT_2$ antagonists are expected to help develop personalized treatments of high efficacy and low toxicity for physiologic disorders.

In this chapter, 106 5-HT_2 antagonists are collected manually from different references. Moreover, 1272 chemicals which bind to a wide range of proteins, excluding 5-HT_2 homological proteins, were selected as a diversified sample of "non-" 5-HT_2 antagonists. These data are further split into training and testing data sets according to the method of Kennard and Stone[77].

Different machine learning classifiers, namely decision tree, $k$nn and SVM are built using training data sets. The classification results are measured and compared on original testing data and testing data sets consisting of the first 113, 58, and 23 principal components. The results show:

- The overall performance is decent. The most important measurement $Fmeasure$ from decision tree, $k$nn and SVM reach are 83.33%, 83.33% and 92.31% respectively on the optimal data set—the data set consisting of 113 principal components. SVM beats decision tree and $k$nn by nearly 9%.

- PCA seems useful in dimensionality reduction. The data set that gives the best results by all the three approaches is made up of the first 113 principal components, which corresponds to the minimum dimensions that keep 100% of the original variances.

- The fact that the PCA processed data set keeping 100% variances give better results than those keeping 99% and 90% variances demonstrates the descriptors I used in this work are capable of characterizing these chemicals for this task. But it might be necessary to reorganize the 159 descriptors and remove those redundant ones since 113 principal components are able to represent all the 100% variances in the training data set.

In later chapters, these methods will be tested for the inhibitor prediction for an enzyme target–cholinesterase and an ADME associated protein–CYP3A4.

Chapter 4

Inhibitor prediction of the therapeutic target–Cholinesterase

Enzymes are known to cover 44% of the total drug targets available[23]. Therefore they can not be missed out for the test of new lead identification techniques. In this chapter, cholinesterase is selected as the representative of enzyme targets. A set of compounds that have inhibitory effects on cholinesterase are collected. Decision tree, $k$nn and SVM are evaluated for their predictive capacity of identifying this class of compounds.

Specifically, in section 4.1, the chemistry and biology of cholinesterase and therapeutic application of cholinesterase inhibition are introduced. Then the detailed description of data preparation procedure and experimental design is given in section 4.2. The results obtained are then presented and discussed in section 4.3. In section 4.4, a short summary is given.

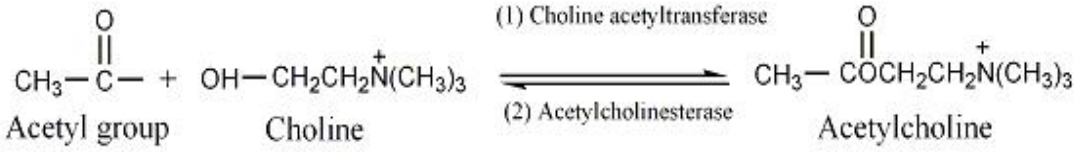

Figure 4.1: The chemistry of acetylcholine

4.1 Introduction

4.1.1 Cholinergic transmission

Acetylcholine is the chemical mediator of nerve impulses at all (sympathetic and parasympathetic) autonomic ganglia, the post ganglionic parasympathetic neuroeffector junction, the neuromuscular junction, and some parts of the central nervous system.

Acetylcholine's synthesis is controlled by the enzyme choline acetyltransterase, which mediates the transfer of an acetyl group from acetyl coenzyme A to choline, a normal constituent of the diet. This synthetic reaction (1) is depicted as going from left to right in Figure 4.1.

Following its release from vesicles at the nerve endings, acetylcholine interacts with the cholinergic receptor to initiate a response. Acetylcholine is then very rapidly hydrolyzed and inactivated by the enzyme acetylcholinesterase. This inactivation reaction (2) is depicted as going from right to left in Figure 4.1. Choline, one of the products of this reaction, is taken up by the nerve endings responsible for its release, and is reused for the synthesis of new molecules of acetylcholine.

4.1.2 Biological function of Cholinesterase

Cholinesterase (ChE) are a family of related enzymes that hydrolyze choline esters at a very fast rate. The major biological function of ChE is to terminate the impulse

transmission by acetylcholine at cholinergic nerve endings in synapses or in effector organs.

These enzymes have been further classified on the basis of substrate specificity and sensitivity to various inhibitors. Those enzymes which preferentially hydrolyse acetyl esters such as acetylcholine (ACh) are called acetylcholinesterase (AChE) or acetylcholine acetylhydrolase (EC 3.1.1.7), and those which prefer other types of esters such as butyrylcholine are termed butyrylcholinesterase (BChE) or acylcholine acylhydrolase (EC 3.1.1.8). BChE is also known as pseudocholinesterase, non- specific cholinesterase, or simply cholinesterase. The last term has led to confusion, and in this thesis the term cholinesterase will refer to all choline ester hydrolysing enzymes, irrespective of their substrate specificity.

The main function of AChE is the rapid hydrolysis of the neurotransmitter ACh at cholinergic synapses. The hydrolysis reaction proceeds by nucleophilic attack of the carbonyl carbon, acylating the enzyme and liberating choline. This is followed by a rapid hydrolysis of the acylated enzyme yielding acetic acid, and the restoration of the esteratic site[78].

The function of BChE remains a puzzle. It has no known specific natural substrate, although it is capable of hydrolysing ACh. It has been suggested that BChE acts as a scavenging enzyme in the detoxification of natural compounds[79]. Certain human individuals have a mutant BChE which lacks the ability to hydrolyse succinyl choline. In rare individuals the complete BChE gene is missing. Neither of these cases result in any apparent physiological consequence. There is however an important clinical implication; succinyl choline is commonly used during tracheal intubation in the administration of inhalation anaesthetics, and causes post operative apnoea in these people[80].

4.1.3 Cholinesterase inhibitions

Cholinesterase inhibitors (also called anticholinesterase agents) inhibit cholinesterase and thus slow down the hydrolysis of acetylcholine. This inhibition permits the buildup of acetylcholine at the receptor site and causes more intensive and prolonged cholinergic activation. The resulting pharmacological effects are qualitatively similar to those observed after simulation of cholinergic nerves, although quantitatively of far greater magnitude. Cholinesterase inhibition have both desirable therapeutic effects and undesirable therapeutic effects, which are discussed separately below.

Some cholinesterase inhibitors have been used as medicine for a long time. For example, Physostigmine is used to treat certain types of glaucoma[1] for more than a century. Neostigmine was developed in the early 1930s for management of myasthenia gravis; Ambenonium was developed later for this same purpose, as well as Pyridostigmine bromide[81]. All these compounds belong to the chemical class of carbamate.

More recently, cholinesterase inhibitors are developed to treat Alzheimer's disease (AD). Alzheimer's disease is a neurodegenerative condition characterized by progressive deficits in memory and cognition, together with impairment in the ability to perform activities of daily living[82, 83]. It is prevalent in aged people worldwide and has stimulated many scientists to focus on the research aiming at identifying pathogenesis of this disease and at discovering effective pharmaceuticals for it. However, despite these efforts, a cure for this disease remains to be found. Currently only one class of medications has been extensively evaluated and showed efficiency for AD symptoms. These are cholinesterase inhibitors. Tacrine, donepezil, rivastigmine and galantamine are the representatives that have been approved by the US Food and

[1]It is a disease characterized by increased intraocular pressure, which if untreated, will ultimately result in damage to the optic nerve.

Drug Administration (FDA) and other governmental agencies for the treatment of AD.

On the other hand, cholinesterase inhibitors are also widely used as agricultural and household insecticides, and even chemical warfare agents (e.g. nerve gases such as sarin and soman). These compounds belong to the chemical class of organophosphate. Their acute toxicity is mainly due to the buildup of ACh at cholinergic synapses. Signs and symptoms of overexposure to these cholinesterase inhibitor include tiredness, dizziness, nausea and blurred vision; headache, sweating, tearing, drooling, vomiting, tunnel vision, and twitching; abdominal cramps, urinating, diarrhea, muscular tremors, staggering gait, pinpoint pupils, hypotension, slow heartbeat, breathing difficulty, and possibly death, if not promptly treated by a physician[84, 85].

4.2 Data preparation

The data preparation procedures I took in this chapter are similar to those in Chapter 3. Examples of both classes, cholinesterase inhibitors (positive examples) and compounds having no inhibitory effect on cholinesterase (negative examples) are required to train a classification model. The information about cholinesterase inhibitors are scattered in numerous references. A total of 132 cholinesterase inhibitors were manually collected from available literatures with the help of a few simple automated text retrieval programs. The text retrieval programs I use are a set of in-house developed Perl programs that can automatically search PubMed database with keyword query. Specifically, they include a NCBI interface, a text formatter, and a job planner. They are able to download relevant abstracts and modify them to highlight certain keywords, such as “inhibitor”, “inhibitory”, “IC50”, and “cholinesterase” etc. They also sort the retrieved abstracts by frequencies of key

words of our interests in them. This small toolkit can improve the efficiency of inhibitors searching in literatures. However, the techniques underpinning them belong to the IT field and thus beyond the scope of this thesis.

Then a total of 1588 negative examples ("non-" cholinesterase inhibitors) are selected from the inhibitors of 1170 "negative proteins". The source of "negative proteins" is the therapeutic target databse (TTD)[23]. The composition of the "negative proteins" encompasses major pharmacological important proteins, such as enzymes, receptors, transporters, and antibodies et al. The 2D structures of these compounds are downloaded from ChemIDplus[2]. Then to calculate the 159 descriptors described in Chapter 2, the 2D structures are converted into 3D structures using Concord 4.0.1.

In this process, the structures of the cholinesterase inhibitors and "non-" cholinesterase inhibitors undergo a series of cleaning procedures: a) those whose nomenclature are not standard and hence impossible to find their structures are firstly removed; b) duplicates within each class are removed; c) compounds shared by both classes are removed from negative class because the positive examples are confirmed in references and the quality of positive data is believed to be better; d) counterions and solvent molecules are removed, as well as obviously unsuitable compounds, e.g. heavy metal containing compounds, compounds whose molecular weight below 150 or above 1000.

After the data cleaning step, a 159-dimensional data set is successfully generated

[2]ChemIDplus (http://chem.sis.nlm.nih.gov/chemidplus/) is a free, web-based search system that provides access to structure and nomenclature authority files used for the identification of chemical substances cited in National Library of Medicine (NLM) databases. ChemIDplus also provides structure searching and direct links to many biomedical resources at NLM and on the Internet for chemicals of interest. The database contains over 368,000 chemical records, of which over 206,000 include chemical structures, and is searchable by Name, Synonym, CAS Registry Number, Molecular Formula, Classification Code, Locator Code, and Structure.

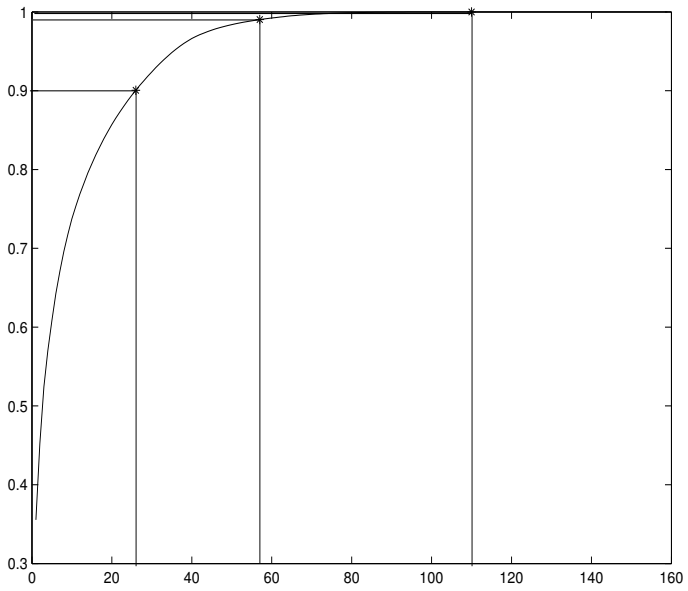

Figure 4.2: The number of principal components VS the percentage of total variance they represent.

with 116 positive examples and 1405 negative examples. This data set is further split into training and testing sets with the ratio of 8:2. Training data are used to tune free parameters and train a classification model. Testing data are used to measure and evaluate the performance of different classification models.

Table 4.1: The training and testing data sets

Data Set	No. of positive examples	No. of negative examples	Total No.
Training set	92	1125	1217
Testing set	24	280	304
Total No.	116	1405	1521

Both the training data and test data undergo the preprocessing techniques—normalization and PCA discussed in Chapter 2. Figure 4.2 shows the PCA analysis of the training data. From it, we can see that without loss of any variances encoded in the data set, 49 dimension redundant descriptors can be reduced. Keeping 99% of the variances, 102 dimensions can be reduced. Keeping 90% of the variances, 133 dimensions can be reduced. To be consistent with the approach used in last chapter, I construct new training and testing data sets with the first n principal components while $n = 26$, $n = 57$, and $n = 110$.

4.3 Prediction results and analysis

After scaling and PCA preprocessing, we are ready to train classification models. Different machine learning algorithms were explored on the four sets of training data. Their performances are measured on the testing data sets. The results and discussion are given in this section:

4.3.1 Decision tree prediction

The unpruned decision tree C4.5 generated for original data is shown in Figure 4.3. Its interpretation method is identical to that of section 3.3. The threshold values used in this decision tree are the same as those after scaling.

This time, the decision tree is more complicated. It has a maximal depth of 15 layers, whereas the longest branch in the tree for 5-HT_2 antagonist prediction is only 8 layers. 12 descripors, more than 1/3 of the descriptors tested in decision nodes of this tree are also tested in the nodes of in the decision tree for 5-HT_2 antagonist prediction. But the remaining 2/3 of the descriptors, about 20 attributes, were never used in the decision tree for 5-HT_2 antagonist prediction.

Table 4.2 presents the final results on testing data sets. The precision, recall, F

```
x11 <= -1:
|   x133 <= 0.036842:
|   |   x136 > -0.55309: 1 (13/1)
|   |   x136 <= -0.55309:
|   |   |   x27 > -0.77227: -1 (19)
|   |   |   x27 <= -0.77227:
|   |   |   |   x25 > -1: -1 (3)
|   |   |   |   x25 <= -1:
|   |   |   |   |   x13 <= -1: -1 (7)
|   |   |   |   |   x13 > -1:
|   |   |   |   |   |   x85 > -1: -1 (2)
|   |   |   |   |   |   x85 <= -1:
|   |   |   |   |   |   |   x59 > 0.10652: -1 (4)
|   |   |   |   |   |   |   x59 <= 0.10652:
|   |   |   |   |   |   |   |   x12 > -1: 1 (4)
|   |   |   |   |   |   |   |   x12 <= -1:
|   |   |   |   |   |   |   |   |   x135 > -0.58802: 1 (9/1)
|   |   |   |   |   |   |   |   |   x135 <= -0.58802:
|   |   |   |   |   |   |   |   |   |   x60 <= -0.30548: 1 (3/1)
|   |   |   |   |   |   |   |   |   |   x60 > -0.30548: -1 (8)
|   x133 > 0.036842:
|   |   x134 <= -0.20205:
|   |   |   x70 > -0.83216: -1 (3)
|   |   |   x70 <= -0.83216:
|   |   |   |   x106 > -0.35764: -1 (3)
|   |   |   |   x106 <= -0.35764:
|   |   |   |   |   x92 > -1: 1 (2)
|   |   |   |   |   x92 <= -1:
|   |   |   |   |   |   x74 > -0.9931: -1 (2)
|   |   |   |   |   |   x74 <= -0.9931:
|   |   |   |   |   |   |   x75 <= 0.16536: 1 (6/1)
|   |   |   |   |   |   |   x75 > 0.16536:
|   |   |   |   |   |   |   |   x84 > -0.84804: 1 (3/1)
|   |   |   |   |   |   |   |   x84 <= -0.84804:
|   |   |   |   |   |   |   |   |   x20 <= -0.93142: 1 (2)
|   |   |   |   |   |   |   |   |   x20 > -0.93142:
|   |   |   |   |   |   |   |   |   |   x135 <= -0.56734: -1 (21/1)
|   |   |   |   |   |   |   |   |   |   x135 > -0.56734: 1 (2)
|   |   x134 > -0.20205:
|   |   |   x16 <= -1:
|   |   |   |   x59 > 0.21812: 1 (3)
|   |   |   |   x59 <= 0.21812:
|   |   |   |   |   x85 <= -0.6303: -1 (37/1)
|   |   |   |   |   x85 > -0.6303: 1 (3/1)
|   |   |   x16 > -1:
|   |   |   |   x24 > -0.67422: -1 (498)
|   |   |   |   x24 <= -0.67422:
|   |   |   |   |   x86 <= -0.28816:
|   |   |   |   |   |   x17 > -1: -1 (343/2)
|   |   |   |   |   |   x17 <= -1:
|   |   |   |   |   |   |   x24 > -0.82777: -1 (45)
|   |   |   |   |   |   |   x24 <= -0.82777:
|   |   |   |   |   |   |   |   x85 > -1: 1 (2)
|   |   |   |   |   |   |   |   x85 <= -1:
|   |   |   |   |   |   |   |   |   x59 <= 0.34653: -1 (6)
|   |   |   |   |   |   |   |   |   x59 > 0.34653: 1 (2)
|   |   |   |   |   x86 > -0.28816:
|   |   |   |   |   |   x6 > -1: -1 (8)
|   |   |   |   |   |   x6 <= -1:
|   |   |   |   |   |   |   x80 > -1: -1 (6)
|   |   |   |   |   |   |   x80 <= -1:
|   |   |   |   |   |   |   |   x56 > -1: -1 (5)
|   |   |   |   |   |   |   |   x56 <= -1:
|   |   |   |   |   |   |   |   |   x18 <= -0.97143: -1 (30/1)
|   |   |   |   |   |   |   |   |   x18 > -0.97143:
|   |   |   |   |   |   |   |   |   |   x41 <= -0.9007: 1 (3)
|   |   |   |   |   |   |   |   |   |   x41 > -0.9007:
|   |   |   |   |   |   |   |   |   |   |   x63 <= 0.78439:
|   |   |   |   |   |   |   |   |   |   |   |   x55 > -0.78331: 1 (2)
|   |   |   |   |   |   |   |   |   |   |   |   x55 <= -0.78331:
|   |   |   |   |   |   |   |   |   |   |   |   |   x85 > -1: 1 (2)
|   |   |   |   |   |   |   |   |   |   |   |   |   x85 <= -1:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x12 <= -1: -1 (10/1)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x12 > -1: 1 (3/1)
|   |   |   |   |   |   |   |   |   |   |   |   x63 > 0.78439:
|   |   |   |   |   |   |   |   |   |   |   |   |   x154 <= -0.67073: 1 (2)
|   |   |   |   |   |   |   |   |   |   |   |   |   x154 > -0.67073: -1 (38/1)
x11 > -1:
|   x49 <= -0.90472: 1 (16)
|   x49 > -0.90472:
|   |   x6 > -1: 1 (3)
|   |   x6 <= -1:
|   |   |   x87 <= -0.26603: 1 (3)
|   |   |   x87 > -0.26603:
|   |   |   |   x75 <= 0.19344: -1 (29/2)
|   |   |   |   x75 > 0.19344: 1 (2)
```

Figure 4.3: The decision tree generated for cholinesterase inhibitors

Table 4.2: Summary of the decision tree's performance on original and PCA pre-processed data sets

Measurement	On original data	On the data sets after PCA		
		110-dimension	57-dimension	26-dimension
Precision	78.95%	93.75%	71.43%	76.92%
Recall/Sensitivity	62.50%	62.50%	41.67%	41.67%
F measure	69.77%	70.00%	52.62%	54.06%
Specificity	98.57%	99.64%	98.57%	98.93%

measure, sensitivity and specificity are compared on the four data sets. From it, we can see that the 57-dimension PCs data set and 26-dimension PCs data set do not give a satisfactory results. The recall value (or sensitivity) is less than 50%, which means the prediction is worse than random guesswork for positive examples. The results on 110-dimension PCs data set is the best among the four group of results and slightly better than those on original data set. The number of true positive, false negative, true negative and false positive are 15, 9, 279 and 1 respectively for 110-dimension PCs data set whereas 15, 9, 276 and 4 respectively for original data set. Thus recall and sensitivity does not increase. However, with the improvement on negative data prediction, the specificity increased about 1%, from 98.47% to 99.64%. More importantly, with the decrease of false negatives, the specificity on the data set consisting of 110 principal components is 15% higher than the specificity on original data set.

These results confirmed the idea brought up in last chapter that the descriptors are highly correlated. 57 principal components are able to represent 99% of the original variances and 110 principal components are able to represent the original

data without any loss of information. Also, the information needed for classification does not necessarily reside in directions with big variances. The addition of the last 1% of the variance lead to 18% of improvement to the F measure.

4.3.2 K-Near Neighbor prediction

As there are 1217 examples in training data set, the parameter k used in $k$nn is scanned in the range of $1, 2, \ldots, 34$, which is approximately the square root of 1217. Such a big range ensures that the optimal value for the parameter of k. The experiments in previous chapter indicates that $Fmeasure$ gives a more objective measurement for unbalanced data sets than Precision, Recall, Sensitivity and Specificity. Hence in order to find the optimal k, $Fmeasure$ is plotted against k. Figure 4.4 illustrates the parameter tuning process. From it, we can see the best $Fmeasure$ on the original training data, 110-dimension PCs training data, 57-dimension PCs training data and 26-dimension PCs training data are 55.48%, 55.48%, 54.90% and 59.63% respectively. The results are the average values on 10-fold cross-validation training data set. The maximal $Fmeasure$ are reached when k is set to 3, 3, 3 and 4 respectively.

The classification capability are consequently measured on testing data sets with their optimal k values (shown in Table 4.3). From it, we can see that:

- As the decision tree method, K nearest neighbor gives better prediction on negative data than on positive data. The specificity values are about 30% higher than sensitivity (Recall) values. Previous prediction of 5-HT_2 antagonist shows the same trend. Cai et al reported the same phenomenon in protein function predictions[59, 60]. This is mainly due to the shortage of positive examples.

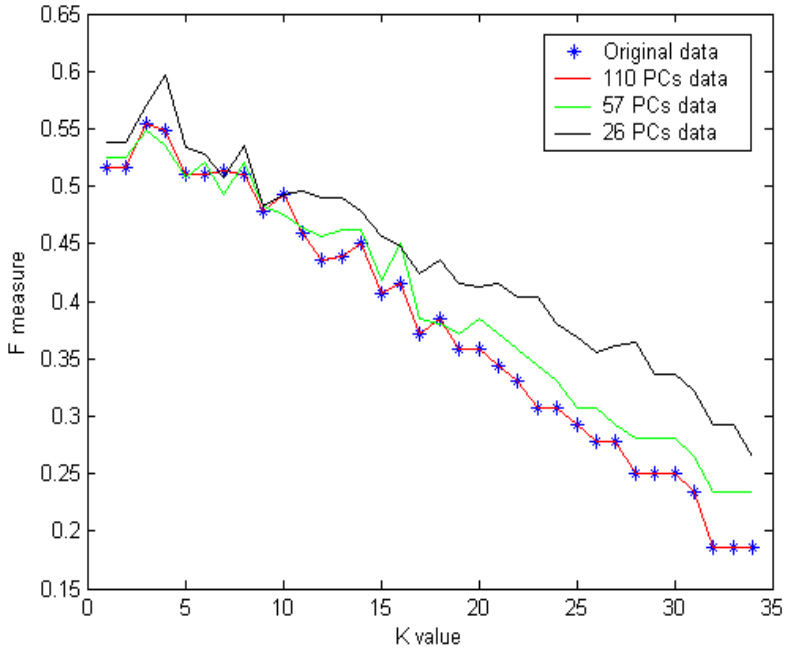

Figure 4.4: The *k*nn parameter tuning process based on 10-fold cross validation

Table 4.3: Summary of the *k*nn's performance on original and PCA preprocessed testing sets

Measurement	On original data	On the data sets after PCA		
		110-dimension	57-dimension	26-dimension
Precision	89.47%	89.74%	85.00%	80.00%
Recall/Sensitivity	70.83%	70.83%	70.83%	66.67%
F measure	79.07%	79.07%	77.27%	72.73%
Specificity	99.29%	99.29%	98.57%	98.57%

- Original data set and 110-dimension PCs data set (keeping the 100% variances in original data), give exactly the same results. 57-dimension PCs data set

(keeping the 99% variances in original data) gives results inferior to that from the former two and 26-dimension PCs data set (keeping the 90% variances in original data) gives results inferior to that from 57-dimension PCs data set. Such results are consistent with previous experiments.

- K nearest neighbor gives better results than decision tree does on cholinesterase inhibitor prediction. The difference between two groups of results are larger than that of 5-HT_2 antagonist prediction. In case of original data set and 110-dimension PCs data set, $Fmeasure$ and Sensitivity (or Recall) are improved by about 8% and 9% respectively.

4.3.3 SVM prediction

SVMs are trained with the kernel parameter σ scanned in the range of $[0, \ldots, 8]$ with an interval of 0.2. After $\sigma = 8$, the $Fmeasure$ becomes stable and does not change much with increasing σ. Figure 4.5 gives the plot of $Fmeasure$ against σ for the four training sets. The best σ found are 7, 7, 0.4 and 0.2, corresponding to the optimal $Fmeasure$ values: 60.24%, 60.24%, 54.78%, and 54.12% respectively.

Afterward the classification performances are measured on four testing sets with their optimal σ values. Table 4.4 gives the summary. Among all the three approaches, SVM produces the best results. The best $Fmeasure$ is as high as 89.36% on original data set and 110-dimension PCs data set. By contrast, decision tree only gives the highest $Fmeasure$ of 70.00% and $k$nn 79.07%. Furthermore, it is found that the achievement of SVM is mainly contributed by the improvement of Sensitivity (Recall) values, or in other words, the prediction improvement on positive examples. Among 24 true positive examples, SVM is able to pick out 21 examples correctly. This is amazingly high.

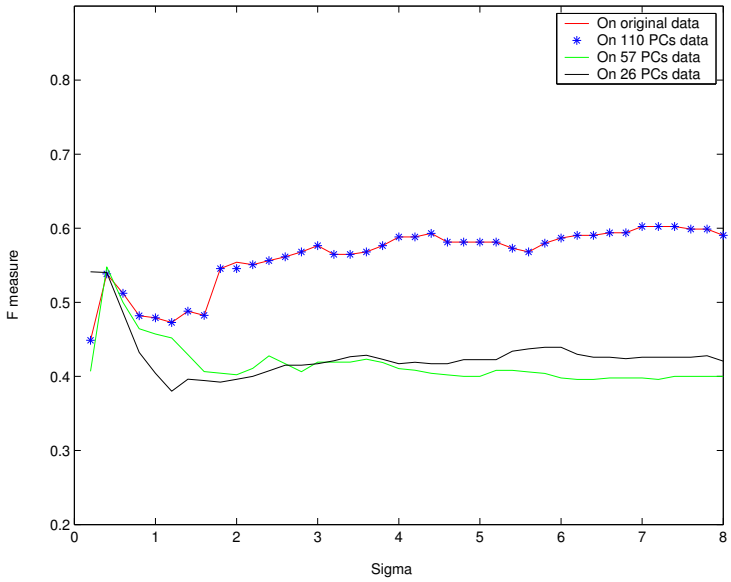

Figure 4.5: The SVM parameter tuning process based on 10-fold cross validation

Table 4.4: Summary of the SVM's performance on original and PCA preprocessed testing sets

Measurement	On original data	On the data sets after PCA		
		110-dimension	57-dimension	26-dimension
Precision	91.30%	91.30%	94.12%	94.44%
Recall/Sensitivity	87.50%	87.50%	66.67%	70.83%
F measure	89.36%	89.36%	78.05%	80.95%
Specificity	99.29%	99.29%	99.64%	99.64%

4.4 Summary

In this chapter, we use decision tree, $k$nn and SVM to predict the inhibitory effects of different compounds on the enzyme target–cholinesterase. Cholinesterase is the

enzyme responsible for the rapid hydrolysis of the neurotransmitter–acetylcholine. Cholinesterase inhibitors, also called anticholinesterase agents, block the catalysis of cholinesterase. Consequently acetylcholine is accumulated at the receptor and causes more intensive and prolonged cholinergic activation. This effect can be desirable in the case of treatment of glaucoma, myasthenia gravis, and Alzheimer's disease. This effect, if too strong, can also lead to toxicity, in the case of insecticides and chemical warfare agent. Therefore extra caution should be paid when designing new cholinesterase inhibitors for medications.

The procedure begins with the search of cholinesterase inhibitors and non-cholinesterase-inhibitor chemicals. A total of 92 cholinesterase inhibitors and 1125 non-cholinesterase-inhibitor representatives are collected to train different classification models. Then 24 unseen cholinesterase inhibitors and 280 non-cholinesterase-inhibitor chemicals are used to test the different models. In order to train a good model, other pre-processing techniques, such as normalization and principal components analysis are used.

From the algorithmic evaluation perspective, the results obtained in this chapter are similar to that of 5-HT_2 antagonist prediction.

SVM outperforms decision tree and $k$nn. The best $Fmeasure$ achieved by SVM is 89.36%, which corresponds to a precision value of 91.30% and a recall value of 87.5%. This $Fmeasure$ is 19.59% higher than that of decision tree, and 10.29% higher than that of $k$nn.

110-dimension PCs data set that keeps the 100% variances in the training data set, gives exactly the same good results as the original data set in the case of $k$nn and SVM, and slightly better results than the original data set in case of decision tree. This once again confirmed that PCA is useful in dimensionality reduction.

Chapter 5

Inhibitor prediction for the ADME associated protein– CYP3A4

Although the speed of lead compound screening has been increased greatly during the early stages of drug discovery, rapid optimization of parameters that determine whether a potent inhibitor will become a successful drug remains a challenge in improving the efficiency of the drug discovery process. The speed with which drugs are screened for properties such as absorption, cytochrome P450 (CYP) inhibition, and metabolic stability is still several orders of magnitude lower than those for high-throughput methods used in lead identification. Parameters that define absorption, distribution, metabolism, and excretion properties of drug candidates are essential for therapeutic efficacy, and thus should be optimized during early stages of drug discovery.

In this chapter, machine learning techniques are tested for their applicability to expedite identifying drug candidates with potential to inhibit cytochrome P4503A4 . This is important in the drug discovery process because metabolism by CYP represents an important clearance mechanism for the vast majority of drugs, thus affecting their oral bioavailability and/or duration of action.

5.1 Introduction

5.1.1 Drug metabolism

Metabolic transformation of drug molecules represents a key process by which drugs are cleared from the body. Metabolic transformations have traditionally been divided into two phases. Phase I reactions (biotransformation) include oxidation, reduction, and hydrolysis which primarily serve to increase the hydrophilicity and enhance the excretion of a drug by unveiling or incorporating a polar functional group into the molecule (-OH, -SH, -NH_2, or -COOH). Phase II reactions (conjugation) further increase the polarity of a drug by modifying a functional group to form O- or N-glucuronides, sulfate esters, α-carboxyamides, and glutathionyl adducts.

Metabolic stability is one of several major concerns in defining the oral bioavailability and systemic clearance of a drug. After a drug is administered orally, it first encounters metabolic enzymes in the gastrointestinal lumen and the intestinal epithelium. After it is absorbed into the bloodstream through the intestinal epithelium, it is delivered to the liver via the portal vein. A drug can be effectively cleared by intestinal or hepatic metabolism before it reaches systemic circulation, a process known as first-pass metabolism. The stability of a compound toward metabolism within the liver as well as extrahepatic tissues will ultimately determine the concentration of the drug found in the systemic circulation and affect its half-life and residence time within the body.

5.1.2 Cytochrome P450s

The cytochrome P450s (CYP) are a superfamily of enzymes which are found in all forms of living organism. They are responsible for the metabolism of many

endogenous compounds, participate in the activation/deactivation of many carcinogens and detoxify many xenobiotics. In particular, in human body they metabolize many drugs and hence are of great interest to pharmacologists and toxicologists.

The cytochrome P450 mixed function monooxygenases are located on the smooth endoplasmic reticulum of cells throughout the body, but the highest concentrations are found in the liver (hepatocytes) and small intestine[86]. These enzymes are responsible for the oxidative (Phase I) metabolism of a large number of compounds, including many medications. They biotransform lipophilic drugs to more polar compounds that can be excreted by the kidneys[87]. The metabolites are usually less active than the parent compound, although some drugs undergo biotransformation to become pharmacologic active agents. In some cases the metabolites can be toxic, carcinogenic or teratogenic[87].

At least 12 cytochrome P-450 gene families have been identified in humans, although only 3 families are involved in the majority of the drug biotransformations; they are the cytochrome P-450 1, 2 and 3 (CYP1, CYP2 and CYP3). A single hepatocyte can contain a variety of cytochrome P-450 enzymes. An individual enzyme of cytochrome P-450 may be able to metabolize many different drugs, but a given drug may be primarily metabolized by a single enzyme[87].

Members of the CYP3A subfamily are the most abundant cytochrome enzymes in human, accounting for 30% of the cytochrome enzymes in the liver and 70% of those in the gut. CYP3A4 is the major form of cytochrome P-450 in the adult liver and metabolizes the greatest proportions of drugs[87]. This enzyme and CYP3A3, which are 97% identical and cannot be distinguished from each other based on the substrates that they metabolize, are the major enzymes expressed in the small intestine, while CYP3A5 is the major enzyme expressed in the stomach[88]. CYP3A5 is present in only 20% $\sim$ 30% of Caucasians, but being deficient in CYP3A5 poses

no problem because the CYP3A4 enzyme is available to assume its functions[89].

5.1.3 CYP3A4 metabolism-based drug interactions

Metabolic stability of a drug is a major factor that will ultimately determine the concentration of the drug found in the systemic circulation. It is quite common for two or more drugs to be co-administered to a patient to increase the chances for a drug-drug interaction to occur. Many drug-drug interactions are metabolism-based and result from two or more drugs competing for the same enzyme, with the majority of these interactions involving CYP[90, 92]. For example, if a new chemical entity is a potent cytochrome P450 inhibitor, it may inhibit the metabolism of a co-administered medication. Thus, much higher plasma concentrations of the second drug are attained. For a drug with a narrow therapeutic index, this would lead to an adverse reaction. This problem has prompted the need to assess drug safety early during drug discovery/development, and to identify and eliminate compounds that may exhibit a potential for undesirable drug interactions. Assessing the safety of new drug candidates during drug discovery can save considerable amount of time and money, and prevent the exposure of patients to unnecessary risk, especially if a drug must later be removed from the market because of safety issues[91].

As the most abundantly expressed CYP isoform, CYP3A4 is responsible for the metabolism of more than 50% of pharmaceuticals[92, 93, 94]. As a consequence many important drug-drug interactions observed in the clinic are associated with drugs which are principally metabolized by CYP3A4. The two major reasons for drug-drug interactions involving CYP3A4 are induction and inhibition, with inhibition appearing to be the more important in terms of known clinical problems.

As a means of avoiding disasters in vivo drug interactions, the FDA requires identification of the specific metabolic pathways from which potential inhibition or

induction interactions may be inferred and, most recently, the effect of the new drug on hepatic P450. As a consequence, many pharmaceutical companies employ in vitro drug-drug assays early in drug discovery to predict potential interactions of new drug candidates in an attempt to minimize undesirable characteristics associated with novel compounds. Many articles have been published over the past several years outlining the advances in high-throughput CYP inhibition screens.

5.2 Data preparation

Table 5.1: The training and testing data sets

Data Set	No. of positive examples	No. of negative examples	Total No.
Training set	140	1180	1320
Testing set	37	318	355
Total No.	177	1498	1675

In this chapter, we mainly focus on the inhibitor prediction of the most important representative of the CYP family—CYP3A4. In order to build machine learning models, 194 CYP3A4 inhibitors are first collected from available literature from PubMed. Then a total of 1795 negative examples (non-CYP3A4-inhibitor chemicals) are selected from the inhibitors of a group of "negative proteins". The method of defining negative proteins is adopted from C.Z. Cai et al[60]. Specifically these negative proteins are selected from seed proteins of the curated protein families in the Pfam database[1]. Those seed proteins known to not belong to the family of CYP3A4

[1]Pfam (http://www.sanger.ac.uk/Software/Pfam/) is a large collection of multiple sequence alignments and hidden Markov models covering many common protein domains and families.

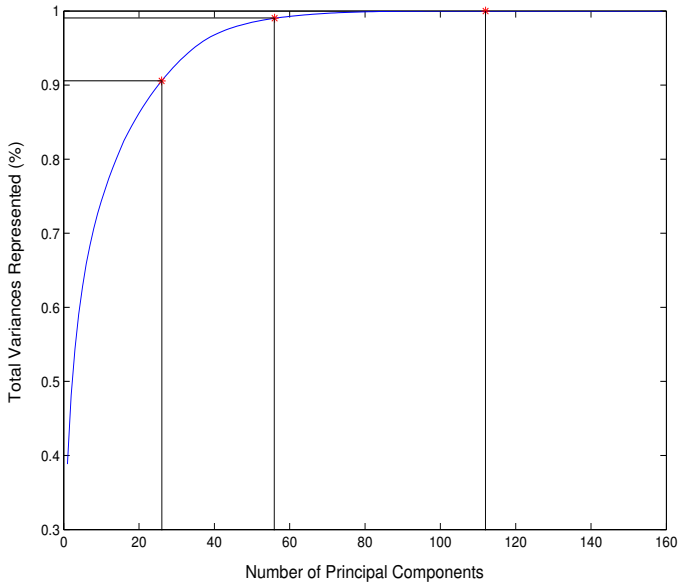

Figure 5.1: The number of principal components VS the percentage of total variance they represent.

(EC 1.14) are used as negative proteins. Negative protein representatives include proteins in all protein families other than EC 1.14. After collecting the positive and negative examples, the 3D structures of the inhibitors and non-inhibitors are generated by Concord based on their 2D structures.

After going through the same data purification procedures as described in Section 3.2, a 159-dimensional data set is successfully generated with 177 positive examples and 1498 negative examples. This data set is further split into training and testing sets. After scaling and PCA preprocessing, it is ready to construct different classification models.

Figure 5.1 shows the principal component analysis results of the training data. From it, we can see that without loss of any variances encoded in the data set, we can remove 47 dimension redundant descriptors. Keeping 99% of the variances, 103 dimensions can be removed. Keeping 90% of the variances, 133 dimensions can be

removed. In consistence with the approach used in previous chapters, I choose to build new training and testing data sets with the first n principal components while $n = 26$, $n = 56$, and $n = 112$.

5.3 Prediction results and analysis

The original data set and the PCA processed data sets consisting of the first 112, 56 and 26 principal components are analyzed by different machine learning approaches. The results are presented as follows.

5.3.1 Decision tree prediction

Table 5.2: The most frequently used descriptors

No.	Name	Description
6	N_F	Count of F atoms
12	N_S	Count of S atoms
17	N_{donr}	Number of H-bond donors
27	${}^5\chi_{CH}$	Simple molecular connectivity Chi indices for cycles of 5 atoms
75	$S(25)$	Atom-type Estate sum for =C<
86	$S(36)$	Atom-type Estate sum for >N-
106	$S(56)$	Atom-type Estate sum for -S-
134	q^+	Atomic charge on the most positively charged H atom
136	μ	Molecular dipole moment

The unpruned decision tree C4.5 generated for the original data is shown in Figure 5.2. This chart follows the convention described in section 3.3. A prominent characteristic of the tree is that it is much more complicated, compared to the decision trees from previous two chapters. It has 76 nodes and 20 layers of branches at most, while the tree generated for cholinesterase inhibitors has 43 nodes and 15

```
x138 <= -0.73689:
|   x44 <= -0.95044:
|   |   x64 > 0.17902: 1 (2)
|   |   x64 <= 0.17902:
|   |   |   x77 > -0.72139: 1 (2)
|   |   |   x77 <= -0.72139:
|   |   |   |   x4 > -0.92: -1 (17/1)
|   |   |   |   x4 <= -0.92:
|   |   |   |   |   x13 > -1: 1 (6)
|   |   |   |   |   x13 <= -1:
|   |   |   |   |   |   x141 <= -0.253: 1 (2)
|   |   |   |   |   |   x141 > -0.253: -1 (4)
|   x44 > -0.95044:
|   |   x26 > -1: -1 (39)
|   |   x26 <= -1:
|   |   |   x104 <= 0.67672: -1 (84)
|   |   |   x104 > 0.67672:
|   |   |   |   x134 <= 0.14206:
|   |   |   |   |   x75 <= 0.016282: -1 (121)
|   |   |   |   |   x75 > 0.016282:
|   |   |   |   |   |   x154 <= -0.74323:
|   |   |   |   |   |   |   x48 > 0.46711: 1 (5)
|   |   |   |   |   |   |   x48 <= 0.46711:
|   |   |   |   |   |   |   |   x87 > 0.022718: 1 (4/1)
|   |   |   |   |   |   |   |   x87 <= 0.022718:
|   |   |   |   |   |   |   |   |   x152 > -0.70774: 1 (4/1)
|   |   |   |   |   |   |   |   |   x152 <= -0.70774:
|   |   |   |   |   |   |   |   |   |   x90 <= -0.89483: -1 (35)
|   |   |   |   |   |   |   |   |   |   x90 > -0.89483:
|   |   |   |   |   |   |   |   |   |   |   x22 <= -0.82857: -1 (5)
|   |   |   |   |   |   |   |   |   |   |   x22 > -0.82857: 1 (2)
|   |   |   |   |   |   x154 > -0.74323:
|   |   |   |   |   |   |   x132 > 0.1728: -1 (124)
|   |   |   |   |   |   |   x132 <= 0.1728:
|   |   |   |   |   |   |   |   x7 > -0.66667: -1 (44/1)
|   |   |   |   |   |   |   |   x7 <= -0.66667:
|   |   |   |   |   |   |   |   |   x14 <= -0.78947:
|   |   |   |   |   |   |   |   |   |   x6 > -1: -1 (35/1)
|   |   |   |   |   |   |   |   |   |   x6 <= -1:
|   |   |   |   |   |   |   |   |   |   |   x7 <= -1:
|   |   |   |   |   |   |   |   |   |   |   |   x37 > -0.59181: -1 (28)
|   |   |   |   |   |   |   |   |   |   |   |   x37 <= -0.59181:
|   |   |   |   |   |   |   |   |   |   |   |   |   x134 <= -0.38004:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x158 <= -0.90322: -1 (4)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x158 > -0.90322: 1 (7/1)
|   |   |   |   |   |   |   |   |   |   |   |   |   x134 > -0.38004:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x14 <= -1:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x58 > -0.43954: 1 (3/1)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x58 <= -0.43954:[S0]
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x14 > -1:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x14 <= -0.89474:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x76 <= -0.11933:[S1]
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x76 > -0.11933:[S2]
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x14 > -0.89474:[S3]
|   |   |   |   |   |   |   |   |   |   |   |   |   x7 > -1:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x106 > 0.28361: -1 (3/1)
|   |   |   |   |   |   |   |   |   |   |   |   |   x106 <= 0.28361:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x68 <= -0.54189: -1 (40)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x68 > -0.54189:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x16 <= -0.88889: 1 (2)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x16 > -0.88889: -1 (2)
|   |   |   |   |   |   |   |   |   |   |   |   x14 > -0.78947:
|   |   |   |   |   |   |   |   |   |   |   |   x88 <= -0.081802:
|   |   |   |   |   |   |   |   |   |   |   |   |   x88 > -1: -1 (12)
|   |   |   |   |   |   |   |   |   |   |   |   |   x88 <= -1:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x6 <= -1:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x7 > -1: -1 (35/2)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x7 <= -1:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x14 <= -0.68421: -1 (73)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x14 > -0.68421:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x12 > -1: -1 (5)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   x12 <= -1:[S4]
|   |   |   |   |   |   |   |   |   |   |   |   |   x6 > -1:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x155 <= -0.83351: -1 (22)
|   |   |   |   |   |   |   |   |   |   |   |   |   x155 > -0.83351:
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x17 <= -0.90476: 1 (2)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   x17 > -0.90476: -1 (2)
|   |   |   |   |   |   |   |   |   |   x88 > -0.081802:
```

Figure 5.2: The decision tree generated for cholinesterase inhibitors

```
|   |   |   |   |   |   |   |   |   |   |   x72 <= 0.90832: -1 (10/1)
|   |   |   |   |   |   |   |   |   |   |   x72 > 0.90832: 1 (3)
|   |   |   |   x134 > 0.14206:
|   |   |   |   |   x11 > -1: -1 (13)
|   |   |   |   |   x11 <= -1:
|   |   |   |   |   |   x14 <= -0.89474:
|   |   |   |   |   |   |   x12 > -1: -1 (2)
|   |   |   |   |   |   |   x12 <= -1:
|   |   |   |   |   |   |   |   x72 <= 0.9047:
|   |   |   |   |   |   |   |   |   x70 <= -0.57526: 1 (8)
|   |   |   |   |   |   |   |   |   x70 > -0.57526: -1 (3/1)
|   |   |   |   |   |   |   |   x72 > 0.9047:
|   |   |   |   |   |   |   |   |   x136 <= -0.76869: -1 (6)
|   |   |   |   |   |   |   |   |   x136 > -0.76869: 1 (2)
|   |   |   |   |   |   x14 > -0.89474:
|   |   |   |   |   |   |   x55 <= -0.82466:
|   |   |   |   |   |   |   |   x52 <= -0.85117:
|   |   |   |   |   |   |   |   |   x70 <= -0.74789:
|   |   |   |   |   |   |   |   |   |   x15 <= -0.53846: -1 (56)
|   |   |   |   |   |   |   |   |   |   x15 > -0.53846:
|   |   |   |   |   |   |   |   |   |   |   x136 <= -0.90627: 1 (2)
|   |   |   |   |   |   |   |   |   |   |   x136 > -0.90627: -1 (29/1)
|   |   |   |   |   |   |   |   |   x70 > -0.74789:
|   |   |   |   |   |   |   |   |   |   x52 <= -0.95024: -1 (5)
|   |   |   |   |   |   |   |   |   |   x52 > -0.95024: 1 (3)
|   |   |   |   |   |   |   |   x52 > -0.85117:
|   |   |   |   |   |   |   |   |   x86 > -0.027022: 1 (5)
|   |   |   |   |   |   |   |   |   x86 <= -0.027022:
|   |   |   |   |   |   |   |   |   |   x23 <= -0.87964: 1 (2)
|   |   |   |   |   |   |   |   |   |   x23 > -0.87964: -1 (10/1)
|   |   |   |   |   |   |   x55 > -0.82466:
|   |   |   |   |   |   |   |   x3 <= -1: -1 (3)
|   |   |   |   |   |   |   |   x3 > -1: 1 (3)
x138 > -0.73689:
|   x74 > -0.54223: 1 (10)
|   x74 <= -0.54223:
|   |   x134 <= 0.17267:
|   |   |   x2 <= -0.88321: 1 (8)
|   |   |   x2 > -0.88321:
|   |   |   |   x11 <= -1: -1 (68/3)
|   |   |   |   x11 > -1:
|   |   |   |   |   x12 <= -0.75: -1 (3)
|   |   |   |   |   x12 > -0.75: 1 (2)
|   |   x134 > 0.17267:
|   |   |   x104 <= 0.64507: -1 (5)
|   |   |   x104 > 0.64507: 1 (20/1)

Subtree [S0]
x60 <= -0.46496: 1 (2)
x60 > -0.46496:
|   x137 > -0.025157: -1 (28)
|   x137 <= -0.025157:
|   |   x59 <= -0.45683: -1 (6)
|   |   x59 > -0.45683: 1 (3)

Subtree [S1]
x72 <= 0.89801: -1 (3)
x72 > 0.89801: 1 (5)

Subtree [S2]
x12 > -1: -1 (9)
x12 <= -1:
|   x25 > -1: -1 (4)
|   x25 <= -1:
|   |   x27 > -1: -1 (21)
|   |   x27 <= -1:
|   |   |   x86 <= 0.4134: -1 (38/2)
|   |   |   x86 > 0.4134: 1 (3/1)

Subtree [S3]
x18 > -0.95918: -1 (57/2)
x18 <= -0.95918:
|   x17 > -0.90476: 1 (2)
|   x17 <= -0.90476:
|   |   x62 <= 0.9721: -1 (11)
|   |   x62 > 0.9721: 1 (2)

Subtree [S4]
x3 > -1: -1 (4)
x3 <= -1:
|   x63 > 0.7921: -1 (62/1)
|   x63 <= 0.7921:
|   |   x1 <= -0.34357: 1 (2)
|   |   x1 > -0.34357: -1 (2)
```

Figure 5.3: The decision tree generated for cholinesterase inhibitors(Continued)

layers of branches and the tree for 5-HT_2 antagonist has only 37 nodes and 8 layers.

Table 5.3: Summary of frequently used descriptors

Descriptors used twice (24 in total)					
2 (A, C)	4 (A, C)	7 (A, C)	13 (B, C)	14 (A, C)	16 (B, C)
25 (B, C)	48 (A, C)	49 (A, B)	55 (B, C)	59 (B, C)	60 (B, C)
62 (A, C)	63 (B, C)	64 (A, C)	68 (A, C)	70 (B, C)	74 (B, C)
87 (B, C)	132 (A, C)	133 (A, B)	135 (A, B)	138 (A, C)	154 (B, C)
Descriptors used once (38 in total)					
1 (C)	3 (C)	11 (C)	15 (C)	20 (B)	22 (C)
23 (C)	24 (B)	26 (C)	29 (A)	37 (C)	41 (B)
44 (C)	47 (A)	52 (C)	56 (B)	58 (C)	72 (C)
76 (C)	77 (C)	80 (B)	84 (B)	85 (B)	88 (C)
89 (A)	90 (C)	91 (A)	92 (B)	93 (A)	104 (C)
137 (C)	141 (C)	147 (A)	148 (A)	152 (C)	153 (A)
154 (C)	158 (C)				

Table 5.4: Summary of the decision tree's performance on original and PCA pre-processed data sets

Measurement	On original data	On the data sets after PCA		
		112-dimension	56-dimension	26-dimension
Precision	45.16%	48.28%	46.43%	62.50%
Recall/Sensitivity	37.84%	37.84%	35.14%	13.51%
F measure	41.18%	42.43%	40.00%	22.22%
Specificity	94.65%	95.28%	95.28%	99.06%

A further analysis of the descriptors shows that the decision tree for CYP3A4

inhibitors uses 53 descriptors, the tree for cholinesterase inhibitors uses 32 descriptors, and the tree for 5-HT_2 antagonist uses 30 descriptors. There are 9 common descriptors used by all the three decision trees. Table 5.2 gives the summary list. There are 24 descriptors appeared twice and 38 descriptors appeared once in the three decision trees. Due to the space limitation, only their serial numbers are listed in Table 5.3. The letter in the bracket tells where the descriptor appears. "A" stands for 5-HT_2 antagonist prediction, "B" stands for cholinesterase inhibitor prediction and "C", for CYP3A4 inhibitor prediction.

Nonetheless, the complexity compromises the generalization capability of the tree, which is illustrated by the prediction on the unseen testing data set. In table 5.4, the summary of the results on testing data sets is presented. The precision, recall, F measure, sensitivity and specificity are compared on the four data sets. From it, we can see that although the prediction for negative examples are very high (the specificity values are all above 90%), this complex decision tree could not give an acceptable prediction accuracy for positive examples, which are our main focus. The recall / sensitivity value are only 37.84%, 37.84%, 35.14% and 13.51% for the original data set, the 112-dimension PCs data set, the 56-dimension PCs data set, and the 26-dimension PCs data set respectively. This means among 37 positive examples in the testing data set, only 14, 14, 13 and 5 are predicted correctly for the four data sets.

The poor performance on positive examples may be explained by the dramatic increase of the complexicity in the training data. The positive class and negative class probably overlapped a lot in the input space. The following *k*nn and SVM prediction prove this idea.

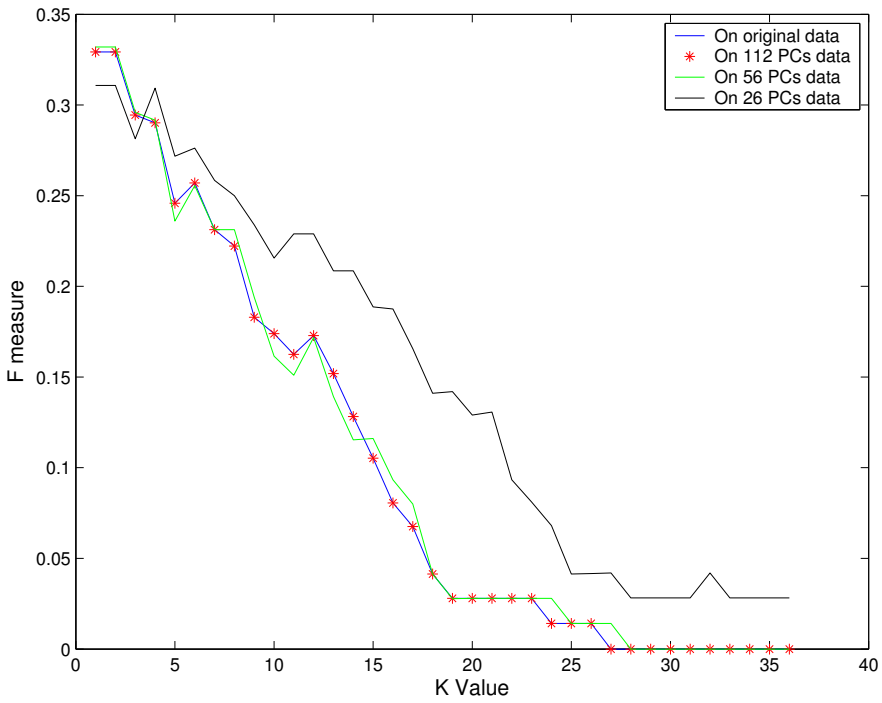

Figure 5.4: The $k$nn parameter tuning process based on 10-fold cross validation

5.3.2 K-Near Neighbor prediction

During training, the number of how many nearest neighbors to use, denoted by k is scanned in the range of $1, 2, \ldots, 36$. 36 is calculated as the square root of approximately 1320, the number of training examples. Figure 5.4 gives the plot of $Fmeasure$ against k. From it, we can see that the overall trend is that the bigger the k value, the smaller the $Fmeasure$. The maximum $Fmeasure$ are reached when k is set to 1 in all the four cases. The best $Fmeasure$ on the original data, 112-dimension PCs data, 56-dimension PCs data and 26-dimension PCs data are 32.92%, 32.92%, 33.20% and 31.08% respectively.

Finally the classification results are measured on testing data sets with their optimal k values (See Table 5.5). Obviously $k$nn gives better overall prediction than decision tree. The improvement for positive data are most obvious. On original

Table 5.5: Summary of the $k$nn's performance on original and PCA preprocessed testing sets

Measurement	On original data	On the data sets after PCA		
		112-dimension	56-dimension	26-dimension
Precision	84.00%	84.00%	83.33%	75.00%
Recall/Sensitivity	56.76%	56.76%	54.05%	56.76%
F measure	67.74%	67.74%	65.57%	64.00%
Specificity	98.74%	98.74%	98.74%	97.80%

data set and 112-dimension PCs data set, the Sensitivity (Recall) value increased by about 20%, on 56-dimension and 26-dimension PCs data sets, increased even more. But due to the shortage of positive data, their prediction is still much worse than that of negative data. The Specificity values are all around 98%, more than 40% higher than the Sensitivity values. This applies to the decision tree prediction too. But different from the previous experiments, all the four data sets give similar results this time. The original data set and 112-dimension PCs data give exactly the same performance, slightly higher measurements than the other two data sets.

5.3.3 SVM prediction

SVMs are trained with the kernel parameter σ scanned in the range of $[1, \ldots, 150]$ with an interval of 2. Such a large range is scanned because this time for three data sets of four, the $Fmeasure$ shows a dramatic increase trend with respect to σ in the range of $[0, \ldots, 8]$, but as σ gets larger, the performances gradually become stable with only a small fluctuation. Only a large scan range of σ can give a whole picture

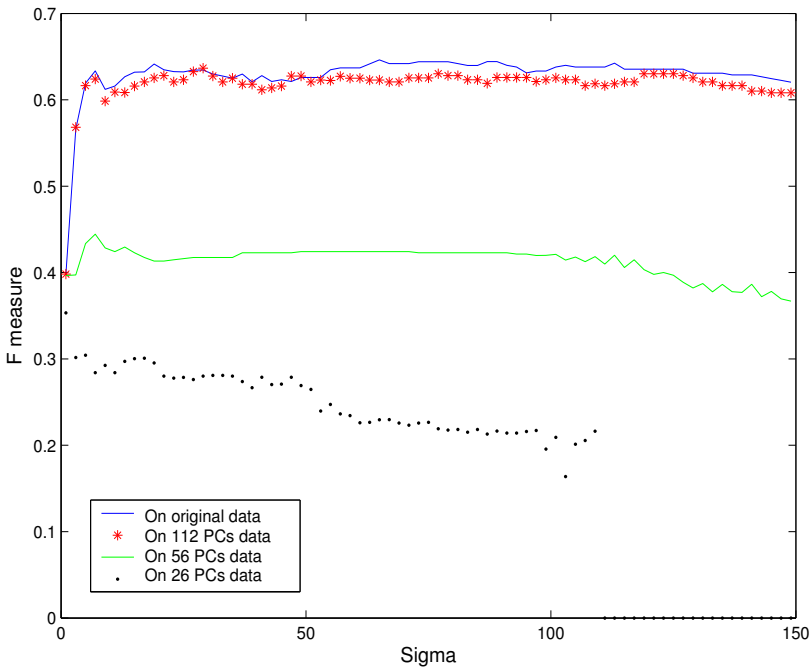

Figure 5.5: The SVM parameter tuning process based on 10-fold cross validation

of the trend. $Fmeasure$ are plotted against σ for the four training sets in Figure 5.5. The best σ found are 33, 15, 4 and 1 respectively. The classification performance are measured on four testing sets with their optimal σ values. Table 5.6 gives the summary.

Such result is inferior to its counterparts presented in Chapter 3 and Chapter 4. As far as $Fmeasure$ is concerned, the best $Fmeasure$ for CYP3A4 data set is about 10% less than that of cholinesterase, and more than 12% less than that of $5HT_2$ (See Figure 5.6). And the best Recall (Sensitivity) also decrease 8% and 10% respectively, compared with that of $5HT_2$ and cholinesterase. The best measurements are achieved on original data set, then on 112-dimension PCs data set, 56-dimension PCs data set and 26-dimension PCs data set. This demonstrates that the CYP3A4 data set is really more complicated. Simple methods like decision tree and PCA do not work well on it.

Table 5.6: Summary of the SVM's performance on original and PCA preprocessed testing sets

Measurement	On original data	On the data sets after PCA		
		112-dimension	56-dimension	26-dimension
Precision	82.86%	83.87%	75.00%	79.17%
Recall/Sensitivity	78.38%	70.27%	64.86%	51.35%
F measure	80.81%	76.47%	69.56%	62.30%
Specificity	98.11%	98.43%	97.48%	98.43%

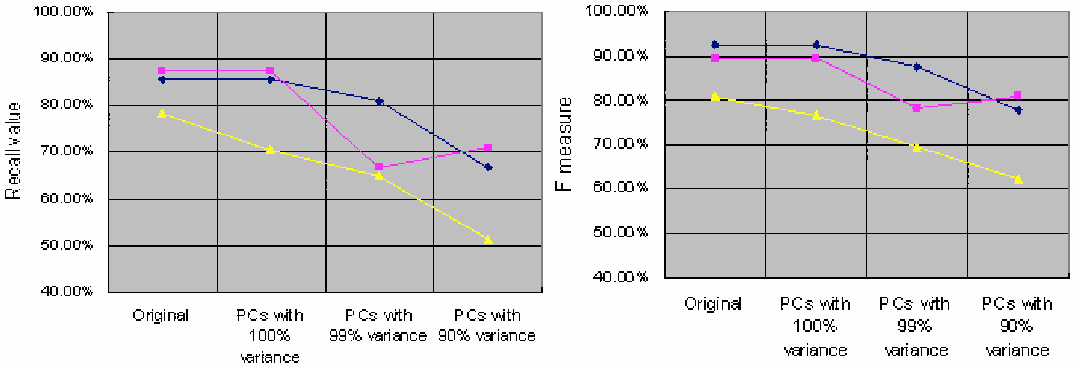

Figure 5.6: SVM results comparison on 3 data sets (Blue — $5HT_2$; Orange — Cholinesterase; Yellow — CYP3A4)

5.4 Summary

CYP3A4 is an important enzyme responsible for metabolism of more than 50% of oral drugs. In this chapter, decision tree, $k$nn and SVM are applied to analyze the potential of a compound to inhibit CYP3A4.

Results show that the prediction for CYP3A4 inhibitors is not as good as that of 5-HT_2 and cholinesterase. The decision tree does not work in this case. Its

prediction accuracy for positive data cannot reach 50%, which is less than random guesswork. $k$nn and SVM give just acceptable results. The best $Fmeasure$ are 67.74% and 80.81% respectively. PCA does not work well neither. 112-dimension PCs data which keeps 100% variance of information fail to give the same results as the original data set.

Possible reason is that CYP3A4 is a very broad functional enzyme. It can metabolize more than 50% of the oral pharmaceuticals. Therefore it is very possible that there are some examples in the negative sample data, which can interact with CYP3A4 as its substrates. On the other hand, its inhibitors may lack specificity in structures. Or it is very difficult to tell CPY3A4 inhibitors from its substrates.

So far, we have finished the main work in the thesis. Three inhibitor/antagonist prediction problems are explored for different proteins of different biological significance. Conclusion and reflections are made in next chapter on the basis of the results we have got in Chapter 3, 4 and 5.

Chapter 6

Conclusion and future work

The drug discovery process has evolved over the past 60 years from serendipitous findings of biologically active natural products, to rational design of potent and selective pharmacologically active compounds based on elucidation of three-dimensional structure of target proteins, to high-throughput screening against cloned and expressed enzymes and receptors, and to the construction of enormously diverse combinatorial libraries for ultra high-throughput screening.

However, even with rapid and efficient technologies to identify drug leads, it takes several years, in some cases up to 15 years, to bring a drug from discovery to market with an estimated price of US$880 million per individual drug. These high costs cannot be solely attributed to inflation or extensive clinical testing required by federal agencies; they also reflect the high rate of failure in the preclinical and clinical development of drugs. The ultimate success of a compound is defined not only by its biological activity and potency, but also by its ADME/toxicity related properties.

In this work, three machine learning methods, namely decision tree, k-nearest neighbor and support vector machine and preprocessing techniques such as normalization and principal component analysis were explored for potential application in

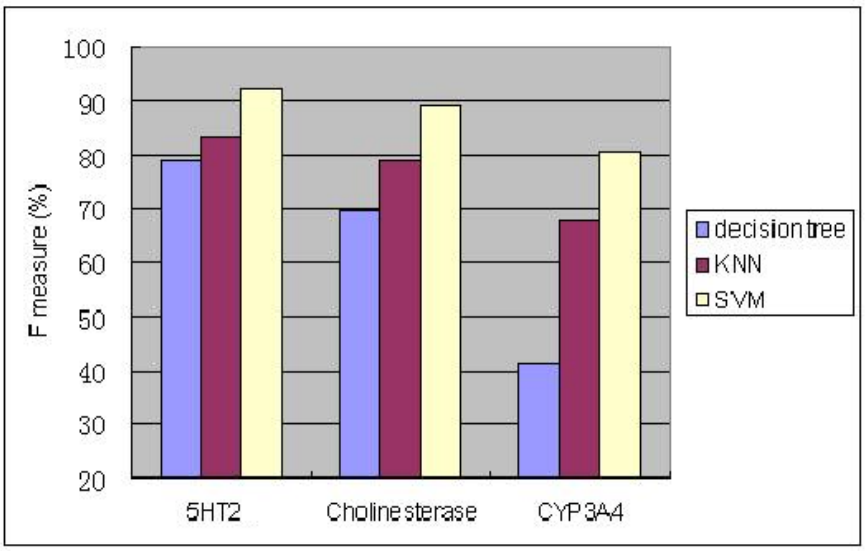

Figure 6.1: *Fmeasure* comparison on original data sets

drug lead identification and its ADME/toxicity analysis. Specifically, these machine learning methods were investigated to predict the inhibitors/antagonists for the receptor target—5-HT_2, the enzyme target—cholinesterase and the ADME related target—CYP3A4. The following results were obtained:

- Figure 6.1 shows the comparison among different machine learning approaches using *Fmeasure* on original data sets (PCA preprocessed data sets give similar results and therefore the figure is omitted). *Fmeasure* balances between how many positive examples are recognized and how many examples are true positive among those predicted positive ones. Thus it works as an effective indicator when tuning the free parameters in *k*nn and SVM, as well as an objective measurement for the capability of each classification model. It can be concluded that in this application, SVM always outperforms *k*nn and decision tree. In the case of 5-HT_2 antagonist prediction, SVM gives *Fmeasure* of 92.31% whilst *k*nn gives *Fmeasure* of 83.33% and decision tree, only 78.95%.

The difference becomes even bigger for cholinesterase and CYP3A4 inhibitor prediction.

- Simple models give better prediction than complex models. A good example is the decision trees generated for 5-HT_2 antagonist, cholinesterase inhibitor and CYP3A4 inhibitor prediction. On the original data sets, the decision tree generated for 5-HT_2 antagonist has 37 nodes in total and 8 layers of branches at most; the tree for cholinesterase inhibitor has 43 nodes and 15 layers of branches at most; and the tree for CYP3A4 inhibitor has 76 nodes and 20 layers of branches at most. By contrast, the $Fmeasure$ for 5-HT_2 antagonist, cholinesterase inhibitor and CYP3A4 inhibitor prediction are 78.95%, 69.77% and 41.18% respectively, in a dramatic decreasing trend. The $k$nn and SVM predictions show the same trend too (See Figure 6.1). On the one hand, this phenomenon once again confirmed the Ockham's razaor theorem. On the other hand, this may reflect the different complexities and quality within the three data sets.

- With the exception of SVM prediction on CYP3A4 inhibitors, most PCA preprocessed data sets with 100% variances give the same, or slightly better results as the original data sets. These results are also the best among the original data set, PCA data set with 100%, 99% and 90% variances. This demonstrates that PCA is helpful to reduce the dimension of the data sets and speed up the CPU computation without compromising in the performance. This also gives us a hint that the descriptors we have used are capable of catching the necessary characteristics of thousands of compounds with regard to inhibitor classification. But there may be certain degree of redundancy. Around 46$\sim$50 descriptors can be removed without losing any variances in the original data. Of course, in order to further improve the prediction accuracy,

new descriptors which can describe and distinguish compounds from a different perspective may be introduced.

- For the three inhibitor prediction problems, all the three machine learning approaches give significantly better prediction on negative examples than on positive examples. This may be due to the fact that the number of negative examples is more than ten times that of positive examples. That is the negative examples might represent a more comprehensive sample in the input space and provide more useful information about its classification.

  In order to improve the prediction performance of positive examples, more positive examples in training data sets are desirable which can cover more space in the input space. If equal numbers of positive and negative examples of equal quality are given in the training data set, the difference between the prediction accuracy on the two classes would become much smaller. This is determined by the nature of machine learning algorithms. Currently almost all the machine learning methods for two-class classification problems assume that both of the two classes are well studied and equally distributed in the hyper space. Therefore, unbalanced training data set will result in a classifier that is better trained for one class over the other class.

  In addition to adding more examples to balance the training set, new algorithms are expected which can give different consideration to different classes in very unbalanced data sets, like our cases in which the positive examples are more than ten times that of negative examples.

The results obtained are quite exciting, which illuminate a novel approach towards drug lead identification and toxicity/ADME analysis at early stage. However, the methods adopted in this work are far from perfect. Further study is needed to

deepen our understanding of the different chains in the machine learning screening pipeline. Several directions are worth of our further exploration, such as:

- Feature selection

  Using all the available descriptors to build statistical models may not be helpful, as the inclusion of irrelevant descriptors can cause the classical problem of the "curse of dimensionality"[95], which leads to expensive computation and compromised generalization performance. The selection of a good set of features is an essential prerequisite in the design of an accurate predictive model.

  From the PCA analysis in this work, it is noted that there is certain redundancy in the 159 QSAR descriptors used. For example, the first descriptor "molecular weight" can be calculated linearly from the value of count of hydrogen atoms, halogen atoms, ..., which correspond to the 2nd to 14th descriptors. Besides, the 111th to 125th descriptors describe the electrotopological state (E state) for the elements of Germanium (Ge), Arsenic (As) and Selenium (Se), which are seldom shown in drug compounds. Therefore these kind of descriptors give little useful information in the application of drug lead prediction. In order to improve the performance of these machine learning models, further work are needed to select a set of more informative descriptors, which requires strong knowledge in chemistry and QSAR.

- SVM regression

  In this work, the SVM classification method has been successfully applied to inhibitor prediction. But, in essence, protein-ligand binding is not a simple binary classification problem. Different inhibitors may have different binding affinity. Chemical and biological experiments have provided some quantitative

data regarding the binding affinity of various compounds to a particular protein, such as effective concentration 50% (EC50 value)[1], which reveal valuable information to us. Once this kind of information is available, SVM regression may be a more powerful tool to analyze quantitatively the potential of a compound to become an inhibitor of a particular protein. Song M. H et al[96] has reported to use SVM regression in protein retention time prediction.

- Rule extraction

  A limitation of SVM is that it generates black-box-like models, which are not interpretable to medicinal chemists. Haydemar et al has reported a "prototype" based method to derive "if–then" rules from SVM models[97]. Another possible approach to look into the black box might be training a decision tree that mimics the behavior of SVM.

After solving these problems, an expert system to facilitate the decision making of the pharmaceutical industry during lead discovery can be built in the future. Currently most machine learning scientists only focus on how to improve the performance of algorithms based on measurements of recall/sensitivity and precision/specificity et al, without any consideration of the particular field in which an application lies. The ultimate goal of drug design software should be to find an optimal experimental design during the lead discovery period, in order to maximize pharmaceutical companies' profit with lowest risk. Therefore, the profit and risk analysis should be taken into consideration during model building. That is, the prediction accuracy on both classes should be given different weights. Figure 6.2 gives the flowchart of a simple expert system. Some operation research knowledge should

[1]The EC50 is commonly defined as the drug concentration at which the response has decreased (increased) to 50% of the initial response, assuming that the response is a decreasing (increasing) function of drug concentrate.

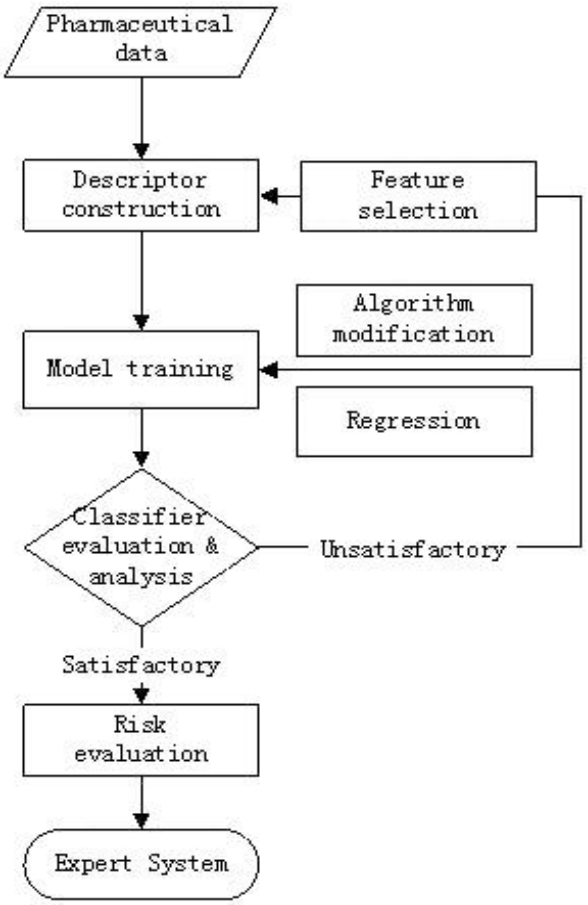

Figure 6.2: Blueprint for an expert system to facilitate the decision making of the pharmaceutical industry during lead discovery.

be introduced to the final model selection step to get the most practical model.

Bibliography

[1] DiMasi J. A., Hansen R. W., Grabowski H. G. The price of innovation: new estimates of drug development costs. *J Health Economics*, vol.22, 151–185, 2003.

[2] DiMasi J. A. Risks in new drug development: approval success rates for investigation drugs. *Clin Pharmacol Ther*, vol.69, 297–307, 2001.

[3] Hertzberg R. P, Pope A. J. High–throuphput screening: new technology for the 21st century. *Curr Opin Chem Biol*, vol.4, 445–451, 2000

[4] Wolcke J, Ullmann D. Miniatured HTS technologies–uHTS. *Drug Discov Today*, vol.6, 637-646, 2001

[5] Drews J. Drug discovery today and tommorrow. *Drug Discov Today*, vol.5, 2–4, 2000

[6] Schneider G, Bohm H. J. Virtual screening and fast automated docking methods. *Drug Discov Today*, vol.7, 64–70, 2002

[7] Toledo–Sherman L. M, Chen D. High–throughput virtual screening for drug discovery in parallel. *Curr Opin Drug Discov Devel*, vol.5, 414–421, 2002

[8] Kuntz I. D, Blaney J. M, Oatley S. J *et al* A Geometric Approach to Macromolecule–ligand Interaction. *J Mol Biol*, vol.161, 269–279, 1982

[9] Alexander Hillisch and Rolf Hilgenfield. Modern Methods of Drug Discovery. Birkhauser Verlag, 2003

[10] Maurizio Pellecchia, Daniel S. Sem and Kurt Wuthrich. NMR in Drug Discovery *Nature Reviews Drug Discovery*, vol.1, 211–219, 2002

[11] Van Drie J. H. Pharmacophore discovery-lessons learned. *Current Pharmaceutical Design*, vol.9, 1649–1664, 2003

[12] Barreca M. L, Gitto R, Quartarone S, De Luca L, De Sarro G, and Chimirri A. Pharmacophore modeling as an efficient tool in the discovery of novel noncompetitive AMPA receptor antagonists. *J Chem Inf Comput Sci*, vol.43, 651–655, 2003

[13] Green D. V. Virtual screening of virtual libraries. *Prog Med Chem*, vol.41, 61–97, 2003

[14] Langer T and Krovat E. M. Chemical feature-based pharmacophores and virtual library screening for discovery of new leads. *Curr Opin Drug Discov Devel*, vol.6, 370–376, 2003

[15] Schleifer K. J and Tot E. Pharmacophore modelling of structurally unusual diltiazem mimics at L-type calcium channels. *J Comput Aided Mol Des*, vol.14, 427–433, 2000

[16] Springsteel M, Galietta L. J, Ma T, By K, Berger Go Yang H, and et al. Benzoflavone activators of the cystic fibrosis transmembrane conductance regulator: towards a pharmacophore model of flavone-CFTR binding. *Bioorg Med Chem*, vol.11, 4113–4120, 2003

[17] Akamatsu M. Current state and perspectives of 3D-QSAR. *Curr Top Med Chem*, vol.2, 1381–1394, 2002

[18] Smith P. A, Sorich M. J, McKinnon R. A, and Miners J. O. Pharmacophore and quantitative structure-activity relationship modeling: complementary approaches for the rationalization and prediction of UDP-glucuronosyltransferase 1A4 substrate selectivity. *J Med Chem*, vol.46, 1617–1626, 2003

[19] Brennan M. B. Drug Discovery: filtering out failures early in the game. *Chemical & Engineering News*, vol.78, 63–73, 2000

[20] Prentis R. A, Lis Y, Walker S. R. Pharmaceutical innovation by the seven UK-owned pharmaceutical companies (1964–1985)*Br J Clin Pharmacol*, vol.25, 387–396, 1988

[21] Cunningham M. J. Genomics and proteomics:The new millennium of drug discovery and development. *J Pharmacol Toxicol Methods*, vol.44, 291–300, 2000

[22] Drews J. and Ryser S. Classic Drug Targets: Special Pullout. *Nat Biotechnol*, vol.15, 1997

[23] Chen X., Ji Z. L. and Chen Y. Z. TTD: Therapeutic Target Database. *Nucleic Acids Res*, vol.30, 412–415, 2002

[24] Thomas M. Frimurer, Robert Bywater, Lars Narum and etc. Improving the Odds in Discriminating "Drug-like" from "Non-Drug-like" Compounds. *J. Chem. Inf. Comput. Sci*, vol.40, 1315–1324, 2000

[25] Markus Wagener and Vincent J. van Greerestein. Potential Drug and Nondrugs: Prediction and Identification of Important Structural Features. *J. Chem. Inf. Comput. Sci*, vol.40, 280–292, 2000

[26] Hall L. H., Kier L. B. The molecular connectivity chi indices and kappa shape indices in structure-property modeling. In: Lipkowitz, K. B., Boyd, D. B. (Eds.), *Reviews of Computational Chemistry*, vol.2, 367–412, 1991

[27] Kier L. B., Hall L. H. Molecular Connectivity in Structure-Activity Analysis. John Wiley and Sons, New York, 1986

[28] Hall L. H., Mohney B. K., Kier L. B. The electrotopological state: structure information at the atomic level for molecular graphs. *J. Chem. Inf. Comput. Sci.*, vol.31, 76, 1991

[29] Hall L. H., Mohney B. K., Kier L. B. Electrotopological state: An atom index for QSAR. *Quant. Struct.-Act. Relat.*, vol.10, 43, 1991

[30] Kellogg G. E., Kier L. B., Gaillard P., Hall L. H. The E-State fields–Applications to 3D QSAR. *J. Comp. Aid. Mol. Des.*, vol.10, 513–520, 1996

[31] Hall L. H., Kier L. B. Electrotopological state indices for atom types: a novel combination of electronic, topological, and valence state information. *J. Chem. Inf. Comput. Sci.*, vol.35, 1039-1045, 1995

[32] Famini G. R., Penski C. A., Wilson L. Y. Using theoretical descriptors in quantitative structure activity relationships: some physicochemical properties. *J. Phys. Org. Chem.*, vol.5, 395–408, 1992

[33] Famini G. R., Wilson L. Y. Using theoretical descriptors in quantitative structure-activity relationships: application to partition properties of alkyl(1-phenylsulfonyl) cycloalkane-carboxylates. *Chemosphere*, vol.35, 2417–2447, 1997

[34] Karelson M., Lobanov V. S. Quantum-chemical descriptors in QSAR/QSPR studies. *Chem. Rev.*, vol.96, 1027–1043, 1996

[35] Thanikaivelan P., Subramanian V., Raghava J., Rao J. R., Nair B. U. Application of quantum chemical descriptors in quantitative structure activity and structure property relationship. *Chem. Phys. Lett.* vol.323, 59–70, 2000

[36] Tsodikov O. V., Record M. T., Jr. Sergeev Y. V. Novel computer program for fast exact calculation of accessible and molecular surface areas and average surface curvature. *J. Comp. Chem.*, vol.23, 600–609, 2002

[37] Chang C. C., Lin C. J. LIBSVM: a Library for Support Vector Machines. Department of Computer Science and Information Engineering, National Taiwan University, 2003

[38] Bellman R. E. Adaptive control process: a guided tour. Princeton University Press, Princeton NJ, 1961

[39] Jolliffe I. T. Principal Component Analysis 2nd Edition, Springer-Verlag New York, 2002

[40] Tom Mitchell. Machine Learning. McGraw-Hill, 1997

[41] Quinlan J. R. Induction of decision trees. *Machine Learning*, vol.1(1), 86–106, 1986

[42] Breiman L., J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Wadsworth Inc, 1984

[43] Quinlan J. R. C4.5: Programs for machine learning. Morgan Kaufmann Publishers Inc, 1993

[44] Vapnik V. The Nature of Statistical Learning Theory. Springer-Verlag New York Inc, 1995

[45] Kim K.I., Jung K., Park S.H. and Kim H.J. Support vector machine-based text detection in digital video, *Pattern Recognition*, vol.34, 527–529, 2001

[46] de Vel O., Anderson A., Corney M. and Mohay G. Mining e-mail content for author identification forensics, *Sigmod Record*, vol.30, 55–64, 2001

[47] Vapnik V. N. Statistical learning theory. Wiley New York, 1998

[48] Ben-Yacoub S., Abdeljaoued Y. and Mayoraz E. Fusion of face and speech data for person identity verification. *Ieee Transactions on Neural Networks*, vol.10, 1065–1074, 1999

[49] Karlsen R. E., Gorsich D. J. and Gerhart G. R. Target classification via support vector machines. *Optical Engineering*, vol.39, 704–711, 2000

[50] Liong S. Y. and Sivapragasam C. Flood stage forecasting with support vector machines. *Journal of the American Water Resources Association*, vol.38, 173–186, 2002

[51] Brown M. P. S., Grundy W. N., Lin D., Cristianini N., Sugnet C. W. et al. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*. vol.97, 262–267, 2000

[52] Ding C. H. and Dubchak I. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, vol.17, 349–358, 2001

[53] Hua S. and Sun Z. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *J Mol Biol*, vol.308, 397–407, 2001

[54] Bock J. R. and Gough D. A. Predicting protein-protein interactions from primary structure. *Bioinformatics*, vol.17, 455–460, 2001

[55] Platt J. C. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Microsoft Research. Technical Report MSR-TR-98-14, 1998.

[56] Osuna E., Freund R. and Girosi F. An improved training algorithm for support vector machines. *Neural Networks for Signal Processing VII–Proceedings of the 1997 IEEE Workshop*, 276–285, 1997

[57] Tax D. M. J. and Duin R. P. W. Support Vector Domain Description. *Pattern Recognition Letters*, vol.20, 1191–1199, 1999

[58] Ron Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *the International Joint Conference on Artitificial Intelligence(IJCAI)*, 1995

[59] Cai C.Z., Han L.Y., Ji Z.L., Chen X., and Chen Y.Z. SVM-Prot: Web-Based Support Vector Machine Software for Functional Classification of a Protein from Its Primary Sequence. *Nucl. Acids Res.*, vol.31, 3692–3697, 2003

[60] Cai C.Z., Han L.Y., Ji Z.L., and Chen Y.Z. Enzyme Family Classification by Support Vector Machines. *Proteins*, vol.55, 66–76, 2004

[61] Olivier B., Vanwijngaarden I. and Soudijm W. Serotonin Receptors and Their Ligands. Elsevier, 1997.

[62] Peroutka S. J and Snyder S. H. Multiple serotonin receptors: differential binding of [^{3}H]5-hydroxytryptamine, [^{3}H]lysergic acid diethylamide and [^{3}H]spiroperidol. *Mol Pharmacol*, vol.16, 687–699, 1979

[63] Glennon R. A. and Malgorzata D. Serotonin Receptors and Drugs Affecting Serotonergic Neurotransmission. *Foye's Principals of Medicinal Chemistry*, Chapter.12, 317–337, 2002

[64] Olivier B., Van Wijingaarden I, Soudin W, et al. Serotonin receptors and their ligands. *Amsterdam: Elsevier*, 1997

[65] Zifa E. and Fillion G. 5-hydroxytryptamine receptors. *Pharmacol Rev*, vol.44, 401–458, 1992

[66] Kennett GA, Bailey F, Piper DC, et al. Effect of SB 200646A, a 5-HT_2C/5-HT_2B antagonist in two conflict models of anxiety. *Psychopharmacology*, vol.118, 178–182, 1995

[67] Cohen M. L., Fuller R. W., Wiley K. S. Evidence for 5-HT2 receptors mediating contraction in vascular smooth muscle *J Pharmacol Exp Ther*, vol.218, 421–425, 1981

[68] Watts S. W., Baez M., and Webb R. C. The 5-hydroxytryptamine$_{2B}$ receptor and 5-HT receptor signal transduction in mesenteric arteries from deoxycorticosterone acetate- salt hypertensive rats. *J Pharmacol Exp Ther*, vol.277, 1103–1113, 1996

[69] Duxon M. S, Flanian T. P, Reavley A. C, et al. Evidence for expression of the 5-hydroxytryptamine-2B receptor system in the rat central nervous system. *Neuroscience*, vol.76, 323–329, 1997

[70] Roth B, Willins D. L, Kristiansen K, et al. 5-hydroxytryptamine$_2$ family receptors (5-hydroxytryptamine$_{2A}$, 5-hydroxytryptamine$_{2B}$, 5-hydroxytryptamine$_{2C}$): Where structure meets function. *Pharmacol. Ther*, vol.79, 231–257, 1998

[71] Branchek T. Site-directed mutagenesis of serotonin receptors. *Med Chem Res*, vol.3, 287–296, 1993

[72] Westkaemper R. B. Guest Editor for Serotonin receptors: Molecular genetics and molecular modeling. *Special issue of Med Chem Res*, vol.3(5/6), 269–418, 1993

[73] Aulakh C. S., Mazzola-Pomietto P., Hill J. L., and Murphy D. L. Role of various 5-HT receptor subtypes in mediating neuroendocrine effects of 1-(2,5-dimethoxy-4-methylphenyl)-2-aminopropane (DOM) in rats. *J Pharmacol Exp Ther*. vol.271(1), 143–148, 1994

[74] Janssen P.A., Niemegeers C.J., Awouters F., Schellekens K. H., Megens A. A., and Meert T. F. Pharmacology of risperidone (R 64766), a new antipsychotic with serotonin-S2 and dopamine-D2 antagonistic properties. *J Pharmacol Exp Ther*, vol.244, 685–693, 1998

[75] Cai C.Z., Wang W.L., Sun L.Z., Chen Y.Z. Protein function classification via support vector machine approach. *Math Biosci*, vol.185, 111–122, 2003

[76] Sadowski J. Kubinyi H. A scoring scheme for discriminating between drugs and nondrugs. *J. Med. Chem*, vol.41, 3325–3329, 1998

[77] Kennard R. W. and Stone L. A. Computer aided design of experiments. *Technometrics*, vol.11, 137–149, 1969

[78] Wilson I. B., Bergmann F. and Nachmansohn D. Acetylcholinesterase X, Mechanism of the catalysis of acylation reactions. *J. Biol. Chem.*, vol.186, 781–790, 1950

[79] Massoulie J., Pezzementi L., Bon S., Krejci E. and Vallette F.-M. Molecular and Cellular Biology of Cholinesterases. *Prog. Neurobiol.*, vol.41, 31–91, 1993

[80] McGuire M., Nogueira C. P., Bartels C. F., Lightstone H., Hajra A. and et. al. Identification of the structural mutation responsible for the dibucane-resistant (atypical) variant form of human serum cholinesterase. *Proc. Natl. acad. Sci.*, vol.86, 853–957, 1989

[81] Augustinsson K. B. Cholinesterases and anticholinesterase agents. *Springer*, Berlin, 1963

[82] Small G. W., Rabins P. V., Barry P. P., et al. Diagnosis and treatment of Alzheimer's disease and related disorders: consensus statement of the American Association for Geriatric Psychiatry, the Alzheimer's Association, and the American Geriatrics Society. *JAMA*, vol.278, 1363–1371, 1997

[83] Perry R. J. and Hodges J. R. Relationship between functional and neuropsychological performance in early Alzheimer's disease. *Alzheimer Dis Assoc Disord*, vol.14, 1-10, 2000

[84] Mileson B. E., Chambers J. E., Chen W. L., Dettbarn W, Ehrich M, and et al. Common mechanism of toxicity: a case study of organophosphorus pesticides. *Toxicol Sci*, vol.41, 8–20, 1998

[85] Imbimbo B. P. Pharmacodynamic-Tolerability Relationships of Cholinesterase Inhibitors for Alzheimer's Disease. *CNS Drugs*, vol.15, 375–390, 2001

[86] Buck M. L. The cytochrome P450 enzyme system and its effect on drug metabolism. *Pediatric Pharmacotherapy*, vol.3, 1997

[87] Meyer J. M. & Rodvold K. A. Drug biotransformation by the cytochrome P-450 enzyme system. *Infect Med*, vol.13, 452,459,463-464,523, 1996

[88] Kolars J. C., Lown K. S., Schmielin-Ren P., Ghosh M., Fang C. CYP3A4 gene expression in gut epithelium. *Pharmacogenetics*, vol.4, 247–259, 1994

[89] Wrighton S. A., Ring B. J., Watkins P. B. & Vandenbranden M. Identification of a polymorphically expressed member of the human cytochrome P-450III family. *Mol Pharmacol*, vol.36, 97–105, 1989

[90] Murray M. P450 enzymes: Inhibition mechanisms, genetic regulation and effects of liver disease. *Clin Pharmacokinet*, vol.23, 132–146, 1992

[91] Mullins M. E, Horowitz B. Z, Linden D. H, Smith G. W, Norton R. L, Stump J. Life-threatening interaction of mibefradil and beta-blockers with dihydropyridine calcium channel blockers. *JAMA*, vol.280, 157–158, 1998

[92] Guengerich F. P. Characterization of human cytochrome P450 enzymes. *FASEB J*, vol.6, 745–748, 1992

[93] Guengerich F. P. Cytochrome P450: Advances and prospects. *FASEB J*, vol.6, 667–668, 1992

[94] Guengerich F. P. In vitro techniques for studying drug metabolism. *J Pharmacokinet Biopharm*, vol.24, 521–533, 1996

[95] Bellman R. Adaptive Control Processes: A Guided Tour, Princeton University Press. 1961

[96] Song M., Breneman C. M., Ji B., Sukumar N.,Bennett K. P., Cramer S. and Tugcu N. Prediction of Protein Retention Times in Anion-Exchange Chromatography Systems Using Support Vector Regression. *J. Chem. Inf. Comput. Sci.*, vol.42, 1347–1357, 2002

[97] Haydemar N., Cecilio A. and Andreu C. Rule extraction from support vector machines. *European Symposium on Artificial Neural Networks*, Bruges(Belgium), 2002

INHIBITOR PREDICTION BY MACHINE LEARNING APPROACHES

YAO LIXIA

NATIONAL UNIVERSITY OF SINGAPORE

2004