TEACHING STATEMENTS

Ya Zhang

To teach is to touch a life forever. Teaching is not only the dissemination of knowledge but to foster the desire of learning and help students to develop learning skills through planned guidance. I have been greatly impacted by the teachers I have had and are inspired to serve others with the gifts they had given me.

My practical teaching experience includes lab lecturing and teaching assistantships in several undergraduate and graduate core courses at Pennsylvania State University. I have served for various capacities, including weekly lab lectures, project and exercise design, and one-on-one and small group assistance. The courses span broad topics (databases, discrete mathematics, information retrieval, etc) and embody students with variant backgrounds, which gives me ample opportunities to practice teaching in a variety of settings.

I believe that learning is more effective when students are internally motivated about the subjects, and the teacher plays a major role in increasing students' interests and motivation. This principle has always been reflected in my lab lecturing. At the beginning of the lab lectures, I usually present some real-world applications of the technologies which are covered in the lectures. For example, when lecturing on Javascript, I started with showing several applications of Javascripts and impressed the students with what Javascripts can do. When the students gain interests in the technologies, their learning process becomes active and full of enthusiasm.

Knowledge is exploding in the new information era. As the half-life of knowledge becomes increasingly shorter, teaching should create an active learning environment for students to "learn how to learn". As a teaching assistant, helping student with their homework and projects is part of my responsibility. When students approach me for questions, I picture myself as one of their classmates. I always discuss the potential solutions with them rather than directly telling them the solutions. This principle is also reflected in lab lectures, I emphasize the importance of understanding the contents and "know how". The students are required to reflect their thinking in their solutions to homework and exams. Knowing how to find the solution is always considered more important than knowing what the solution is.

As technologies are best learned through examples and practice, my lecture notes are usually supported with abundant examples. For instance, the lab tutorial about SQL is filled with sample queries and each sample query is accompanied by explanations of the key points embedded in the query. Moreover, the students are often given some small problem sets to exercise at the end of each lab lecture. This strategy has been proven very effective and I have received good feedbacks from my students.

I believe that my teaching experience and my educational background have well prepared me for the journey of an academic position. My primary teaching interests are in bioinformatics, data mining and machine learning, at both undergraduate and graduate levels, but I would also enjoy teaching other fundamental computer science courses such as database, algorithm, and information retrieval. For the graduate level courses, I expect to integrate ideas from on-going research into lecture because bioinformatics is a fast developing area. I will help students to identify research topics, develop research methodology, and inspire their creativity. For the undergraduate courses, in addition to teaching students fundamental materials and fostering critical thinking, I will emphasize more on problem solving skills and integration of different subjects and frameworks. Students will be

provided with real-world case studies, which can provide them the opportunity of learning in practical contexts.

Another key component of teaching is student advising. Quite different from other fields in computer science, research in computational biology and bioinformatics traverses many disciplines, such as biology, computer science, mathematics, and statistics. Depending on their particular specialties and talents, different students will have different sets of skills and their own favorite ways of solving problems. The advisor ought to provide the students tailored guidance based on their background and interests. Moreover, the focus of research in bioinformatics and computational biology are fast shifting with advancements in biotechnology. A striking example is the shift of focus from DNA sequencing to genomics and proteomics due to the success of whole genome sequence techniques. Helping students to choose appropriate research problems to tackle and allowing students to maintain independence in research are important aspects of student advising.

In summary, I am looking forward to contributing to cultivate students into independent scientists and researchers since I regard teaching as a crucial component of an academic career. I have the motivation, the qualities and the necessary background to take the journey to academia.

Sample Teaching Material

Ya Zhang

Attached below is the tutorial for a SQL lab that I designed and used for the course *Organization of Data*. The purpose of this lab was to introduce advance SQL queries. Students had learned some fundamentals of SQL before the lab. Simply reciting the SQL syntax in class would be tedious, and students might lose interests in learning. My strategy is to include abundant examples in the tutorial. A sample database *ist210* was provided so that students could try the given examples. During the lab, I first provided an overview of the topics covered in the class, and then introduce each concept by examples.

IST 210 Lab 4 Tutorial: SQL: multi-table queries and Data Definition Language

Objective: To learn how to work with multiple tables using JOIN operator (inner join and outer join). We will also discuss subqueries.

Note: The examples throughout this tutorial used tables from ist210 database.

Inner Join

Inner Join, or we simply call *join*, operation combines information from two or more tables where the matching column/s in each of the table have the same value. To perform a join, we simply include more than one tables in the FROM clause, using a comma as a separator and specify the matching (or join) column/s in the WHERE clause.

Inner Join Using Where Clause

Example 1: Get students' information including student_id, names, department name and building.

```
SQL:    SELECT student_id, lname, fname, dept_name,
                CONCAT(building_no, ' ', building) AS Office
        FROM students, department
        WHERE students.dept_id = department.dept_id
        ORDER BY lname
```

Result:

student_id	lname	fname	dept_name	Office
871992345	Aniston	Cynthia	Statistics	210 Thomas
200776969	Barry	Keith	Psychology	322 Thomas
222343456	Brown	James	Statistics	210 Thomas
654934000	Cavalier	Jenifer	Marketing	411 BAB
343567892	Go	William	Psychology	322 Thomas
223656743	Hosch	Jason	Industrial Engineering	207 Hammond
124579943	Kumara	Tirupati	Statistics	210 Thomas
801761698	McDougal	John	Industrial Engineering	207 Hammond
322561903	McNabb	Jason	Computer Science	289 Hammond
365768907	Miller	Johannes	Physics	108 Davey
023457691	Montgomery	Donald	Marketing	411 BAB
334128799	Nicholas	Theodore	Computer Science	289 Hammond
455112324	Rider	Susan	Marketing	411 BAB
547682303	Santoro	Richard	Physics	108 Davey
876223698	Smith	Kelly	Computer Science	289 Hammond
561498810	Vicario	Hector	Industrial Engineering	207 Hammond

Note:
In this example we join students and department tables. As can be seen from the WHERE clause, the matching or join column is dept_id. The important points to remember in the above example are:

1. The semantics of the corresponding join columns must be identical. This means both columns must have the same logical meaning and column field **data type** has to be compatible (e.g. text column cannot be used as matching column for numeric column).
2. For readability of the design, the matching columns should be kept having a same name. Normally, the table name is used as a qualifier to differentiate the columns of different tables. In above example, students.dept_id and department.dept_id are used to specify the dept_id column from students table and department table, respectively.
3. CONCAT operator serves as a concatenation operator to join char/text data type. In the example, building and building no are joined together to form 'Office'. column.

Cartesian Product
In cartesian product, each row of one table is combined with each row of another table. In general, the cartesian product of two tables with n and m rows, respectively, will produce a result with n times m rows.

Example 2: Remove the WHERE clause of the SQL in the example 1 and see the query result.

```
SQL:  SELECT student_id, lname, fname, dept_name,
            CONCAT(building_no , ' ', building) AS Office
      FROM students, department
      ORDER BY lname
```

Result:

student_id	lname	fname	dept_name	Office
871992345	Aniston	Cynthia	Statistics	210 Thomas
871992345	Aniston	Cynthia	Marketing	411 BAB
871992345	Aniston	Cynthia	Computer Science	289 Hammond
871992345	Aniston	Cynthia	Industrial Engineering	207 Hammond
871992345	Aniston	Cynthia	Psychology	322 Thomas
871992345	Aniston	Cynthia	Physics	108 Davey
200776969	Barry	Keith	Statistics	210 Thomas
			.	
			.	
			.	
561498810	Vicario	Hector	Computer Science	289 Hammond
561498810	Vicario	Hector	Industrial Engineering	207 Hammond
561498810	Vicario	Hector	Psychology	322 Thomas
561498810	Vicario	Hector	Physics	108 Davey

Note:
As can be observed from the result above, each row of students table is combined with each row of the department table. In the above example, students table has **16** rows and department table has **6** rows, the result is **16*6 = 96** rows. By adding the WHERE clause as in example 1, only rows that have the same value of dept_id are retained in the solution set (the rows that are grayed in the result above).

Recall that in order to avoid ambiguity due to columns with the same name from different tables, we need to put a qualifier in front of the column name. The qualifier is generally the table name itself. However, sometime using table names as the qualifiers makes the SQL unnecessary long and messy. To overcome this problem we can use an alias as shown in the following example.

Example 3: Get the names of those students who are working on projects which are supervised by (PI =) either Dr. Enrique Salvatore or Dr. Steven Yu. Sort the result by last name in ascending order.

```
SQL:    SELECT CONCAT(lname, ', ', fname)  AS Student, PI as Professor
        FROM students s, res_member rm, research r
        WHERE (s.student_id = rm.student_id AND r.res_id = rm.res_id)
                AND (PI = 'Dr. Enrique Salvatore' OR PI = 'Dr. Steven Yu')
        ORDER BY lname
```

Result:

Student	Professor
Aniston, Cynthia	Dr. Enrique Salvatore
McNabb, Jason	Dr. Steven Yu
Miller, Johannes	Dr. Steven Yu
Montgomery, Donald	Dr. Enrique Salvatore
Santoro, Richard	Dr. Enrique Salvatore

Note:

1. Table students is aliased with ‘s’, res_member with ‘rm’ and research with ‘r’. We then can use the aliases as the qualifiers.
2. From the above example, you may notice that there are two kinds of WHERE clauses, one used for JOIN operation (s.student_id = rm.student_id AND rm.res_id = r.res_id) and another for criteria specification (PI = ‘Dr. Enrique Salvatore’ OR PI = ‘Dr. Steven Yu’). In order to increase readability of queries, SQL-92 standard introduced explicit **JOIN** keyword to replace the former WHERE clause as the following example.

Inner Join Using JOIN keyword

Example 4: Rewrite the SQL in example 3 using JOIN keyword

```
SQL:    SELECT CONCAT(lname, ', ',   fname)  AS Student, PI as Professor
        FROM (students s JOIN res_member rm ON s.student_id = rm.student_id)
        Join research r ON r.res_id = rm.res_id
        WHERE PI = 'Dr. Enrique Salvatore' OR PI = 'Dr. Steven Yu'
        ORDER BY lname
```

Result: same as example 3

Note:
In some database systems, the keyword INNER should be explicitly specified. But generally, the keyword INNER can be omitted and the system understands that the keyword JOIN means inner join. In the subsequent sections, outer join will be discussed. The keywords LEFT, RIGHT or FULL need to be specified to denote that the queries are outer join.

Outer Join

Inner Join operation returns only rows that have matched values. Sometime it is necessary to retrieve, in addition to the matching rows, the unmatched rows from one or both of the tables. Such an operation is called an *outer* join. There are three kinds of outer join: *Left Outer Join*, *Right Outer Join* and *Full Outer Join*. Left outer join, or simply left join, returns all rows from the table specified on the left of JOIN keyword regardless whether or not the rows are matched. Thus, left join operation returns the matched rows plus the unmatched rows from left table. On the other hand, right join operation returns all rows from the table specified on the right of JOIN keyword regardless whether or not the rows are matched. Full join operation is the union of left and right join operation. Examples below will clarify the idea.

Example 5: A charity organization has a table that lists all of its members (last name, first name and SSN). The table name is 'charity_org'. Some of the students belong to this organization, others don't. Identify students who are not members of this organization.

```
SQL:    SELECT students.lname, students.fname
        FROM students LEFT JOIN charity_org
        ON students.student_id = charity_org.ssn
        WHERE charity_org.ssn IS NULL
```

Result:

lname	fname
Brown	James
Kumara	Tirupati
McNabb	Jason
Vicario	Hector
Hosch	Jason
Go	William
Aniston	Cynthia
Cavalier	Jenifer

Note: After left joining table students and charity_org, for the students who are members of this organization, their charity_org.ssn are not null. Students who aren't members of this organization have NULL charity_org.SSN value. Therefore, in order to get students who are not members of this organization, the WHERE clause above is added. Without the clause, the result of the SQL is all students from students table.

Subqueries

Example 7: Get the names of students whose office building is BAB.

```
SQL:    SELECT lname, fname
        FROM students
        WHERE dept_id =
                (SELECT dept_id
                 FROM department
                 Where building = 'BAB')
```

Result:

lname	fname
Montgomery	Donald
Rider	Susan
Cavalier	Jenifer

Note: The inner query returns one value, the dept_id of Marketing department that is located in BAB building. Then the dept_id is used as the criterion by the outer query. If the criterion of the inner query is replaced by Hammond building, which is used by two departments, the SQL above should be slightly changed as shown in example 8 below.

Example 8: Get the names of students whose office building is Hammond.

```
SQL:    SELECT lname, fname
        FROM students
        WHERE dept_id  IN
```

```
                (SELECT dept_id
                 FROM department
                 Where building = 'Hammond')
```

Result:

lname	fname
McNabb	Jason
Vicario	Hector
Hosch	Jason
Smith	Kelly
McDougal	John
Nicholas	Theodore

Note: IN operator is used in WHERE clause of the outer query because there are 2 dept_id returned by the inner query (refer to tutorial 2 for explanation of IN operator).

Example 9: Get the names of the students whose department has a dept_id less than the one of computer science department.

```
SQL:    SELECT lname, fname
        FROM students
        WHERE dept_id <
                (SELECT dept_id
                 FROM department
                 WHERE dept_name = 'Computer Science' )
```

Result:

lname	fname
Brown	James
Montgomery	Donald
Kumara	Tirupati
Rider	Susan
Aniston	Cynthia
Cavalier	Jenifer

RESERCH STATEMENTS

Ya Zhang

My research interests spans the fields of bioinformatics and computational biology, system biology, data mining and machine learning. I have been exploring statistical and computational methods to analyze microarray data and protein interaction data, and to model and discover biological pathways and networks. In the following two sections, I will briefly description of my current research and outline future directions and goals of my research.

Research Experiences

Development in biotechnology has revolutionized the way that biological research is performed and pressed for the use of powerful computational methods for systems modeling and data analysis. Bioinformatics and computational biology has emerged as a new discipline to address the ever-increasing computational needs. Research efforts in this field include to develop and apply computational tools for acquiring, storing, analyzing, and visualizing biological data, many of which involves the use of machine learning and data mining techniques. With the recent success of whole genome sequence techniques, the focus of research in bioinformatics and computational biology has been shifted from DNA sequencing to genomics and proteomics. My research has been focused on developing innovative quantitative methods for solving puzzles in functional genomics and proteomics, such as protein-interaction prediction and microarray data analysis.

(1) Inferring Protein-Protein Interaction Networks

It has been proposed that all proteins in a given cell are connected through an extensive network where non-covalent interactions are continuously forming and dissociating. Finding interactions between proteins provides a broader view of how they work cooperatively in a cell and is the key to solve the functional genomics puzzle. High-throughput experimental approaches such as Yeast two-hybrid system have brought us an unprecedented opportunity to decipher the protein interaction networks. However, the high throughput data are inherently noisy and a large portion of the interactions are absent in the data. Computational methods have been employed for predicting protein-protein interactions from various sources. As proteins are assumed to interact through their interaction domains, domain-domain interactions provide a more general representation of protein-protein interactions. Several studies adopted a domain-based framework to infer protein-protein interactions: protein interaction data is used to infer domain interactions, which are then used to predict unobserved interactions. However, existing studies tend to oversimplify the problem by introducing two biologically unfounded assumptions about interacting domains: domain interactions are between two individual domains, and domain interactions are independent of each other.

I propose a new framework of learning to overcome the above limitation. Based on the underlying assumption that two proteins interact if and only if at least one pair of domains from the two proteins interact, the relationship between domain-domain interactions and protein-protein interactions can be expressed in conjunctive normal forms. The problem of interaction inference is then naturally modeled as a constraint satisfiability problem. This problem, known to be NP hard, essentially is to find a set of domain interactions that fit into the observed protein interactions. Because the data may conflict to each other, I try to maximize the number of protein interactions that are satisfied according to the assignment of domain interactions. Linear programming is used to solve the problem. Experimental results, based on a combined yeast data set and a multi-organism data set, have demonstrated the robustness of the algorithm.

(2) Comparative Mapping of Sequence-based and Structure-based Protein Domains

The notion of protein *domains* has gained increasing interest from the biology research community because of its importance in protein classification, protein function assignment, and protein engineering. Protein domains are generally considered as the building blocks of proteins. Classifying proteins based on their constituent domains is one of the most effective and efficient approaches to organize protein data both by structure and by evolutionary relationships. However, such a classification requires the identification of domain composition for proteins, which is by no means an easy task. The challenges lie in the ambiguity of domain definition as well as the lack of useful structural information about most proteins.

Both structure-based and sequence-based domain definitions have been widely used. But whether these types of models alone can capture all essential features of domains is still an open question. I attempt to provide insight on domain definition through comparative mapping of two domain classification databases: sequence-based (Pfam) and structure-based (SCOP). Two approaches are employed for the domain mapping: an indirect mapping approach based on bipartite graphs and a direct mapping approach using the location information of each domain instances. While the latter approach more accurately captures the mapping, the former method requires no location information and is computationally less expensive.

Mapping results from both approaches reveal a general agreement between the two types of domain definition. To further analyze the problem, I introduce several subcategories (one/many SCOP domain to one/many Pfam domain, and vice versa), and provide detailed studies on the mapping using examples from each category. In the cases of disagreement, Information from past literature, such as known domain functions, is used for external validation. In addition, I have proposed to use the evolutionary correlation between domains to measure the fitness of the domain classification.

(3) Revealing Co-regulated Genes through Time-Course Biclustering

Microarray experiments usually measure the expression levels of thousands of genes along time or under different cellular conditions. An important objective in analyzing the gene expression data is to discover co-regulated genes which are deemed as members of the same regulatory network. Initial attempts to interpret gene expression data start with grouping genes according to their overall expression profiles, with the underlying assumption that co-regulated genes behave similarly. However, it is unlikely that related genes behave similarly across all conditions because of the complexity in gene expression regulation. Discovering genes that co-regulated under part of the given experimental conditions (biclustering) is a biologically more meaningful. Existing biclustering algorithms target on condition-based Microarray data. For time-series gene expression data, the internal sequential relationship between time points is crucial. Several complex relationships such as time shifting and inverting are observed between co-regulated gene pairs based on their time-course expression profiles. Among them, time shifting is a unique property of time-series. It may reflect the regulator/regulatee relationship between a pair of genes. Thus, when applying biclustering algorithms to time series Microarray data, additional consideration needs to be given to the inherent sequential relationship between time points.

I attempt to discover locally co-expressed protein clusters by biclustering the time-series gene expression data. To reveal the importance of the internal sequential relationship among time points in biclustering time-series gene expression data, I extend the biclustering algorithm by Cheng & Church to time-series Microarray data. The discovered biclusters cover a continuous time interval in the whole time course. The algorithm is applied to a set of yeast cell cycle data, and the Gene Ontology is then used to annotate the gene biclusters. Based on the result of cluster annotation, our algorithm was able to discover several significant clusters which existing algorithm failed to find.

(4) Other Projects

Mining motifs from yeast promoters based on instance density (Bioinformatics/Data mining): Motifs are short consensus patterns embedded in biological sequences. Motifs play important roles in biological systems. Protein motifs are generally closely related to protein functions and structures. For example, enzyme catalytic sites of a protein usually have a certain conserved pattern. One important type of DNA motifs serve as binding sites for regulators of gene expression such as the transcriptional factors. With the increasing accessibility of sequential information such as DNA and protein sequences, in-silicon motif discovery has been an important task of bioinformatics. I developed a motif discovery algorithm based on instance density, where a motif is seen as a unique feature that distinguishes the set of sequences with the motif from other sequences. The algorithm implicitly takes into account information about both the background sequences and the motif instances. The algorithm is applied to discovering the transcriptional factor binding sites in yeast promoter regions. The results show that the proposed algorithm outperforms a benchmarked method.

Progressively display of very high resolution images (Medical Informatics): Images are produced and used widely in modern medicine. Medical images such as pathology images usually have very high resolutions, which make them difficult or impossible to display. It is desirable to develop a multi-resolution display method with which users can freely browse the contents of very high resolution images. I developed a progressive image display system based on wavelet transform. The system was designed to transmit and display high-resolution images (i.e. several gigabytes per image) with great fidelity. Particular regions of the image could be retrieved from the database and displayed in various resolutions.

Correlating summarization of multi-source news articles (Text Mining): The fast growth of the World Wide Web has been accompanied by the explosion of online information. How to make the overwhelming amount of information accessible to users is an important task. Online news represents an important part of online information. One striking feature of online news is that they are highly redundant -- hundreds or even thousands of news articles may be found on almost any topic/event. Discovering and utilizing the correlation among the news represents an important research issue in text summarization. I explored the research issue and methodology of correlated summarization for a pair of news articles. A machine learning method is constructed to highlight similarities of multi-source news with shared (sub)topics by modeling pairs of news articles as weighted bipartite graphs. Considering two news articles may share several subtopics, spectral K-way clustering is applied to aligns the (sub)topics. A mutual reinforcement principle is then applied to extract topic sentences within each subtopic group.

Research Plans

My thesis work focuses on predicting protein interaction networks. I will continue to expand this field of research. However, the linkage between bio-molecules is beyond proteins. For example, cellular signaling pathways are highly interconnected networks of biochemical reactions acting as molecular circuits that relay signals to genetic machinery. The approach used by biochemists in studying these pathways has been traditionally dependent solely upon experiments. The unforeseen complexity of these networks calls for computational assistance to determine the myriad of interconnections. Particularly desired are quantitative analytical tools that complement the suite of current bioinformatics capabilities. I will specifically target on the following two aspects:

(1) **Pathway discovery:** generating a set of potential connection maps and mathematical models from known and predicted interactions and function. This thrust consists of predicting pathways and networks with unsupervised learning methods (e.g. clustering) and modeling the systems (e.g. with Boolean or Bayesian network models). It also involves reverse engineering of networks using methodologies from system identification.

(2) **Pathway analysis:** quantifying models for fitness to experimental data. This thrust will focus on parameter estimation and dynamic model fitting for scoring model fitness. It may also involve characterizing sub-network dynamic responses through pattern discovery from existing known pathways.

The research closely matches the themes of NSF funding programs *Quantitative Systems Biotechnology, Genes and Genome Systems,* and *Biomolecular Systems.* I would like to actively seek research funds to support research.