論文
2-2-1

관계형 데이터베이스에서 효과적 데이터 마이닝 정보 추출을 위한 관계 연산자의 정의
(Definition of Relational Operators for Effective Extracting Data Mining Information from Relational Relational Database)

송 지 영*
(Song-Gy Young)

요 약

데이터베이스의 크기가 증대함에 따라, 데이터의 분석 및 데이터베이스로부터의 지식 습득 필요성이 대두되고 있다. 데이터 마이닝 기법은 그 대표적인 예이다. 대부분의 마이닝 대상 데이터 집합은 규모가 매우 크고, 데이터베이스 내에 저장되어 있다. 효과적인 마이닝 기능을 구현하기 위해서는 기존의 데이터베이스로부터 분석 대상 데이터 집합을 추출하고, 일반화시켜 함께 유지 관리함이 요구된다.

본 논문에서는 새로운 마이닝 연산자를 정의함으로써 기존 SQL 언어를 확장하여 릴레이션으로부터 분석 대상 데이터를 도메인 중심 방법으로 추출 후 일반화시킨다. 분석 대상 애트리뷰트 값과 일반화된 정보를 포함하는 배경지식은 관계형 데이터베이스의 릴레이션과 동일한 구조로 저장 및 관리된다. 또한 본 논문에서 제안된 배경지식 추출을 수행하는 SQL 유사 연산자와 집단 함수를 예제를 통하여 그 사용 예를 보임으로써, 마이닝 표현력을 나타낸다.

ABSTRACT

As the growth of database volume, it has required a need and an opportunity of data analysis and extracting knowledge from database. Data mining method is the representative example. The size of most minable data set is huge, and stored in a database. To implement effective mining function, we must extract minable data set to be analyzed from existing relational database, and it must be managed with its generalized information.

In this paper, the new mining operator is defined in a similar manner to the existing SQL operators and SQL is extended to extract data subset from relations and to generalize it using domain-oriented method. The background knowledge includes attribute values, which will be mind and generalized information, and it is managed as the same structure with a relation in relational database. These functions are implemented by defining some SQL - like operators and aggregated functions, and we describe the expressive powers of these new operators and functions through examples.

* 정회원 : 안산 1대학 웹프로그래밍과 교수

논문접수 : 2000. 1. 5.
심사완료 : 2000. 1. 26.

1. 서론

데이터 마이닝(Data Mining)은 규모가 큰 데이터 집단 내의 일부 데이터를 채취, 분석하여 지식을 얻는 과정이다. 이와 같은 데이터 마이닝 기능은 주로 통계와 인공 지능 분야에서 많이 사용되었고, 데이터베이스와는 독립적으로 개발되어 왔다[1]. 그러나 요사이는 데이터 마이닝 기능과 RDBMS는 서로 통합되어가는 추세이다. 데이터베이스를 기반으로 하는 데이터 마이닝 알고리즘을 개발하기 위해 보조 기억장치에 저장된 관계형 데이터베이스의 특성에 적합한 방식으로 데이터 마이닝 알고리즘을 개발하거나, 검색 언어를 확장하는 방법으로써 SQL 검색과 유사한 방식으로 마이닝 데이터를 검색 하는 연구가 행해지고 있다[2,3]. 후자의 검색 언어 확장 방법을 사용하면 마이닝 기능 구현 시 이미 개발된 기존 RDBMS의 SQL 엔진과 사용자 인터페이스의 기능상 장점을 이용할 수 있다.

본 논문에서는 기존의 RDBMS가 지원하는 SQL에서와 유사한 형식으로 마이닝 기능을 수행하는 단위 연산자(operational primitives)와 집단 함수(aggregate functions)를 정의한다. 이들 연산자나 집단 함수는 데이터베이스로부터 분석 대상 데이터를 추출하고 일반화하는 데 사용된다. 일반화된 데이터 집합은 릴레이션 형태로 배경지식 데이터베이스에서 유지 관리된다. 이 방법을 사용하면 SQL 검색과 유사한 방식으로 분석 대상 데이터를 추출 후, 데이터 값의 특성에 따라 그룹핑 후, 배경지식으로 관리함으로써 적은 비용으로 데이터 마이닝을 수행할 수 있다. 또한 RDBMS의 상위 부분에 구현된 마이닝 알고리즘들은 RDBMS의 기본 연산자처럼 사용되므로, 마이닝 기능을 기존 RDBMS의 SQL 질의 처리 기능을 확장함으로써 적은 비용으로 구현할 수 있는 장점이 있다[4].

본 논문의 구성은 다음과 같다. 2장은 관련 연구로서 RDBMS와 밀결합된 데이터 마이닝 시스템의 특징에 대해 알아본다. 또한 SQL을 확장하거나, SQL 구문과 유사한 구조로 마이닝 기능을 구현한 예를 살펴본다. 3장은 본론으로써 SQL을 확장하여 필요한 정보를 기존 릴레이션으로 부터 검색 후, 일반화 시켜 릴레이션으로 정의하는 방법을 구체적으로 기술한다. 이를 위해 3.1절에서는 기존 릴레이션으로 부터 분석 대상 애트리뷰트 값을 추출하고, 일반화시키는 단계를 기술한다. 3.2절은 마이닝시 필요한 배경지식 정보를 얻기 위해 단위 연산자와 집단 함수를 새롭게 정의한 후, 확장된 관계 대수를 사용하여 연산자의 절차적 의미를 나타낸다. 4장에서 결론을 맺는다.

2. 관련 연구

기존의 데이터 마이닝 과정은 맨 먼저, 데이터베이스로부터 마이닝 대상의 데이터 집합을 검색하고, 둘째 단계에서 패턴 추출 및 분석 등의 마이닝 기능을 수행하는 두 단계로 구성된다. 일반적으로 첫째 단계의 데이터 집합 검색 기능은 RDBMS에 의해 수행되고, 둘째 단계의 데이터 분석 기능은 고유의 마이닝 시스템에 의해 수행되는 데, 이러한 두 단계의 마이닝 수행 기능은 디스크에 저장된 대용량의 데이터 집합에 대한 특성을 충분히 고려하지 못했기 때문에 성능 상의 문제점을 야기 시켜 왔다 [5]. 이 문제점을 보완하기 위해, SQL과 같은 강력한 질의 도구를 사용하여 대용량 데이터 집합에 대해 동적으로 마이닝 연산이 수행되는 대화식 마이닝 기능을 지원하기 위한 RDBMS와 밀결합된 마이닝 시스템의 구현이 시도되었다. 이제 까지 구현된 시스템들은 크게 두 부류로 구분되는 데, 첫째는 마이닝 연산을 구현하기 위한 고유의 대화식 언어를 정의하고, 사용하는 것이고, 다른 하나는 기존의 SQL을 확장하여 마이닝 연산을 구현한 것이다.

[6]에서 제안된 DBMiner는 독자적인 마이닝 언어를 개발한 대표적인 예이다. DBMiner는 관계형 데이터베이스를 대상으로 다단계 지식을 대화식으로 마이닝하기 위해 개발된 시스템이다. 이 시스템은 SQL과 유사한 구조의 데이터 마이닝 언어인 DMQL[7]을 사용하여 데이터 마이닝 기능을 수행한다. DMQL은 구조면에서 SQL과 유사하게 정의되었지만 데이터베이스의 검색은 SQL 검색문을 사용하고, 마이닝 기능은 DMQL의 독립적인 언어를 사용하므로 일관된 데이터 검색 및 분석 기능을 지원할 수 없다. [8]에서 제안된 시스템도 SQL에서와 유사한 연산자인 MINE RULE을 정의하고 이를 이용하여 데이터 집합 간의 연관성 규칙을 규정한다. 그러나 Mine Rule에서 정의된 연산자들은 모두

연관성의 특성만을 구현함으로써 그 외의 마이닝 기능인 분류(classification), 데이터의 특징(characterization), 순차적 패턴 (sequential pattern)을 구현할 방법을 지원하지 않는다.

[4]에서 제안된 시스템은 위의 DBMine의 단점을 보완하기 위하여 SQL 언어를 확장하여 마이닝 기능을 지원한다. 전처리기가 마이닝 연산-특히 상호 연관성(associative rule) 연산을 포함한-확장된 SQL문을 입력으로 받아 마이닝 연산을 기존의 데이터베이스 검색 연산과 분리한다. 분리된 마이닝 연산은 다단계의 조인 연산을 수행하여 분석 대상 데이터 집합을 추출한다. 이로 인해 RDBMS는 마이닝 수행 시 빈번하게 발생하는 조인 수행 부담을 갖는다. 이러한 조인 연산은 질의 수행시 대부분의 비용을 차지하므로 성능 저하의 요인이 된다. [9]. MS-SQL Server 97도 SQL 질의 구문을 확장하여 마이닝 연산을 구현한다. SQL의 GROUP BY 절을 사용하여 분석 대상 데이터 집합을 추출하고, CUBE와 ROLLUP 연산자[10]를 사용하여 추출된 데이터 집합의 값을 일반화한다. 이들 SQL을 확장하여 데이터 마이닝 연산을 지원하는 RDBMS는 기존의 릴레이션 구조를 변경하지 않음으로써 기존 관계형 데이터베이스로부터 마이닝 대상 데이터를 동적으로 추출하고 분석하여 지식을 얻을 수 있다.

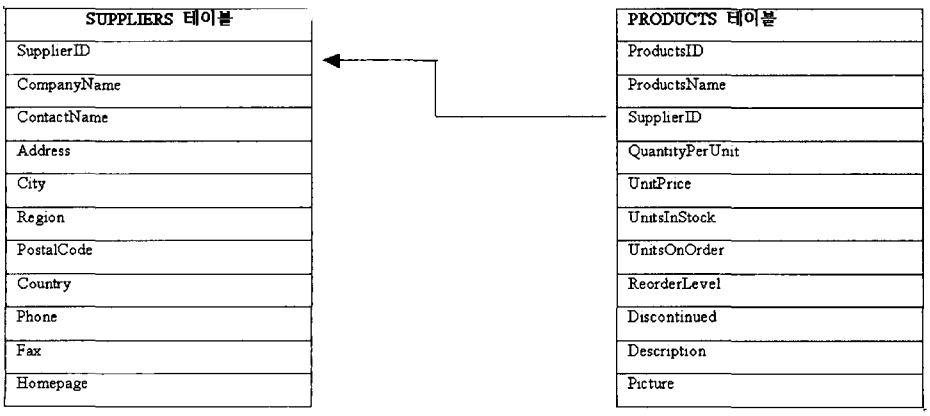
본 논문은 배경지식을 생성하고 저장 및 관리하기 위하여, 마이닝 대상 데이터를 추출하고 새롭게 정의한 SQL 연산자와 집단 함수를 사용하여 일반화시킨 후 그 결과를 기존의 릴레이션과 동일한 형태로 배경지식 데이터베이스에 저장한다.

3. 배경지식의 구축

데이터 분석 연산은 데이터베이스 내의 데이터들의 연관성(association), 분류(classification), 특성화(characterization) 등의 패턴 탐색 기능을 포함한다. 이 기능을 수행하기 위해 대상 데이터를 추출한 후, 데이터 합계를 구하거나, 통계 정보를 추출하거나, 데이터를 그룹화 시켜 그룹 간 비교 연산을 수행 후, 분석 결과를 출력한다. 본 논문에서는 분석 대상 데이터를 추출하고 일반화시키는 집단 함수를 정의하여 SQL문과 함께 사용함으로써 데이터 마이닝 기능을 구현한다. 3.1절은 분석 대상 데이터를 추출하여 배경지식을 저장하는 기능을 단계별로 예제를 통하여 기술한다.

3.1 배경지식의 생성

다음의 [그림 1]과 같은 물품 관리 데이터베이스를 고려하자. 재고 창고에서 관리되는 물품의 종류와 주문자 리스트에 관한 정보를 포함한다. PRODUCTS 테이블은 물품 리스트를 포함하고SUPPLIERS 테이블은 물품 공급자에 관한 정보를 포함한다. ProductID와



[그림 1] 물품 관리 데이터베이스

[Fig 1] Products Management Database

SupplierID 애트리뷰트는 각각의 테이블의 기본 키이다. PRODUCTS 테이블의 SupplierID는 SUPPLIERS테이블과의 관련성을 나타내는 외래 키이다. 재고 창고에서 관리되는 세분화된 개별 물품에 대해 품목별 물품 정보를 일반화시켜 배경지식으로 정의해 보자.

구체적으로 분석 데이터를 추출하여 배경지식을 저장하는 단계는 다음과 같다.

(단계 1) 첫번째 데이터베이스 스캔을 통해 분석 대상 튜플을 추출한다. 예로서, 품목별 물품 정보를 관리하기 위해 PRODUCTS 테이블을 분석 대상으로 선택한다. 초기 PRODUCTS 테이블로부터 첫번째 데이터베이스 스캔을 통해 아래의 셀렉션(selection) 조건을 만족하는 튜플의 수는 66개이다.

$\sigma_{(\text{UnitPrice} >= 10)}(\text{PRODUCTS})$

(단계 2) 새로운 테이블을 생성 시 분석 대상에서 제외시킬 애트리뷰트들과 외래 키를 결정한다. [그림 1]의 PRODUCTS 테이블의 외래키는 SupplierID (SUPPLIERES 테이블의 기본키)이다. 물품 재고에 관한 정보를 배경지식으로 구성하기 위한 분석 대상 애트리뷰트는 ProductName 이다.

$\pi_{\text{ProductName}}(\text{PRODUCTS})$

ProductsName
Chai
Chang
Aniseed Syrup
Chef Anton's Cajun Seasoning
Chef Anton's Gumbo Mix
Grandma's Boysenberry Spread
Uncle Bob's Organic Dried Pears
Northwoods Cranberry Sauce
Mishi Kobe Niku
Ikura

[그림 2] 마이닝 대상 테이블 추출

[Fig 2] Abstracted table for data mining

(단계 3) 분석 대상 애트리뷰트를 포함하는 테이블의 값을 일반화시킨다.

[그림 3]은 분석 대상 테이블의 ProductName애트리뷰트를 값에 따라 일반화시킨 결과(CategoryName 애트리뷰트)를 나타낸다.

CategoryName	ProductName
Beverages	Chai
	Chang
	Laughing Lumberjack Lager
Condiments	Chef Anton's Gumbo Mix
	Genen Shouyu
	Louisiana Hot Spiced Okra
Dairy Products	Queso Manchego La Pastora
	Raclette Courdavault
Grains/Cereals	Wimmers gute Semmelknodel
Meat/Poultry	Alice Mutton
Produce	Longlife Tofu
Seafood	Nord-Ost Matjeshering
	Rod Kaviar
	Spegesild

[그림 3] 일반화된 테이블:view1

[Fig 3] Generalized table : view1

(단계 4) 배경지식을 얻기 위해 집단 함수를 실행한다. [그림4]에서 사용된 집단 함수의 정의와 그의 실행 알고리즘은 3.2절에서 기술한다.

3.2 배경지식 연산자와 집단 함수의 정의

본 절에서는 마이닝 대상의 일반화 테이블을 얻기 위해 SQL과 유사한 연산자를 정의한다.

- POWERSET($\wp$(relname)(T)) - 단일 애트리뷰트로 구성된 relname의 새로운 릴레이션을 생성한다. 새로 생성된 릴레이션인 relname은 기존 T와 동일한 구조를 갖으며 튜플들도 T로부터 유도된다.
- GROUPBY($\mathcal{G}$(attr1, attr2)(T)) - 릴레이션 T의 애트리뷰트 attr1의 값을 스캔하여 각기 다른 값들에 대해 그룹핑 후, 더 나아가 attr1의 값

에 따라 분할 된 부그룹들을 일반화시켜 새로운 애트리뷰트인 attr2를 생성한다.

- APPEND(À (attr, expr)(T)) - 릴레이션 T에 새로운 애트리뷰트인 attr을 첨가하여 스키마를 확장한다. attr의 튜플은 expr을 수행한 결과로서 얻어진다.

이들 연산자를 사용하여 [그림 3]에 나타난 일반화 테이블을 표현해보자.

$$\mathrm{n}_{\text{view1}}(\Pi_{\text{productname}}(\sigma_{\text{(unitprice>10)}}(\text{PRODUCTS})))$$

$$\mathrm{G}_{\text{(Categoryname, productname)}}\ \text{view1}$$

기존의 SQL 검색 문은 GROUP BY 절을 사용하는 애트리뷰트 값의 특성에 따라 애트리뷰트를 집단 화하여 부 그룹으로 분할한다. 각 부 그룹에 대해 집단 함수가 실행되는 데, 본 논문에서 제안된 함수는 모두 도메인 중심의 함수(domain specific function)이므로 각 부 그룹에 대해 단일 값을 결과로서 환원한다. 예를 들어, [그림 3]의 일반화된 테이블의 CategoryName 애트리뷰트 값에 따라 생성된 각각의 부 그룹에 대하여 집단 함수를 실행시키면 [그림 4]와 같이 제품의 개수(NumberOfProducts), 재고량이 가장 많은 제품 이름(MaxProductName)과 수량 애트리뷰트(MaxUnitsInStock)를 포함하는 배경지식(view1)을 구성할 수 있다.

[그림 4]의 배경지식을 얻기 위한 집단 함수를 새롭게 정의해 보자. 일반적으로 집단 함수는 문자 (를 사용하여 다음과 같이 표현될 수 있다.

<애트리뷰트 리스트> Ø <함수 이름, 애트리뷰트 리스트><릴레이션 이름>

첫번째 <애트리뷰트 리스트>는 일반화된 애트리뷰트의 결과 값을 포함하는 애트리뷰트 이름의 집합이다. <릴레이션 이름>은 집단 함수가 수행될 피연산자 릴레이션을 나타내고, <함수 이름, 애트리뷰트 리스트>의 함수 이름은 수행될 집단 함수 이름을 나타낸다. 애트리뷰트 리스트는 집단 함수가 적용될 애트리뷰트 이름인데, 집단 함수의 실행 후, <결과 애트리뷰트 리스트>와 일대일 대응된다.

다음은 본 논문에서 정의된 집단 함수 이다.

가. CountAllGroups (attr, Rel) : attr의 전체 부 그룹의 개수 (cardinality of attr)

나. CountAllValues (Attr, Rel) : attr의 값을 구성하는 부 그룹 내의 서로 다른 값의 개수를 환원한다.

다. Rank(attr, Rel) : attr의 서로 다른 값의 개수가 N일 때, attr내의 Rank 함수의 결과 N을 환원하고, 최저 값은 1을 환원한다. 이 때, N의 값은 CountAllGroups(attr, Rel)의 결과와 동일하다.

CategoryName	NumberOfProducts	MaxProductsName	MaxUnitsInStock
Beverages	10	Sasquatch Ale	111
Condiments	12	Grandma's Boysenberry Spread	120
Confections	11	NuNuCa Nu-Nougat-Creme	76
Dairy Products	9	Queso Manchego La Pastora	86
Grains/Cereals	5	Gustaf's Knackebrod	104
Meat/Poultry	5	Pate chinois	115
Produce	5	Tofu	35
Seafood	9	Boston Crab Meat	123

[그림 4] 배경지식 뷰:view1

[Fig 4] View with background Knowledge

라. MaxSub(attr, Rel) : 릴레이션 내의 attr의 부 그룹 내에서 최저 값을 환원한다.

마. Ratio(expr, Rel) : expr의 수행 결과 부 그룹을 구성하는 튜플이 전체 튜플(cardinality of relation)에 대해 차지하는 비율을 나타낸다.

[그림 4]의 테이블에서 우선 CategoryName과 NumberOfProducts 애트리뷰트를 얻기 위해 view1 테이블에 대해 다음 함수를 실행한다.

$$\grave{A}_{(numberOfProducts,\ NumberOfProducts} CountAllValues_{(Categoryname,\ View1))}\ view1)$$

$$\grave{A}_{(MaxUnitsInStock,\ MaxUnitsInStock} MaxSub_{(UnitsInStock,\ Products))} view1)$$

[그림 5]는 [그림 4]에서 얻어진 view1에 대해 Rank()함수를 실행시킨 결과이다.

$$\grave{A}_{(rank,\ rankRank(CategoryName,\ view1))} view1)$$

$$\grave{A}_{(ration,\ ratioRatio(CategoryName,\ view1))} view1)$$

4. 결론

RDBMS와 마이닝 기능을 효과적으로 통합하여 기존의 SQL 검색 기능과 마이닝 기능을 최적의 방법으로 실행하기 위해서는 마이닝 기능 중 RDBMS의 질의 검색기와 통합될 기능과 RDBMS의 상위 단계에서 실행될 기능을 분리하는 것이 중요하다.

관련 연구에서 RDBMS에서의 마이닝 데이터를 추출하기 위해 개발된 언어들에 대한 비교 분석을 함으로써 연구 범위를 한정 시켰다. 본 논문에서는 데이터 값을 사용한 유사 집단화 연산자와 구별 연산자를 정의하였다. 또한 관계 대수를 사용하여 그 구현 방법을 나타내었다. 본 논문에서 제안한 마이닝 연산자들과 집단 함수들은 RDBMS의 질의 검색기와 통합되어 배경지식 데이터베이스를 생성하는 데 유용하다. 배경지식 데이터베이스는 기존 관계형 데이터베이스로부터 유도되는 데, 원시 테이블로부터 분석 대상 데이터를 추출하고, 일반화시킨 후, 본 논문에서 정의된 집단 함수를 사용하여 배경지식을 얻는다. 이들 배경지식은 powerset연산을 사용하여 기존 테이블과 동일한 구조로 생성되고 유지, 관리된다.

CategoryName	NumberOfProducts	MaxProductsName	Rank	MaxUnitsInStock
Beverages	10	Sasquatch Ale	3	111
Condiments	12	Grandma's Boysenberry Spread	1	120
Confections	11	NuNuCa Nu-Nougat-Creme	2	76
Dairy Products	9	Queso Manchego La Pastora	4	86
Grains/Cereals	5	Gustaf's Knackebrod	6	104
Meat/Poultry	5	Pate chinois	7	115
Produce	5	Tofu	8	35
Seafood	9	Boston Crab Meat	4	123

[그림 5] 결과 테이블 : view1

[Fig 5] Result table : view1

본 논문에서 제안한 배경지식 데이터 집합의 추출 및 일반화 연산은 단일 테이블에만 수행되지만, 앞으로는 본 논문에서 제안된 연산자와 함수를 확장시켜 다중 테이블로부터 배경지식을 얻을 경우를 구현한다. 이때, 배경지식을 효과적으로 구현하기 위한 조인 연산 알고리즘의 개발이 요구된다. 뿐만 아니라 원시 데이터베이스가 변경된다면 그에 따라 배경지식도 변경되어야 한다. 이와 같이 데이터베이스의 변경 시는 배경지식의 관리가 실행 부담으로 남을 수도 있으므로, 최저의 비용으로-데이터베이스에 수행되는 연산 중 검색연산이 대부분임을 고려하여 마이닝 정보를 신속하게 얻는 효과적인 배경지식의 유지 관리 방법이 개발되어야 한다.

※ 참고문헌

[1] U. Fayyad, G. Piatetsky-Shapiro et. Al. (Eds.), Advances in Knowledge Discovery and Data Mining, AAA/MIT Press, 1996.

[2] R. Agrawal, "Data Mining : Crossing the Chasm", Proc. of the 5th ACM SIGKDD., 1999.

[3] S. Chaudhuri, "Data Mining and Database Systems: Where is the Intersection?", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 1998

[4] S. Sarawagi, S. Tomas, and R. Agrawal, "Intergrating Association Rule Mining with Relational Database Systems: Alternatives and Implications", Proc. ACM SIGMOD, 1998.

[5] H. Mannila, "Data mining: machine learning, statistics, and database", http://www.cs.helsinki.fi/~mannila/

[6] J. Han, J. Chiang , Sonny Chee, et. Al. "DBMiner: A System for Data Mining in Relational Databases and Data Warehouses", http://db.cs.sfu.ca/DBMiner.

[7] J. Han, Y. Fu, W. Wang et. Al. "DMQL:A Data Mining Query Language for Relational Databases", Proc. ACM SIGMOD, 1996.

[8] R. Meo, G. Psaila, and S. Ceri, "A New SQL-like Operator for Mining Association Rules", Proc. of the 22st VLDB Conf., Mumbai, India, pp.122-133, 1996.

[9] F. Manachuca et. al., "Enhancing the explanation of data mining in relational database systems via the rough sets Theory including Precision variables", Proc. of the 1998 ACM symposium on Applied Computing, 1998.

[10] S. Ramaswamy et. al., "Efficient Algorithms for Mining Outliers from Large Data Set", Proc. of the 2000 ACM SIGMOD., 2000.

송 지 영

1987년 인하대학교 전산학 학사
인하대학교 전산학 석사
인하대학교 전산학 박사
현재 안산 1대학 웹프로그래밍과
교수