공학석사학위논문

Support Vector Machines를 이용한 문서 정보 기반의 단백질 기능 분류

A Literature Based Method for Protein Function Classification via Support Vector Machines

2004년 2월

서울대학교 대학원
컴퓨터공학부

어 상 준

Support Vector Machines를 이용한 문서 정보 기반의 단백질 기능 분류

A Literature Based Method for Protein Function Classification via Support Vector Machines

지도교수 장 병 탁

이 논문을 공학석사 학위논문으로 제출함

2003 년 10 월

서울대학교 대학원
컴퓨터공학부

어 상 준

어상준의 공학석사 학위논문을 인준함

2003 년 12 월

위원장 유 석 인 印
부위원장 장 병 탁 印
위 원 문 병 로 印

초 록

생물학자들이 단백질의 기능을 연구함에 있어 처음부터 끝까지 실험적으로 유전자나 단백질을 일일이 다룬다는 것은 그 실험 범위가 너무 넓어서 현실적인 어려움을 지니고 있다. 이에 본 논문에서는 생물학자들이 단백질의 기능에 관련된 실험에 앞서서 그 기능을 미리 예측해 볼 수도 있고 실험 후에는 그 실험 결과도 검증해 볼 수도 있는 실험의 보조적인 정보를 제공하고자 신뢰할 만한 성능의 문서 정보 기반 단백질 기능 분류 방법론을 제시하였다.

문서 정보만을 이용하여 단백질의 기능을 예측해 볼 수 있는 이유는 기존에 이미 방대한 양의 단백질 기능 관련 연구들이 문서로 공개되어 있기 때문이다. 즉, 어떤 단백질에 대해 직접적으로 그 기능에 대하여 언급을 한 문서가 없다고 하여도 최소한 그 단백질의 여러 가지 실험 결과에 관하여 쓰여진 문서들이 존재하며 그러한 문서들의 정보를 이용하여 해당 단백질의 기능을 예측할 수 있다.

본 논문에서는 단백질의 기능 분류 문제와 문서의 분류 문제가 같다는 것을 보였고, 문서의 분류 처리를 위하여 support vector machines를 사용하였다. 문서처럼 많은 수의 자질을 갖는 데이터에서 특히 좋은 성능을 보여주고 있는 support vector machines 는 학습데이터를 최대 마진으로 분류하는 초평면(hyperplane)을 찾아 학습데이터를 두 부분으로 분류한다.

실험적으로는 실제 효모균과 대장균의 단백질 기능 정보를 가지고 본 논문에서 제시한 방법론의 성능을 확인하였다.

주요어 : 단백질 기능 분류, Support Vector Machines, 문서 분류, 효모균, 대장균
학 번 : 2002-21556

목 차

Ⅰ. 서론 ……1

1.1 연구 배경 ……1

1.2 논문의 구성 ……3

Ⅱ. 관련 연구 ……4

2.1 MEDLINE 데이터베이스 ……4

2.2 COGs 개요 ……6

2.3 기존의 단백질 기능별 분류 연구들 ……10

Ⅲ. 문서 정보 기반의 단백질 기능 분류 방법론 ……15

3.1 문서 분류와 단백질 기능 분류 ……15

3.2 Support Vector Machines (SVMs) ……20

3.2.1 비선형 가설 공간 ……25

3.2.2 최상의 파라미터값 ……26

3.2.3 분류할 수 없는 문제들 ……27

3.3 알고리즘 ……28

Ⅳ. 실험 및 분석 ……33

4.1 효모균 (Saccharomyces cerevisiae) ……33

4.2 대장균 (Escherichia coli) ……43

Ⅴ. 결론 및 향후 과제 ……49

참고 문헌 ……51
ABSTRACT ……54

Ⅰ. 서론

1.1 연구 배경

게놈(genome:유전체) 프로젝트의 결과물로 많은 수의 시퀀스 데이터가 밝혀졌고 그것으로부터 잠재적으로 이용 가능한 정보를 이해하고 활용하기 위하여 기능을 예측하고 시퀀스에 주석을 다는 일(sequence annotation)은 무척 중요해지고 있다. 새롭게 밝혀지는 시퀀스들을 미리 정의된 유사한 기능을 가지는 범주로 분류하는 일이 그러한 일일 것이다. 그러한 유전자 기능에 관한 정보를 가지고 있는 유용한 정보원 중 하나가 텍스트이다. 여러 개체의 유전자 산물들의 많은 기능적 특징들이 텍스트로 기록되어 있고 현재도 많은 수의 텍스트들이 연구자들에 의하여 쓰여지고 있다.

그러나 텍스트로부터 공통된 기능을 가지는 단백질들을 분류해내는 일은 쉬운 일이 아니다. 우선, 관련 텍스트의 방대한 양을 그 이유로 들 수 있다. 예를 들어, 효모균에 관련된 논문들을 찾아보면 대략 119,526 편의 논문을 찾을 수 있고 그 수도 매일 늘어난다. 이러한 방대한 양의 텍스트를 일일이 사람의 손으로 처리한다는 것은 거의 불가능하다. 그 다음 이유로는 유전자나 단백질의 동의어 문제를 들 수 있다. 유전자나 단백질은 종종 여러 개의 이름을 가질 수 있고 더 많은 동의어들이 새로운 기능적, 구조적 정보가 발견됨에 따라 생겨나고 있다. 관련 논문의 저자들은 그 중에서 임의의 동의어 하나를 선택할 수 있고 이 이름이 논문에서 사용된다. 따라서, 관련 유전자나 단백질 정보를 텍스트로부터 모두

이용하려면 이러한 동의어 문제를 고려해야 한다. 또 다른 이유로, 내용적인 측면에서 보면 어떤 유전자들은 아주 많이 연구되고 있는 반면에 몇몇 유전자들은 최근에 발견되고 있고 또한 대부분의 유전자들이 복수의 기능들을 가지고 있기 때문이다 [Raychaudhuri et al. 2002].

본 논문에서는 이러한 문서데이터 특징에 적합한 support vector machines를 이용한 문서 정보 기반의 단백질 기능 분류 방법론을 제시한다. 이를 통하여 실질적으로 생물학자들에게 도움을 줄 수 있는 생물학 실험에 보조적인 정보를 제공할 것이다.

1.2 논문의 구성

본 논문의 구성은 다음과 같다. 2 장에서는 본 논문에서 사용된 실험 데이터인 MEDLINE 문서 데이터베이스, 단백질의 기능적 분류정보를 가지고 있는 COGs 데이터와 기존에 연구된 단백질의 기능적 분류 방법들에 대해 설명한다. 3 장에서는 본 논문에서 제시한 방법론의 근거, 사용된 방법론과 전체 알고리즘을 설명한다. 4 장에서는 실제 효모균 단백질과 대장균 단백질을 가지고 실험한 방법을 기술하고 결과를 분석한다. 마지막으로 5 장에서는 연구 내용을 요약하고 앞으로의 연구 과제와 함께 결론을 맺는다.

Ⅱ. 관련 연구

2.1 MEDLINE 데이터베이스

미국 국립 의학 도서관(US Natinal Library of Medcine : NLM)에서 제작하여 공급하는 MEDLINE(MEDlARS ONLINE)은 생화학 및 의학분야의 자료들을 중심으로 전 세계의 주요 논문들을 검색할 수 있게 해 주는 데이터베이스로서 생화학과 의학 외에도 독성학, 영양학, 약물학, 수의학, 간호학, 치과학, 정신의학, 의료공학, 병리학, 스포츠의학 등에 대해서 다루고 있다.

미국과 전세계 70 개국에서 출판된 3800 종류가 넘는 최신의 의학 및 생물학 저널에서 인용한 내용과 초록이 실려 있고 1966 년 이후의 약 1200 만건의 데이타가 모두 들어 있으며 대부분 영어로 되어 있고 일부 영어로 번역된 요약도 있다.

이 데이타베이스는 Index Medicus, Index to Dental Literature, International Nursing Index 3 개의 주요 서적을 그 원전으로 하고 있는데, Index Medicus에서 발간되지 않은 저작물은 MEDLINE 내의 전염병과 개체번식생물학 분야에 포함되어 있다.

또한 이 데이터베이스는 MeSH(Medical Subject Headings)라는 고유의 검색용어로 인덱스되어 있고, 이것은 수록시기별로 2 개 DB로 나뉘어 있다. 1975 년 이전에 수록된 정보는 초록이 없으며, 그 이후의 것은 약 47%정도가 초록을 갖고 있다. 1984 년부터 현재까지의 레코드는 약 59%가 초록을 갖고 있다. 매년 거의 380,000 레코드가 추가되고 있으며, 그 중 75%정도가 영어로 씌여져

있다. 선별된 전공논문의 chapter 나 article 의 요약은 1976 년에서 1981년까지 수록되어 있다.

2.2 COGs 개요

단백질의 기능들을 분류하는 범주(class)는 한 개체에 대해서도 여러 가지가 있을 수 있고 기존의 단백질 기능 분류를 위한 범주 데이터로 유전자 온톨로지 콘소시엄(Gene Ontology Consortium)과 MIPS (Munich Information Center for Protein Sequences) 데이터를 이용한 연구가 있었다 [Ashburner et al. 2000; Mewes et al. 2000]. 그러나 본 연구에서는 COGs 데이터 [Tatusov et al. 1997; Tatusov et al. 2001]를 사용하여 단백질의 기능들을 새롭게 분류, 정리했고 실험, 검증 데이터로 사용하였다.

COGs는 Clusters of Orthologous Groups of Proteins의 약자로 미국 NCBI(National Center for Biotechnology Information)가 계통 발생학 분야에서 이미 알려진 일곱 개의 서로 다른 생명체의 유전체들을 가지고, 보존된 유전자들을 분류하고 그들 사이의 진화적 관계를 조사하기 위하여 고안한 새로운 체계이다. 여기서 orthologs라는 용어는 두 생물체에서 상동성을 나타내는 서열을 나타내는 말로서 공통 조상이 가지고 있던 한 서열의 직접적인 (즉, 유전자 중복이 수반되지 않은) 후손인 경우를 말한다.

진화 생물학자들은 모든 생물체의 유전적 구성이 일련의 공통된 선조 유전자들로 거슬러 올라갈 수 있다고 생각하고 이런 추측은 과학자들이 다른 종의 유전자들 사이에서 염기서열을 비교하여 이들 사이의 멀고도 미묘한 관계를 확인하도록 하였다. 완벽하게 염기서열이 결정된 유전체의 수가 점점 많아짐에 따라 주요 계통발생집단이나 특별한 생물체들 사이에 예견치 못했던 광범위한

비교가 가능해졌고 이들의 관계에 대한 개략적인 정보를 얻을 수 있었다. 이러한 광범위한 조망은 진화과정에 대한 지식을 증가시킬 것이고, 일부 생명체에 보존되어 있지만 다른 생명체에는 보존되어있지 않은 단백질의 기능을 확인시켜 줄 수 있을 것이다. 이러한 관점에서 NCBI 에서는 단일 유전자를 가지고 시작하여 모든 다른 유전체에서 가장 잘 일치하는 염기서열을 조사했다. 그들은 거의 18,000 개의 염기서열을 비교할 때 까지 각 단백질 서열을 모든 유전체에 있는 모든 다른 서열에 대하여 일대일 서열 비교를 수행하였다. 결과로 얻어진 단백질의 기능적 범주는 다음과 같다.

Code	COGs	Domains	Description
Information storage and processing			
J	217	6449	Translation, ribosomal structure and biogenesis
K	132	5438	Transcription
L	184	5337	DNA replication, recombination and repair
Cellular processes			
D	32	842	Cell division and chromosome partitioning
O	110	3165	Posttranslational modification, protein turnover, chaperones
M	155	4079	Cell envelope biogenesis, outer membrane
N	133	3110	Cell motility and secretion
P	160	5112	Inorganic ion transport and metabolism
T	97	3627	Signal transduction mechanisms
Metabolism			
C	224	5594	Energy production and conversion
G	171	5262	Carbohydrate transport and metabolism
E	233	8383	Amino acid transport and metabolism
F	85	2364	Nucleotide transport and metabolism
H	154	4057	Coenzyme metabolism
I	75	2609	Lipid metabolism
Q	62	2754	Secondary metabolites biosynthesis, transport and catabolism
Poorly characterized			
R	449	11948	General function prediction only
S	750	6416	Function unknown

표 1: COGs (Functional annotation)

표 1에서 보듯이 COGs 데이터는 정확히 그 기능이 명시된 16 개의 범주와 아직 그 기능이 확실치 않은 2 개의 범주로 분류될 수 있다. 위 표에서 COGs 항목은 각 기능적 분류에 속하는 COGs의 개수를 나타내고 domains 항목은 각 기능적 분류에 속하는 알려진 단백질의 개수를 나타낸다. 즉, 본 논문에서 제시한 방법론과는 틀리게 COGs 데이터의 특징은 단백질 기능 분류의 방법론적 관점에서 보면 단백질 서열 자체의(형태적) 유사성에 기반을 두고 기능을 분류한 것이다.

2.3 기존의 단백질 기능별 분류 연구

우선 가장 확실한 단백질의 기능을 밝히는 방법은 생물학자들이 실험실에서 실험적으로 단백질의 기능을 연구, 실험하는 것이다. 대표적인 실험방법으로는 yeast two hybrid 실험이 있다 [Criekinge, 1999]. 그러나 다양한 게놈과 cDNA 서열분석 사업들이 놀라운 속도로 새로운 서열자료들을 방출하고 있어 일일이 생물학적 실험으로 이를 처리하기에는 한계가 있다.

따라서 단백질의 대규모 기능적 분류에 사용되는 일반적인 방법은 현존하는 단백질 관련 데이터베이스를 토대로 단백질의 유사성을 비교, 분류하는 것이다 [Cai et al. 2003]. 이러한 데이터베이스들을 표 2 에 정리하였다 [Wu et al. 2003].

내용	데이터베이스명
Protein sequences	PIR-PSD, PIR-NREF, Swiss-Prot, TrEMBL, GenPept, RefSeq
Families	InterPro, Pfam, ProSite, Blocks, Prints, COG, MetaFam, PIR-ASDB, ProClass
Functions and pathways	EC-IUBMB, KEGG, BRENDA, WIT, MetaCyc, EcoCyc
Interactions	DIP, BIND
Post-translational modifications	RESID, PhosphoSite DB
Protein expression and proteomes	PMG
Structures and structural classifications	PDB, PDBSum, SCOP, CATH, FSSP, MMDB
Genes and genomes	GenBank, EMBL, DDBJ, LocusLink, TIGR, SGD, FlyBase, MGI, GDB, OMIM, MIPS, GenProtEC
Ontologies	GO
Taxonomy	NCBI Taxonomy

표 2: 단백질 관련 데이터베이스들

또 다른 단백질 기능 분류 방법으로는 DNA microarray 칩이나 단백질 칩등의 실험적 결과와 단백질 데이터베이스의 형태적 정보를 동시에 이용하는 방법이 연구되고 있다 [Pavlidis et al. 2001].

마지막으로 본 연구와 가장 비슷하다고 할 수 있는 문서들 사이의 유사성에 기초해서 비슷한 기능들을 가진 유전자들을 군집화한 연구가 있었다 [Raychaudhuri et al. 2002; Raychaudhuri and Altman, 2003; Raychaudhuri et al. 2003]. 이 연구에서는 다음과 같은 단백질 기능 분류표를 만들고 이를 기준 데이터로 보고 실험을 하였다.

Functional classification	Gene ontology code	Genes	Total article references
Signal_transduction	GO:0007165	94	3484
Cell_adhesion	GO:0007155	6	82
Autophagy	GO:0006914	16	110
Budding	GO:0007114	74	1692
Cell_cycle	GO:0007049	341	8399
Biogenesis	GO:0016043	459	6439
Shape_size_control	GO:0007148	54	1629
Cell_fusion	GO:0006947	89	2495
Ion_homeostasis	GO:0006873	43	667
Membrane_fusion	GO:0006944	6	212
Sporulation	GO:0007151	27	646
Stress_response	GO:0006950	94	2603
Transport	GO:0006810	313	4559
Amino_acid_metabolism	GO:0006519	78	1594
Carbohydrate_metabolism	GO:0005975	90	2719
Electron_transport	GO:0006118	8	205
Lipid_metabolism	GO:0006629	90	1035
Nitrogen_metabolism	GO:0006807	15	264
Nucleic_acid_metabolism	GO:0006139	676	12345

표 3: GO codes 로부터 만들어진 효모균의 gold standard functional gene groups.

표 3 은 GO 의 데이터를 이용해서 만들어진 효모균 관련 19 가지 기능들이다. 각 기능적 분류에 속한 유전자들에 관련된

MEDLINE 문서는 total article references에 그 개수를 표시하였고 본 논문과는 틀리게 Saccharomyces Genome Database[Cherry et al. 1998]를 참조하여 작성되었다. 여기서 GO(gene ontology)는 생물학적 기능을 묘사하는데 사용될 수 있는 ontology를 만들기 위해 대다수의 organism specific database들이 참여하여 만들어진 consortium으로, 이들의 목적은 functional computation에 사용되기 위한 controlled vocabulary를 확립하는 것이다.

이 연구에서 행해진 실험방법은 우선 표 3의 각 기능에 속한 유전자 그룹에 대하여 같은 크기의 유전자 그룹을 임의로 100 개씩 총 1900 개의 그룹을 생성하였다. 만약, 어떤 유전자 그룹이 서로 비슷한 기능을 가진 유전자들로 우연히 서로 모아졌다면 그 그룹에 속하는 문서들도 서로 연관성이 깊을 거라는 게 이 연구의 기본 아이디어다. 이렇게 해서 1900 개의 각 그룹에 대하여 그 그룹에 속한 문서들의 유사성을 조사해서 그 결과값이 높은 그룹을 같은 기능들을 가지는 유전자들이 모인 그룹으로 본 것이다. 여기서, 문서들간의 유사성은 inverse document frequency weighted word vectors[Manning and Schutze, 1999]를 각 문서에 대해 계산하여 두 문서간의 cosine 각을 계산하는 방법으로 구하였다. 그러나, 이 연구는 단순히 기존 단백질 분류 정보를 문서 정보를 가지고 확인하는 수준에 머물렀다는 한계가 있다.

Ⅲ. 문서 정보 기반의 단백질 기능 분류 방법론

3.1 문서 분류와 단백질 기능 분류

본 연구에서 새롭게 발견하고 실험에 이용한 문서 정보와 단백질 기능 정보와의 상관 관계는 COGs 데이터에서 단백질의 기능적 분류 문제는 단순히 그 단백질 명만을 포함하는 문서의 분류 문제와 거의 일치한다는 것이다. 즉, 같은 기능들을 가지는 단백질들을 분류한 COGs 범주 결과를 가지고 그 단백질 명들을 포함한 문서들을 분류해보면 전체 관련 문서들이 COGs 의 기능적 범주 결과와 거의 일치 되게 분류되는 것을 알 수 있었다.

Code	Total Abstracts Number	Duplicated Abstracts Number	Uniqueness of Abtracts (%)
J	1117	205	81.7
K	1064	196	81.6
L	4821	232	95.2
D	64	0	100
O	1627	41	97.5
M	264	32	87.9
N	151	0	100
P	1271	24	98.1
T	3410	49	98.6
C	1305	48	96.3
G	1843	42	97.7
E	2965	117	96.1
F	1298	32	97.5
H	726	27	96.3
I	518	5	99
Q	653	9	98.6

표 4: 효모균의 검색 결과 분석

표 4에서 J 기능에 속하는 효모균 단백질들의 명칭이 들어있는 문서들의 총 개수는 1117개이고 이 문서들 중 다른 기능에도 포함된 문서의 총 개수는 205개 였다. 따라서, J 기능에 속하는 모든 문서들이 순수하게 J 기능에만 속할 확률은 81.7% 이다.

그러나 COGs 데이터에서는 종종 여러 단백질들이 여러 개의 기능적 범주에 동시에 속한다 (예로 J 기능에 속하는 효모균의 단백질 총 324 개중에서 49 개가 다른 기능에 동시에 속함). 이를 감안하여 좀더 노이즈가 적은 SVMs 학습 데이터를 만들기 위하여 각 기능적 범주에 유일하게 속하는 단백질들 만을 다시 간추려서 그 단백질 명을 포함하는 문서들을 다음과 같이 분류하였다.

Code	Total Abstracts Number	Duplicated Abstracts Number	Uniqueness of Abtracts (%)
J	877	21	97.6
K	551	16	97.1
L	4097	57	98.6
D	64	0	100
O	1329	33	97.5
M	219	0	100
N	151	0	100
P	863	15	98.3
T	3291	32	99
C	1181	38	96.8
G	1433	27	98.1
E	2260	56	97.5
F	1159	12	99
H	563	14	97.5
I	498	2	99.6
Q	461	6	98.7
R	3161	59	98.1
S	3729	66	98.2
SUM	25887	454	98.3

표 5: 효모군의 각 기능적 범주에 고유하게 속하는 단백질 명들만을 이용한 검색 결과 분석.

표 5의 결과에서 보듯이 단순히 해당 단백질 명을 포함하는 문서 분류결과는 기능적 분류와 거의 일치되게 나왔고 (각 기능적 분류에 속하는 문서들의 그 한 기능에만 유일하게 속할 확률이 평균 98.3% 임) 심지어는 poorly characterized인 R과 S 클래스에서도 이와 같은 결과가 나왔다. 따라서, 위의 문서 분류결과를 기능적 분류결과와 거의 같다고 봐도 무방할 것이다. 본 결과로 유추할 수 있는 사실은 단백질에 관한 논문은 일반적으로 한가지의 기능에 대하여만 쓰여지는 경향이 있다는 것이다. 따라서, 단백질의 기능적 분류 문제를 해당 단백질 명을 포함하는 문서 분류 문제로 치환하여 간주할 수 있다.

3.2 Support Vector Machines (SVMs)

본 논문은 단백질 분류 문제(문서 분류 문제)를 풀기 위해 support vector machines를 사용하였다. Support vector machines는 기본적으로 두 범주를 갖는 객체들을 분류하는 방법이다. 이 방법은 1979년 Vapnik에 의하여 발표된 바 있으나 최근에 와서야 그 성능을 인정받아 각광을 받게 되었으며 문서 분류 문제에서 좋은 성능을 보이고 있다 [Vapnik, 1995; Cortes and Vapnik, 1995; Boser et al., 1992; Joachims, 1997].

Support vector machines는 다항식 분류기(polynomial classifier), 신경망 분류기 또는 Radial Basis Function 분류기를 학습하는 방법으로 볼 수 있다. 기존의 분류기들 대부분이 경험적 위험(empirical risk)을 최소화한다는 아이디어에 기초하는 반면에 SVMs는 일반화 에러의 상한(upper bound)을 최소화하는, 구조적 위험 최소화(structural risk minimization) [Vapnik, 1995]라 불리는 또 다른 추론원리(induction principle)에서 동작한다. 구조적 위험 최소화는 가장 낮은 true error를 보장할 수 있는 가설 h를 찾는 것이다. 가설 h의 true error는 무작위로 선택되어지고 미리 알 수 없는 테스트 예제에 대한 h의 에러 생성 확률이라 볼 수 있다. 아래의 상한은 가설 h의 true error를 학습 집합에서의 h의 에러와 h 복잡성의 결합으로 보고 있다 [Vapnik, 1995].

$$P(error(h)) \le train\_error(h) + 2\sqrt{\frac{d(\ln\frac{2n}{d}+1)-\ln\frac{\eta}{4}}{n}} \qquad (1)$$

여기서 학습률 η 는 학습자에 의해 지정된다. n은 학습 예제들의 수를 나타내고 d는 가설공간의 속성과 표현정도를 나타내는 VC-Dimension (VCdim)이다 [Vapnik, 1995]. 식 (1)은 최소 $1-\eta$의 상한 확률을 가진다. 따라서, 가설공간의 복잡도와 학습에러의 trade-off 관계를 보여주고 있다. 단순한 가설공간(작은 VCdim)은 성능이 좋은 예측함수를 포함하지 못하여 높은 학습에러로 이어질 것이다. 반대로 너무 복잡한 가설공간(큰 VCdim)은 적은 학습에러를 내겠지만 식 (1)의 부등호 오른쪽 편 두 번째 항이 커질 것이다. 이러한 상황을 과학습(overfitting)이라 한다. 따라서 옳은 복잡도를 갖는 가설공간를 고르는 문제는 어렵다고 볼 수 있다.

구조적 위험 최소화는 가설공간들 H_i의 구조를 그들의 대응되는 VC-Dimension d_i가 증가되게 정의하는 것에 의해 행해진다.

$$H_1 \subset H_2 \subset H_3 \subset ... \subset H_i \subset ... \quad \text{and} \quad \forall i : d_i \le d_{i+1} \tag{2}$$

결국 목적은 식 (1)이 최소화되는 인덱스 i^*를 찾는 것이다.

VCdim이 증가되는 구조를 만들기 위해 아래와 같은 타입의 선형 임계 함수(linear threshold functions)를 생각할 수 있다.

$$h(\vec{d}) = sign\{\vec{w} \cdot \vec{d} + b\} = \begin{cases} +1, & if \quad \vec{w} \cdot \vec{d} + b > 0 \\ -1, & otherwise \end{cases} \tag{3}$$

자질 선택 방법(feature selection strategy)인 자질들의 개수에 기초한 구조를 만드는 대신에 support vector machines는 텍스트 분류에서 대부분의 자질들은 관련이 있다는 사실을 내포하는 정제된 구조를 사용한다.

Lemma 1. [Vapnik, 1982] 초평면(*Hyperplanes*) $h(\vec{d}) = sign\{\vec{w}\cdot\vec{d}+b\}$ *를 가설이라고 간주하자. 만약 모든 예제 벡터* $\vec{d}_i$ *는 반지름이 R 인 구모양에 포함되어있고 모든 예제들* $\vec{d}_i$ *에 대해 아래의 수식이 성립한다면*

$$\left|\vec{w}\cdot\vec{d}_i+b\right| \geq 1, \quad with \quad \left\|\vec{w}\right\| = A, \tag{4}$$

이 hyperplane 집합은 아래와 같이 제한되는 VCdim d 를 갖는다.

$$d \leq \min([R^2A^2], n)+1 \tag{5}$$

이 hyperplane 들의 VCdim 은 자질들의 개수에 종속되지 않고 가중치 벡터 $\vec{w}$ 의 유클리드 길이(Euclidean length) $\left\|\vec{w}\right\|$ 에 종속된다. 이것은 만약 우리의 가설이 하나의 작은 가중치 벡터를 갖는다면 고차원의 공간을 잘 일반화 시킬 수 있다는 것을 의미한다.

이러한 기본적인 형태에서 support vector machines 는 학습데이터를 분류하고 가장 작은 가중치 벡터를 가지는 hyperplane 을 찾는다. 이 hyperplane 이 positive 학습 예제들과 negative 학습 예제들을 최대 마진(margin)으로 분류한다. 그림 1 에서 이것을 보여주고 있다.

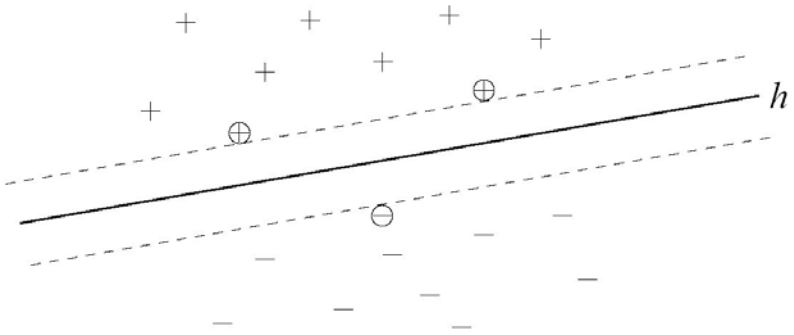

그림 1: Support vector machines 는 정답 학습 예제들과 오답 학습 예제들을 최대 마진(margin)으로 분류하는 hyperplane 을 찾는다. Hyperplane 에 가장 가까운 예제들은 *Support Vectors* (원으로 표시)라 불린다.

이 hyperplane 을 찾는 것은 다음의 최적화 문제로 해석될 수 있다.

$$\text{Minimize: } \left\|\vec{w}\right\| \tag{6}$$

$$\text{such that: } \forall i : y_i[\vec{w}\cdot\vec{d_i} + b] \geq 1 \tag{7}$$

y_i 는 만약 문서 d_i가 클래스 + (−)에 있다면 + 1 (−1)이다. 제약 조건 (7)은 모든 학습 예제들이 옳게 분류되는 것을 필요로 한다. 그러므로, 우리는 lemma 1 을 사용하여 분류 hyperplane을 만들어 내는 구조적 요소의 VCdim에 관한 결론을 이끌어 낼 수 있다. 즉, 식 (1)과 유사한 범위적 제약[Shawe-Taylor et al., 1996]은 분류 문제에 대하여 이 hyperplane의 true error에 범위적 제약을 지울 수 있다.

위와 같은 최적화 문제는 숫자상으로 처리하기는 어렵기 때문에 lagrange multipliers를 이용하여 동치의 이차(quadratic) 최적화 문제로 변환하여 생각할 수 있다[Vapnik, 1995].

$$\text{Minimize: } -\sum_{i=1}^{n}\alpha_i + \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \overrightarrow{d_i}\cdot\overrightarrow{d_j} \tag{8}$$

$$\text{such that: } \sum_{i=1}^{n}\alpha_i y_i = 0 \text{ and } \forall i : \alpha_i \geq 0 \tag{9}$$

이런 종류의 최적화 문제에 대하여 전역 최적해를 찾는 것을 보장하는 효율적인 알고리즘이 존재한다. 최적화 과정의 결과는 식 (8)을 최소화 시키는 계수 α_i^* 들의 집합이다. 이 계수들은 식 (6)과 (7)을 만족시키는 hyperplane을 만드는데 사용될 수 있다.

$$\vec{w}\cdot\vec{d} = (\sum_{i=1}^{n}\alpha_i^* y_i \overrightarrow{d_i})\cdot\vec{d} = \sum_{i=1}^{n}\alpha_i^* y_i(\overrightarrow{d_i}\cdot\vec{d}) \text{ and } b = \frac{1}{2}(\vec{w}\cdot\overrightarrow{d_+} + \vec{w}\cdot\overrightarrow{d_-}) \tag{10}$$

식 (10)은 hyperplane의 결과 가중치 벡터가 학습예제들의 선형조합으로 만들어 지는 것을 보여준다. 여기서 계수 α_i가 양수가 되는 예제들만 이용된다. 그러한 벡터들을 *Support Vectors* 라 한다. 그림 1에서 support vectors는 원으로 표시되어 있다. 그것들은 hyperplane으로부터 최소의 거리를 가지는 학습예제들이다. b 를 계산하기 위해서 두개의 임의의 support vectors $\overrightarrow{d_+}$ 와 $\overrightarrow{d_-}$ (하나는 클래스 +에서 하나는 클래스 −에서)가 이용될 수 있다.

3.2.1 비선형 가설 공간

비선형 가설을 학습하기 위해서 SVMs 은 convolution 함수를 사용한다. Convolution 함수의 타입에 따라서 SVMs 는 polynomial 분류기, radial basis 함수(RBF) 분류기, 2-layer sigmoid 신경망을 학습할 수 있다.

$$K_{poly}(\overrightarrow{d_1} \cdot \overrightarrow{d_2}) = (\overrightarrow{d_1} \cdot \overrightarrow{d_2} + 1)^d \quad (11)$$

$$K_{rbf}(\overrightarrow{d_1} \cdot \overrightarrow{d_2}) = \exp(\gamma(\overrightarrow{d_1} - \overrightarrow{d_2})^2) \quad (12)$$

$$K_{sigmoid}(\overrightarrow{d_1} \cdot \overrightarrow{d_2}) = \tanh(s(\overrightarrow{d_1} \cdot \overrightarrow{d_2}) + c) \quad (13)$$

이 convolution 함수들은 Mercer의 가설을 만족시킨다[Vapnik, 1995]. 즉, 벡터 $\overrightarrow{d_1}$ 와 $\overrightarrow{d_2}$ 가 비선형 매핑 Φ 에 의하여 새로운 “feature” 공간에 매핑된 후 내적이 계산됨을 의미한다.

$$\Phi(\overrightarrow{d_1}) \cdot \Phi(\overrightarrow{d_2}) = K(\overrightarrow{d_1} \cdot \overrightarrow{d_2}) \quad (14)$$

Convolution 함수를 사용하기 위하여는 식 (8), 식 (10)에 나와있는 모든 내적을 희망하는 convolution 함수로 대체시키면 된다. 그러면 support vector machine은 비선형 feature 공간에서 가장 넓은 마진을 가지고 학습 데이터를 구분하는 hyperplane을 찾을 수 있다.

3.2.2 최상의 파라미터값

Convolution 함수를 사용함에 있어 파라미터들이 도입되었다. Polynomial convolution 에 있어서는 degree d 가 그것이고 RBFs 에 대해서는 variance γ 등이다. 그러면 적당한 파라미터값들을 어떻게 자동적으로 고를 수 있을까? 식 (1)에 의해서 유추된 다음의 절차 [Vapnik, 1995]가 사용될 수 있다. 우선 support vector machine 을 d, γ 의 몇 가지 다른 값들에 대해 학습시킨다. 그 다음 식 (5)를 사용하여 찾은 가설들의 VCdim 을 예측하고 가장 작은 값의 VCdim 을 고른다.

가중치 벡터의 길이를 계산하기 위해 아래 식을 사용할 수 있다.

$$\|w\|^2 = \sum_{i,j \in SupportVectors} \alpha_i \alpha_j y_i y_j K(\overrightarrow{d_i}, \overrightarrow{d_j}) \qquad (15)$$

그리고 모든 문서 벡터들이 단위길이로 정규화되었다면 모든 학습예제들을 포함하는 구의 반지름 R이 식 16처럼 제한된다.

$$\text{Polynomial: } R^2 \le 2^d - 1 \qquad \text{RBF: } R^2 \le 2(1-\exp(-\gamma)) \qquad (16)$$

적당한 파라미터값을 선택하기 위한 이 절차는 완전히 기계적이고 테스트 데이터를 참고할 필요도 없고 비용이 높은 cross-validation도 필요치 않다.

3.2.3 분류할 수 없는 문제들

지금까지는 학습 데이터가 에러없이 분류가능하다는 전제를 두었었다. 만약 선택되어진 가설공간에 대해 이것이 불가능하다면 어떻게 할 것인가? Cortes와 Vapnik [Cortes and Vapnik, 1995]이 slack 변수라는 것을 통한 방법에 대해 제안을 하였다. 이 방법에서는 하나의 단순한 절차가 사용되었다. 즉, 식 (8)의 최적화 과정에서 계수 α_i의 값들이 모니터 된다. 높은 α_i을 가지는 학습 예제들이 데이터의 분류 불가능성에 많이 기여함을 알았고 α_i의 값이 어떤 임계치 C 값(예로 $\alpha_i \geq 1000$)을 초과하면 그에 대응되는 학습 예제가 학습 집합에서 제거된다. 이런 방식으로 SVMs는 남은 데이터에 대해 학습이 가능하게 된다.

3.3 알고리즘

본 논문에서 제시한 support vector machines 를 이용한 문서 정보 기반의 단백질 기능 분류 방법론의 전체 알고리즘은 다음의 그림과 같다.

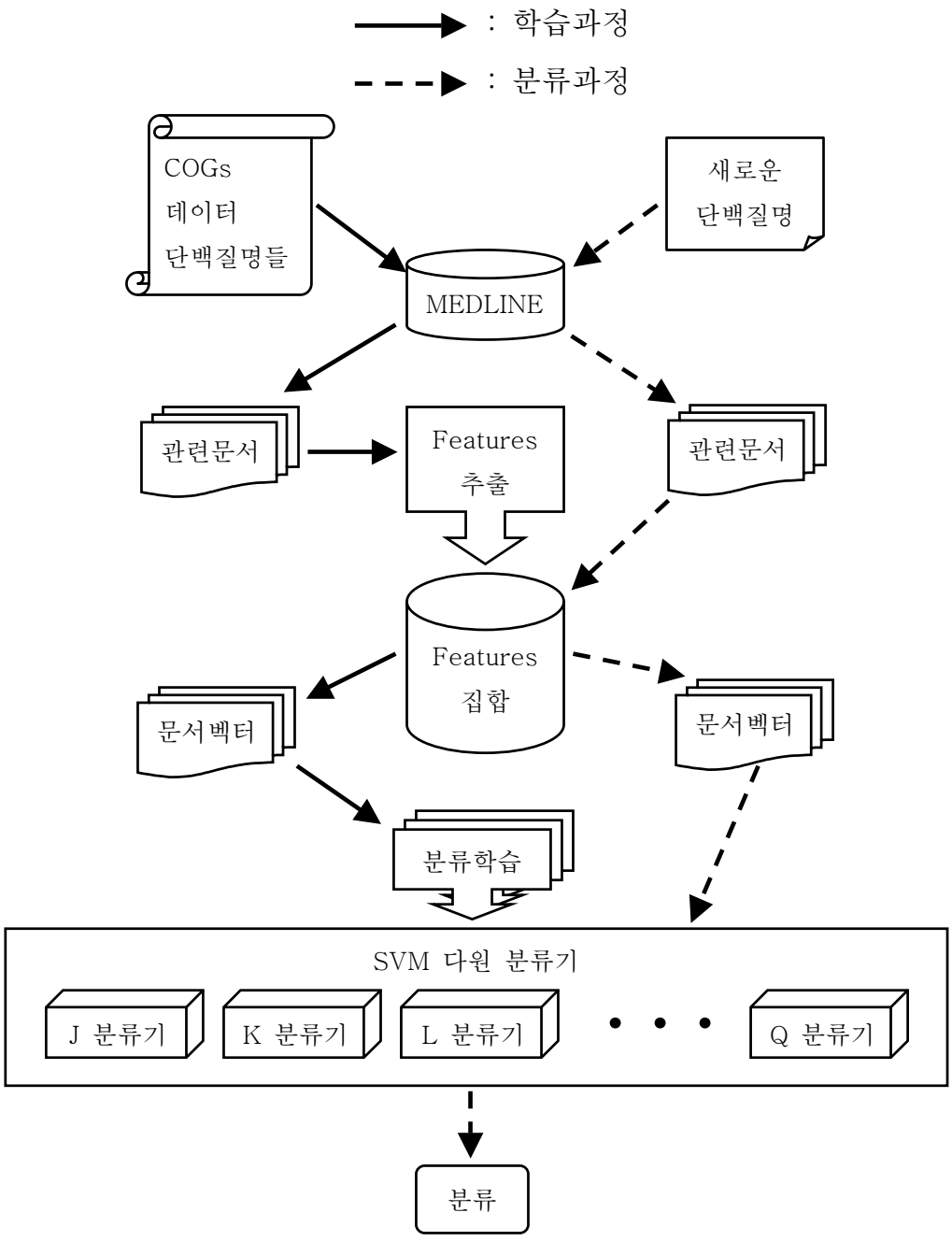

그림 2: Support vector machines 를 이용한 문서 정보 기반의 단백질 기능 분류.

그림 2 는 본 논문에서 제시한 단백질 기능별 분류 방법론의 전체 알고리즘을 나타낸다. 우선 실선은 학습 단계를 나타내며 점선은 학습 후에 실제 데이터가 들어왔을 때의 분류 단계를 나타낸다.

알고리즘은 COGs 데이터의 각 기능별(J~Q) 단백질명들을 가지고 MEDLINE을 검색하는 것에서 시작한다. 그 단백질명들을 포함하는 문서들을 찾아낸 후 그 문서들의 features를 추출한다. 이때 stopwords를 제거하거나 stemming 작업을 하여 features 수를 줄일 수 있다. SVMs는 매우 많은 수의 features를 성능에 지장을 주지않으면서 처리할 수 있기 때문에 본 논문에서는 편의상 stopwords 제거 작업만을 하였다 [Joachims, 2001].

이렇게 만들어진 features 집합을 가지고 각 문서들을 벡터공간 모델로 표현하였다. 이때, 문서의 값은 features의 가중치에 의해 결정되기 때문에 가중치 부여 기법은 분류 결과에 중요한 요소로 작용할 수 있다. 이렇듯 features 에 가중치를 부여하기 위하여 출현빈도(term frequency), 문서빈도(document frequency)의 두 가지 요소를 고려할 수 있다. 출현빈도는 문서 내에서 자주 출현하는 features 에 보다 높은 가중치를 부여한다. 문서빈도는 전체 문서들 중에서 적은 수의 문서에 출현하는 features 에 보다 높은 가중치를 부여한다.

벡터공간 모델에서 사용하고 있는 벡터 원소의 가중치를 결정하는 *tf-idf* 방법을 설명하면 다음과 같다. 벡터 공간의 각각의 차원(dimension)은 각 단어에 해당하는 가중치를 나타낸다. 벡터의

각 값(vector element)은 각 문서에 대한 단어 빈도수 (term frequency) $TF(w,d)$와 문서 빈도수(document frequency) $DF(w)$의 조합으로 계산된다. $TF(w,d)$는 단어(word) w가 문서(document) d에 나타난 횟수를, $DF(w)$는 단어 w가 한 번 이상 나온 문서의 수를 나타낸다. 문서 빈도수로부터 역 문서 빈도수(inverse document frequency) $IDF(w)$를 다음의 식과 같이 계산한다.

$$IDF(w) = \log \frac{|D|}{DF(w)} \tag{17}$$

여기서 $|D|$는 문서의 총 개수이고 벡터의 i번째 값 $d^{(i)}$는 다음과 같이 두 값의 곱으로 계산된다.

$$d^{(i)} = TF(w_i, d) \times IDF(w_i) \tag{18}$$

이렇게 만들어진 문서 벡터들을 가지고 SVMs 분류기를 J 기능, K 기능, L 기능, …, Q 기능 총 16 개의 범주에 관하여 각각 학습을 시켰다. 결과로 16 개의 SVMs 분류기를 얻을 수 있었고 이를 이용하여 SVMs 다원 분류기를 구현하였다.

다음으로 실제 분류 단계로 새로운 단백질 명이 주어졌을 때 이것의 기능을 찾아내는 과정을 보면 학습할 때와 마찬가지로 우선 MEDLINE 데이터베이스를 검색하여 그 단백질 명을 포함하고 있는 문서들을 찾아낸다. 다음 단계로 학습 시 정해진 features 집합을 이용하여 각 문서들을 벡터형태로 표현한다. 이 문서 벡터들을

16 개의 SVMs 분류기에 입력으로 넣어 결과로 나온 값들을 모두 합하여 가장 높은 결과치를 보인 SVMs 분류기를 찾아낸다. 이를 다음과 같이 수식으로 나타내었다.

$$c_{COGs} = \arg\max_{c \in COGs} \sum_{i=1}^{m} S_c(d_i) \tag{19}$$

여기서 c는 COGs의 16 개의 범주 중 하나이고 m은 문서의 개수를 나타내며 S 함수는 SVMs 분류기를 나타낸다. 이렇게 하여 문서 정보만을 이용하여 단백질 하나 하나의 기능들을 찾아 낼 수 있는 다원 분류기를 구현할 수 있었다.

Ⅳ. 실험 및 분석

4.1 효모균 (Saccharomyces cerevisiae)

실험에 사용한 텍스트 데이터는 NCBI의 MEDLINE 데이터이다. 실험 방법은 우선 MEDLINE 에서 119,526 개의 효모균 관련 논문 abstracts를 모았다. 이렇게 모아진 텍스트 데이터에서 모든 단백질 명을 추출하는 대신 COGs 데이터에 나와있는 효모균 단백질 명 (총 2306 개)을 이용하여 COGs 에 속하는 단백질 명만을 스트링 매칭 방법을 사용하여 검색하였다. 그러나 실제로 효모균 COGs 데이터에는 단백질명이 직접 나와 있지 않고 COGs 번호만 나와있어 각 COGs 번호에 해당되는 효모균 단백질 명을 알아내기 위해 NCBI 내의 효모균관련 시퀀스 정보를 사용하였다. 본 실험에서 단백질의 동의어는 검색 결과 거의 문서 데이터에 나오지 않았고 한 종류의 단백질명 만을 가지고도 실험에 필요한 충분한 수의 문서 데이터를 얻을 수 있었기 때문에 전체 실험 결과에 큰 영향을 안 준다고 볼 수 있어서 본 실험에서는 실험의 효율을 위하여 단백질의 동의어는 사용하지 않았다.

효모균 COGs 데이터에 나와있는 16 개의 기능적 범주에 속하는 단백질명을 이용하여 119,526 개의 관련 문서들을 각각의 범주로 나누었고 이를 학습 및 검증 데이터로 이용하였다. 이때 학습 데이터와 테스트 데이터의 구성은 아래와 같다.

Code	Training Abstracts	Test Abstracts	Total Abstracts Number
J	653	224	877
K	410	141	551
L	3049	1048	4097
D	48	16	64
O	989	340	1329
M	163	56	219
N	112	39	151
P	642	221	863
T	2450	841	3291
C	879	302	1181
G	1067	366	1433
E	1682	578	2260
F	863	296	1159
H	419	144	563
I	371	127	498
Q	343	118	461
SUM	14140	4857	18997

표 6: 효모균의 학습 및 검증 데이터

위 표 6에서 보듯이 초기 검색된 119,526 개의 효모균 관련 문서중 실제로 COGs 데이터의 효모균 단백질 명을 포함하고 있는 문서는 25,433 개 (R, S 기능 포함)였다. 또한, R 기능과 S 기능은 아직 그 기능이 확실히 밝혀진 것이 아니기 때문에 학습 및 테스트 데이터에서 제외시켰다.

이렇게 모아진 전체 문서 데이터에서 stopwords를 제거하고 61701 개의 features를 추출하였다. 이를 통하여 문서 데이터를 벡터 공간 모델로 표현하였다. 여기서 각 feature에 해당되는 가중치 값을 계산하기 위하여 아래표와 같이 여러가지 TF 계산식을 고려할 수 있었다.

종류	공식
이진값	1 (if tf>0), 0
단순 TF	$TF = tf$
Log TF	$TF = 1 + \log(tf)$
Root TF	$TF = \sqrt{tf}$
보정 TF	$TF = (1-w) + w \times \frac{tf}{\max \quad tf}$
Okapi TF	$TF = \frac{tf}{2+tf}$

표 7: TF (term frequency)

본 논문에서는 여러 TF 계산식들에 따라 분류 성능 차이를 비교하기 위하여 각각의 TF 계산식에 대하여 TF 값을 계산하고 문서벡터를 생성하였다.

이렇게 문서벡터 집합을 만든 후 16 개의 범주 각각에 대하여 그 범주에 속한 표 6의 문서 분류 정보를 이용하여 support vector machines 를 학습시켰다.

실험 결과의 성능 평가를 위하여는 accuracy, recall, precision, F_1 값을 고려할 수 있었다. Accuracy 는 정확도로 전체 테스트 데이터에서 옳게 분류된 데이터 개수의 비율이다. Recall 은 재현율로 정답 범주에 속하는 문서가 정답으로 분류될 확률이고 precision은 정답 범주일거라고 예측된 문서가 실제로 정답일 확률이다. F_1 값은 recall 값과 precision 값을 구해 평균을 한 값이다. 그러나 본 논문에서는 각기 다른 조건에서의 분류 성능을 쉽게 비교하기 위하여 F_1 값 만을 사용하였다. 아래에 이들 식을 나타내었다.

$$Accuracy = \frac{f_{true\_positive} + f_{true\_negative}}{f_{true\_positive} + f_{true\_negative} + f_{false\_positive} + f_{false\_negative}} \quad (19)$$

$$\mathrm{Re}call = \frac{f_{true\_positive}}{f_{true\_positive} + f_{false\_positive}} \quad (20)$$

$$\mathrm{Pr}ecision = \frac{f_{true\_positive}}{f_{true\_positve} + f_{true\_negative}} \quad (21)$$

$$F_1 = \frac{2 f_{true\_positive}}{2 f_{true\_positive} + f_{true\_negative} + f_{false\_negative}} \quad (22)$$

실험은 우선 여러 TF 값에 따른 분류 성능 평가를 위하여 표 7 의 각 TF 계산식들에 대하여 16 개 범주를 대상으로 학습 및 테스트를 하였고 실험결과는 아래와 같았다. 단, 이때 C=1000 으로 하는 선형 SVMs 를 사용하였다.

	이진값		단순		LOG		Root		보정		Okapi	
	tf	*tfidf*	*tf*	*tfidf*	*tf*	*tfidf*	*tf*	*tfidf*	*tf*	*tfidf*	*tf*	*tfidf*
J	91.5	92.4	92.3	92.6	90.4	90.6	93.1	92.8	92.7	91.8	93.3	92.1
K	92.1	90.4	91.8	90.5	91.3	88.1	92.0	91.6	91.6	87.4	92.4	89.7
L	89.6	92.7	90.4	93.1	89.2	88.7	93.7	93.3	93.8	89.1	94.1	91.9
D	87.4	87.6	87.8	88.7	87.6	86.9	89.1	88.9	88.6	86.7	89.3	87.5
O	91.7	91.4	92.1	92.6	90.8	89.4	93.2	93.0	93.4	89.6	93.7	90.3
M	89.7	88.8	90.6	89.7	90.7	88.7	91.8	90.3	91.5	88.2	92.1	87.5
N	90.8	89.7	91.6	90.8	89.9	87.5	91.3	91.2	92.1	88.5	91.5	89.4
P	89.2	89.5	88.6	90.3	88.1	88.9	90.9	90.4	89.7	89.3	91.8	88.7
T	91.1	90.9	91.8	91.1	90.6	86.9	92.7	91.9	92.6	87.5	93.1	89.9
C	89.6	91.2	90.7	92.5	89.7	90.1	92.9	92.8	91.8	90.3	93.4	90.7
G	90.3	89.7	91.4	90.9	91.1	88.7	92.4	91.9	92.5	88.9	92.7	89.3
E	91.5	91.9	92.1	92.2	90.3	90.9	92.8	92.6	91.4	91.0	93.6	90.5
F	92.1	92.4	91.8	93.4	91.4	89.7	93.7	93.7	92.7	90.7	94.2	91.3
H	89.7	88.8	90.3	89.7	88.8	86.9	91.5	90.4	90.8	86.8	91.8	87.4
I	87.6	88.7	88.7	89.1	89.6	88.1	89.9	89.5	89.4	87.5	90.7	87.6
Q	88.8	88.1	89.4	89.8	87.9	86.8	90.9	90.3	90.1	87.4	91.4	87.9
평균	90.2	90.3	90.7	91.1	89.8	88.6	92.0	91.5	91.5	88.8	***92.4***	89.5

표 8: TF 계산식들에 따른 분류 성능

표 8 에서 볼 수 있듯이 전체적인 SVMs 분류 성능은 높게 나왔고 여러 TF 계산식에 따른 TF 값들이 분류 성능에는 크게는 영향을 안주었다. 특히, *tf-idf* 에 비해 *tf* 만을 features 값으로 사용하여도 우수한 분류 성능을 보였다. 그러나 그 중에서 Okapi TF 계산식이 가능 좋은 성능을 보여 이후의 실험에서는 이 계산식을 사용하였다.

다음 실험으로는 학습 데이터가 정확히 두 범주로 나누어지지 않는 경우일 때 학습 데이터에 어느 정도의 오류를 허용하기 위한 임의의 C 값에 따른 분류 성능의 실험 결과치를 비교하였다. 이 실험에서도 마찬가지로 선형 SVMs 를 사용하였다.

C	*0.05*	*0.1*	*0.5*	*1.0*	*5*	*10*	*1000*
J	54.2	63.8	90.3	95.4	95.3	94.4	93.3
K	53.1	61.9	89.9	94.3	94.2	93.8	92.4
L	54.3	62.3	91.2	95.2	95.0	94.7	94.1
D	52.3	59.9	88.5	91.8	91.5	90.3	89.3
O	55.8	62.5	90.4	94.3	94.2	93.9	93.7
M	51.9	63.0	90.1	93.5	93.2	92.5	92.1
N	50.3	61.2	89.3	92.9	92.3	91.8	91.5
P	51.4	59.5	92.1	93.2	92.9	92.2	91.8
T	52.1	60.8	92.9	94.5	94.2	93.4	93.1
C	51.0	61.2	92.6	94.5	94.1	93.6	93.4
G	52.1	62.9	92.3	94.7	94.2	93.7	92.7
E	49.9	62.8	93.0	95.2	94.4	94.1	93.6
F	53.7	63.1	93.4	96.7	95.2	94.5	94.2
H	50.3	60.4	91.7	93.5	92.9	92.3	91.8
I	51.5	60.6	91.5	92.1	91.4	91.0	90.7
Q	52.1	61.2	92.2	93.5	92.8	91.9	91.4
평균	52.3	61.7	91.3	***94.1***	93.6	93.0	92.4

표 9: *C* 값에 따른 분류 성능 비교

표 9 에서 보듯이 *C* 값이 0.5 미만에서는 분류 성능이 나쁘게 나오다가 *C* 값이 1.0 인 부분에서 가장 좋은 성능을 보였다. 그러나 그 이후로는 전체적으로 큰 차이는 보이지 않았다.

다음으로 비선형 SVMs인 다항식 분류기를 학습하기 위하여 몇 가지 *d* 값에 대하여 분류 성능을 실험, 비교하였다. 이 때 *C* 값은 1.0 으로 고정 시켰다.

d	*2*	*3*	*4*	*5*
J	90.1	93.2	93.3	93.1
K	88.9	93.0	93.4	92.4
L	90.2	92.5	92.6	90.5
D	89.5	90.2	91.2	90.8
O	91.4	92.4	93.1	92.3
M	90.3	91.5	92.4	91.4
N	88.2	91.7	92.0	91.2
P	91.0	91.2	92.2	91.5
T	92.5	92.3	93.4	92.2
C	92.1	93.1	93.9	91.4
G	91.3	92.7	93.2	92.5
E	92.9	92.1	92.3	92.2
F	92.4	93.5	94.3	92.5
H	90.8	91.1	92.2	91.9
I	93.2	90.5	91.2	91.2
Q	91.6	91.4	92.1	92.3
평균	91.0	92.0	**92.7**	91.8

표 10: *d* 값에 따른 분류 성능 비교

표 10 에서 보듯이 SVMs 다항식 분류기의 전체적인 분류 결과는 고른 결과값을 보였으며 *d* 값이 4 일 때 가장 좋은 성능을 보였다.

효모균에 대한 마지막 실험으로는 또 다른 비선형 SVMs인 RBF 분류기의 파라미터값 γ에 따른 성능 평가를 위하여 6 가지의 γ 값에 대하여 각 범주를 학습시키고 분류 결과를 비교하였다. 마찬가지로 C값은 1.0으로 고정 시켰다.

γ	*0.01*	*0.03*	*0.1*	*0.3*	*1.0*	*3.0*
J	94.2	92.5	93.1	93.4	90.3	69.5
K	93.4	91.1	91.5	92.1	89.6	67.7
L	94.3	92.3	92.6	93.5	89.3	68.9
D	90.4	88.4	89.1	89.8	85.6	69.2
O	93.3	92.2	92.4	92.6	89.8	70.3
M	92.5	90.3	90.9	91.3	88.7	67.7
N	91.4	89.7	90.3	90.9	87.3	65.5
P	92.0	89.9	90.0	90.4	86.4	66.1
T	93.2	91.3	91.7	92.3	89.8	68.6
C	92.3	89.9	90.2	91.4	87.4	66.3
G	93.9	91.5	91.9	92.5	90.3	70.0
E	94.2	91.2	91.4	92.9	89.8	65.5
F	95.5	92.6	92.9	93.8	90.4	69.4
H	92.4	90.6	91.1	91.5	88.2	67.6
I	91.9	89.0	89.5	89.9	87.7	67.4
Q	92.3	90.7	91.4	91.8	86.4	68.5
평균	***93.0***	90.8	91.3	91.9	88.6	68.0

표 11: γ 값에 따른 분류 성능 비교

표 11 에서 볼 수 있듯이 γ이 3.0 과 1.0 일 때를 제외하고는 SVMs 다항식 분류기와 대체로 비슷한 성능을 보였고 γ이 0.01일 때 가장 좋은 성능을 나타내었다.

결과로 전체적인 SVMs의 성능을 비교해 보면 C=1.0인 선형 SVMs 에서 가장 좋은 분류 성능을 나타내었고 비선형 SVMs 인 다항식 분류기와 RBF 분류기는 큰 차이는 보이지 않았지만 γ 값이 0.01인 RBF 분류기가 다항식 분류기보다 좋은 분류 성능을 나타내었다. 또 다른 사실은 4 가지 실험 모두에서 범주 D의 분류 성능이 다른 범주들보다 조금 낮은 결과를 보였다. 이는 표 6 에서 보듯이 범주 D의 학습데이터 수가 다른 범주보다 적은 것이 영향을 주는 것 같았다.

4.2 대장균 (Escherichia coli)

본 논문에서 제시한 단백질 기능 분류 방법론의 일반성, 범용성을 밝히기 위하여 효모균 문서 정보만을 가지고 학습한 support vector machines 분류기를 대장균 COGs 데이터를 가지고 검증하였다. 우선 마찬가지로 대장균 단백질 관련 문서를 MEDLINE 데이터베이스에서 검색하여 총 195,000 개의 문서를 얻었다. 일반적으로 대장균은 두가지 종류로 나눌 수 있는데 O157과 K12이다. O157은 식중독 원인균으로 유명해진 대장균의 일종이고 K12는 유전공학 실험에 널리 쓰이는 대장균이다. 본 논문에서는 해당 단백질 수가 K12보다 조금 더 많은 O157 대장균 COGs 정보를 가지고 실험을 하였다.

이에 대장균 O157 에 대하여 총 2867 개의 단백질 명을 COGs 데이터에서 얻을 수 있었고 이를 가지고 관련된 MEDLINE 문서를 검색한 결과 각 범주에 속하는 단백질 명을 포함하는 문서들을 아래와 같이 분류할 수 있었다.

Code	Total Abstracts Number	Duplicated Abstracts Number	Uniqueness of Abtracts (%) and (Unique Abstracts)
J	20350	1807	91.1 (18543)
K	7674	1806	76.5 (5868)
L	34319	4651	86.5 (29668)
D	741	108	85.4 (633)
O	2982	363	87.8 (2619)
M	10040	1005	90.0 (9035)
N	1100	133	87.9 (967)
P	8793	862	90.2 (7931)
T	3131	612	80.5 (2519)
C	1510	219	85.5 (1291)
G	12861	1817	85.9 (11044)
E	28693	3887	86.5 (24806)
F	3471	458	86.8 (3013)
H	66914	9718	85.5 (57196)
I	505	80	84.2 (425)
Q	154	21	86.4 (133)
R	2666	501	81.2 (2165)
S	105	17	83.8 (88)
SUM	206009	28065	86.4 (177944)

표 12: 대장균 O157 의 각 기능적 분류에 속하는 단백질 명들을 이용한 검색 결과 분석.

위 표에서 보듯이 대장균에서는 총 195,000 개의 대장균 관련 문서 중 실제로 COGs 데이터의 대장균 O157 단백질 명을

포함하고 있는 문서는 177,944 개 였다. 대장균 O157 단백질의 개수가 2867개였고 효모균 단백질이 2306 개로 그 개수 차이가 별로 없다는 사실을 고려한다면 각 개체의 관련 문서 개수가 2 배정도 차이 (효모균 관련 문서 119,500 개)가 나고 실제로 각 단백질 명을 포함하는 문서의 개수 (효모균 단백질 명 포함 문서 25,433 개)는 7 배나 차이를 보인 것은 효모균보다 대장균의 연구가 더욱 활발했다는 것을 보여주는 것이다. 이런 연유로 대장균에서는 단순히 해당 단백질 명을 포함하는 문서들을 18 개의 기능적 범주로 분류할 경우 각 기능적 분류에 속하는 문서들이 그 한 기능에만 유일하게 속할 확률이 평균 86.4% 였다. 이는 효모균의 98.3% 보다 11.9% 가 감소한 수치였다.

실험은 우선 앞 절에서 효모균 문서 정보만으로 생성한 분류기를 대장균 문서 정보에도 적용했을 때의 결과를 확인하기 위하여 앞 절의 실험과는 틀리게 표 12 에서 보듯이 각 기능에 유일하게 속하는 문서들 만을 다시 간추려서 각 SVMs 분류기의 분류 성능을 확인하였다.

	선형 SVMs		비선형 SVMs	
	C=1.0	*C=1000*	*γ =0.01*	*γ =0.3*
J	89.7	87.4	89.1	86.8
K	88.2	86.6	87.9	87.0
L	90.4	88.1	89.2	88.5
D	85.7	84.5	85.1	84.3
O	89.6	87.4	88.4	86.9
M	88.8	87.2	88.3	87.3
N	87.2	85.6	87.3	86.2
P	88.1	86.9	86.9	87.1
T	89.7	87.4	89.2	86.8
C	89.5	88.0	88.7	87.9
G	88.3	86.6	88.1	87.2
E	89.9	88.3	88.6	87.7
F	90.1	88.5	89.0	88.6
H	87.4	86.1	86.4	86.3
I	86.3	84.0	87.5	85.1
Q	88.2	86.2	87.8	86.5
평균	***88.6***	86.8	88.0	86.9

표 13: 효모균으로 학습한 SVMs 를 사용하여 대장균을 분류한 결과 비교.

표 13 에서 보듯이 각각의 SVMs 분류기의 성능은 효모균의 분류 결과와 비교하여서는 그 수치가 떨어졌다. 그러나 본 논문의 목적인 문서 정보만으로 하나 하나의 단백질 기능을 밝히는 문제에 있어서는 만족할만한 수치였다. 왜냐하면 본 논문이 제시한 방법론은

승자독식방법(winner-takes-all)을 이용하여 support vector machines 다원 분류기(multi-class classifier)를 구현하는 것인데 이는 만약 어떤 단백질을 COGs의 기능적 범주로 분류한다면 16 개의 범주 중 반드시 하나에는 포함되기 때문이다.

즉, 아무리 표 13 의 분류 성능이 평균 88.6 (C=1.0 인 경우) 이지만 각각의 분류기는 상대적으로 16 개의 분류기중 자기 범주에 속하는 문서를 자기 범주로 옳게 분류할 확률은 다른 범주의 분류기보다는 월등히 높다는 것이다.

실제로 J 범주에 속하는 문서 18543 개를 모든 범주에 대하여 정답 문서 데이터라 가정하고 각 SVMs 분류기의 분류 성능을 C=1.0 인 선형 SVMs 를 사용하여 비교해봤다.

J	***89.7***	T	25.4
K	27.8	C	32.1
L	30.2	G	23.2
D	25.6	E	28.3
O	28.4	F	25.8
M	29.4	H	31.5
N	27.9	I	24.8
P	33.6	Q	29.5

표 14: 다원 분류기 예제 실험

표 14 에서 보듯이 비록 J 범주 하나만 본다면 89.7%의 분류 성능을 내지만 나머지 분류기의 결과값과 비교해 보면 월등한 차이를 보여 승자독식방법에 의하여 가장 큰 결과값을 낸 J 범주로 옳게 분류되었고 결과적으로 본 논문에서 제시한 다원 분류기의 성능은 신뢰할 만한 수준이라 할 수 있다.

V. 결론 및 향후 과제

본 논문에서는 문서 정보 기반의 단백질 기능 분류 방법론을 제시하였다. 단백질의 기능적 분류 문제를 해당 단백질 명을 포함하는 문서 분류 문제로 치환하였는데 이는 효모균은 98.3%의 확률로, 대장균은 86.4%의 확률로 하나의 문서는 하나의 단백질 기능에 관하여만 쓰여졌다는 사실에 그 근거를 두었다. 여기서 두 개체간 확률이 차이가 나는 것은 실제 생물학 실험실에서 대장균이 상대적으로 더 다루기 쉽고 연구하기에도 간단한 반면에 효모균은 진핵 생물이기 때문에 훨씬 더 중요한 실험 생물로서의 잠재능력을 지니고는 있지만 실험적으로 다루기 어렵기 때문에 연구결과도 광범위하지 않기 때문이다. 대다수의 개체가 실험적으로 다루기가 어렵다는 사실을 볼 때 생물학자들이 실험적으로 연구를 할 때는 주로 한번에 단백질의 한가지 기능에 관하여만 연구하는 경향이 있다고 말하여도 무난할 것이다. 또한, 승자독식방법을 취하는 본 논문의 다원 분류기의 관점에서 보더라도 위의 대장균의 확률은 신뢰할 만한 분류 결과를 준다.

SVMs 분류기의 성능을 비교해 보면 효모균 실험에서는 C=1.0인 선형 SVMs에서 가장 좋은 분류 성능(F_1 =94.1%)을 나타내었고 비선형 SVMs인 다항식 분류기와 RBF 분류기는 큰 차이는 보이지 않았지만 γ 값이 0.01인 RBF 분류기가 다항식 분류기보다 좋은 분류 성능(F_1=93.%)을 나타내었다.

효모균 실험에서 학습한 SVMs 분류기의 대장균 실험은 효모균 실험 결과 보다는 분류 성능(F_1 =88.6%)이 떨어졌지만 본

논문의 목적인 다원 분류기 구현 측면에서는 신뢰할 만한 성능을 나타내었다.

지난 몇 년간에 걸쳐 축적된 많은 양의 게놈과 cDNA (EST) 서열자료들 때문에 genome-wide expression 연구들에 대하여 생각하는 것이 가능해졌다. 예를 들어 거의 20 개 정도의 원핵생물들과 단세포인 효모균의 게놈들의 완전한 서열들이 이용 가능해졌다. 후생동물인 Caenorhabditis elegans 의 게놈분석도 완료되었으며, 작년에는 생쥐 게놈 물리지도도 완성되었으며, 올해는 인간 게놈지도도 완성되었다. 이러한 서열들은 서열상동성 연구에 의해 추론된 것들 말고는 기능적인 정보가 결여되어 있다. 분명히 21 세기의 생물학자는 이런 모든 서열자료들에 내제하는 기능들을 연구하고 이해할 수 있는 새로운 기술을 요구할 것이다.

본 논문에서 연구된 문서 정보 기반의 단백질 기능 분류기는 새로운 개체에서 새롭게 밝혀지는 단백질을 미리 문서 정보만을 가지고 COGs 의 기능적 범주로 분류해 보는데 사용할 수 있고 더욱이 단백질 서열 자체의(형태적) 유사성에 기반을 두고 기능을 분류한 COGs 데이터에서 27%의 단백질들이 아직 그 기능의 분류가 안된 상황임을 볼 때 본 연구는 COGs 데이터의 기능적 분류에 도움을 줄 수 있는 보조적 기능을 충분히 수행할 수 있을 거라 기대된다.

참고 문헌

[Ashburner et al. 2000] Ashburner, M., Ball, C.A., blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Gene Ontology: Tool for the unification of biology. The Gene Ontology Consortium, *Nat. Genet,* 25:25-29, 2000.

[Boser et al. 1992] Boser, B., Guyon, M., and Vapnik, V., A training algorithm for optimal margin classifiers, *Proceedings of Computational Learning Theory (COLT),* 144-152, 1992.

[Cai et al. 2003] Cai, C.Z., Wang, W.L., Sun, L.Z., Chen, Y.Z., Protein Function Classification via Support Vector Machine Approach, *Mathematical Biosciences*, 185:111-122, 2003.

[Cherry et al. 1998] Cherry, J.M., Adler, C., Ball, C., Chervitz, S.A., Dwight, S.S. et al., SGD: Saccharomyces Genome Database, *Nucleic Acids Research*, 26:1, 73-79, 1998.

[Cortes and Vapnik, 1995] Cortes, C., and Vapnik, V., Support-vector networks, *Machine Learning*, 20:273-297, 1995.

[Criekinge and Beyaert, 1999] Criekinge, W.V., and Beyaert, R., Yeast Two-Hybrid: State of the Art, *Biological Procedures Online*, 2:1-38, 1999.

[Cristianini and Shawe-Taylor, 2000] Cristianini, N., and Shawe-Taylor, J., *An Introduction to Support Vector Machines and other*

kernel-based learning methods, Cambridge University Press, 2000.

[Joachims, 1997] Thorsten Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features, *Proceedings of ECML-98*, 1997.

[Joachims, 2001] Thorsten Joachims, *Learning to Classify Text Using Support Vector Machines : Methods, Theory and Algorithms*, Kluwer Academic Publishers, 2001.

[Manning and Schutze, 1999] Manning, C.M., and Schutze, H., *Foundations of Statistical Natural Language Processing*, Cambridge: The MIT Press, 1999.

[Mewes et al. 2000] Mewes, H.W., Frishman, D., Gruber, C., Geier, B., Haase, D., Kaps, A., Lemcke, K., Mannhaupt, G., Pfeiffer, F., Schuller, C., MIPS: A database for genomes and protein sequences, *Nucleic Acids Res,* 28: 37-40, 2000.

[Moriyama and Kim, 2003] Moriyama, E.N., and Kim, J., Protein family classification with discriminant function analysis, *Proceedings of Stadler Genetics Symposium*, 2003.

[Pavlidis et al. 2001] Pavlidis, P., Weston, J., Cai, J., Grundy, W.N., Gene Functional Classification from Heterogenous Data, *Proceedings of the Fifth Annual International Conference on Computational Biology (RECOMB)*, 249-255, 2001.

[Raychaudhuri et al. 2003] Raychaudhuri, S., Schutze, H., and Altman, R.B., Inclusion of Textual Documents in The Analysis of

Multidimensional Data Sets: Application to Gene Expression Data, *Machine Learning* 52:119-145, 2003.

[Raychaudhuri and Altman, 2003] Raychaudhuri, S., and Altman, R.B., A Literature-Based Method for Assessing The Functional Coherence of A Gene Group, *Bioinformatics* 19:396-401, 2003.

[Raychaudhuri et al. 2002] Raychaudhuri, S., Schutze, H., and Altman, R.B., Using Text Analysis to Identify Functionally Coherent Gene Groups, *Genome Research*, 1582-1590, 2002.

[Tatusov et al. 1997] Tatusov, R.L., Koonin, E.V., Lipman, D.J., A genomic perspective on protein families. National Center for Biotechnology Information, *Science,* Oct 24;278(5338):631-7, 1997.

[Tatusov et al. 2001] Tatusov, R.L., Natale, D.A., Garkavtsev, I.V., Tatusova, T.A., Shankavaram, U.T., Rao, B.S., Kiryutin, B., Galperin, M.Y., Fedorova, N.D., Koonin, E.V., The COG database: new developments in phylogenetic classification of proteins from complete genomes. National Center for Biotechnology Information, *Nucleic Acids Res,* Jan 1; 29(1):22-28, 2001.

[Vapnik, 1995] Vapnik, V., *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[Wu et al. 2003] Wu, C.H., Huang, H., Yeh, L.L., Barker, W.C., Protein Family Classification and Functional Annotation, *Computational Biology and Chemistry*, 27:37-47, 2003.

Abstract

When biologists make an experiment in revealing protein function, they are confronted by practical difficulties such as the variety of experiments. For this reason, I propose a literature based method for protein function classification. With this method, biologists can predict protein function in advance of the experiment, and also verify its result after the experiment.

The reason they can predict protein function using only literature information is that enormous amount of protein function related researches are already opened to the public in forms of documents. In other words, even though there is not such a document that contains no clear mention of a certain protein function, at least there exist documents written about various experiment results of that protein. In that way, they can predict protein function using only the information of documents.

In this paper, I showed the equality between protein function classification and document classification, and support vector machines were used for documents classification. Support vector machines show good performance with the data having many features like documents, and they find a hyperplane which classifies training data with maximum margin. In that way, they divide training data into two parts.

The performance of the proposed method was tested using protein function data of Saccharomyces cerevisiae and Escherichia coli.

Keywords : Protein Function Classification, Support Vector Machines, Document Classification, Saccharomyces cerevisiae, Escherichia coli
Student Number : 2002-21556