ETRI

2012
Electronics and
Telecommunications
Trends

정보통신 미래원천기술 특집

빅데이터 활용을 위한 기계학습 기술동향

Machine Learning Technology Trends for Big Data Processing

임수종 (S.J. Lim) SW 원천기술연구팀 선임연구원
민옥기 (O.K. Min) SW 원천기술연구팀 팀장

빅데이터 시대를 맞이하여 이를 분석하여 지능형 서비스로 활용할 수 있는 기술로 인공지능 기술이 다시 관심을 받고 있다. 본고에서는 인공지능의 여러 요소 기술 중 기계학습(machine learning) 분야의 빅데이터 처리를 위한 동향을 소개한다. 현재 사용 가능한 병렬처리 기반의 기계학습, 빅데이터를 이용한 기계학습 기반으로 진행되고 있는 프로젝트, 다양한 분야에 쉽게 기계학습을 적용할 수 있는 domain adaptation 기술에 대해서 정리한다.

Ⅰ. 머리말

각종 디지털 기기, 특히 스마트폰 사용이 폭증함과 더불어 만들어지는 데이터의 양도 함께 늘어나고 있다. 이러한 다양하고 많은 종류의 데이터를 '빅데이터'라고 부르며 이를 활용하여 인간생활을 편리하게 하고자 하는데 관심이 쏟아지고 있다.

빅데이터를 활용하기 위한 기술로 인프라 기술, 분석 기술 등이 제시되는데 본고에서는 분석 기술에 유용한 기계학습 연구 동향에 대해서 살펴보려 한다.

기계학습에 바탕을 두고 빅데이터를 충분히 활용할 경우 SF영화에 등장하던 사람과 비슷한 생각을 갖고, 사람과 의사소통이 가능하며 심지어는 사람을 지배하려고 하는 인공지능(artificial intelligence)이 등장할 수도 있다. 인공지능은 그 동안 주목을 받으면서도 실현 가능성에 대해 의문이 존재하였으나, 방대한 양의 데이터가 인공지능의 실현 가능성 및 연구의 신뢰성을 높여주고 있다. 애플의 Siri(Speech Interpretation and Recognition Interface)와 제퍼디 퀴즈쇼에서 사람을 이긴 IBM의 Watson은 최근에 인공지능의 실현 가능성을 보여주는 사례라 할 수 있다.

인공지능을 구성하는 기술은 패턴인식(pattern recognition), 자연어처리(natural language processing), 데이터 마이닝(data mining) 등 여러 가지 관련 기술 분야가 있지만, 기계학습 방법은 다른 기술 분야의 가장 기초가 되는 기술이라 할 수 있다.

기계학습(machine learning)이란, 인간과 같은 학습 능력을 기계를 통해 구현하는 여러 가지 방법들에 대해 연구하는 것을 말하며, 주어진 데이터를 분석하여 분석된 결과에서 학습 가능한 규칙이나 새로운 지식을 자동적으로 추출해 궁극적으로는 기계가 학습하는 효과를 얻도록 한다.

기본적인 수준에 머물러 있던 기계학습과 관련된 기법들은, 기계학습 기법의 잠재력을 최대치로 끌어낼 수 있는 수많은 데이터(빅데이터)의 출현으로 인해 점점 실현 가능성이 높아지고 있다. 그러나, 빅데이터를 처리하기 위해서는 기존의 기계학습 방법은 대규모성(scalability)을 갖고 있지 못하기 때문에, 기존에 병렬처리 기법을 이용한 접근 방법을 많이 사용하였으며, 대표적으로 Hadoop, Google Percolator, 야후에서 개발한 분산 스트림 처리 시스템 S4(Simple Scalable Streaming System)와 같은 예가 있다.

그러나, 이 경우는 기계학습 방법 그 자체를 바꾸기보다는 연산 능력(computing power)과 저장 공간을 병렬적으로 처리하는 것으로 확장에 한계가 존재한다. 이를 극복하기 위해서 기계학습 기법 자체를 개선하여 빅데이터 시대에 대응할 수 있게 하는 움직임이 병렬처리 방법 기반 위에서 빅데이터를 처리할 수 있는 연구가 진행되고 있다.

구글은 대규모 분산 컴퓨팅 인프라를 사용해 자가학습이 가능한 인공 신경망을 만들었으며, 구글X 연구소는 1만 6천 개에 이르는 CPU 코어와 10억 건 이상의 데이터 연결을 처리하는 모델을 도입해, 고양이에 대한 특징을 학습하지 않아도 스스로 고양이를 인지하는 인공 신경망을 개발했다. 이는 사실상 대규모 분산 컴퓨팅 인프라가 사람의 뇌 역할을 할 수 있음을 밝혔으며, 빅데이터의 미래가 인공지능 분야로 연결될 수 있음을 보여줬다.

Ⅱ. 병렬처리 기반 기계학습

기계학습 알고리즘을 툴이나 패키지 형태로 제공하려는 시도는 1999년에 Java 라이브러리 형태로 제공된 뉴질랜드 Waikato 대학의 Weka(Waikato Environment for Knowledge Analysis)를 비롯한 PyBrain 등 많은 툴이 존재하지만, 거의 대부분이 실험실 수준의 데이터 처리를 할 수 있다는 한계를 보였다.

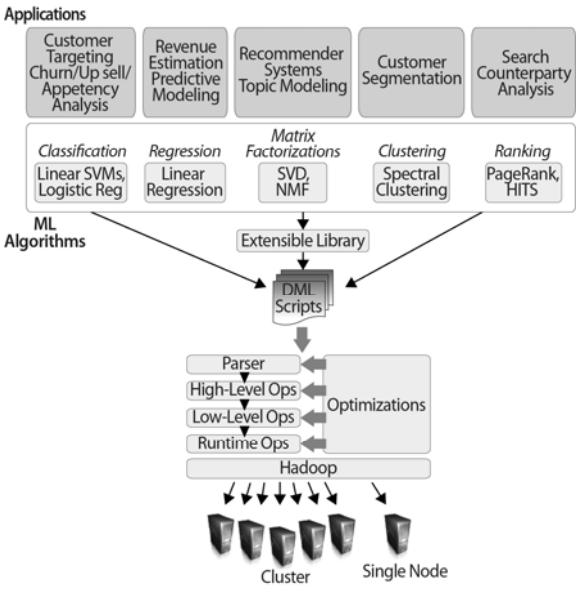
그러나, 빅데이터 처리가 관심을 받으면서 자연스럽

게 대용량의 빅데이터를 처리할 수 있는 기계학습 알고리즘을 지원하는 패키지에 대한 관심이 늘어나고 있고, Hadoop 같은 병렬처리에 기반하여 실제 많은 양의 데이터를 처리할 수 있도록 능력을 갖춘 Skytree 등 다양한 시도가 있는데, 본고에서는 상용제품인 IBM의 SystemML과 오픈 소스 진영에서 개발한 Mahout에 대해서 소개하고자 한다.

1. SystemML

IBM에서 다양한 기계학습 알고리즘을 map-reduce 기반의 분산처리 환경에서 실행하도록 개발한 시스템 패키지로[1], 빅데이터 처리를 위해서 기존 알고리즘을 분산처리 환경에서 구동 가능하도록 구현했다는 점에서 의미가 있다고 하겠다.

통계분석 오픈 소스인 R의 syntax를 차용하여 개발된 DML(Declarative Machine learning Language)을 이용하여 구현이 되도록 하였으며, Java 기반의 API와 프로그래밍 프레임워크를 이용하여 Random decision tree, Stochastic gradient descent for Matrix Factorization과



〈자료〉: IBM Cooperation, 2011.

(그림 1) System ML을 이용한 응용 프로그램 구성[2]

같은 외부 분산 패키지도 이용할 수 있도록 하였다.

DML script 형태로 지정된 기계학습 작업은 컴파일 HOP(High-Level Operator), LOP(Low-Level Operator)를 거쳐서 컴파일되며, 병렬처리를 위해 Map-Reduce 환경에서 실행이 된다. Linear Regression, Descriptive Statistics, Linear SVMs와 같은 기계학습 방법이 준비되어 있으며, IBM에서는 (그림 1)과 같이 SystemML을 사용한 애플리케이션 대해서 소개하고 있다.

2. Mahout

Mahout은 원래는 코끼리를 조련하는 사람이라는 뜻인데, 대용량의 데이터를 필요로 하는 지능형 애플리케이션 개발을 위한 분산/병렬처리가 가능한 기계학습 라이브러리로, ASF(Apache Software Foundation)에서 추진하는 오픈 소스 프로젝트이다[3]. Mahout을 통해 다양한 ML 알고리즘을 라이브러리 형태로 제공하고 이를 Apache Hadoop을 사용하여 클라우드 환경에서 효과적으로 확장하여 기존 기계학습 알고리즘의 한계 중의 하나인 대용량 학습 데이터 처리 시간 등의 문제를 해결하도록 하고자 한다.

Mahout에서 구현된 기계학습이 적용될 작업으로는 추천(recommendation), 군집화(clustering), 분류(categorization), FPM(Frequent Pattern Mining)이 있다.

2012년 6월 현재 0.7 버전이 배포됐고, Java로 개발되어 cross platform을 지원하는 특성을 갖고 있다. 유명한 클라우드 서비스 중 하나인 아마존의 EC2에서 사용 가능하다.

GSOC(Google Summer of Code)를 통해서 기계학습 알고리즘을 구현하여 Mahout에 추가하고 있다. 관련된 오픈 소스로는 기계학습의 전처리 역할을 담당하는 Lucene, 기계학습 알고리즘이 분산처리 환경에서 수행되도록 하는 Hadoop, MapReduce를 효율적으로 활용하도록 하는 Hama가 있다.

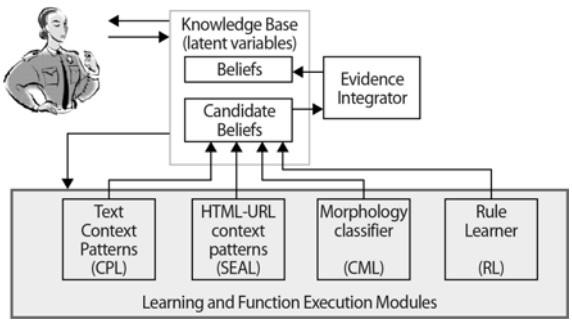
Ⅲ. 빅데이터 기반 기계학습 프로젝트

빅데이터에 기반한 프로젝트는 주로 대용량 색인/검색과 같이 현재 처리하기 힘든 대용량의 빅데이터를 효율적으로 처리하는 것에 대한 것이 많기 때문에, 기계학습 기법에 기반한 프로젝트는 상대적으로 많이 진행되고 있지는 못하고 있다. 관련하여 NELL(Never Ending Language Learner) 프로젝트, Google Prediction API, BigML에 대해 소개한다.

1. NELL Project

CMU의 Tom Mitchell이 주관하는 NELL 프로젝트는 영속적인(never-ending) 기계학습 시스템을 구축하는 것을 목표로 하는데, 이 시스템은 비구조 웹 페이지로부터 구조화된 정보를 추출할 수 있는 능력을 갖추고 있다. 이러한 능력은 시스템 구축 초기에 주어지는 person, sportsTeam, fruit, emotion과 같은 범주(category)를 정의한 ontology와 이에 대한 관계들을 정의한 seed로부터 시작하여 스스로 검증과정을 통해 구조화된 정보를 추출할 수 있는 belief를 자가 학습한다. NELL 시스템의 구조는 (그림 2)와 같다.

이러한 학습을 통해 웹상의 모든 콘텐츠를 반영한 지식베이스(Knowledge Base: KB)를 구축한다[4]. 이 프로젝트는 2010년 1월부터 전 세계 웹을 대상으로 다음 2가지 종류의 작업을 지속하며 지속적으로 확장 중이다.



(그림 2) NELL Architecture[4]

첫 번째는, 수백만의 웹 페이지를 대상으로 다음과 같은 facts를 추출한다.

playsInstrumnet (George_harrison, guitar)

두 번째는 좀 더 정확하고, 좀 더 많은 facts를 추출하기 위한 능력을 향상시키기 위해 끊임없이 학습을 한다. 초기에 주어진 ontology와 relation을 seed로 하고 Lemur project에서 구축한 The ClueWeb09 Dataset[5]를 초기 웹 페이지로 입력 받아 지식베이스를 구축하는데, The ClueWeb09 Dataset는 10개국 언어의 약 10억 개 페이지로 25TB 용량이며, 연결된 웹 페이지(outlinks)가 약 80억 페이지에 이른다. 이러한 페이지를 대상으로 〈표 1〉과 같은 seed belief에 기반하여, facts 추출 방법에서 고유한 특성을 갖는 CPL(Coupled Pattern Learner), CSEAL(Coupled Set Expander for Any Language), CMC(Coupled Morphological Classifier), RL(Rule Learner)과 같은 서브시스템에서 독자적으로 후보 facts를 추출한다. 추출된 후보 facts는 Evidence Integrator를 통해 자동으로 통합되고 최종적으로 beliefs가 되기 위해서 준교사학습(semi-supervised learning) 방법 중 하나인 bootstrap 기법을 이용하여 belief에 추가된다. 이렇게 추가된 beliefs를 이용하여 다시 웹 페이지를 대상으로 후보 facts를 추출하고 beliefs를 확장하는 과정을 반복하면서 자가 학습된 belief를 이용하여 점차 지식베이스의 커버리지를 넓혀간다.

현재 이렇게 구축된 지식베이스는 87% 정도의 정확도로 추정되고 있으며 815개 범주에 대해서 약 130만여 개의 instance로 구성되어 있다.

〈표 1〉 NELL의 Belief[4]

Predicate	Instance	Sources
female	Kate Mara	CPL, CMC
sport	BMX bicycling	CSEAL, CMC
river	Fording River	CPL, CMC
cityInState	(troy, Michigan)	CSEAL
productType	(Acrobat Reader, FILE)	CPL

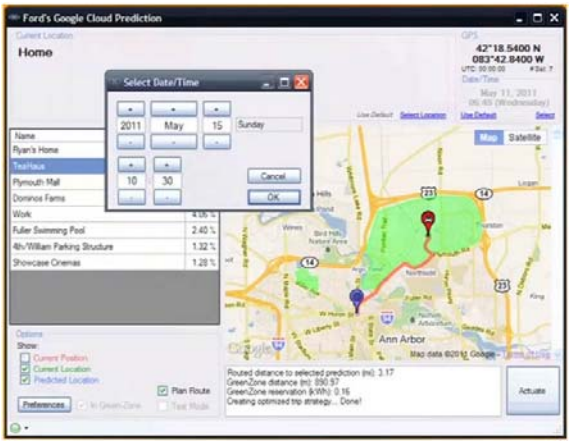
NELL 프로젝트는 준교사학습 방법인 bootstrap 기법을 사용하였으나, 특정 instance에 대해서 1차원으로 판단하지 않고 다차원(multi-view)으로 판단할 수 있는 coupling constraints를 도입하여 사람의 개입을 최소화하였고, 실제로 하루 5분 정도 사람의 개입으로 초기 71%였던 구축 정확도를 87%까지 개선했다는 점에서 장점이 있다. 이러한 장점을 바탕으로 실험실 수준의 데이터 대상이 아닌 실제 빅데이터를 대상으로 기계학습 기법을 도입하여 최초로 실질적인 결과를 내고 있다는 점에서 의의를 찾을 수 있다.

2. Google Prediction API

구글에서 'Machine Learning for your business!'라는 목표로 기계학습 기법에 대해서 잘 모르는 개발자들도 API로 제공된 기계학습 알고리즘을 이용하여 다양한 종류의 애플리케이션에 지능적인 요소를 추가할 수 있는 Google Prediction API를 제공한다[6]. 보유하고 있는 데이터를 이용하여 오프라인으로 학습하여 학습 모델을 구성하고 이를 실시간 서비스에 적용하여 prediction 결과를 실시간으로 얻을 수 있도록 한다. 2011년 일반에게 공개되었고, 2011년에 버전 1.5를 발표하면서 일정 용량 이상 사용할 경우 유료 정책이 적용된다.

Google Prediction API는 다양한 종류의 기계학습 알고리즘을 구현하여 제공한다는 면에서는 앞에서 설명한 Mahout과 유사할 수 있으나, 단순히 기계학습 알고리즘을 제공하는 라이브러리가 아니라 사용자가 데이터만 보유하고 있다면 구글의 cloud storage/computing, 웹 서비스를 위한 애플리케이션 엔진을 사용할 수 있다는 점이 다르다.

2010년에 최초로 실험실 수준에서 제안됐을 때는 입력된 텍스트에 대해서 사용된 언어를 판별해주는 수준이었으나, 현재는 추천, 스팸 필터, sentiment analysis, 메시지 라우팅 등 다양한 응용이 있으며, 궁극적으로 Google Prediction API에서 목표로 하는 것은 (그림 3)

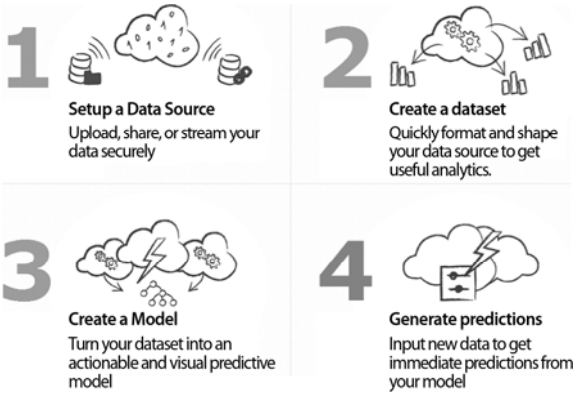


(그림 3) 구글 Prediction API를 이용한 포드 자동차 서비스 개념[7]

과 같은 서비스를 목표로 하고 있다. 포드와 공동으로 개발하고 있는 서비스로 운전습관, 운전환경, 소요 시간, 기름의 소모량과 같은 운전자의 과거 데이터를 기반으로 운전자에게 최적의 경로를 예측하여 제공한다.

3. BigML

BigML은 Google Prediction API와 유사하게 machine learning as a service를 목표로 기계학습에 대한 지식이 없더라도 필요한 데이터를 확보하고 있는 사용자라면 누구나 쉽게 기계학습 기법을 이용하여 지능형 서비스나 애플리케이션을 개발할 수 있도록 하는 것을 목표로 하고 있다[8].



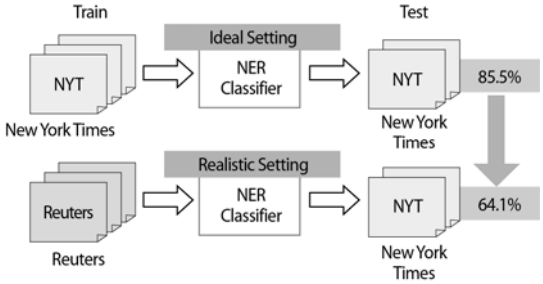
(그림 4) BigML 기반 지능형 애플리케이션 구축 과정[8]

(그림 4)는 BigML을 이용하여 데이터를 보유한 사용자가 지능형 애플리케이션을 구축하는 과정이다.

Ⅳ. Domain Adaptation 기법

기계학습의 경우 정답을 알려주는 교사학습(supervised learning)과 정답을 알려주지 않는 비교사학습(unsupervised learning) 방법으로 크게 나눌 수 있는데, 일반적으로 같은 문제일 경우 교사학습이 비교사학습에 비해서 좀 더 나은 성능을 보인다. 그러나, 모든 경우에 교사학습을 적용할 수 없는데, 가장 큰 문제는 교사학습을 위한 학습 데이터를 구축하는 것이 시간과 비용이 많이 든다는 데 있다. 그리고 이렇게 구축된 학습 데이터도 적용 분야(domain)가 변경될 경우 다시 재구축 비용이 발생한다.

자연언어처리 기술 중 인명, 지명, 장소명 등을 인식하는 개체명 인식(Named Entity Recognition: NER) 문제에 교사학습 방법을 적용하기 위해 뉴스 데이터(NewYork Times, Reuters)을 이용하여 개체명 인식 분류기(NER Classifier)를 교사학습 방법으로 구축했는데, 각각의 분류기를 뉴욕타임즈 데이터에 적용했을 때 (그림 5)에서 보여지듯이 뉴욕타임즈로 학습한 분류기보다는 Reuters로 학습한 분류기가 약 20% 정도 성능 저하를 보여준다. 성능 저하가 없으려면 적용하고자 하는 모든 domain에 대해서 학습 데이터를 구축하고 각각의 분류기를 구축해야 하는데, 이것은 현실적으로 불가능하다. 이런 문제를 도메인 적응 문제(domain adaptation problem)로 정의하고 이를 해결하기 위한 연구방법을 소개하고자 한다.



(그림 5) Domain Adaptation Problem[9]

먼저 가장 간단한 방법으로 (그림 5)와 같이 소스 도메인(source domain)의 데이터만을 이용해서 기계학습에 이용하는 source only 방법과 원하는 타깃 도메인(target domain)의 데이터를 일일이 구축하는 target only 방법이 있으나, 특별한 기술이 필요하지는 않으며, 어떤 방법을 선택할지는 단지 학습 데이터 보유 여부에 달려있다.

아래에서 설명할 방법은 source domain에 대해서는 기계학습 방법을 적용할 만큼의 학습 데이터가 확보가 되어 있고, target domain에 대해서는 소량의 학습 데이터만 확보되어 있는 경우를 가정한다.

1. All and Weighted Model

이 모델은 source domain, target domain에 해당하는 모든 학습 데이터를 사용한다. 이 경우에는 보유하고 있는 모든 데이터를 사용한다는 측면에서는 긍정적이나 source domain 학습 데이터가 target domain 학습 데이터에 비해 너무 큰 경우에는 target domain의 특성이 반영되지 못하고 source domain 특성만 반영되어 실제로는 target domain 학습 데이터를 구축한 효과를 전혀 보지 못한다.

이러한 경우에 해결 방법으로 제시된 것이 Weighted 모델로, source domain 학습 데이터에 대해서 가중치를 줄여주는 것이다. 가장 직관적인 방법은 source domain 학습 데이터가 target domain 학습 데이터의 10배라고 하면, source domain에서 학습된 모델의 가중치를 0.1로 하는 식이다.

2. PRED Model

PRED 모델은 source 모델을 통해 구축된 분류기의

인식 결과를 target 분류기를 위한 학습 모델의 feature로 이용한다. 먼저 source 학습 데이터만으로 분류기를 학습한 후에, 이 분류기를 target data를 대상으로 실행한 결과 데이터를 얻는다. 그 다음 결과 데이터를 추가 feature로 하여 기존의 target data를 이용하여 학습을 하면 PRED 모델을 구축할 수 있다. PRED 모델은 source, target data를 구분하여 학습에 이용하는 장점이 있으나, 모델을 구축하기 위한 과정이 늘어나서 구축 속도가 느려지는 단점도 있다.

3. Linear Interpolation Model

Linear Interpolation(LININT) 모델은 source, target 모델을 각각 독립적으로 구축한 후에, 선형보간법(linear interpolation)을 적용하여 하나의 모델로 통합하는 것으로 수식은 다음과 같다.

LININT Model
= λ * Source Model + (1−λ) * Target Model

일반적으로 보간법의 파라미터 λ(0≤λ≤1)는 target data에 특성에 맞게 조정이 된다.

LININT 모델은 사용자가 파라미터 λ를 이용하여 source, target 데이터의 모델 반영 비율을 조정할 수 있어 데이터 상황에 맞게 유연하게 모델을 구축할 수 있는 장점은 있으나, 이 역시 구축 속도가 느려지는 단점이 있다.

4. Prior Model

Prior 모델[10]은 최초로 domain adaptation 문제 해결을 위해 제안된 것으로 source 모델의 인식 결과를 target 모델의 추가 feature로 채택하는 PRED 모델과 다르게 source 모델의 weights를 target 모델의 학습에서 사용한다. Prior 모델은 maximum entropy 모델에 적용되었으나, Prior 모델의 기본 아이디어는 어떤 통계 모델에도 적용 가능하다. 학습의 과정은 기본적으로는 데이터에서 추출된 feature에 대해 문제 해결을 위한 최적의 가중치 벡터 w를 구하는 것인데, 이것은 $\lambda\|w\|^2$ 형태로 표현할 수 있다. Prior 모델에서는 먼저 학습된 source 모델의 가중치 벡터 w^s를 기준점으로 삼아 target 모델의 가중치 벡터 w를 $\lambda\|w-w^s\|^2$ 방식으로 구한다.

Prior 모델은 target model의 가중치 벡터를 구하기 위해서 상대적으로 소량의 target 데이터를 이용하기보다는, source 데이터에서 학습된 가중치 벡터를 기준으로 하여 target 데이터의 가중치 벡터가 얼마나 이동했는지를 구하기 때문에 상대적으로 적은 양의 학습 데이터로도 좋은 성능을 보일 수 있다.

5. Feature Augmentation model

Feature Augmentation 모델은 학습을 위해 추출하는 feature를 다음과 같이 3가지로 분류하여 각각의 feature를 이용한 모델을 독립적으로 학습하여 구축한다[11].

- General: source, target domain 모두에 사용
- Source-specific: source domain의 특징이 반영되어 source domain에서만 사용
- Target-specific: target domain의 특징이 반영되어 target domain에서만 사용

예를 들어 스마트폰 도메인과 호텔 도메인에 대해서 'horrible'이라는 단어는 부정적인 뜻으로 두 도메인에서 일반적으로 사용하면 되지만, 'small'이라는 단어는 스마트폰 도메인에서는 긍정적인 뜻으로, 호텔 도메인에서는 부정적인 뜻으로 feature로 간주하여 학습하시는 식이다.

Feature Augmentation 모델과 Prior 모델을 가중치 벡터 관점에서 비교하면 다음과 같이 표현할 수 있다.

$$\|w_g\|^2 + \|w_s - w_g\|^2 + \|w_t - w_g\|^2$$

즉, source 도메인에서 학습된 가중치 벡터를 중심으

로 삼았던 Prior 모델과 다르게 Feature Augmentation 모델에서는 일반적(general)인 feature에서 추출된 모델을 중심으로 삼아서 source, target 가중치 벡터를 이동하는 개념이다.

이 모델을 확장하면 n개의 도메인에 대해서도 확장이 가능하며 다음과 같이 표현할 수 있다.

$$\|w_g\|^2 + \|w_1 - w_g\|^2 + \cdots + \|w_n - w_g\|^2$$

Ⅴ. 맺음말

다양한 분야에서 다양한 종류의 데이터가 쏟아져 나오는 빅데이터 시대를 맞아, 이러한 데이터를 효과적으로 이용하는 것에 대한 관심이 증폭되고 있다. 빅데이터를 효과적으로 이용할 수 있는 인공지능 기술의 기본이라 할 수 있는 기계학습 분야가 현재 빅데이터 시대를 맞아 어떤 동향을 보이고 있는지 알아보았다.

기계학습 기법이 효과적으로 동작하기 위해서는 충분한 학습 데이터가 필요하지만, 학습 데이터가 많아질 경우 학습 및 처리 시간이 늦어지는 것을 방지하기 위해서 기계학습 알고리즘을 수정하기보다는 병렬처리 기반으로 처리 속도를 향상시킨 IBM의 SystemML이나 Mahout에 대해 알아봤다. 그리고 NELL, Google Prediction API와 같은 현재 진행되고 있는 기계학습 기반의 프로젝트에 대해서 소개하였다.

이러한 노력들도 빅데이터 시대에 기계학습을 활용하기 위해 중요한 방향이지만, 궁극적으로는 외부의 기술과 접목하려는 노력과 병행하여 domain adaptation 기법의 예와 같이 기계학습 그 자체를 개선하여 좀 더 효율적인 기법으로 향상시키는 것이 빅데이터 시대를 맞는 기계학습 기법의 나아갈 방향이라고 전망한다.

용어해설

학습 데이터(training data) 기계학습에서 원하는 정보를 추출하기 위해서 사용된 데이터의 집합

교사학습(supervised learning) 원하는 결과가 표현된 학습 데이터를 이용한 기계학습 방법. 일반적으로 비교사학습 방법에 비해 성능은 좋으나, 원하는 결과를 데이터에 포함하기 위한 시간과 구축 비용이 증가함.

비교사학습(unsupervised learning) 교사학습과 반대로 원하는 결과가 표현되지 않은 학습 데이터를 이용한 기계학습 방법. 교사학습에 비해 성능은 좋지 않으나, 학습 데이터 구축이 용이하기 때문에 클러스터링과 같은 비교사학습 방법이 적합한 문제에 적용하는 것이 효율적임.

준교사학습(semi-supervised learning) 원하는 결과가 표현된 학습 데이터를 seed로 하여 교사학습을 한 후에 이 결과를 비교사학습 방법을 이용하여 확장하는 방법

약어 정리

KB	Knowledge Base
ASF	Apache Software Foundation
CMC	Coupled Morphological Classifier
CPL	Coupled Pattern Learner
CSEAL	Coupled Set Expander for Any Language
DML	Declarative Machine learning Language
FPM	Frequent Pattern Mining
GSOC	Google Summer of Code
HOP	High-Level Operator
KB	Knowledge Base
LININT	Linear Interpolation
LOP	Low-Level Operator
NELL	Never Ending Language Learner
RL	Rule Learner
Siri	Speech Interpretation and Recognition Interface
Weka	Waikato Environment for Knowledge Analysis

참고문헌

[1] SystemML. http://www.almaden.ibm.com/cs/projects/systemml/

[2] B. Reinwald, “Expressing and Running Big Data Analytics,” *3rd Workshop Large-scale Data Min.*, 2011.

[3] Mahout. http://mahout.apache.org/

[4] A. Carlson et al., “Toward an Architecture for Never-Ending Language Learning,” *Prec. Conf. Assoc. Advancement Artif. Intell.*, 2010.

[5] The ClueWeb09 Dataset. http://lemurproject.org/clueweb09/

[6] Google Prediction API. https://developers.google.com/prediction/

[7] WRIED, "Ford, Google Team Up to Make Smarter Cars," May 10th, 2011. http://www.wired.com/autopia/2011/05/ford-google-prediction-api/

[8] BigML. http://bigml.com

[9] 이창기, "자연어 분석 – 도메인 적응 기술 동향," KCC 자연어처리 및 정보검색 최근 동향 워크샵, 6 .29. 2011.

[10] C. Chelba and A. Acero, "Adaptation of Maximum Entropy Classifier: Little Data Can Help a Lot", Proc. EMNLP, 2004.

[11] A. Saha, P. Rai, and H. Daumé, "Active Supervised Domain Adaptation," *Eur. Conf. Mach. Learning*, 2011.