

공학석사학위논문

기계 학습을 이용한 내용 기반의 음악
장르 분류

Content-Based Classification of Musical
Genre using Machine Learning

2002년 8월

서울대학교 대학원
컴퓨터공학과
박상준

초 록

음악 장르는 음악을 설명하기 위한 범주적 설명 방법이다. 음악 장르는 현대와 같이 네트워크 상에서 수많은 정보들 가운데 특히 음악 정보에 대한 분류 구조를 만들기 위한 자연스런 방법으로 채택되었으며, 음악의 정보 추출에서도 중요한 역할을 수행한다. 장르는 전통적으로 수 작업으로 구분되었지만, 리듬이나 악기 혹은 멤버들과 같은 특징적인 음악의 한 부분에 대해, 장르마다의 특징적인 부분이 있다. 이 사실에 착안하여 본 논문에서는 음악 장르를 빠르게 구분하여 실제 웹에서의 음악 정보 추출에 사용할 수 있는 음악 장르의 구분법을 연구한다.

음악 장르의 구분은 필연적으로 전처리와 feature selection 이 핵심이기 때문에 미디어 음악의 특성에 맞게 feature를 추출하였다. 즉, 기본적인 feature인 템포, 박자, 키(key)와, 좀더 특수한 feature인 멜로디 높낮이, 멜로디 간격, 음표 빈도, 장르에 특화된 박자(리듬)등에 대한 feature selection 과정을 수행한다. 이렇게 만들어진 29개의 feature에 대해, Artificial Neural Network, Decision Tree, AdaBoost를 이용한 Decision Tree, K-nearest Neighbor Model 의 4가지 방법으로 음악 장르를 구분하였다. 일반적으로 특징적인 Feature에 대한 충분한 데이터 셋이 주어지는 경우, 그 동안 연구된 기계 학습 기법을 사용하면 사람이 경험적으로 데이터를 분류하는 것과 유사한 방법으로 컴퓨터를 통한 빠른 학습과 분류가 가능하기 때문이다. 위의 4개의 기계 학습 방법은 다량의 데이터를 분류하기 위한 대표적인 기계 학습 방법들이다.

소프트락, 재즈, 클래식, 팝송, 한국가요의 5개의 장르에서 각각 50곡씩 모두 250개의 미디어 음악에 대해 cross validation으로 분류 실험을 수행한 결과 AdaBoost를 이용한 Decision Tree가 가장 높은 확률로 음악 장르를 구분하였으며, 무작위 분류의 20%를 상회하는 72.85%의 정확도가 나왔다. 이는 음악 분류와 직접적으로 관계되는 feature의 특징을 잘 잡아주는 Decision Tree의 특성에 AdaBoost를 추가하여 학습 효과를 증대시킨 결과라 분석되며, 이를 이용해 향후 사용자의 취향에 맞는 웹상에서의 장르별 음악 제시등, 좀더 능동적인 기계학습의 응용이 가능하리라 생각된다.

주요어 : 음악 장르 분류, 기계 학습, 미디어(MIDI)

학 번 : 2000-21206

목 차

1장 서론	1
2장 관련연구	3
2-1절. 컴퓨터 음악의 역사	3
2-2절. 인간의 컴퓨터 인지과정	7
2-3절. 장르구분	8
2-4절. 미디 데이터	12
3장 분석을 위한 기계학습 기법	23
3-1절. Artificial Neural Network	23
3-2절. Decision Tree Model	26
3-3절. Decision Tree Model with Adaboost	27
3-4절. K-nearest neighbor Classifier	28
4장 실험 및 결과	29
4-1절. 학습 데이터	29
4-2절. 실험 설계 및 결과	29
5장 결론	37

표 목차

[표 1] ID3v1[Id3, 2001] 에서 제안된 장르 구분표	9
[표 2] Winamp에서 제안된 음악 장르의 확장	10
[표 3] 표준 미디어에서 제안한 각 드럼 파트의 대응 숫자 표	19
[표 4] MIDI 곡 Easy Lover에 대한 대한 비트 정보	21
[표 5] Artificial Neural Net을 이용한 실험 결과 (Confusion Matrix)	29
[표 6] Decision Tree을 이용한 실험 결과 (Confusion Matrix)	30
[표 7] AdaBoost를 이용한 Decision Tree의 실험 결과 (Confusion Matrix)	32
[표 8] K-nearest Neighbor Classifier를 이용한 실험 결과 (Confusion Matrix)	33
[표 9] 4개의 실험에 대한 최종 결과 분석표	33

그림 목차

[그림 1] The research framework : a sketch of data processing/analysis steps	6
[그림 2] Happy Birthday 의 악보	14
[그림 3] 0부터 127까지의 음표 빈도를 그대로 나타낸 그래프	17
[그림 4] 옥타브를 무시하고 같은 음들의 빈도를 합쳐서 표시한 그래프	17
[그림 5] 가장 높은 값을 가지는 key을 가장 왼쪽으로 shift한 그래프	18
[그림 6] 락의 기본 리듬	18
[그림 7] 재즈의 기본 리듬	18
[그림 8] 시퀀서에서의 드럼 정보	20
[그림 9] 일반적인 다층 퍼셉트론에서의 신경망 구조	22
[그림 10] k-최근점 알고리즘의 동작 예	27

1 장

서론

컴퓨터가 맨 처음 발명된 이후, 컴퓨터는 실로 수많은 분야에서 없어서는 안될 존재로 사용되고 있다. 특히 인터넷이 활성화되고 그 사용자의 수가 증가하면서 컴퓨터의 응용 범위는 날로 확장되어, 인문·사회 등 컴퓨터와 특별히 관련 없을 것 같은 분야에도 날로 사용이 증가되고 있다.

음악도 그 대표적인 분야 중 하나로 들 수 있다. 이미 대중 음악에서 컴퓨터 음악의 입지는 이미 절대적이며, 테크노·레이브 등 컴퓨터 음악만을 사용한 음악 장르까지 나오고 있는 추세이다. 컴퓨터를 음악에 도입하려는 시도는 처음에는 굉장히 이상한 것으로 받아들여졌다. 하지만 컴퓨터 음악으로 쉽게 연주를 편집 녹음할 수 있게 되고, 인터넷의 도입으로 개인의 음악이 빠르게 전파될 수 있음을 알게 되자, 수많은 사람들이 컴퓨터를 이용하여 음악을 만들고 또 들으며 자신의 음악생활을 향유하고 있다. 사람에게 있어서 음악은 본능이다.

우리들 대부분은 작곡이건, 연주가건, 청취이건 간에, 음악 활동에 참여하고 있다. 우리가 음악 활동에 참여하는 이유는 음악이 우리 마음속에 깊고 의미 있는 감정을 불러일으키기 때문이다. [Sloboda, 1983]

위의 인용을 보더라도 사람은 음악을 추구하며, 거대화 된 웹에 산재하는 웹 음악 정보는 적당한 탐색과 분류 작업을 통해 개인이 필요한 고급 정보로 탈바꿈 할 수 있다. 따라서 다수의 웹 정보 안에서 자신이 원하는 음악을 빠르게 찾거나 원하는 장르를 분류해주는 연구는 실용성이 아주 크다고 할 수 있다. 미디어 데이터나 음악을 자동적으로 찾는 연구는 많이 진행되어 왔으나 [Ghias et al, 1995] 음악 장르를 구분하는 연구는 심도 있게 진행되지 않았다. 음악은 장르로 구분되므로, 장르를 분류해주는 방법을 제시하는 연구는 방대한 데이터에서 원하는 음악을 찾는 좀더 효율적인 방법이라고 할 수 있다. 장르를 구분해 놓으면, 음악이 필요한 효과음향 부분이나 음악 치료 등 전문적인 분야에서도 데이터를 좀 더 빠르게 유용할 수 있으며, 전통적으로 데이터베이스가 불가능한 음악을 인류의 지식체계 범주 속에 넣기 용이해진다.

본 논문에서는 컴퓨터음악으로 많이 활용되는 MIDI 데이터에서 기계 학습 개념을 이용하여 자동적으로 장르 구분을 하는 방법을 제안한다. 연구는 실험에 필요한 미디 데이터를 인터넷에서 임의로 구하여 실험하는 제한적 실험으로, 첫 번째로 미디 데이터에서 장르 구분에 중요한 정보를 어떤 방법으로 추출할 것인지에 대한 분석을 해 보고, 두 번째로 추출된 feature set을 어떤 기계 학습적인 classification 방법으로 장르 구분을 할 것인지를 고찰해본다.

논문 구성은 다음과 같다. 2장에서는 컴퓨터 음악의 역사와, 인간의 음악 인지 과정을 알아보고, 이와는 별도로 가장 분류기준에 부합되는 장르 구분에 대한 연구를 살펴본 뒤, 마지막으로 직접 분석의 대상이 되는 미디 데이터의 구조와 feature 중요도에 따른 feature 벡터 생성법에 대해 설명한다. 3장에서는 주어진 데이터를 분석할 주요 방법인 Neural Net Classifier, Decision Tree, AdaBoost, K-nearest Neighbor Classifier 에 대해 알아보며, 4장에서는 주어진 데이터를 통하여 학습을 수행한 후 결과를 최종적으로 확인한다. 마지막 5장에서는 실험의 의미를 고찰해본다.

2장

관련연구

2-1절. 컴퓨터 음악의 역사

처음에 컴퓨터로 음악을 다룰 수 있게 됨을 알았을 때 사람들은 이를 대수롭지 않게 여겼다. 태초이후 인간이 음악과 함께 한 시간의 무게에 비하면 컴퓨터가 만들어내는 다양한 사운드와 기법은 너무나 가볍고 즉흥적이며 따라서 버려야 할 것으로 여겨졌기 때문이다. 따라서 초기 컴퓨터 음악은 대학 내에서 실험적으로나 쓰였지만, 재밌게도 이를 음악의 안쪽에 도입한 그룹은 모던 락의 거장 Queen 이었다. Queen 은 그들이 만드는 우아하고 풍부한 사운드를 전자 오케스트라 사운드로 만들어냈고, 이런 음악이 새로운 음악으로 표현될 수 있음을 알자, 여러 그룹들이 점차 다양한 방법을 통해 컴퓨터 음악을 발전시키기 시작했다. 사운드뿐만 아니라, 편집을 컴퓨터로 하자는 시도도 그 후에 자연스럽게 연구되어 갔는데 이에 MIDI 라는 규약이 새롭게 생겨나게 되었다. 아래는 미디 기초 개론[김태희, 1999]에서의 미디의 정의이다.

MIDI란-Musical Instrument Digital Interface 의 약어이다. 해석해보면, 'Musical Instrument'는 음악 기기(기계로써의 악기)로 해석이 되고, 'Digital Interface'는 디지털 적 연결 장치를 뜻하는 것을 알 수 있다. 즉 미디란 디지털 적 전자 악기나 기기 들을 서로 연결하기 위해 만들어진 세계 공통의 규약이란 뜻이다.

미디는 1983년 이후 제정된 음악적 규약을 뜻하는 용어이다. 이 규약이 제정되면서 모든 디지털 방식의 악기와 기기 들은 이 규약에 맞추어 나오게 되었고, 미디에 관련된 장비를 사용하는 사람들은 세계 어디에서 사용하던 간에 이 규약에 따라 사용하게 되어 있다.

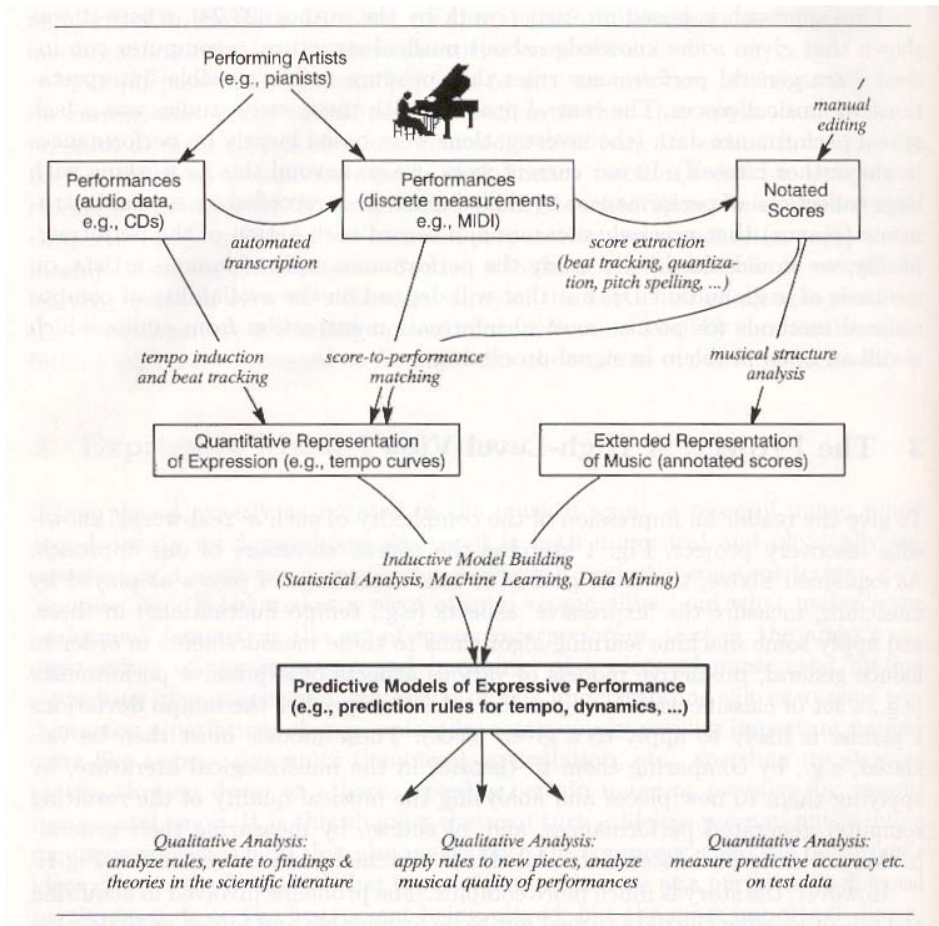
예를 들어, 한국사람인 A란 사람이 B라는 사람과 정보를 주고받으려면 대화가 필요하다.'사랑, 믿음'과 같은 단어도 필요하고, '사랑합니다, 믿습니다' 와 같은 문법도 필요하다. A란 사람과 B라는 사람이 단어, 문법 등의 약속을 할 때 비로소 대화가 통하고 정보를 주고받을 수 있게 되는 것이다. 이것과 동일한 약속이 바로 미디이다. 즉, 디지털 악기와 기기 간의 연주 정보를 교환할 수 있도록 만든 하나의 약속이다. 그래서 전세계 모든 악기 제조회사에서도 '미디'란 규약을 만들어, 하나의 약속으로서 디지털 악기와 기기 들이 대화할 수 있게 만든 것이다. 미디 규약에는 여러 가지 규격이 나와 있다. 신디사이저 외의

악기나 기기 들의 내부와 외부의 규격, 미디 포트 규격, 시퀀서와 시퀀서 프로그램에 대한 규격, 그리고 미디 메시지와 패치(악기음색)등이 바로 그것들이다. 이 규격은 세월이 흐르면서 계속적으로 바뀌고, 새로 만들어지고 있으며, 이에 따라 미디 규약 역시 계속 변화하면서 조금씩 성장하고 있다. 세월이 지날수록 미디의 발전하는 속도는 더욱 빨라지고 있다. 미디는 미완성된 것으로, 아직도 발전하고 있는 단계인 것이다.

처음 초기 8비트 컴퓨터 (APPLE등) 시절에는 오직 컴퓨터 내부의 조그만 스피커로 단음 연주만이 가능했다. 컴퓨터 게임을 위주로 만들어졌던 MSX 컴퓨터는 게임에 사용되던 음원을 그대로 사용해 시뮬레이트된 3중 화음을 내는데 그쳤지만, 그 후 16비트 컴퓨터에서 사용 가능한 야마하사의 YM-3812 칩을 이용한 Adlib 카드가 등장했고, 간이 작곡 프로그램을 통해 ROL 파일, IMS 파일 연주가 가능해졌다. 개인용 컴퓨터 음악의 혁명을 만든 것은 크리에이티브 (<http://www.creative.com>)사의 사운드 블레스터 카드이다. 이 카드가 나오기 전까지만 하더라도 음악을 전문적으로 구현하는 컴퓨터가 아닌, 일반 컴퓨터를 통한 녹음과 음원 재생은 거의 불가능한 수준이었다. 하지만, 이 카드의 기술을 이용하여 아날로그 데이터를 wave 형식 등의 디지털 데이터로 바꾸어 이를 바로 연주하는 연주가 가능해지게 되었고, 이 카드가 널리 퍼지면서 점차 컴퓨터를 사운드 기기로 응용하는 것에 대한 인식이 생기기 시작했다. 부분적으로 이런 wave기능을 이용하여 고가의 컴퓨터 장비에서나 가능한 sampling을 흉내낸 S3M 파일 형식 등으로 개인적으로 작곡을 하는 기술도 나오고, 전용 MIDI카드와 컴퓨터 사운드 카드의 융합으로 국내에서 개발된 옥소리 카드 등도 출시되었지만, 결정적으로 Mpeg 표준기술 중 하나로 컴퓨터에서 CD의 음원을 10분의 1 정도의 용량으로 줄여 연주가 가능한 MP3 압축 방식이 개발되자, 작곡된 각종 음악들은 mp3로 압축되어 웹을 통해 쉽게 전파되었고, 점점 컴퓨터는 음악에 없어서는 안되는 도구가 되어갔다. 후에 사운드 블레스터의 사운드 폰트나 야마하에서 자신의 미디 음원을 소프트웨어 적으로 사용 가능하게 한 야마하 미디 에뮬레이션 등 현재 놀라운 소프트 컴퓨팅 능력에 힘입어 개인의 음악 데이터는 점점 다양해지고 고급화되고 있다.

인류와 함께한 음악의 역사만큼, 음악과 인공지능을 연결하려는 연구는 활발히 이루어졌다. 지금까지의 연구를 정리해보면, 우선 대부분의 음악 논문의 참고문헌에 참조되는 [Ghias et al, 1995] 의 허밍을 통한 음악 정보의 추출에 대한 연구가 있다. 이 논문은 처음으로 아날로그 데이터를 통해 방대한 디지털 데이터에

대한 정보를 추출하려는 시도를 한 연구로 큰 의미를 지니고 있다. pitch 정보를 추출하기 위해 Autocorrelation 과 Maximum Likelihood 그리고 Cepstrum Analysis를 사용한 것이 특징이다. 그 다음으로 음악정보에 대한 인공지능 관련 논문은 크게 데이터 추출과 분류로 나뉘어 연구되는데, [Lemström, 2000] 에서는 텍스트 데이터에서 많이 사용되는 string matching 기법을 음악의 정보 추출에 사용하는 방법을 제시하였다. 데이터 분류를 위해서는 [Cook and Tzanetakis, 2002]에서처럼 아날로그 데이터를 Fast Fourier Transform 알고리즘을 사용하여 가공한 뒤 분류하는 방법이 있다. 정보 추출이나 분류를 위한 음악의 feature를 체계적으로 정리한 논문으로는 [Lap and Kao, 1999] 가 있으며, [Pollastri and Simoncelli, 2001] 에서는 이미 알려진 작곡가의 작곡 패턴을 학습하여 음악 분류가 가능한지를 Hidden Markov Model를 사용하여 실험한 결과, 비교적 좋은 실험 데이터를 추출하였다. 인공지능과 인간을 직접 연결하려는 시도로는, [Widmer, 2001]에서 클래식 곡을 세 명의 피아니스트가 연주한 뒤, 각 피아니스트의 미묘한 연주법 차이를 기계 학습으로 분석하였다. 이에 대한 분석법은 [그림1] 과 같다. 마지막, 재미있는 논문으로 Carnegie Mellon University의 Belinda Thom은 음악의 Interactive Improvisation을 컴퓨터의 기계 학습을 통해 시뮬레이션 하는 연구를 수행한 뒤, 실제 Autonomous Agent 2000 conference 에서 컴퓨터가 연주자가 연주한 음악을 토대로 자신의 연주를 되받아 연주하는 모습을 들려주어, 음악과 인공지능과의 연결에 대한 새로운 모습을 제시하였다. [Thom, 2000]



[그림 1] The research framework : a sketch of data processing/analysis steps

정리하면 음악과 인공지능의 관계는 인간의 행동을 컴퓨터와 연결하려는 시도 이래로 가장 활발히 연구되었던 분야 중에 하나이며, 지금도 여러 분야에서 새로운 방법으로 음악을 인공지능과 연결하고, 방대한 음악 정보를 인간이 편한 방법으로 빠르게 찾게 하기 위해 각종 실험을 진행중이라고 말할 수 있다.

2-2절. 인간의 컴퓨터 인지과정

2-2절의 대부분의 내용은 The Musical Mind 1장 ‘인지 기술로서의 음악’에서 [Sloboda, 1983] 발췌하였다.

물리학이라는 차가운 눈으로 본다면, 음악적 사건은 다양한 음고, 시가 그리고 그 이외의 측정 가능한 성질들을 가진 소리의 집합일 뿐이다. 하지만, 인간은 이 소리들에 의미를 부여한다. 그 소리들은 단순한 소리 이상의 어떤 것, 다시 말해 우리들을 울거나 웃게, 좋아하거나 싫어하게, 감동 받게 하거나 무관심하게 하는 어떤 것의 상징이 된다. 감정적 반응에 대하여 지적해야 할 첫 번째 인지적 요소는 음악에 대한 우리 대부분의 반응이 습득된 것이라는 점이다. 물론 웅장하고 빠른 음악은 분위기를 고양시키는 반면, 부드럽고 조용한 음악은 분위기를 가라앉힌다. 그러나 이러한 원초적인 경향들은 음악에 대해 인간이 가지는 정교하고 다차원적인 본성을 설명하지 못할 뿐만 아니라 반응에 있어서의 많은 중요한 문화적인 차이들도 설명하지 못한다.

감정적 반응에 있어서 두 번째로 지적해야 할 인지적 요소는 감정적 반응이 단순히 ‘조건화’ 라는 말로 설명되지 않는다는 사실이다. 음악의 형식과 내용은 그 음악이 획득한 감정적 특성과는 상관이 없다고 한다. 다만 영화음악처럼 음악이 들려졌던 상황적 맥락만이 중요하다는 뜻이다. 음악으로 감정적 반응이 오려면 음악을 인지하는 과정이 선행되어야 하며, 이런 과정은 유머를 이해하고 웃게 되는 과정과 비슷하다. 유머를 듣고도 웃지 않을 수 있듯이, 듣고 있는 음악을 이해하고도 감동 받지 않을 수도 있다. 즉 인지적 단계가 추상적인 또는 상징적인 음악의 내적 재현의 표상을 형성한다는 뜻이다.

사람들이 음악을 자신에게 떠올리는 즉, 재현하는 방식이 그들이 음악을 얼마나 잘 기억할 수 있으며 연주할 수 있는가를 결정해 준다. 작곡과 즉흥 연주는 그러한 재현의 생성을 요구하며, 지각은 청자로 하여금 재현을 구성하지 않을 수 없게 만든다. 이런 과정은 직접 관찰되지 않기 때문에 음악을 듣고 기억하며, 연주하고, 작곡하고, 그것에 반응하는 방법을 관찰함으로써 이런 과정을 추론해야 한다.

2-3절. 장르구분

자동적 장르 구분에서 구분 방법만큼 중요한 연구분야는 장르 자체를 어떻게 나눌 수 있는가 하는 점이다. [Pachet and Cazaly, 2000] 에서는 음악관련 산업분야에서의 장르구분법과, 인터넷 Mp3 사이트에서의 장르 구분법을 따랐고, [Cook and Tzanetakis, 2002] 에서는 장르의 하부 구분을 두어서 좀더 정확한 장르 구분이 가능하도록 했다. 장르 구분에서 가장 중요한 요소는 장르의 독립성과 동시성이다. hierarchy를 포함시켜서 새로운 분야의 음악을 어떤 장르 안으로 포함시켜보는 것도 좋겠지만, 장르구분이 너무 복잡해지면 데이터의 분류 기준을 잃어버리는 문제를 가지고 있다.

0. Blues	27. Trip-Hop	54. Eurodance
1. Classic Rock	28. Vocal	55. Dream
2. Country	29. Jazz+Funk	56. Southern Rock
3. Dance	30. Fusion	57. Comedy
4. Disco	31. Trance	58. Cult
5. Funk	32. Classical	59. Gangsta
6. Grunge	33. Instrumental	60. Top 40
7. Hip-Hop	34. Acid	61. Christian Rap
8. Jazz	35. House	62. Pop/Funk
9. Metal	36. Game	63. Jungle
10. New Age	37. Sound Clip	64. Native American
11. Oldies	38. Gospel	65. Cabaret
12. Other	39. Noise	66. New Wave
13. Pop	40. AlternRock	67. Psychadelic
14. R&B	41. Bass	68. Rave
15. Rap	42. Soul	69. Showtunes
16. Reggae	43. Punk	70. Trailer
17. Rock	44. Space	71. Lo-Fi
18. Techno	45. Meditative	72. Tribal
19. Industrial	46. Instrumental Pop	73. Acid Punk
20. Alternative	47. Instrumental Rock	74. Acid Jazz
21. Ska	48. Ethnic	75. Polka
22. Death Metal	49. Gothic	76. Retro
23. Pranks	50. Darkwave	77. Musical
24. Soundtrack	51. Techno-Industrial	78. Rock & Roll
25. Euro-Techno	52. Electronic	79. Hard Rock
26. Ambient	53. Pop-Folk	

[표 1] ID3v1[Id3, 2001] 에서 제안된 장르 구분표

위의 표는 mp3 음악의 공인 기관인 ID3의 mp3태그 v1버전에서 사용되는 장르 구분 표이다. 이밖에 Winamp 라는 mp3 재생 컴퓨터 프로그램에서 제안된 장르의 확장은 125개의 장르까지 포함하며, 구성은 아래와 같다.

80. Folk	96. Big Band	
81. Folk-Rock	97. Chorus	112.Club
82. National Folk	98. Easy Listening	113.Tango
83. Swing	99. Acoustic	114.Samba
84. Fast Fusion	100.Humour	115.Folklore
85. Bebob	101.Speech	116.Ballad
86. Latin	102.Chanson	117.Power Ballad
87. Revival	103.Opera	118.Rhythmic Soul
88. Celtic	104.Chamber Music	119.Freestyle
89. Bluegrass	105.Sonata	120.Duet
90. Avantgarde	106.Symphony	121.Punk Rock
91. Gothic Rock	107.Booty Bass	122.Drum Solo
92. Progressive Rock	108.Primus	123.A capella
93. Psychedelic Rock	109.Porn Groove	124.Euro-House
94. Symphonic Rock	110.Satire	125.Dance Hall
95. Slow Rock	111.Slow Jam	

[표 2] Winamp에서 제안된 음악 장르의 확장

위와 같이 인간이 구분해내는 장르는 엄청나게 다양하지만, 이는 어느 특정한 기준을 가지고 있는 영역이 아닌 미묘하고 주관적인 경우가 많다. 따라서 장르 구분을 너무 복잡하게 하는 것은 실험에 안 좋은 영향을 미칠 수 있다. 이에 본 논문에서는 사람이 인지하는데 크게 무리가 없는 장르를 구분해냈는데 표현하면, 소프트 락, 재즈, 클래식, 팝송, 한국가요의 5개의 장르이다.

1. 소프트 락

소프트 락은 미국의 락의 한 분야로서, 엄밀히 말하면 아트락(Art Rock) 과 대비되는 락의 한 종류이지만, 실험에서는 팝보다는 좀 더 비트가 강한, 듣기 편한 락 위주의 곡을 소프트락(Soft Rock) 이라 분류하였다.

2. 재즈

재즈는 미국의 뉴 올리언스 지방에서 흑인 영가와 영국의 클래식이 융합되어 만들어진 장르이다. 재즈는 전형적으로 2,4박의 리듬에 악센트가 있고 스윙감을 느낄 수 있는 비트를 쓴다. [Collier, 1991] 실험에서는 1980년대 이후의 팝이 가미되어 새롭게 정의된 퓨전 재즈적인 장르가 아닌, 1920년대 빅 밴드 시절에서부터 보사노바 까지의 순수한 재즈의 계보를 잇고 있는 곡들을 위주로 분류하였다.

3. 클래식

클래식은 가장 뚜렷이 구분되는 장르로, 바이올린 혹은 피아노 등의 악기를 사용하며, 모차르트, 베토벤, 바하 등 대중적인 클래식 작곡가의 음악을 이 장르로 분류하였다. 이밖에 유명한 민요등, 재즈와는 달리 즉흥연주(improvisation)가 섞이지 않은 악보위주의 연주곡들도 클래식의 범주에 포함시켰다.

4. 팝송

팝송은 말 그대로 우리 귀에 친숙한 비틀즈, 마이클 잭슨 등의 팝 음악 장르이다. 단순한 비트와 귀에 익은 멜로디를 사용하기 때문에 어느 한 장르로 분류될 수 있을 것이라 생각되어 실험의 장르에 포함시켰다.

5. 한국 가요

한국 가요는 위의 장르에 포함되지 않은 우리 나라 곡들을 모두 포함시켜 보았다. 엄밀히 하자면, 한국 가요는 위의 음악들 중 팝쪽에 가깝고, 트로트는 아예 한 장르로 구분을 시켜놓아야겠지만, 한국적인 무언가를 실험을 통해 분류할 수 있지 않을까 하여 실험 데이터 안에 포함시켜보았다.

이렇게 9개의 장르로 실험 데이터를 구성하였고, 실험 데이터는 모두 국내 4대 통신사중 하나누리, 하이텔의 공개자료실과 세계의 각종 미디 관련 인터넷 사이트에서 구하였다. 미디 데이터 자체만으로는 Feature Selection 이 불가능하므로 실험을 위해서 이 데이터들을 heuristic 하게 가공해주어야 한다. 4절에서 미디 데이터의 구성과 데이터 가공 작업에 대해 서술한다.

2-4절. 미디 데이터

MIDI (Musical Instrument Digital Interface) 포맷은 신디사이저나 시퀀서처럼 음악 하드웨어나 소프트웨어를 컨트롤하는 명령 순서의 포맷이다. 즉, 이 포맷 자체가 사운드를 나타내는 것은 아니지만, 특정한 악기들에게 특정한 시간에 소리를 내게 한다. 예를 들면 악기 #1 (Acoustic Grand Piano) 을 선택하여 #60 의 노트를 (C5) #127의 속도로 연주하게 한다는 정보를 담을 수 있다. 따라서 미디 포맷으로 음성정보는 담을 수 없지만, MIDI 파일로 저장된 음악에 대해서는 쉽게 악기를 바꾸거나 음표를 조정할 수 있다. [Akoff]

미디 데이터는 단순히 바이너리로 되어있는 악보라고 볼 수 있다. 하지만, 음원의 선택과 여러 효과를 삽입하기 위해 내부 바이너리는 MIDI Manufacturers Association에서 제정된 규약에 맞게 구성 되어있다.

잠깐 내부를 살펴보자. 아래 표는 바이너리로 저장되어 있는 기본적인 미디 형식을 MIDI File DisAssembler [Jeff] 라는 프로그램을 사용하여 텍스트 형태로 변환한 것이다.

```
Track #1 *****
Time      Event
1: 1: 0 |Track Name | len=0      |
        |Program      | chan= 1   | pgm #- 2 Bright Piano
        |On Note       | chan= 1   | pitch=B 2      | vol=100
3: 60 | (Off) Note | chan= 1   | pitch=b 2
        |On Note       | chan= 1   | pitch=E 3      | vol=100
4: 60 | (Off) Note | chan= 1   | pitch=e 3
        |On Note       | chan= 1   | pitch=D#3      | vol=100
2: 1: 0 | (Off) Note | chan= 1   | pitch=d#3
        |On Note       | chan= 1   | pitch=B 2      | vol=100
3: 60 | (Off) Note | chan= 1   | pitch=b 2
        |On Note       | chan= 1   | pitch=A 2      | vol=100
        |On Note       | chan= 1   | pitch=F#3      | vol=100
4: 60 | (Off) Note | chan= 1   | pitch=f#3
        | (Off) Note | chan= 1   | pitch=a 2
        |On Note       | chan= 1   | pitch=E 3      | vol=100
3: 1: 0 | (Off) Note | chan= 1   | pitch=e 3
        |On Note       | chan= 1   | pitch=A 2      | vol=100
3: 60 | (Off) Note | chan= 1   | pitch=a 2
        |On Note       | chan= 1   | pitch=A 2      | vol=100
4: 0 | (Off) Note | chan= 1   | pitch=a 2
        |On Note       | chan= 1   | pitch=A 2      | vol=100
        |On Note       | chan= 1   | pitch=E 3      | vol=100
```



```

60 | (Off) Note | chan= 1 | pitch=e 3
    | (Off) Note | chan= 1 | pitch=a 2
    | On Note   | chan= 1 | pitch=A 2 | vol=100
    | On Note   | chan= 1 | pitch=D#3 | vol=100
4: 1: 0 | (Off) Note | chan= 1 | pitch=d#3
    | (Off) Note | chan= 1 | pitch=a 2
    | On Note   | chan= 1 | pitch=E 2 | vol=100
    | On Note   | chan= 1 | pitch=G#2 | vol=100
5: 1: 0 | (Off) Note | chan= 1 | pitch=g#2
    | (Off) Note | chan= 1 | pitch=e 2
    | End of track|

```

위의 결과와 같이 MIDI 데이터의 내부는 가장 먼저 노트가 찍히는 시간 단위가 표현되며 내부적으로는 variable length quantity라는 기법을 사용하여 필요한 비트의 수를 효율적으로 줄인다. 그 다음으로 노트의 시작과 끝을 알리는 데이터가 들어가며, 이 데이터가 저장되는 channel, 마지막으로 음의 높낮이를 결정하는 pitch 가 저장되며, 부가적으로 volume 값이 저장된다. [Sonicspot]

이제 실험에 쓰일 수 있게 미디 데이터를 feature 의 set으로 바꾸어주는 작업이 필요하다. 우선 어떤 feature set을 포함시킬 것인지를 생각해보자.

1. Tempo

음악의 구성 중 가장 기본적인 수치로, 1분당 비트 수를 의미한다. 즉 음악의 빠르기를 의미하며, 기본적으로 어떤 정보도 없는 경우 템포는 180이다. 클래식 의 경우 일반적으로 템포가 느리고, 테크노의 경우 템포가 빠르므로 장르를 나누는 중요한 수치라 할 수 있다.

2. 박자

보통 음악에서 말하는 몇 분의 몇 박자에 해당하는 데이터이다. 내부적으로는 Beats per Measure / Beat Value로 나뉘며, 이 정보는 나중에 리듬을 분석할 때 중요한 자료로 쓰인다. Beats per Measure는 Beat Value 가 한 마디에서 몇 번 울리는가 하는 것이고, Beat Value 는 2의 자승의 숫자를 갖는 음표의 기본 단위이다. 기본적으로 어떤 정보도 없는 경우, 4/4를 기본 수치로 갖는다.

3. 키 (Key)

음악의 조성을 나타내는 수치이다. 하지만, 이는 컴퓨터 내부에서는 별로 중요하지 않는데, 이는 음의 구성 원리 때문이다. 즉, 음과 음 사이가 단일한 주파수로 올라가는 것이 연산에 더 편리하기 때문에 기본적으로 미디 데이터는 0에서 255까지의 수치에 사운드 모듈이 표현해낼 수 있는 주파수의 범위를 쪼개어 놓는다. 이는 아래 멜로디 높낮이 부분에서 더 자세히 설명할 것이다. 데이터 상으로 중요하지는 않지만, 장르별로 선호되는 키가 존재할 수 있지 않을까 하여 실험의 feature 안에 포함시켜 보았다.

4. 멜로디 높낮이

모든 미디 곡의 멜로디를 순차적으로 읽어 그 특성을 파악하는 것도 좋은 분석의 방법이겠지만, 음악에서의 멜로디는 화성에 의존적인 부분이다. 즉, 화성을 정확하게 분석해서 멜로디의 추이를 따라가야 하지만, 화성자체를 분석하는 것은 논문의 의도와는 또 다른 방대한 작업이므로 본 실험에서는 멜로디의 변화의 추이를 분석하여 수치적으로 나타내는 것으로 멜로디에서 데이터를 추출하였다. 이는 음악 패턴을 분석하기 위한 일반적인 수치 분석법이며 [Yip and Kao, 2000] 간단하게 서술하면 아래와 같다.



[그림 2] Happy Birthday 의 악보

(a). Interval Sequence, (b). Profile, (c). Note Duration Ratio Sequence

위의 [그림 2]에서 (a) 부분의 Interval Sequence 는 각각의 음표 사이에서의 음표의 높낮이의 변화를 정수로 나타낸 것이다. 여기서의 음계의 개념이 필요한데, 음계라는 것은 음들이 연속적으로 높아지거나 낮아지는 순서를 수치로 정렬한 것을 뜻한다. 대부분의 작곡은 음계를 기반으로 이루어지는데, 음의 구성 개수에 따라 5음 (pentatonic), 12음 (chromatic), 7음 (major or minor diatonic) 등

이 일반적으로 쓰인다. 서양 음악에서는 한 옥타브를 반음계 (semitone) 라고 부르는 12개의 단계로 나눈다.

음계에서 음들 사이의 높이의 값을 어떻게 책정하느냐에 따라 순정율(Just scale), 피타고리안 음계 (Pythagorean scale), 평균율 (equal temperament) 의 세 가지가 있는데 [Rossing, 1989] 이 중 컴퓨터 음악에서는 평균율이 일반적으로 쓰인다. 평균율에서는 C에서 1옥타브 높은 C로 가려면 +12를 하고, C에서 G로 가기 위해서는 +7을 한다. F에서 E로 가면 -1로 가는 식의 계산인 것으로, 위의 [그림 2]에서 (a)에 적용된 정수가 이 간격차이를 수치화 시킨 숫자들이다. 이렇게 계산을 하면, 보통 음악에서 쓰이는 key를 신경 쓰지 않아도 되기 때문에 MIDI 이외의 컴퓨터 음악에서도 널리 통용된다. 그리고, key 자체를 바꾸는 것도 간단해지기 때문에 역시 컴퓨터 음악에서의 key 조정에 쓰이며 재미있는 예이지만, 노래방에서 키 조정이 간단하게 이루어지는 이유도 컴퓨터 음악이 이런 평균율을 사용하기 때문이다.

두 번째로 (b) 부분은 각각의 음표가 이전의 음표보다 높거나, 낮거나 아니면 음정변화 없이 같은 세 가지 범주 안에 들어간다는 사실에 착안하여 이 값을 수치화 시킨 것이다. 음정을 수치화 시킬 수 있는 간단한 방법이므로 이전의 논문에서 많이 사용되었다. [Dowling, 1978] [The, 1981] 인지 과학적으로 보면, 이런 방식은 사람이 음악을 머릿속에 기억하는 방법과 많이 연결되어있다고 한다. [Dowling, 1978] 본 논문에서는 멜로디 음정 변화에 대한 심도 여부만을 판단 할 수 있게, (a)에서처럼 수치화 된 높낮이 간격을 정규화 하여 그 값들의 표준편차를 계산하였다. 즉, 아래와 같은 수치를 사용하였다.

$$\text{멜로디높낮이} = \sqrt{\sum_{i=1}^{N-1} \frac{(m_{i+1} - m_i)^2}{N}}$$

m_i 는 전체 멜로디에서 i 번째의 멜로디 pitch 수치이고, N은 멜로디의 총 개수이다.

5. 멜로디 간격

음이 얼마만큼의 시간동안 소리를 내는 가도 기본적인 feature중에 하나이다. 수치화 시키는 여러 가지 방법이 있는데, 위의 [그림 2]에서 (c) 부분의 수치처

럼 각 음 사이의 음 길이의 비율을 수치화 하는 것이 가장 일반적이다. 보통 대량의 데이터에서 원하는 음악을 추출하는 Musical Information Retrieval 에서는 sequential 한 음표의 흐름에 대한 정보가 중요하지만, 본 논문에서는 음악의 장르를 다루는 것이므로 전체 데이터에서 주 멜로디의 간격의 변화 추이를 수치화 시킨 값이 의미 있다고 판단되어, 역시 아래와 같은 수식으로 멜로디 간격을 수치화 하였다.

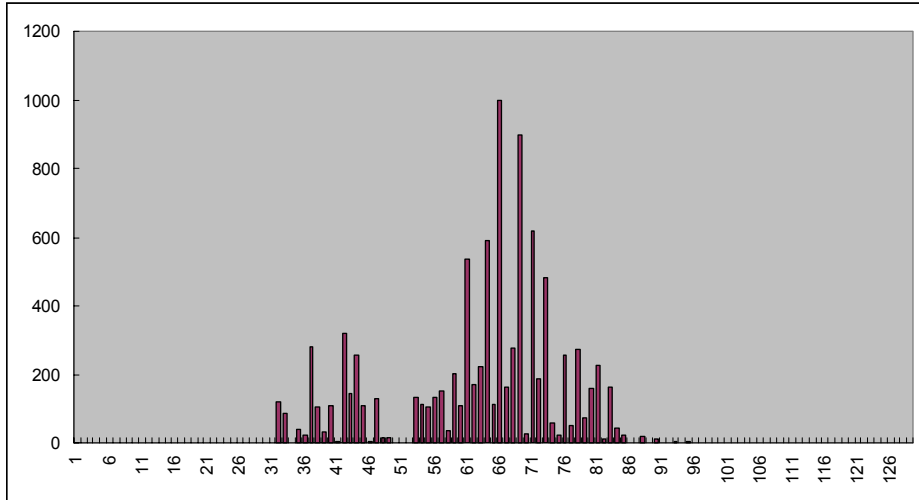
$$\text{멜로디간격} = \sqrt{\frac{\sum_{i=1}^{N-1} (w_{i+1} - w_i)^2}{N}}$$

위와 마찬가지로 w_i 는 전체 멜로디에서 i 번째의 멜로디 pitch 수치이고, N 은 멜로디의 총 개수이다.

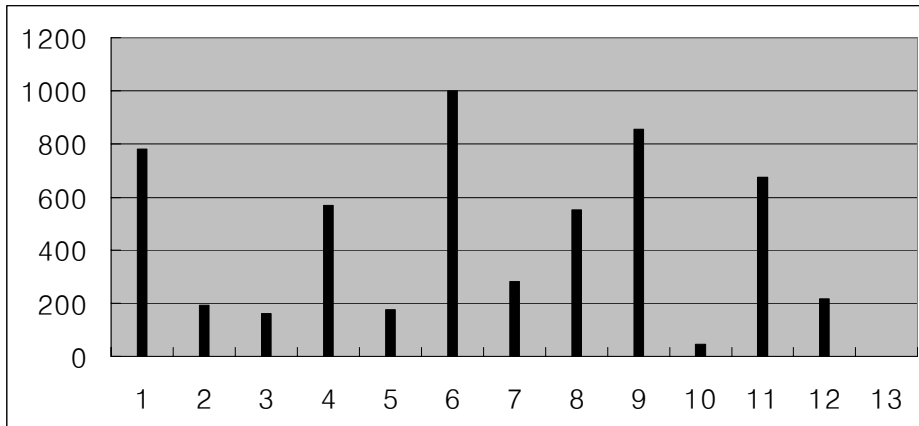
6. 음표 빈도

얼마나 많은 코드를 쓰는 가도 음악의 장르를 나누는 중요한 기준이다. 텐션 코드(tension chord)가 많은 재즈의 경우, 음과 음 사이가 완전음이 아닌 불안정한 음의 간격을 가지게 되므로 음표의 빈도를 계산할 경우 주로 나타나는 음들의 pitch 사이 간격이 장르를 나누는 훌륭한 기준이 될 수 있다. [Cook and Tzanetakis, 2002]

음표의 빈도는 두 가지 방법으로 계산이 가능한데, 첫 번째는 미디에서 다루는 0에서 127까지의 음표 범위를 배열로 잡아 나타난 음표들을 각각의 배열에 카운트를 하는 방법이고, 두 번째는 mod 12 연산을 수행하여 각 음표들의 옥타브 차이를 없애서 카운트하는 방법이다. 첫 번째 방법은 각 음악들마다의 특성을 정확하게 표현할 수 있는 장점이 있지만, 127개의 feature 가 생성되므로 학습이 잘 되지 않는 문제점이 있다. 반면에 두 번째 방법을 사용하면 음악 특성이 흐려지는 대신, 하나의 옥타브 내에서 충분한 정보가 쌓이게 되므로 학습이 용이한 장점이 있다. 실험에서는 두 번째 방법을 사용하여 12개의 feature를 추가하여 실험하였다.

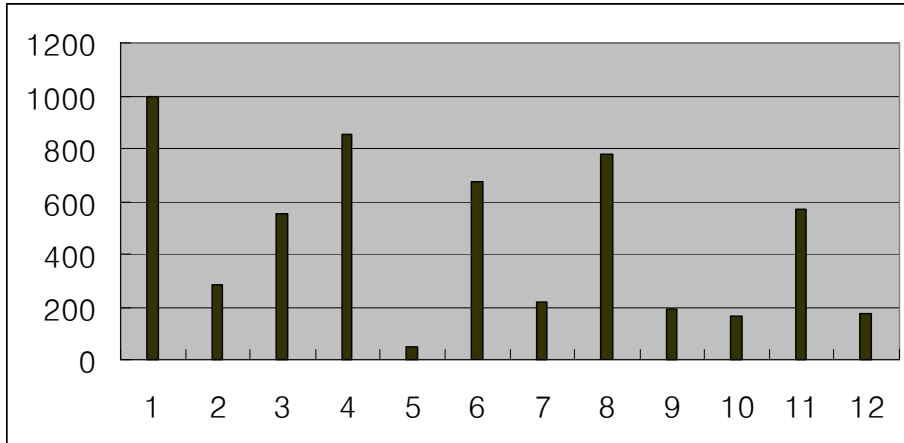


[그림 3] 0부터 127까지의 음표 빈도를 그대로 나타낸 그래프



[그림 4] 옥타브를 무시하고 같은 음들의 빈도를 합쳐서 표시한 그래프

그래프에서 나타나듯이, 각 음악들마다 음표의 사용 회수는 다를 수 밖에 없으므로 데이터의 최대값이 1000이 되도록 정규화 하여 수치를 표현하였다. 여기서 한가지 유의해야 할 점은, 최대값이 나오는 부분은 그 음악의 key 가 될 확률이 높으며, 따라서 이 key를 기본으로 음악의 조성 즉, major 인지 minor 인지를 구분할 수 있기 때문에 최대값을 가장 왼쪽으로 shift하여야 각 음악에 대한 동등한 비교가 가능하다는 점이다. 이때 최대값인 1000은 어느 음악에서나 같은 수치를 가지므로 의미가 없다고 판단되어 생략했다.



[그림 5] . 가장 높은 값을 가지는 key을 가장 왼쪽으로 shift한 그래프

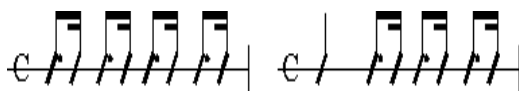
7. 박자 (리듬)

실험에서 가장 중요시되는 데이터 부분이다. MIDI 파일에서 리듬 데이터는 채널 10번을 사용하며, 각 채널에서 볼 때의 키 값, 즉 음표값이 드럼 소리에 대한 정보를 담고 있다. 아래는 General MIDI에서 각 키 값에 대응되는 드럼 정보이다.

드럼의 정보를 feature 화하기 위해서는 드럼의 형식을 분석 할 필요가 있다. 아래 표에서 보이듯, 각 드럼은 Snare 나 hihat 등 특정한 드럼파트로 분류가 되며, 이 드럼파트가 울리는 패턴은 장르마다 조금씩 차이가 있다. 따라서 한 마디에서 드럼이 찍히는 시간의 차이를 feature 화 할 수 있다면, 드럼 정보를 분석 대상으로 삼을 수 있다. 실험에서는 대표적인 3개의 드럼 파트와 나머지 드럼 파트를 구분하여 각 박자의 어느 위치에 드럼 정보가 출력되는지를 표시하는 방법을 사용하였다. 각 박자의 드럼 출력 위치가 중요한 이유는, 이 드럼의 위치를 통해 리듬의 정보가 바뀌며, 리듬의 정보는 특정 장르를 구분하는데 중요한 정보가 되기 때문이다. 음악적으로 볼 때, 정통 락의 일반적인 비트 구조는 [그림 6]과 같다.



[그림 6] 락의 기본 리듬



[그림 7] 재즈의 기본 리듬

반면 재즈의 리듬은 [그림 7] 과 같이, 같은 4박에서의 비트의 위치가 다르다. 따라서 비트에서의 리듬의 위치 정보는 장르 구분의 좋은 기준이 된다. 이를 바탕으로 Snare, Drum, Hihat, Tom 의 4개의 드럼에 대해 각각 4차원의 벡터 정보를 표현하는 방법이 아래에 설명되어있다.

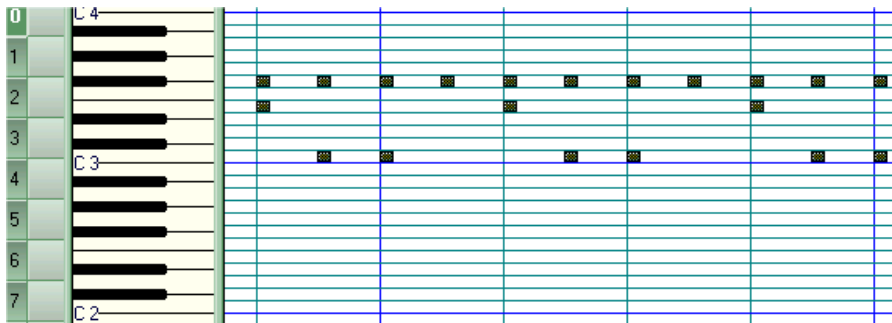
Key Midi DRUM SOUND	Key Midi DRUM SOUND
35 Acoustic Bass Drum	59 Ride Cymbal 2
36 Bass Drum 1	60 Hi Bongo
37 Side Stick	61 Low Bongo
38 Acoustic Snare	62 Mute Hi Conga
39 Hand Clap	63 Open Hi Conga
40 Electric Snare	64 Low Conga
41 Low Floor Tom	65 High Timbale
42 Closed Hi-Hat	66 Low Timbale
43 High Floor Tom	67 High Agogo
44 Pedal Hi-Hat	68 Low Agogo
45 Low Tom	69 Cabasa
46 Open Hi-Hat	70 Maracas
47 Low-Mid Tom	71 Short Whistle
48 Hi-Mid Tom	72 Long Whistle
49 Crash Cymbal 1	73 Short Guiro
50 High Tom	74 Long Guiro
51 Ride Cymbal 1	75 Claves
52 Chinese Cymbal	76 Hi Wood Block
53 Ride Bell	77 Low Wood Block
54 Tambourine	78 Mute Cuica
55 Splash Cymbal	79 Open Cuica
56 Cowbell	80 Mute Triangle
57 Crash Cymbal 2	81 Open Triangle
58 Vibraslap	

[표 3] 표준 미디에서 제안한 각 드럼 파트의 대응 숫자 표

위의 표는, 표준 미디에서 제안한 각 드럼 파트의 대응 숫자 표이다. 각 숫자

는 보통 시퀀서에서의 음높이를 의미하지만, 드럼을 위한 채널인 11번 채널에서는 파트를 구분하는 기준이 된다. [그림 8] 은 Cakewalk 라는 (<http://www.cakewalk.com>) 가장 대표적인 미디 시퀀서의 캡처 화면이다. 원래 시퀀서(Sequencer)란 부호화된 정보의 입력으로 미디 악기를 연주시키는 기계 장치를 의미했다. 예를 들어 일반 녹음기에 기타 연주를 녹음한 후 재생을 하면 녹음된 연주가 흘러나오듯이, 시퀀서에 연주에 대한 정보(음정, 박자...등)을 녹음(입력)하면 재생 시 그 정보를 연주로 나타낸다. 하지만 컴퓨터 음악이 확대됨에 따라서 시퀀서는 소프트웨어적으로 개발되기 시작했고, 최근에 시퀀서라 함은 시퀀스 기능을 지닌 컴퓨터용 프로그램을 지칭하는 경우가 많다. [박운영, 2000] 즉, Cakewalk 는 소프트웨어적인 시퀀서이고, 여기서의 정보는 그대로 미디 데이터로 저장되어 음악을 연주하는 정보로 쓰이게 된다는 뜻이다.

[그림 8] 의 음악은 Phill Collins 의 Easy Lover 라는 팝송의 미디 정보이다. [그림 8]에서 보면 이 음악의 드럼 부분은 총 3개의 악기를 사용하였으며, 위의 코드표와 대조를 해 보면, 이 음악에 사용된 악기는 맨 아래부분 36번 key의 Bass Drum 1 과 40번 key의 Electric Snare , 마지막으로 가장 윗 부분의 42번 key 인 Closed Hi-Hat 이다.



[그림 8] 시퀀서에서의 드럼 정보

이 정보를 MIDI File DisAssembler 로 분석한 결과는 아래와 같다.

```

14: 1: 48 |On Note      | chan=10  | pitch=F#1      | vol=30
      49 |(Off) Note   | chan=10  | pitch=f#1     |
      96 |On Note      | chan=10  | pitch=E 1     | vol=100
          |On Note      | chan=10  | pitch=F#1     | vol=67
      97 |(Off) Note   | chan=10  | pitch=f#1     |
          |(Off) Note   | chan=10  | pitch=e 1     |
  
```



```

2: 48 |On Note      | chan=10 | pitch=C 1      | vol=79
      |On Note      | chan=10 | pitch=F#1     | vol=42
  49 |(Off) Note   | chan=10 | pitch=f#1     |
      |(Off) Note   | chan=10 | pitch=c 1     |
  96 |On Note      | chan=10 | pitch=C 1     | vol=99
      |On Note      | chan=10 | pitch=F#1     | vol=67
  97 |(Off) Note   | chan=10 | pitch=f#1     |
      |(Off) Note   | chan=10 | pitch=c 1     |
3: 48 |On Note      | chan=10 | pitch=F#1     | vol=38
  49 |(Off) Note   | chan=10 | pitch=f#1     |
  96 |On Note      | chan=10 | pitch=E 1     | vol=100
      |On Note      | chan=10 | pitch=F#1     | vol=67
  97 |(Off) Note   | chan=10 | pitch=f#1     |
      |(Off) Note   | chan=10 | pitch=e 1     |
4: 48 |On Note      | chan=10 | pitch=C 1     | vol=79
      |On Note      | chan=10 | pitch=F#1     | vol=42
  49 |(Off) Note   | chan=10 | pitch=f#1     |
      |(Off) Note   | chan=10 | pitch=c 1     |
  96 |On Note      | chan=10 | pitch=C 1     | vol=99
      |On Note      | chan=10 | pitch=F#1     | vol=67
  97 |(Off) Note   | chan=10 | pitch=f#1     |
      |(Off) Note   | chan=10 | pitch=c 1     |

```

비트 정보는 타격 순간이 중요하기 때문에 Off 노트 부분을 생각하지 않는다면 위에서 F#1의 key에 해당되는 Closed Hihat은 1번째 박자에서 총 1박의 전체 숫자인 96의 division 범위 중 49번과 96번에 균일하게 타격 되는 것을 볼 수 있고, E의 key에 해당되는 Electric Snare는 1,3박의 96번 위치에, 마지막으로 C의 key에 해당되는 Bass Drum은 2박의 48과 96에, 그리고 4박의 48과 96에 타격 되는 것을 볼 수 있다. 이를 표로 정리하면 아래와 같다.

악기 \ 박자	1박	2박	3박	4박
drum		48, 96		48, 96
snare	96		96	
hihat	49, 96	49, 96	49, 96	49, 96

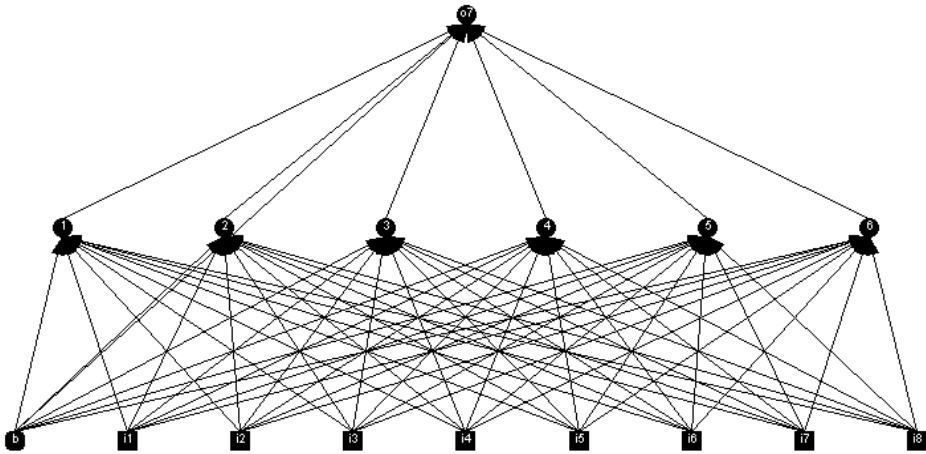
[표 4] MIDI 곡 Easy Lover에 대한 대한 비트 정보

이를 데이터에 수치화 하기 위해, 위의 3개의 악기 이외에 tomtom의 정보까지 총 4개의 악기를 각각 4차원의 벡터로 표현하여 실험 데이터에 16개의 feature를 추가하였다. 이렇게 구성된 데이터는 총 29개의 feature를 갖게 된다.

3장 분석을 위한 기계학습 기법

3-1절. Artificial Neural Network

인공 신경망이란 인간이나 동물의 뇌의 구조를 모방한 계산 모델이다. 신경망 모델은 각 뉴런의 기능, 망의 구조, 데이터에 따른 적응 알고리즘에 의해 구분된다. 모든 신경망은 기본적으로 하나의 뉴런에서 시작하지만, 일반적으로는 [그림 9]와 같이 내부에 숨겨진(hidden) 뉴런을 포함시켜 다층 퍼셉트론으로 학습한다.



[그림 9] 일반적인 다층 퍼셉트론에서의 신경망 구조

다층 퍼셉트론은 비선형 활성화함수를 가지고 있고, 은닉 층으로 불리는 중간 층을 가지며, 각 층의 뉴런들은 인접한 층의 다른 모든 뉴런들과 상호 연결되어 있다는 특징을 지닌다. 학습은 출력 층의 에러 값이 최소가 되는 방향으로 이루어진다. [Haykin, 1998]

$$e_j(n) = d_j(n) - y_j(n) \tag{1}$$

$$E(n) = \frac{1}{2} \sum_{j=1}^N e_j^2(n), \quad E_{av} = \frac{1}{N} \sum_{n=1}^N E(n) \tag{2}$$

식 (1) 은 j번째 출력 뉴런에서의 에러값 e를 구하는 것으로 출력 값 y 와 학

습 값 d 사이의 오차 값을 사용한다. 그리고, 식 (2) 는 출력 층의 모든 뉴런에 대한 에러 값을 더하여 2로 나눈 후, 전체 학습 데이터 패턴에 대한 평균 에러 값을 구하는 과정이다.

실제 학습은 다음의 과정을 거친다. 먼저 입력 층의 뉴런은 입력 신호 값에 가중치를 곱한 후 은닉 층으로 보낸다. 은닉 층의 뉴런은 입력 층의 뉴런들로부터 받은 값들을 모두 더한 후 비선형 활성화 함수 처리를 하고 다시 가중치를 곱하여 출력 층으로 보낸다. 출력 층의 뉴런에서는 은닉 층의 뉴런들로부터 받은 값들을 모두 더하여 다시 활성화 함수 처리를 하여 결과를 내보내게 된다. 이렇게 이전 층에서 다음 층으로 값을 전달하는 방식을 순차 방식(feed-forward) 방식이라고 부른다. 이들 출력 층에서 나오는 결과 값들과 실제 우리가 값을 비교하여 그 오차 값을 계산한 후, 신경망의 weight를 갱신해 주어야 하는데, 출력 층에서 입력 층까지 반대 방향으로 갱신 작업이 이루어진다고 하여 오류 역전파 방식(error-back propagation) 이라 한다.

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)} + \alpha [w_{ji}^{(l)}(n-1)] + \eta \delta_{ji}^{(l)}(n) y_{ji}^{(l)}(n) \quad (3)$$

$$\delta_{ji}^{(l)}(n) = \begin{bmatrix} e_j^{(L)}(n) \phi'_{j'}(v_j^{(L)}(n)) \\ \phi'_{j'}(v_j^{(L)}(n)) \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \end{bmatrix} \quad (4)$$

$$\phi_j(v_j(n)) = \frac{1}{1 + \exp(-av_j(n))}, a > 0 \quad -\infty < v_j(n) < \infty \quad (5)$$

$$\phi'_{j'}(v_j(n)) = ay_j(n) [1 - y_j(n)] \quad (6)$$

각 층의 오류 역 전파를 통한 가중치 갱신은 식(3) 과 같이 이전 층의 뉴런과 현재 층 l(중간 층) 또는 L(출력 층)의 j 번째 뉴런에 대한 계산으로 이루어지며, 이렇게 나온 값으로 식(4)를 이용하여 에러값 δ 를 계산하게 된다. δ 값을 구하기 위해서는 활성화 함수가 미분 가능해야 하는데, Logistic function 또는 Hyperbolic tangent function 등이 사용된다. 식(5)와 (6) 은 전자에 해당하는 시그모이드 함수(sigmoid function)의 원형 식과 미분 식을 보여주고 있다.

실제 학습에 있어서는 크게 순차적인 방법과 일괄적인 방법이 있다. 순차적인

방법은 개개의 입력 값들에 대한 에러 값을 계산하여 바로 가중치를 갱신하는 방식으로 구현이 간편하다는 점과 방대하고 복잡한 패턴을 학습하는데 상대적으로 신속한 장점이 있다. 반면 일괄적인 방법은 모든 입력 패턴의 값을 처리한 후 마지막에 한번 에러값을 통한 갱신을 해 주는 방식으로 비교적 적은 양의 데이터를 빠르게 처리할 수 있는 장점이 있다. 식 (1)은 순차 학습 방법의 에러 계산식이며 식 (7)은 일괄 학습 방법의 계산식이다.

$$E_{av} = \frac{1}{2N} \sum_{n=1}^N \sum_{j=C} e_j^2(n) \quad (7)$$

그 외, 패턴을 효율적으로 학습하기 위해서 입력 데이터를 정규화(Normalize)시키는 방법도 사용되며, 학습 시의 에러의 수렴 속도를 빠르게 하기 위하여 모멘텀 항의 추가, 학습 계수 최적화, 선택적 재학습 방법 등이 사용되기도 하지만, 이번 실험에서는 기본적인 신경망 구조를 통하여 실험 데이터를 학습하였다.

3-2절. Decision Tree Model

결정 트리 학습은 널리 사용되며 귀납적 추론(inductive inference)에 매우 실용적인 방법 중 하나이다. 결정 트리가 노이즈에 강하고 논리합 표현을 학습하는 이산 함수를 유도하는 방법이고 본 논문에서는 이산값으로 표현되는 데이터를 사용하기 때문에, 결정 트리는 이 데이터로부터 규칙을 학습하는 데 적합하다고 할 수 있다. [Mitchell, 1997]

결정 트리에서는 각 비단말 노드가 인스턴스의 어떤 속성(Attribute) 검사를 뜻하고, 그 노드로부터의 가지(branch)는 이 속성이 가질 수 있는 가능한 여러 값 중에서 하나이다. 각 인스턴스는 트리의 루트 노드에서 시작해서 아래로 내려가면서 분류된다. 비단말 노드에서는 인스턴스의 속성값을 검사해서 해당 값에 따라 트리의 가지가 선택된다. 이러한 과정이 새로운 노드를 루트로 하는 서브 트리에서도 반복되어 마지막 단말 노드에까지 이르게 된다. 결정 트리 학습에서 가장 중요한 문제는 트리의 각 노드에서 어떤 속성을 검사할 것인지를 선택하는 것이다. 예제 인스턴스를 분류하는데 가장 유용한 속성을 선택하기 위해서, 정량적 측정 단위인 정보 이득(information gain)이 사용될 수 있다. 이 측정 단위는 결정 트리 학습 알고리즘으로 가장 널리 쓰이는 ID3 알고리즘과 그 후계자라고 할 수 있는 C4.5 알고리즘에서도 채택된 것이다. [Quinlan, 1993] 목적 속성이 c

개의 서로 다른 값을 가질 수 있을 때 예제 인스턴스 집합(S)의 엔트로피는

$$Entropy(S) = \sum_{i=1}^c -p_i \log_2 P_i$$

와 같이 정의된다. 그러면, 예제 인스턴스 집합 S에서의 속성 A에 대한 정보 이득 Gain(S,A)는 다음과 같이 정의된다.

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

여기서 Value(A)는 속성 A가 가질 수 있는 가능한 모든 값을 뜻하고, S_v 는 A의 값이 v인 속성 A에 따라 학습 예제들을 나눔으로써 얻어지는 엔트로피 감소의 평균으로 해석될 수 있다.

3-3절. Decision Tree Model with AdaBoost

AdaBoost는 Classifier를 순차적으로 생성해 낸 후 이를 결합함으로써 더 나은 성능을 보이는 Classifier를 생성해 내는 것을 목적으로 한다. 이 방법은 Schapire에 의해서 제안된 후 많은 연구자들에 의해서 연구되고 실제 문제에 응용되었다.[Freund and Schapire, 1996] 여기서 'boosting'은 알고리즘이 데이터의 중요도(importance factor)를 재조정함으로써 새로운 데이터 중요도를 생성하기 때문에 붙여진 이름이다. 각각의 가설(hypothesis)은 이렇게 새롭게 매겨진 중요도를 바탕으로 훈련을 진행하게 된다. 'Ada'는 이렇게 만들어진 각각의 가설(θ_t)의 투표 가중치(α_t)를 적응적으로 계산할 수 있다는 의미이다.

이때 약학습자($\theta_t(x)$)의 출력의 부호는 x에 해당되는 -1 또는 1의 범주를 가리키며, $|\theta_t(x)|$ 의 값은 이러한 추정에 대한 확신도(Confidence)를 가리킨다. 그러므로 $\theta_t(x)$ 가 0에 가까우면 낮은 확신도를 가지고 추정하는 것이고, 0에서 멀어질수록 높은 확신도를 가지고 추정하는 것이 된다. AdaBoost는 최종적으로 이러한 약학습자를 결합하여 Classifier를 만든다.

$$f(x) = \sum_{t=1}^T \alpha_t \theta_t(x)$$

여기서 T 는 반복 회수를 의미하고 α_t 는 약 학습자의 무게를 나타낸다. 만약 우리가 $\theta_t(x)$ 의 범위를 $[-1,1]$ 로 제한하는 경우라면, α_t 를 원래 AdaBoost 알고리즘의 식을 그대로 사용할 수 있다. 본 논문에서는 약 학습자를 10개의 Decision Tree 알고리즘으로 선택하여 실험을 하였다.

3-4절. K-nearest neighbor Classifier

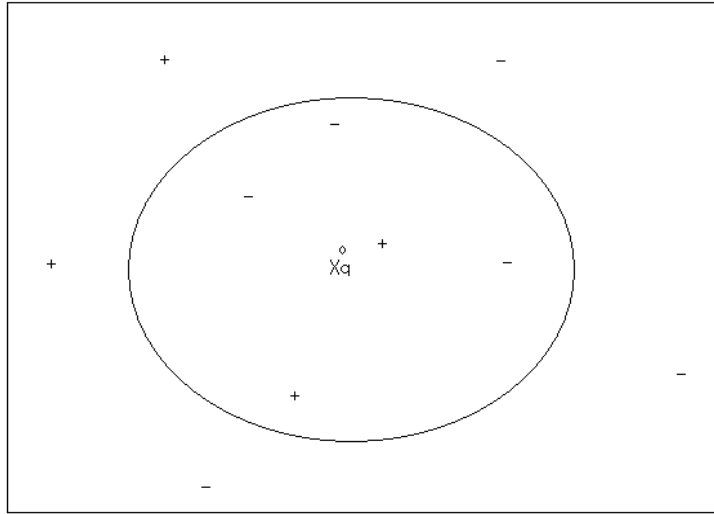
이 분류법은 가장 기초적인 분류 방법 중 하나로, 모든 데이터가 n -차원 공간 R^n 상의 점들로 대응된다고 가정한다. 한 데이터와 다른 데이터 사이의 최근점은 표준 유클리드 거리(standard Euclidean distance)를 통해서 정의된다. r 번째 feature $a_r(x)$ 를 가진 사례 x 가 다음의 feature 벡터를 가진다고 가정하면

$$\langle a_1(x), a_2(x), \dots, a_n(x) \rangle$$

와 같이 표현되고, 두 데이터 x_i 와 x_j 사이의 거리 $d(x_i, x_j)$ 는 다음과 같다.

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

최근점 학습에서 목표 함수는 이산값 (discrete-value) 이거나 실수값 (real-value) 이 된다. [그림 10]은 k -최근점 알고리즘이 데이터가 2차원 공간상의 한 점이고 목표함수가 이항 값을 가지는 경우에 어떻게 작동하는가를 보여준다. [그림 10]에서 '+'는 양성인 데이터를, '-'는 음성인 데이터를 각각 의미한다. 이 그림에서 x_q 는 1-최근점 알고리즘을 사용할 때는 양성으로 분류되지만, 5-최근점 알고리즘을 사용할 때는 음성으로 분류된다. [Mitchell, 1997]



[그림 10] k-최근접 알고리즘의 동작 예

4장 실험 및 결과

4-1절 학습 데이터

실험을 위한 데이터는 우리 나라 대표 4대 통신사중 하나인 나우누리과 하이텔의 공개자료실과 미디 동호회, 그리고 각종 인터넷 사이트에서 모은 미디 곡을 장르별로 각각 50곡씩 선택하였다.

4-2절 실험 설계 및 결과

실험 데이터는 Neural net classifier와 Decision Tree, AdaBoost를 사용한 Decision Tree, 마지막으로 K-nearest Neighbor Model 등의 5개의 모델에 대해 적용하였다. 실험은 각 장르별로 총 250개의 곡을 training 데이터로 하고 이중 랜덤한 10%를 test 데이터로 하여 모두 10번의 10 fold cross validation을 수행하여 결과를 확인하였다. 실험 결과는 아래와 같다.

1. Artificial Neural Net classifier

Correctly Classified Instances	113	45.0331 %
Incorrectly Classified Instances	137	54.9669 %
Kappa statistic	0.3127	
Mean absolute error	0.2298	
Root mean squared error	0.4129	
Relative absolute error	71.8007 %	
Root relative squared error	103.2172 %	
Total Number of Instances	151	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.2	0.091	0.353	0.2	0.255	0
0.419	0.192	0.361	0.419	0.388	1

0.333	0.14	0.37	0.333	0.351	2
0.733	0.14	0.564	0.733	0.638	3
0.567	0.124	0.531	0.567	0.548	4

	Pop	Rock	Jazz	Classic	Korean	CorRate
Pop	10	13	12	6	9	20 %
Rock	5	22	11	5	8	44 %
Jazz	6	14	17	10	3	34 %
Classic	3	2	4	37	5	74 %
Korean	4	10	2	7	28	56 %

[표 5] Artificial Neural Net을 이용한 실험 결과 (Confusion Matrix)

Artificial Neural Net은 기계 학습 방법 중에서 비교적 특성이 뚜렷하지 않은 데이터에 대해서도 학습을 수행하는 특성이 있다. 결과를 살펴보면, 클래식을 맞춘 비율이 74퍼센트로 가장 높다. 하지만 Artificial Neural Net의 학습 능력 만큼의 결과가 다른 장르에서는 나오지 않았다. 이는 실험 데이터가 Artificial Neural Net이 충분히 학습을 할 정도로 많지 않기 때문으로 생각된다. Feature특성이 뚜렷한 Classic 부분은 학습속도가 빨랐지만, 그렇지 않은 Pop 이나 Jazz 등에서 학습 효과가 별로 좋지 않았다.

2. Decision Tree Classifier

Number of Leaves : 21

Size of the tree : 41

Time taken to build model: 0.13 seconds

=== Summary ===

```

Correctly Classified Instances      155          61.5894 %
Incorrectly Classified Instances    95          38.4106 %
Kappa statistic                    0.5197
Mean absolute error                 0.1579
Root mean squared error            0.3751
Relative absolute error            49.3604 %
Root relative squared error        93.7662 %
    
```

=== Detailed Accuracy By Class ===

```

TP Rate  FP Rate  Precision  Recall  F-Measure  Class
 0.667   0.091   0.645     0.667   0.656     0
 0.516   0.158   0.457     0.516   0.485     1
 0.333   0.099   0.455     0.333   0.385     2
 0.967   0.041   0.853     0.967   0.906     3
 0.6     0.091   0.621     0.6     0.61      4
    
```

	Pop	Rock	Jazz	Classic	Korean	CorRate
Pop	34	3	7	3	3	68 %
Rock	7	26	6	1	10	52 %
Jazz	7	21	17	0	5	34 %
Classic	1	0	3	46	0	92 %
Korean	3	7	7	3	30	60 %

[표 6] Decision Tree을 이용한 실험 결과 (Confusion Matrix)

두 번째로 수행한 실험은 Decision Tree를 이용한 실험이었다. Classic 부분에서 92%라는 놀라운 장르 분석 능력을 보여주었는데, 이는 Feature별로 트리의

분기점을 만들며 학습을 하는 Decision Tree의 특성때문으로 생각된다. Classic 부분은 리듬 파트의 Feature값이 거의 제로에 가깝기 때문에 이 특성만으로도 확실한 구분이 가능하기 때문이다. 따라서 리듬별로 구분이 뚜렷한 Pop과 Rock 부분에서도 학습 능력이 좋은 반면, 리듬이 자유로운 Jazz 는 학습을 잘 수행하지 못한 결과를 보여주었다.

3. AdaBoost Decision Tree Classifier

Number of Leaves : 19
 Size of the tree : 37
 Weight: 2.6
 Number of performed Iterations: 10
 Time taken to build model: 1.43 seconds

=== Summary ===

Correctly Classified Instances	183	72.8477 %
Incorrectly Classified Instances	67	27.1523 %
Kappa statistic	0.6606	
Mean absolute error	0.1783	
Root mean squared error	0.293	
Relative absolute error	55.7073 %	
Root relative squared error	73.2422 %	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.733	0.091	0.667	0.733	0.698	0
0.581	0.117	0.563	0.581	0.571	1
0.633	0.041	0.792	0.633	0.704	2
0.967	0.017	0.935	0.967	0.951	3
0.733	0.074	0.71	0.733	0.721	4

	Pop	Rock	Jazz	Classic	Korean	CorRate
Pop	37	7	0	2	4	74 %
Rock	7	30	6	0	7	60 %
Jazz	7	10	32	0	1	64 %
Classic	2	0	0	48	0	96 %
Korean	3	7	2	2	36	72 %

[표 7] AdaBoost를 이용한 Decision Tree의 실험 결과 (Confusion Matrix)

세 번째로 실험 결과가 좋은 Decision Tree에 AdaBoost를 이용하여 학습 능력을 배가시켜 보았다. 예상대로 대부분의 장르에 대해 Decision Tree만을 사용했을 때보다 학습 결과가 좋게 나왔다. 특기할만한 사항은 Jazz에서의 학습 결과의 대폭적인 상승이다. 이는 AdaBoost를 통해 리듬파트의 가중치를 줄여주는 대신에 다른 Feature부분의 분류 가중치를 높여주어 좀 더 학습을 용이하게 수행했기 때문으로 분석된다.

4. K-nearest Neighbor Classifier

=== Summary ===

Correctly Classified Instances	67	26.8477 %
Incorrectly Classified Instances	183	73.2523 %
Kappa statistic	0.4606	
Mean absolute error	0.3783	
Root mean squared error	0.493	
Relative absolute error	35.7073 %	
Root relative squared error	63.2422 %	

	Pop	Rock	Jazz	Classic	Korean	CorRate
Pop	5	18	17	5	5	10 %
Rock	22	13	10	4	1	26 %
Jazz	12	13	8	10	7	16 %
Classic	3	0	10	32	5	64 %
Korean	3	5	12	21	9	18 %

[표 8] K-nearest Neighbor Classifier를 이용한 실험 결과 (Confusion Matrix)

마지막으로 4번째 실험은 K-nearest Neighbor Classifier를 이용한 실험 결과이다. K값을 3에서 7사이로 조종하여 실험을 수행하였으며 결과는 가장 수행결과가 좋았던 K값이 6일때의 결과값이다. 하지만 K-nearest Neighbor Classifier의 가장 좋은 실험결과가 4가지의 실험 가운데 가장 안 좋은 실험결과였다. Feature의 특성들과 가중치가 각각 다른 이번 실험 데이터에서 Feature의 Vector Distance 만으로 분류 작업을 수행하는 K-nearest Neighbor Classifier는 학습 능력이 약할 수 밖에 없었던 것으로 추론되며, 4번째 실험은 같은 데이터에 대해 어떤 기계학습을 적용하는가에 따라 학습의 결과가 많이 달라질 수 있음을 알게 해준 실험이었다.

실험 방법	맞는 장르수	맞춘 비율	가장 잘 맞춘 장르	두번째
Neural Net	113	45.03 %	클래식 (74%)	가요(56%)
DecisionTree	155	65.59 %	클래식 (92%)	팝 (64%)
DT_Boost	183	72.85 %	클래식 (96%)	팝 (74%)
K-nearest	67	26.85 %	클래식 (64%)	락 (26%)

[표 9] 4개의 실험에 대한 최종 결과 분석표

최종 결과를 정리해 보았다. 실험 결과를 통해 알 수 있듯이 각종 인공지능 기법을 사용하여 장르 구분을 수행해 본 결과 최고로 분류를 잘한 방법은

AdaBoost를 이용한 Decision Tree 학습기법이었으며 52.58% 정도의 분류 정확도를 보여주었다. 랜덤하게 선택할 시의 평균 선택 퍼센트인 20% 보다 높은 수치였다. 20개가 넘는 많은 개수를 가지는 feature의 특성상 Neural Net 이 가장 학습결과를 높게 측정할 것이라 생각했는데, Decision Tree 가 가장 높은 값이 나와서 의외였다. 이 학습에 대해 AdaBoost를 수행하여 정확도를 높인 결과 맞춘 비율을 72.85% 까지 올릴 수 있었다.

실험을 통하여 가장 잘 맞춘 장르는 4개의 실험 결과 모두 클래식이었는데, 이는 클래식이 이 실험의 feature selection에서 강조된 드럼 비트를 상대적으로 사용하지 않아 이 부분의 벡터값이 0으로 통일되어 selection 이 쉬워졌기 때문으로 분석된다.

두 번째로 잘 맞춘 장르는 팝이 평균 68% 정도로 많고 그 다음으로 가요와 소프트 락 순 이었다. 일정한 드럼 비트를 사용하는 락이 두 번째로 잘 분류될 줄 알았는데, 팝이 선택된 것은 의외였으나, 이는 오히려 팝이 뚜렷한 1,3박의 드럼 비트를 사용하고 락은 중간 에드립 부분에 변화되는 비트가 많은 성질을 갖기 때문으로 분석된다.

각 Feature 마다의 특징이 있기 때문에 단순히 Feature를 정량화된 수치로 놓고 학습을 하는 Artificial Neural Net이나 K-Nearest Neighbor Classifier에서는 학습 결과가 좋지 않은 반면, 가장 특성이 높은 Feature로 선분류를 하는 Decision Tree에 가중치를 적절히 변형시키는 AdaBoost를 통한 학습을 수행한 결과가 가장 좋게 나왔다고 최종적으로 결론내릴 수 있었다.

5장 결론

음악의 장르라는 것은 2장에서 서술하였듯이 정확한 분류가 어렵다. 하지만, 대표적인 장르에 대해서 사람의 손을 거치지 않고, 알려진 정보를 토대로 feature를 추출한 뒤, 빠른 방법으로 분류를 수행하여 납득할만한 분류 정확도를 나타낸 것은 이런 분석 방법이 웹 상에서의 응용 가능하다는 점에서 유용하다. 이 실험의 궁극적인 목적은, 사람으로 하여금 사람의 취향에 맞는 음악을 방대한 음악 데이터 안에서 선택적으로 추출하도록 하는 것이므로, 가장 기본적인 미디 음악에 대해서 장르 구분을 하는 것이 이런 궁극적 목적의 시작이라 생각된다.

앞으로 미디 음악뿐만 아니라 실제적인 아날로그 음악 파형에 대해서도 빠르게 음악 장르를 구분하여 원하는 음악을 손쉽게 찾고, 또 자신의 취향에 맞는 몰랐던 음악을 찾을 수 있게 하는 알고리즘을 개발한다면, 네트워크를 통하여 음악을 향유하며 행복을 찾는 인간의 문화생활을 더 한층 풍족하게 만들 수 있을 것이다.

참고 문헌

- [김태희, 1999] 김태희, 미디 기초 개론, 정보게이트, pp. 35-36, 1999.
- [박운영, 2000] 박운영, 나의꿈 나의 미디, 혜지원, pp. 10-15, 2000.
- [Akoff] <http://www.akoff.com/music-recognition.html>. Homepage
- [Collier, 1991] J. L. Collier, 재즈음악의 역사, 세광 음악 출판사, pp. 16-19, 1991.
- [Cook and Tzanetakis, 2002] P. Cook and G. Tzanetakis, Musical genre classification based on the analysis of harmonic content features in audio and MIDI, Andrey Ermolinskiy '02, Independent Work Report, 2002.
- [Dowling, 1978] W. J. Dowling, Scale and contour: Two components of a theory of memory for melodies, *Psychological Review*, 85(4):341-354, 1978.
- [Freund and Schapire, 1996] Y. Freund, Robert E. Schapire, Experiments with a new boosting algorithm, *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148-156, Bari, 1996.
- [Ghias et al., 1995] A. Ghias, J. Logan, D. Chamberlain, and B. Smith, Query by humming-musical information retrieval in an audio database. In *Proceedings of ACM Multimedia '95*, San Francisco, 1995.
- [Haykin, 1998] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Second Edition, Prentice-Hall, pp. 156-157, 1998.
- [Jeff] <http://www.borg.com/~jglatt/progs/software.htm>. Homepage
- [Lap and Kao, 1999] Y. C. Lap and B. Kao, A study for musical features for melodies data bases, Technical Report 99-05, University of Hong Kong, 1999.
- [Lemström, 2000] K. Lemström, String matching techniques for music

- retrieval, Doctoral Thesis, Department of Computer Science, University of Helsinki, 2000.
- [Mitchell, 1997] T. M. Mitchell, *Machine Learning*, The McGraw-Hill Companies, 1997.
- [Pachet and Cazaly, 2000] F. Pachet and D. Cazaly, A Taxonomy of musical genre, *Proceedings of Content-Based Multimedia Information Access (RIAO) Conference*, Paris, 2000.
- [Pollastri and Simoncelli, 2001] E. Pollastri, G. Simoncelli, Classification of melodies by composer with hidden markov models, In *Proceedings of the First International Conference on WEB Delivering of Music (WEDELMUSIC'01)*, Florence, 2001.
- [Quinlan, 1993] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1993.
- [Rossing, 1989] T. D. Rossing, *The Science of Sound 2nd edition*, Addison-Wesley, p.171-178, 1989.
- [Sloboda, 1983] J. A. Sloboda, *The Musical Mind*, Oxford University Press, 1983.
- [Sonicspot] <http://www.sonicspot.com/guide/midifiles.html>. Homepage
- [The, 1981] *The Ultimate Fake Book*, Hal Leonard Publishing Corporation, "C" instruments edition, 1981.
- [Thom, 2000] B. Thom, BoB: an Interactive improvisational music companion, In *Proceedings of the Fourth International Conference on Autonomous Agents*, Barcelona, 2000.
- [Widmer, 2001] G. Widmer, The Musical expression project: A challenge for machine learning and knowledge discovery, *Proceedings of the 12th European Conference on Machine Learning (ECML)*, Freiburg, 2001.
- [Yip and Kao, 2000] C. L. Yip, B. Kao, A study for musical features for melodies databases, *COMPSAC-2000*, 2000.

Abstract

Musical genre is naturally used for constructing musical information in the fast-growing recent computer networks and it plays an important role in the information retrieval as well. Traditionally, genres are classified only by a human-hand. Based on the observation that any general music has a special rhythm part, an instrument set, and melodies, this paper proposes a method for fast classification of musical genre that could apply in the real web environments.

Because a key point of musical classification is preprocessing of musical data and feature selection, this article selects tempo, beat per measure, key, and some more specific data like melody width, melody interval, note occurrence, and rhythm data focused on the special genre. After this preprocessing, we have 29 features and made represent them in vector format for the experiment. The classification is made by 5 machine learning method : Artificial Neural Network, Decision Tree, Decision Tree using AdaBoost, and K-nearest Neighbor Model.

We selected 250 MIDI song data, respectively 50 songs, with 5 genres, namely, soft rock, jazz, classic, pop, and korean-pop. We classified and validated these data using cross validation. The best result was Decision Tree using AdaBoost, with accuracy of 72.85% which is far better than the 20% accuracy of random classification

Keywords: musical genre classification, MIDI, machine learning
Student ID: 2000-21206

감사의 글

끝이 없을 것만 같았던 방황의 시기가 지나고, 이제 석사 졸업을 눈앞에 두고 있습니다. 살아오면서 대학원 2년 반의 기간은 가장 힘들고 고통스러운 시기였지만, 그런 가운데서도 가장 행복하고 즐거운 순간이 함께 했던 시기였다고 회상합니다.

먼저, 연구실을 이끌어주시면서, 학문에서 생활까지 많은 가르침을 주셨던 장병탁 교수님께 감사드립니다. 제 논문이 나오기까지 곁에서 좋은 가르침을 아끼지 않아주신 연구실 맏형들 유섭이형, 성동이형, 정호형, 성배형, 종우형, 규백이형, 장민이형에게 감사드립니다. 영적 인연으로 맺어져있는 저의 대부이자 연구실 선배이신 동연이형은 끊임 없이 연구하고 기도하는 자세가 무엇인지 저에게 가르쳐주셨습니다. 연구실의 제2보스 수용이형, 엄(?)하지만 따뜻한 마음의 소유자 재홍이형, 언제나 웃음으로 절 맞아주시던 선이형, 석사 신입생 때부터 지금까지 지켜봐주신 것, 늘 감사하고 있습니다. 아무것도 모르는 신입생 때 늘 사랑으로 절 아껴주셨던 유환이형, 상욱이형, 호식이형, 형주누나, 인영이누나, 논문의 고마운 조연자 희주에게도 고맙고 감사하는 마음뿐입니다. 먼저 졸업해서 늘 아쉽고 그리웠던 우리시대 철학자 현구형, 이제는 애기 아빠인 태진이형, 빼 없는 치킨으로 우정을 다졌던 승준이는 끈끈한 석사 동기의 우정을 계속 이어갈 것이고, 비슷한 사상으로 애기 많이 나누었던 승우, 연구실의 꽃 인희, 호진이는 선배로서 많이 돌봐주지 못한 것에 미안한 마음뿐입니다. 정문이형과 상준이형, 하영이형은 좀더 많은 얘기를 나누지 못한 것이 너무나 아쉽습니다. 멋진 선배 희웅이형, 보기만 해도 든든한 동민이형은 이 인연이 끝까지 갈 것임을 믿어 의심치 않습니다. 진정한 학자의 모습을 보여주시는 양박사님, 오박사님, 제균이형에게 감사드리며, 철부지 신입생 때에 태진이형과 저를 지도해주시느라 힘들었을 영우형과, 재원이형, 419호의 청량제이자 규백이형과의 트윈콤비 유지선씨에게도 무한한 감사를 드립니다.

어떻게 지나갔는지도 모르게 빨리 흘러간 대학원 생활 속에서, 짧은 시간이었지만 많은 것을 배웠습니다. 연구실에 들어온 처음부터 석사 과정을 마칠 때까지, 끊임없이 여러 분들이 가르쳐주시고, 도움을 주신 덕분이었습니다.

고등학교때부터 모자란 나의 푸념과 쓴소리를 깊은 우정으로 들어준 상욱이와 준호 고맙다. 계속 우정 이어갈 포럼 사람들, 세포시 사람들, 스너크 사람들, 중학교때부터의 인연 상현이, 근용이 고맙고, 과친구들 특히 콤보모임 친구들 성현, 백상, 민규, 용수, 호준, 초이, 홍열, 준형, 계리, 석우, 정지, 최지, 현주와 유락 멤버 우일, 철관, 태호, 형기, 권택, 김박사 프로젝트의 주남, 정현, 정호, 홍우형, 대섭, 상태, 석사동기 강일, 동일이, 신영이, 용석, 승현, 채은도 과동기의 인연이 끝까지 갈 것임을 믿습니다. 저의 영원한 선배인 정현이형, 대학때부터 지금까지 못난 후배 챙겨주신 것 정말 고맙습니다.

무엇보다 감사드려야 할 우리 가족들, 어머니, 아버지, 큰누나, 작은누나, 정말 말로 다할 수 없이 존경하고 사랑합니다.

추운 겨울, 병석에 누워서도 결코 힘들지 않았던 까닭은 가족들, 동기 선후배들의 사랑과, 무엇보다도 지금까지 저에게 아낌없는 사랑을 주는 지원이가 있었기 때문입니다. 모든 분들에게 감사의 마음을 바칩니다.