

# SVM 기반 음성/음악 분류기의 효율적인 임베디드 시스템 구현

## Efficient Implementation of SVM-Based Speech/Music Classification on Embedded Systems

임 정 수, 장 준 혁\*

(Chungsoo Lim, Joon-Hyuk Chang\*)

목포대학교, \*한양대학교 융합전자공학부

(접수일자: 2011년 6월 5일; 수정일자: 2011년 8월 4일; 채택일자: 2011년 10월 19일)

제한된 대역폭을 효율적으로 사용하기 위해서 도입된 가변 전송률은 먼저 신호의 정확한 분류를 필요로 한다. 특히 멀티미디어 서비스가 보편화 되면서 음성/음악 신호 분류의 중요성도 높아지게 되었다. 음성/음악 분류기 중, 서포트벡터머신 (SVM)을 이용한 분류기는 높은 분류 정확도로 주목받고 있다. 그러나 SVM는 많은 계산량과 저장 공간을 요구하므로 효율적인 구현이 요구되며, 특히 임베디드 시스템과 같이 자원이 제한 적인 경우에는 더욱 그러하다. 본 논문에서는 먼저 SVM을 이용한 음성/음악 분류기의 임베디드 시스템으로의 구현을 실행시간과 에너지소비의 관점에서 분석하고, 효율적인 구현을 위한 두가지 방법들을 제안한다. 서포트벡터의 판별결과에의 기여도를 바탕으로 기여도가 낮은 벡터들을 제외하는 방법과, 음성/음악 신호에 기본적으로 존재하는 각 프레임간의 상관관계를 이용하여 입력신호의 일부를 건너뛰는 방법이다. 이 기법들은 SVM의 학습 시 사용되는 다른 최적화 기법에 관계없이 적용이 가능하며, 실험을 통해 분류의 정확도, 실행시간, 그리고 에너지소비의 관점에서 그 성능을 증명하였다.

**핵심용어:** 서포트벡터머신, 음성/음악 분류 알고리즘, 임베디드 시스템

**투고분야:** 음성처리 분야 (2)

Accurate classification of input signals is the key prerequisite for variable bit-rate coding, which has been introduced in order to effectively utilize limited communication bandwidth. Especially, recent surge of multimedia services elevate the importance of speech/music classification. Among many speech/music classifier, the ones based on support vector machine (SVM) have a strong selling point, high classification accuracy, but their computational complexity and memory requirement hinder their way into actual implementations. Therefore, techniques that reduce the computational complexity and the memory requirement is inevitable, particularly for embedded systems. We first analyze implementation of an SVM-based classifier on embedded systems in terms of execution time and energy consumption, and then propose two techniques that alleviate the implementation requirements: One is a technique that removes support vectors that have insignificant contribution to the final classification, and the other is to skip processing some of input signals by virtue of strong correlations in speech/music frames. These are post-processing techniques that can work with any other optimization techniques applied during the training phase of SVM. With experiments, we validate the proposed algorithms from the perspectives of classification accuracy, execution time, and energy consumption.

**Keywords:** Support vector machine (SVM), Speech/Music classification algorithm, Embedded system

**ASK subject classification:** Speech Signal Processing (2)

## I. 서 론

어느 때보다도 높아지고 있다. 제한된 통신망을 효율적으로 사용하기 위해 가변적인 전송률을 가지는 다양한 음성 코덱이 개발 되었는데 [1-2], 음성신호의 유형에 따라 다른 전송률을 할당하기 위해서는 음성신호의 유형을 정확히 분별하는 작업이 우선되어야 한다. 특히 음성/음악 분류에 대한 연구도 많이 진행되고 있는데 가변 전송률 코덱 [2]에 사용될 뿐 아니라 멀티미디어 데이터의 색인 (indexing) [3], 라디오 방송에서 음악이 나오는 채널을 찾기 [4]등 다양한 적용분야가 있다. 구현의 견지에서 보면 멀티미디어 데이터의 색인과 같이 실시간 처리의 요구가 필요치 않은 경우와, 가변 전송률의 적용처럼 실시간 처리가 요구되는 경우로 나눌 수 있다. 실시간 처리의 요구에 상관없이 많은 특징추출기법과 분류기법들이 분류의 정확도와 기법의 복잡도를 고려하여 제안되지만 실시간 처리가 요구되는 경우, 복잡도의 중요도가 그렇지 않은 경우보다 더 높다고 할 수 있다. 특별히 코덱의 경우, 실시간 처리를 위한 제한조건 (allowable delay)이 정해져 있고 음성/음악 분류를 포함한 코덱의 각 부분은 이 조건에 맞게 만들어진다. 따라서 코덱에 적용하는 기법들은 성능향상 뿐 아니라 제한조건에 맞는 효율적인 구현도 고려해야 한다.

근래에 서포트벡터머신 (SVM)을 사용하여 3GPP2 표준 코덱인 selectable mode vocoder (SMV) [1-2]의 음성/음악 분류성능을 향상시키는 기법 [5-6]이 제안되었다. 이 기법은 SMV내에서 만들어지는 파라미터들을 사용하여 별도의 특징추출 과정에 해당하는 계산량을 줄일 수 있다 [7]. 또한 이 SVM기반 분류기는 높은 분류성능을 보인다. 하지만 분류성능에 초점을 맞추어 구현에 대해서는 고려하지 않았기 때문에 본 논문에서는 제안되었던 SVM기반 음성/음악 분류기의 효율적인 구현을 주제로 삼았다.

SVM은 일반적으로 많은 계산량과 서포트벡터를 저장하기 위한 넓은 메모리 공간을 요구한다 [8]. 특히 SVM기반 분류기를 임베디드 시스템에서 사용하기에는 임베디드 시스템의 일반적인 특징인 제한된 프로세서의 성능이나 메모리의 크기로 인해 더욱 어려운 실정이다. SVM의 계산량과 메모리 요구량과 직접적으로 관련이 있는 파라미터는 서포트벡터의 개수와 서포트벡터의 차원 (dimensionality)이고, 많은 연구가 분류 성능의 저하 없이 서포트벡터의 개수와 벡터의 차원을 줄이기 위해 수행되었다 [8-11]. 본 논문에서는 SVM의 임베디드 시스템으로의 효율적인 구현을 위하여, 기존의 기법들과 독립적으로 개발되고 SVM 학습단계 후에 적용이 가능하여 병용

이 용이한 두 가지 기법을 제안한다. 첫째로 각 서포트벡터의 판별에 미치는 영향을 바탕으로 영향이 작은 벡터들을 서포트벡터 집합에서 제외시키는 방법이다. 둘째는 음성과 음악 신호에 존재하는 인접한 프레임간의 상관관계에 기초하여 모든 입력 프레임을 분류하는 것이 아니라 일부만 분류를 실행하고 나머지 프레임은 인접한 프레임의 분류를 따르는 기법이다.

이 기법들을 소개하기 앞서 본 논문에서는 서포트벡터머신의 임베디드 시스템으로의 구현에 대한 사전정보를 얻기 위하여 서포트벡터의 개수와 차원에 따른 임베디드 프로세서에서의 실행시간과 에너지 소비량을 프로세서 시뮬레이터를 사용하여 분석하였고, 또한 SVM 분류기의 프로세서 구조 (부동소수점계산 유닛의 개수와 캐쉬의 크기)의 변화에 따른 성능변화를 분석하였다.

## II. SVM기반 음성/음악 분류기의 임베디드 프로세서에서의 행동 분석

이 장에서는 먼저 RBF를 커널함수로 사용하는 SVM을 복습한다. 그리고 SVM기반 분류기의 사용이 SMV의 실시간 요구조건에 적합한지 알아보고 계산량과 메모리 크기에 영향을 미치는 서포트벡터의 개수와 차원이 어떻게 실행시간과 에너지소비량을 변화시키는지 분석하여 본다.

입력벡터  $X$ 가 선형으로 분류가 가능한 경우, SVM의 판별함수는 다음 식과 같다.

$$f(X) = \sum_{i=1}^M \alpha_i^* z_i < X_i^*, X > + b^* \quad (1)$$

$X_i^*$ 는 학습에 의해 구해진 M개의 서포트 벡터 중  $i$ 번째 벡터이다. 최적화 바이어스 (optimization bias)  $b^*$ 와 라그랑제 승수 (Lagrange Multiplier)  $\alpha^*$ 는 학습에 의해 구해지는 quadratic programming problem의 해이다. 입력벡터가 선형으로 분류가 불가능 한 경우 커널함수를 사용한 SVM의 판별함수는 다음과 같다.

$$f(X) = \sum_{i=1}^M \alpha_i^* z_i K(X_i^*, X) + b^* \quad (2)$$

커널함수로 RBF를 사용한다면  $K(X_i^*, X)$ 는 다음과 같다.

$$K(X_i^*, X) = \exp(-\gamma \| X_i^* - X \|^2) \quad (3)$$

음성/음악의 분류기로 RBF를 커널함수로 사용하는

SVM을 선정하였는데 그 이유는 첫째로 RBF는 입력신호가 선형분류가 가능하지 않은 경우 많이 사용되는 커널 함수이고, 둘째로 계산량이 많아 구현을 연구하는데 적합하기 때문이다.

SVM기반 분류기의 임베디드 프로세서에서의 분석을 위하여 사이클 (cycle)단위로 프로세서의 행동을 모델한 시뮬레이터인 sim-panalyzer [12]가 사용되었다. 이 시뮬레이터는 프로세서의 구조 레벨의 모델링 뿐 아니라 프로세서의 각 부분에서의 정적 그리고 동적인 에너지 소비도 예측할 수 있는 에너지 모델도 포함하고 있다. SVM기반 음성/음악 분류기 코드를 ARM instruction set architecture (ISA)로 컴파일 하기 위하여 gcc-2.95 기반의 교차컴파일러 (cross-compiler)를 사용하였고 시뮬레이션 되는 임베디드 프로세서의 구성은 표 1에 정리하였다.

SMV코덱에서 정하고 있는 인코딩시간의 제한시간 (allowable delay)은 한 프레임 당 20 ms이다 [1], 즉 이 제한시간 내에 음성/음악 분류를 포함, 인코딩에 필요한 모든 코드가 실행되어야 한다. 위의 프로세서 파라미터를 가정한 경우, 한 프레임을 인코딩에 걸리는 평균 시간은 5.68 ms로 측정됐다. 따라서 SVM기반의 음성/음악 분류는 14 ms 안에 수행되어야 한다. SMV의 음성/음악 분류는 SMV의 총 인코딩 시간의 0.5%를 차지하는데 반해, SVM기반 음성/음악 분류는 약 16.5 ms가 걸려 총 인코딩 시간의 74.4%를 차지한다. 이는 14 ms의 제한시간을 초과하는 것으로 SVM기반의 분류기는 최적화 없이는 실시간 요구조건을 만족하지 못하는 것을 알 수 있다. 참고적으로 가정된 프로세서로 실시간 조건을 만족시키려면 888 MHz의 동작속도가 필요하다. 따라서 본 논문에서는 SVM기반 분류기를 위한 최적화 기법을 제안함으로써 분류기의 실제 활용가능성을 높이고자 한다.

식 (2)에서 알 수 있듯이 서포트벡터의 개수와 차원은 총 계산량에 직접적인 관계를 가지고 있는데, 서포트벡

터의 개수와 차원을 증가시키며 계산량과 에너지 소비에의 영향을 조사해 보았다. 표 2는 서포트벡터의 개수를 5000에서 60000까지 변화시켰고 서포트벡터의 차원을 2에서 40까지 변화시키며 얻은 결과이다. 이 표는 서포트벡터의 개수가 1000씩 증가했을 때, 그리고 차원이 1씩 증가했을 때 증가된 평균 실행시간과 에너지소비량을 보여준다. 괄호 안의 숫자는 각 샘플데이터 (실행시간과 에너지소비의 증가량)의 표준편차를 나타낸다.

작은 표준편차는 동일한 개수와 차원의 증가량 (1000개, 1차원)에 대하여, 조사구간에서는, 비슷한 정도의 실행시간과 에너지소비의 증가가 이뤄졌음을 나타낸다. 표의 정보로 계산을 해보면, 서포트벡터의 차원을 1 올리는 것은, 실행시간을 기준으로 본 경우, 서포트벡터의 개수를 467개 늘리는 것과 같고, 에너지소비를 기준으로 본다면, 벡터의 개수를 293개 늘리는 것과 동일하다.

이제 SVM기반 음성/음악 분류기가 임베디드 프로세서에서 실행될 때 부동소수점 곱셈기의 개수와 연산수행 시간 (execution latency), 그리고 캐쉬의 크기에 어떻게 영향을 받는지 알아보도록 한다. 그림 1은 부동소수점 유닛의 개수와 성능 (연산수행 시간)에 따라 SVM기반 분류기의 한 프레임을 분류하는데 필요한 시간이 어떻게 변하는지 보여주고 있다. 이 경우 표 1의 프로세서 구성을 사용하였고 부동소수점 곱셈기의 개수와 연산수행 시간만을 변화시켰다. 프로세서의 issue width가 2 이므로 곱셈기를 두 개 이상으로 늘리는 것은 무의미 하다. 곱셈기의 개수와 성능에 따라 표 1의 구성을 사용한 경우보다 최대 14%까지 한 프레임을 분별하는데 걸리는 시간이 단축되었다. 그리고 곱셈기의 개수보다는 연산수행 시간이 실행시간을 줄이는 데 더 효과적이라는 것을 알 수 있다.

그림 2는 캐쉬 크기에 따른 실행시간의 변화를 나타낸다. 역시 표 1의 구성을 사용하였고 캐쉬의 크기만 변화시켰다. 역시 표 1의 구성을 사용한 경우보다 최대 11% 정도 성능이 향상되었다. 주목할 것은 캐쉬의 크기가 512 KB가 되었을 때 성능의 향상이 확실히 이루어지게 된 점이다. 이는 서포트벡터들이 캐쉬가 512 KB가 되어야 모두 저

표 1. 프로세서 시뮬레이터 파라미터  
Table 1. Parameters for the processor simulator.

ISA	ARMv4T
Clock Freq	800 Mhz
Pipeline	5 stages, in-order, dual issue
Branch	Dynamic prediction (bimodal)
Cache	64 KB separate Inst/Data cache,
	4 way, 32B line, 1 cycle latency
Memory	64 bit bus, 100 cycle latency, 1 port
FPU	1 adder (2 cycle, pipelined)
	1 Multiplier (4 cycle, pipelined)

표 2. 서포트벡터의 개수와 차원의 증가에 따른 한 프레임 당 실행 시간과 에너지소비의 증가량  
Table 2. Influence of the number of support vectors and their dimensionality on execution time and energy consumption per a frame.

증가량	증가된 실행시간 (ms)	증가된 에너지소비량 (mj)
1000 SVs	1.09 (8.10E-03)	3.80 (3.20E-03)
1 차원	0.51 (1.27E-02)	1.11 (8.49E-02)

장이 될 수 있다는 것을 의미한다. 캐쉬의 크기가 512 KB 보다 작은 경우 실행시간은 캐쉬 크기에 별 영향을 받지 않는데 이는 서포트벡터가 프로세서에서 읽히는 패턴을 생각해 보면 이해할 수 있다. 하나의 서포트벡터는 한 입력 프레임 벡터에 대해 한 번씩만 필요하게 되고 다음 입력 벡터를 위해 읽힐 때까지 그 서포트벡터를 제외한 모든 서포트벡터들이 프로세서에 의해 메모리로부터 읽히게 된다. 그러므로 한 서포트벡터가 프로세서로 불러들여져서 캐쉬에 저장되었다고 하더라도 캐쉬가 모든 서포트벡터를 저장할 만큼 크지 않다면 다음 입력 벡터를 위해 그 서포트벡터가 다시 읽힐 때는 이미 캐쉬에 존재하지 않을 것이다.

### III. 효율적인 SVM 구현을 위한 기법

이 장에서는 본 논문에서 제안된 두 가지 기법을 소개한다. 이 기법들은 SVM의 효율적인 구현을 위해 서포트벡터의 개수를 줄이고 입력벡터의 클래스를 부분적으로 근사화하는 실용적인 방법들이다.

#### 3.1. 기여도에 따른 서포트벡터 간략화 방법

전 장에서 살펴봤듯이 서포트벡터의 개수는 SVM기반 분류기의 실행속도와 에너지소비에 큰 영향을 미친다. 그러므로 서포트벡터의 개수를 줄임으로써 실행속도와 에너지소비를 크게 줄일 수 있다. 특히 제한된 연산능력과 전원을 가지는 임베디드 시스템에 있어서 서포트벡터의 개수의 영향은 더 크다고 할 수 있다. 따라서 많은 서포트벡터의 개수를 줄이기 위한 기법들이 제안되었다. Burges [8]는 적절한 분류성능을 유지하면서 적은 서포트벡터의 수로 구성되는 근사 초평면을 구하는 기법을 소개했다. Shen et. al. [9]은 초평면을 복잡하게 만드는 서포트벡터를 찾아내 제거한 후 재학습 시켜 분류성능과 서포트벡터의 수와의 적절한 균형을 찾는 기법을 제안했고, Ho [10]는 초평면 (hyperplane)에 적용되는 서포트벡터들의 힘의평형상태 (equilibrium)를 유지하도록 두 개의 서포트벡터를 하나의 서포트벡터로 대체하는 기법을 제안했다. Ayat et. al. [11]은 서포트벡터의 개수를 줄이고 분류성능을 높이기 위한 커널파라미터의 최적화 기법을 소개했다. 이 기법들 모두 목표는 같지만 다른 방법으로 목표를 달성했다. 본 논문에서는 각 서포트벡터의 기여도를 바탕으로 서포트벡터를 간략화 하는 새로운 방법을 제안한다.

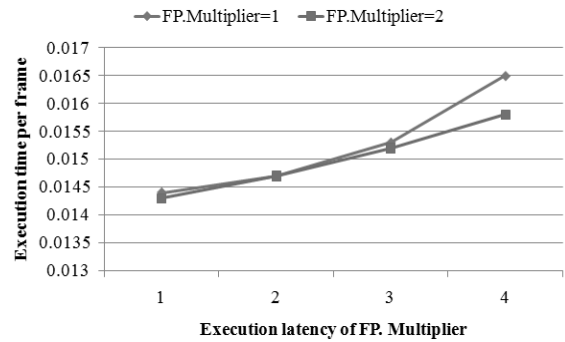


그림 1. 부동소수점 곱셈기의 개수와 성능에 따른 실행시간의 변화  
Fig. 1. Influence of the number of floating point multiplier and its latency on the execution time per frame.

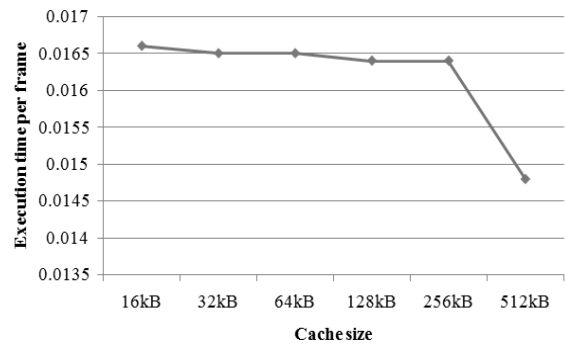


그림 2. 캐쉬 크기에 따른 실행시간의 변화  
Fig. 2. Influence of the cache size on the execution time per frame.

학습에 의해 구해진 서포트벡터들은 각기 다른 정도로 SVM의 판별에 영향을 주고 있다. 입력벡터에 따라서 기여도가 달라지기는 하지만 입력벡터에 관계없이 판별에 영향이 작은 서포트벡터들이 있다. 이런 기여도가 작은 서포트벡터들을 관찰 한 결과, 두 가지 특징이 있는 것을 발견했다. 첫째, 해당 라그랑제 승수의 크기가 작은 서포트벡터들은 판별에 기여도가 작다. 둘째, 자기 외에 모든 서포트벡터들과의 거리의 합이 아주 크면 기여도가 작다. 이는 다른 서포트벡터들과 많이 떨어져 있어서 입력 벡터와의 거리도 클 가능성이 많고, 거리가 크다면 식 (3)에 의해 RBF 커널함수의 결과값이 아주 작은 값이 되기 때문이다.

이 관찰을 바탕으로 한 기여도가 작은 서포트벡터를 제외시키는 판별식은 다음과 같다.

$$|\alpha_i| < \eta_\alpha \tag{4}$$

$$dist_i = \sum_{j \neq i}^M \|sv_j - sv_i\|^2 > \eta_{dist} \tag{5}$$

$\alpha_i$ 는 식 (2)에 있는 라그랑제 승수이고  $dist_i$ 는 모든 벡

터와의 거리의 제곱의 합이다.  $M$ 은 support vector의 개수이고  $\eta_a$ 와  $\eta_{disc}$ 는 각각 라그랑제 승수와 거리의 제곱의 합에 대한 문턱값이다. 두 가지 조건 중 하나만 충족하면 그 support vector는 제외된다. 실험에는  $\eta_a$ 은 0.05로  $\eta_{disc}$ 는 4.0E09로 정하여 사용하였다.

일반적으로 서포트벡터머신은 RBF를 커널함수로 사용한 경우, RBF의 폭 파라미터 (width parameter)  $\gamma$  (식 (3))에 따라 서포트벡터의 개수와 분류능력이 크게 변하게 되는데, 폭 파라미터의 변화에 따른 제안된 기법의 성능변화를 표 3에 나타내었다. ‘original’은 기법 적용 전을 나타내고 ‘pruned’는 기법 적용 후를 나타낸다. 그리고 ‘Num of SV’는 서포트벡터의 개수를 뜻하고 ‘Accuracy’는 음성/음악 분류정확도를 나타낸다. 기법 적용 전을 살펴보면  $\gamma$ 에 따라 분류정확도에는 별 차이가 없으나, 서포트벡터의 개수는 큰 차이가 있음을 알 수 있다. 제안된 기법을 적용한 경우, 서포트벡터의 개수가 많을 때, 더 많은 서포트벡터들이 제거됨을 볼 수 있다. 그러나 제안된 기법은 분류의 정확도에 큰 영향이 없음도 알 수 있다. 이 표는 또한 커널파라미터 최적화 기법 [11]과 제안된 기법이 동시에 적용될 수 있음을 보여준다.

### 3.2. 입력신호의 각 프레임간의 상관관계를 이용한 판별의 근사화 방법

식 (2)에서 알 수 있듯이 하나의 입력벡터  $X$ 에 대하여  $M$ 번의 커널함수의 계산이 이루어지고 커널함수의 계산은 식 (3)에서 보듯이 벡터간의 내적을 포함한다. 즉 하나의 입력벡터에 대하여 판별식의 계산을 생략하게 된다면 많은 계산량을 줄일 수 있다. 일반적인 음성코덱은 매 프레임 연산을 수행하도록 되어 있으나, 음성/음악 신호의 quasi-stationary 또는 stationary 성질을 이용하면 이웃 프레임들 간의 강한 상관관계를 고려해줄 수 있어서 판별식의 계산 대신 인접한 과거 프레임의 판별결과를 가지고 현재의 판별을 근사화 할 수 있다. 실험에서 사용되어진 음성/음악 신호는 음성구간, 음악구간, 그리고 무음구간

으로 구성되어 지는데, 각 구간은 어느 정도의 길이를 가지며 반복된다. 그러므로 각 구간은 실시간 분류를 위해 대체로 많은 수의 프레임으로 구성된다. 그러므로 주어진 구간 안에서는 강력한 상관성으로부터 다음 프레임이 현재 프레임과 같은 종류일 확률이 아주 높다고 할 수 있다. 예를 들어서 현재와 과거의 몇 입력 프레임이 음악 구간을 반영한다고 가정하면 다음프레임도 음악 프레임일 확률이 아주 높다는 것을 의미한다.

이런 상관관계를 이용하여, 모든 입력 프레임 벡터를 분류하지 않고, 현재의 분류 결과를 다음 입력 프레임들의 분류로 사용하는 것이 제안하는 방법의 핵심이다. 현재의 분류 결과를 가지고 몇 개의 입력 프레임을 분류의 정확도의 저하 없이 근사화할 수 있는지 알아보기 위하여 분류를 생략하는 프레임의 수를 변화시키며 전체적인 분류의 정확도와 생략되는 프레임의 전체 프레임 대비 비율을 구해 보았다. 그림 3은 그 결과를 보여주는데,  $X$ 축은 하나의 분류를 가지고 대체하는 (생략하는) 입력 프레임의 개수를 나타내고,  $Y$ 축은 분류의 정확도와 분류가 생략된 입력 프레임의 전체 입력 프레임에 대한 백분율을 나타낸다. 하나의 분류 결과를 가지고 대체되는 프레임의 개수가 많아질수록 분류의 정확도는 천천히 저하되는 것을 볼 수가 있다. 그러나 판별이 생략되는 입력 프레임의 비율은 초반에 급격하게 늘어나서 하나의 분류결과에 의해 생략되는 개수가 10개정도부터는 포화되는 것을 볼 수 있다. 즉 판별의 정확도를 유지하면서 생략되는 프레임으로부터 오는 이득을 취하려면 하나의 분류에 의해 생략되는 프레임의 수는 4~8 정도가 적당하다고 보인다.

본 실험에서 사용된 테스트 데이터는 다음 장에 자세히

표 3. 폭 파라미터의 변화에 따른 제안된 기법의 성능변화  
Table 3. Influence of the RBF kernel width parameter on the number of support vectors and classification accuracy.

	Num of SV	Accuracy
$\gamma=0.1$ original	24775	0.852
$\gamma=0.1$ pruned	12318	0.851
$\gamma=0.01$ original	14731	0.847
$\gamma=0.01$ pruned	12350	0.845

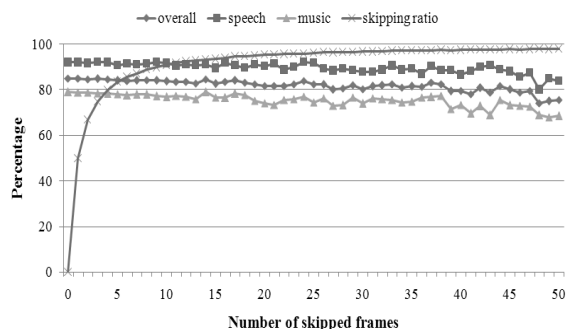


그림 3. 하나의 분류결과에 따라 근사화된 입력벡터의 개수에 따른 음성/음악 분류 정확도와 생략된 총 입력벡터의 전체 입력벡터와의 백분율

Fig. 3. Impact on classification accuracy and the percentage between the number of skipped input frames and the total number of input frames by the number of approximated input frames to the most immediately preceding classification result.

설명이 나오겠지만, 한 세그먼트에 250 프레임 (5초)을 포함하고 있다. 만약 한 세그먼트에 속한 프레임의 개수가 줄어든다면 그림 3에 나온 것 보다 더 빨리 분류의 정확도가 떨어질 것이다. 그러나 음성/음악 신호에서 각 세그먼트의 길이는 5초 이상인 경우가 많으므로 제안된 기법은 일반적인 음성/음악 신호에 적용 가능하다고 볼 수 있다.

### IV. 실험

#### 4.1. 실험 설정

본 논문에서 사용된 음성/음악 데이터를 구성하기 위하여 음성 데이터베이스로 8 kHz로 샘플링 된 약 6 sec 정도의 깨끗한 음성으로 326명의 남자와 138명의 여자 화자에 의해서 화자마다 10개의 파일이 발음된 TIMIT 데이터베이스가 사용되었다 [13]. 음악 데이터베이스는 CD로부터 다섯 가지 장르의 음악을 모바일 폰을 통해서 녹음하였고, 8 kHz로 다운 샘플링 하여 사용하였으며, 각기 약 5분 정도의 길이를 가진다. 학습을 위해서 음성과 음악 데이터베이스에서 20분 정도의 데이터를 선별하여 사용하였으며, 테스트를 위해서는 학습에 사용되지 않은 데이터로 각 장르별로 2 개씩 모두 10개의 파일을 만들었다. 한 개의 파일에는 음성구간 20개 그리고 음악구간 20개가 포함되어 있으며 각 구간은 5초의 길이를 가진다. 음성과 음악구간은 서로 교대로 나타나게 구성되었다.

실험에 사용된 특징벡터로는 SMV [1]에서 추출되어지는 6개의 파라미터로 구성해 사용하였다 [5]. SVM을 학습시키고 테스트 할 때 기본으로 사용되는 커널 width 파라미터  $\gamma$ 를 구하기 위해 여러 값을 시도하였고 그 중 가장 좋은 분류성능과 가장 적은 수의 서포트벡터를 필요로 하는 0.01이 사용되었다.

SVM기반 음성/음악 분류기의 분류성능, 실행시간, 그리고 에너지소비를 구하기 위하여 프로세서 시뮬레이터인 sim-analyzer [12]가 사용되었다. 시뮬레이션 되는 임베디드 프로세서의 구성은 표 1에 정리하였다.

#### 4.2. 실험 결과

제안된 알고리즘을 검증하기 위하여 알고리즘을 적용한 경우와 적용하지 않은 경우의 분류정확도, 실행시간, 그리고 에너지 소비를 비교해 보았다. 그림 4는 그 결과를 보여주는데 10개의 테스트 파일에서 나온 결과들의 평균값이다. X축은 사용된 기법과 파라미터를 나타낸다.

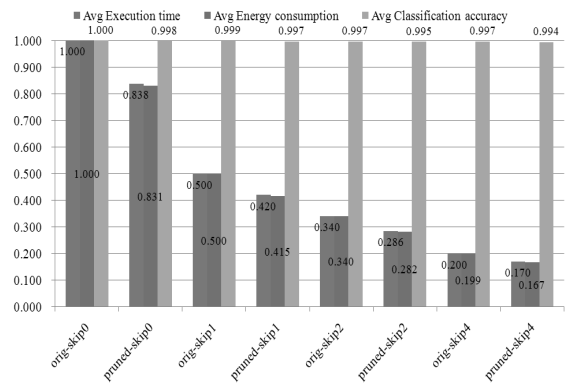


그림 4. 제안된 기법에 의한 실행시간, 에너지소비, 분류정확도의 변화

Fig. 4. Effectiveness of the proposed techniques in terms of execution time, energy consumption, and classification accuracy.

X축의 레이블은 'A-B' 형식으로 표기되었는데 'A'는 SVM의 개수를 줄이는 기법에 해당하는 것으로 'orig'는 이 기법이 사용되지 않은 경우를 뜻하고 'pruned'는 이 기법이 사용된 것을 말한다. 'B'는 입력 프레임을 처리를 생략하는 기법으로 'skip' 다음에 나오는 숫자는 생략된 프레임의 개수를 나타낸다. 즉 'skip0'는 생략된 프레임이 없는 경우, 즉 이 기법이 사용되지 않은 경우를 말하고, 'skip2'는 두 개의 프레임의 판별을 생략하고 가장 근접한 과거 프레임의 판별로 대신한 경우를 나타낸다. X축의 한 레이블 당 세개의 막대가 있는데 각각 한 개의 프레임을 판별하는데 필요한 시간, 에너지, 그리고 판별 정확도이다. 표시된 값들은 제안된 두 가지의 기법이 사용되지 않은 경우 ('orig-skip0')의 값으로 정규화 되었다.

그림에서 제안된 기법은 분류의 정확도에의 영향은 최소화 하면서 효율적으로 실행시간과 에너지소비를 줄일 수 있음을 알 수 있다. 예를 들어 가장 실행시간과 에너지 소비가 적고 동시에 분류 정확도가 가장 낮은 'pruned-skip4'의 경우, 분류정확도의 감소는 0.6%에 불과하다. 또한 실행시간과 에너지소비의 감소는 제거된 서포트벡터의 개수와 생략된 입력벡터의 개수 선형적으로 비례함을 알 수 있다. 입력 벡터의 판별을 생략하고 서포트벡터의 개수를 줄이는 것은 서로 독립적이므로 두 기법에 의해 감소되는 실행시간과 에너지는 두 기법을 모두 사용하였을 때 한 기법만을 사용했을 때보다 늘어나는 것도 알 수 있다.

2장에서 언급되었듯이 SVM기반 분류기는 가정된 임베디드 시스템에서 아무런 최적화기법 없이 사용되면 SMV의 제한시간을 만족시키지 못한다. 그러나 서포트벡터의 간략화 기법만 사용해도 ('pruned-skip0'), 분류기의 한

프레임당 실행시간이 13.8 ms이 되고 총 19.5 ms의 인코딩 시간이 소요되어 20 ms의 조건을 만족시킨다. 또한, 제한된 전원을 가진 임베디드 시스템에서는 필수적인 에너지소비의 감소도 얻어진다.

## V. 결론

본 논문에서는 SVM기반 음성/음악 분류기의 임베디드 시스템으로의 구현을 주제로 실행시간과 에너지소비를 줄이는 기법을 제안하였고, SVM의 주요 파라미터인 서포트벡터의 개수와 차원이 실행시간과 에너지소비에 미치는 영향을 분석하였다. 그리고 SVM기반 분류기 알고리즘이 임베디드 프로세서에서 실행될 때 부동소수점 곱셈기의 개수와 성능, 그리고 캐시의 크기에 어떻게 영향을 받는지 분석하였다. 또한 제안된 기법은 SVM기반 분류기의 실행시간과 에너지소비를 효과적으로 감소시킬 수 있음을 보였다.

앞으로의 연구과제로는 알고리즘 차원의 기법 뿐 아니라 코드 차원이나 프로세서 구조 차원의 기법을 개발, 적용함으로써 SVM기반 분류기의 보다 효율적인 구현을 가능하게 하는 것이다.

## 감사의 글

본 연구는 지식경제부 및 한국산업기술평가관리원의 IT핵심기술개발사업의 일환으로 수행하였음. [2009-S-036-1, 고성능 가상머신 규격 및 기술 개발] 또한 이 논문은 2011년도 정부 (교육과학기술부)의 재원으로 한국연구재단의 대학중점연구소 지원사업으로 수행된 연구임 (2011-0022980).

## 참고 문헌

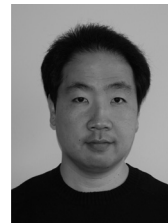
1. 3GPP2 Spec., "Source-controlled variable-rate multimedia wideband speech codec (VMR-WB), service option 62 and 63 for spread spectrum systems," 3GPP2-C.S0052-A, vol. 1.0, Apr. 2005.
2. Y. Gao, E. Shlomot, A. Benyassine, J. Hyssen, Huan-yu Su, and C. Murgia, "The SMV algorithm selected by TIA and 3GPP2 for CDMA applications," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 709-712, 2001.
3. A. Bugatti, A. Flammini, and P. Migliorati, "Audio classification in speech and music: a comparison between statistical and a neural approach," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 4, pp. 372-378, 2002.
4. J. Saunders, "Real-time discrimination of broadcast speech/

musicspeech/music," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 7-10, 1996.

5. S. -K. Kim and J. -H. Chang, "Speech/music classification enhancement for 3GPP2 SMV codec based on support vector machine," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E92-A, no. 2, 2009.
6. S. -K. Kim and J. -H. Chang, "Discriminative weight training for support vector machine-based speech/music classification in 3GPP2 SMV codec," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol. E93-A, no. 1, pp. 316-319, 2010.
7. H. Lee and J. Jeong, "Early termination scheme for binary block motion estimation," *IEEE Trans. Consumer Electronics*, vol. 53, no. 4, pp. 1682-1686, 2007.
8. C. Burges, "Simplified support vector decision rules," in *Proc. International Conference on Machine Learning*, pp. 71-77, 1996.
9. Y. Zhan and D. Shen, "Design efficient support vector machine for fast classification," *Pattern Recognition*, vol. 38, no. 1, pp. 157-161, 2005.
10. T. Ho, "An efficient method for simplifying support vector machines," in *Proc. International Conference on Machine Learning*, pp. 617-624, 2005.
11. N. E. Ayat, M. Cheriet, and C. Y. Suen, "Automatic model selection for the optimization of SVM kernel," *Pattern Recognition*, vol. 38, no. 10, pp. 1733-1745, 2005.
12. T. Austin, T. Mudge, and D. Grunwald, Sim-analyzer. <http://www.eecs.umich.edu/~panalyzer/>
13. W. M. Fisher, G. R. Doddington and K. M. Goude-Marshall, "The DARPA speech recognition research database: Specifications and status," in *Proc. DARPA Workshop Speech Recognition*, pp. 93-99, 1986.

## 저자 약력

### •임 정 수 (Chungsoo Lim)



1996년: 인하대학교 전기공학 학사  
2004년: 매릴랜드 주립대학 전자공학 석사  
2009년: 노스캐롤라이나 주립대학 컴퓨터공학 박사  
2010년: 인하대학교 박사후 연구원  
2011년: 목포대학교 연구교수

### •장 준 혁 (Joon-Hyuk Chang)



1998년 2월: 경북대학교 전자공학과 학사  
2000년 2월: 서울대학교 전기공학부 석사  
2004년 2월: 서울대학교 전기컴퓨터공학부 박사  
2000년 3월 ~ 2005년 4월: ㈜넷디스 연구소장  
2004년 5월 ~ 2005년 4월: 캘리포니아 주립대학, 산타바바라 (UCSB) 박사후연구원  
2005년 5월 ~ 2005년 8월: 한국과학기술연구원 (KIST) 연구원  
2005년 9월 ~ 2011년: 인하대학교 전자공학부 조교수  
2011년 ~ 현재: 한양대학교 융합전자공학부 부교수