Journal of the Korean Institute of Industrial Engineers
Vol. 40, No. 4, pp. 404-414, August 2014.
ISSN 1225-0988 | EISSN 2234-6457
http://dx.doi.org/10.7232/JKIIE.2014.40.4.404

<Original Research Paper>

MCSVM을 이용한 반도체 공정데이터의 과소 추출 기법

박새롬 · 김준석† · 박정술 · 박승환 · 백준걸

고려대학교 산업경영공학과

Under Sampling for Imbalanced Data using Minor Class based SVM (MCSVM) in Semiconductor Process

Sae-Rom Pak · Jun Seok Kim · Cheong-Sool Park · Seung Hwan Park · Jun-Geol Baek

School of Industrial Management Engineering, Korea University

Yield prediction is important to manage semiconductor quality. Many researches with machine learning algorithms such as SVM (support vector machine) are conducted to predict yield precisely. However, yield prediction using SVM is hard because extremely imbalanced and big data are generated by final test procedure in semiconductor manufacturing process. Using SVM algorithm with imbalanced data sometimes cause unnecessary support vectors from major class because of unselected support vectors from minor class. So, decision boundary at target class can be overwhelmed by effect of observations in major class. For this reason, we propose a under-sampling method with minor class based SVM (MCSVM) which overcomes the limitations of ordinary SVM algorithm. MCSVM constructs the model that fixes some of data from minor class as support vectors, and they can be good samples representing the nature of target class. Several experimental studies with using the data sets from UCI and real manufacturing process represent that our proposed method performs better than existing sampling methods.

Keywords: Imbalanced Data, Under-Sampling, MCSVM, Support Vectors, Semiconductor Process

1. 서 론

아날로그 시대에서 최첨단 디지털 시대로 변화하면서 휴대전화를 비롯한 가전제품, 컴퓨터, 제조장비 등 대부분 산업에서는 가장 핵심 부품인 반도체가 필요하다. 이와 함께 반도체 공정은 급속도로 발전하였으며 반도체의 수요 역시 지속해서 증가하고 있는 추세이다(Kim *et al*., 1998). 수요의 증가로 반도체 제조 회사들은 반도체 시장에서의 경쟁력 확보와 제품의 품질 향상을 위해 생산주기, 재작업율, 수율 등 품질 성능지표를 관리하고 있다(Kymal and Patiyasevi, 2006; Kim, 2010; Kim *et al*., 2014). 특히, 반도체 공정은 복잡한 단계를 거치기 때문에 생산과정 중에 발생하는 설비의 고장이나 공정의 변동 등 여러 가지 외부 요인이 수율에 영향을 준다. 그러므로 안정적으로 수율을 관리하고 예측하는 것은 반도체 산업에서 경쟁력을 확보하기 위해 반드시 필요하다(Baek and Han, 2003). 수율예측을 위한 노력으로 Ciciani and Iazeolla(1991)는 불량인 칩(Chip)에 통계적인 분포를 사용하여 단변량 예측모델을 만들어 수율을 예측하는 연구를 수행하였다. 또한, Crosier(1988)는 한 개의 변수가 아닌 여러 변수를 사용하여 수율을 예측하는 다변량 통계 기법을 연구하였다. 그러나 반도체 공정 과정에서 발생하는 데이터는 빅 데이터(Big data)의 형태를 보이며 많은 변수가 존재한다. 따라서 수율 관리를 위해 모든 변수를 통계적 기법으로만 분석하는 것은 많은 시간이 소요되기 때문에(Baek and Han, 2003; An *et al*., 2009) 여러 변수의 성질을 고려하여 학습할 수 있는 기계 학습 알고리즘 기법(Machine learning algorithms)을 사용하여 수율을 예측하는 연구가 진행되었다.

이 논문은 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업이며(NRF-2013R1A1A2010019), BK21 플러스 사업(고려대학교, 제조·물류분야에서의 빅 데이터 운용 사업팀)으로 지원된 연구임.
† 연락저자 : 김준석, 136-701 서울시 성북구 안암동 5가 1번지 고려대학교 산업경영공학과, Tel : +82-2-3290-4802, Fax : +82-2-925-5035, E-mail : bliths@korea.ac.kr
2014년 4월 28일 접수; 2014년 5월 28일 수정본 접수; 2014년 5월 29일 게재 확정.

Jang and Bae(2009)는 반도체 제조 공정에서 투입 변수를 관리하기 위해 1차 도함수(First derivative)를 계산한 후, SVM을 적용하는 하이브리드 데이터 마이닝 기법을 제시하였다. An *et al.*(2009)의 연구는 양품(Pass)과 불량(Fault) 사이의 오분류를 최소화하기 위해 단계적으로 SVM(Support vector machine)을 적용할 수 있는 SSVM(Stepwise-support vector machine)을 제시했다. 또한, Li and Huang(2009)는 SOM(Self-organizing map)과 SVM을 사용하여 웨이퍼 빈 맵(Wafer bin map)을 군집화하고 패턴을 찾아내어 수율을 예측하였다.

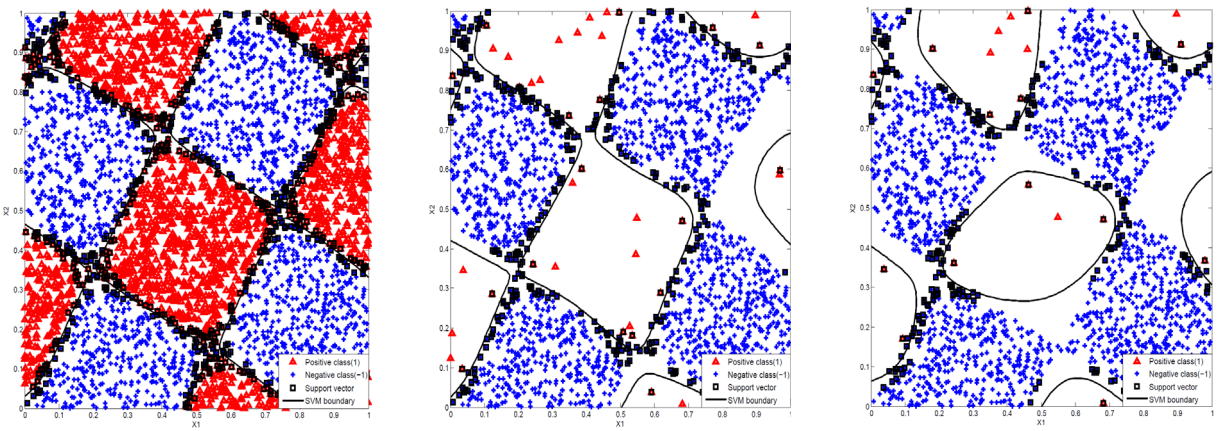
위에서 언급한 연구들은 불량과 양품의 비율이 균형 잡힌 데이터를 이용해 수행되었다. 하지만 반도체 제조공정의 최종 검사 단계에서는 98~99%가 양품인 불균형 데이터(Imbalanced data)가 수집된다. 일반적으로 두 범주(Class)의 불균형 비율이 심할수록 SVM과 대부분의 기계 학습 알고리즘을 적용할 경우 다수 범주와 소수 범주의 비율 차이 때문에 성능의 저하가 발생한다. 즉, 불균형 문제를 해소하지 않고 SVM을 학습할 경우 다수 범주의 데이터들이 소수 범주를 나누는 분류 경계(Decision boundary)가 소수 범주의 영역을 침범하기 때문에 오분류가 발생하는 가능성이 높아지게 된다(Kang and Cho, 2006; Kim, 2012).

<Figure 1>은 두 개의 범주에서 비율에 따른 SVM 학습 후의 분류 경계선의 형태를 나타낸다. <Figure 1>의 (a)는 두 개의 범주의 비율이 동등할 때 SVM 수행 후 나타나는 분류 경계선이다. (b)는 두 개의 범주의 비율이 1:50인 불균형 데이터의 분류 경계선을 나타내고 있다. 마지막 (c)는 두 개의 범주의 비율이 1:100인 불균형 데이터를 SVM에 적용하였을 경우의 분류 경계선의 모습을 나타내고 있다. (a)와 같이 1:1비율로 구성된 데이터에서 SVM 학습 시 생성되는 두 범주의 경계는 원래 경계와 비슷한 곳에 위치한다. 하지만 불균형 데이터로 구성된 (b)의 경우 소수 범주에서 지지 벡터(Support vector)로 선정되는 데이터가 줄어들면서 다수 범주 분류 경계선의 영역이 확대되어 소수 범주의 영역을 침범하게 된다. 특히, (c)와 같이 극심한 불균형을 이루는 경우에는 그 현상이 더욱 심해진다(Wu and Chang, 2003; Kang and Cho, 2006; Kim, 2012).

이러한 극심한 불균형 데이터에 의해 발생하는 문제를 해결하기 위해 크게 과대 표본 추출(Over-sampling)법과 과소 추출(Under-sampling)법 등이 제안되었다. 과대 표본 추출법은 일정한 규칙에 의해 소수 범주의 데이터를 반복적으로 복원 추출하여 다수 범주의 비율을 일정 비율로 맞추어 표본 추출하는 것을 말한다. 이 방법은 모든 데이터의 정보를 이용할 수 있지만, 소수 범주의 데이터가 중복으로 추출되어 데이터의 수가 증가하기 때문에 학습시간이 길어지는 문제가 있다(Kang and Cho, 2006; Kim *et al.*, 2012). 또한, 방법론에 따라 임의로 합성된 데이터가 생성되기 때문에 과잉일반화(Overgeneralization) 문제가 발생할 수 있다(Yen and Lee, 2009).

과소 표본 추출법은 다수 범주의 데이터를 소수 범주의 데이터 수에 비례해 표본 추출하는 방법이다. 이 방법은 다수 범주의 데이터 손실이 존재하지만, 데이터의 수가 감소하므로 학습속도가 빨라지는 장점이 있으며, 소수 범주에 대한 반복 추출로 발생하는 중복 현상을 막을 수 있다(Chawla *et al.*, 2011). 또한, 일반적으로 과대 표본 추출을 이용한 기법들의 성능은 과소 표본 추출법을 통한 기법들에 비해 좋지 않다(Yen and Lee, 2009).

Kang and Cho(2006)는 불균형 문제를 해결하기 위해 과소 표본 추출 기반 앙상블 SVM을 사용하여 성능을 평가하였다. Kang and Cho(2006)의 연구는 불균형 데이터를 SVM에 적용 시 발생하는 분류 경계선의 왜곡 및 소수 범주의 분류 성능이 저하되는 문제들을 극복하였다. 그러나 반도체 공정의 최종 검사 단계에서는 많은 양의 웨이퍼가 검사되기 때문에 웨이퍼별로 최소 1 : 50 이상의 극심한 불균형 데이터가 생성된다. 이 방법을 적용하게 되면 각 웨이퍼 별로 최소 50개 이상의 SVM 모델을 만들어 학습하고 앙상블을 취합해야 하기 때문에 신속히 수율을 예측해야 하는 반도체 공정에는 적용하기 어렵다. 또한, 앙상블 학습을 할 경우 웨이퍼별로 가지고 있는 특징들



(a) 1 : 1 (b) 1 : 50 (c) 1 : 100

Figure 1. Class boundary using SVM algorithm according to the number of sample data for major class

이 요약되어 유실되므로 성능 저하 문제가 발생할 수 있다. 따라서 웨이퍼가 가지고 있는 특징을 고려한 웨이퍼별 과소 표본 추출법이 필요하다.

웨이퍼별 특징을 고려한 표본 추출을 위해 본 논문에서는 SVM 학습 후 서로 다른 범주의 데이터를 가장 잘 나눌 수 있는 지지 벡터만을 사용한 과소 표본 추출법을 제안한다. 지지 벡터는 각 범주를 구분하는 경계를 결정하는 역할을 하므로 소수 범주와 다수 범주를 대표하는 표본으로 생각할 수 있다. 하지만 불균형 데이터는 다수 범주 데이터가 소수 범주의 데이터보다 많기 때문에 SVM 학습 시 다수 범주와 소수 범주의 지지 벡터가 제대로 잡히지 않아 분류 경계의 왜곡이 발생한다(Kang and Cho, 2006; Kim *et al*., 2012). 따라서 기존 SVM의 학습을 통해 산출된 지지 벡터는 표본으로 사용하기에 어려움이 있으므로 소수 범주 데이터를 고려한 SVM 학습이 필요하다. 본 논문에서는 기존 SVM과 다르게 소수 범주를 미리 지지 벡터로 고정할 수 있는 MCSVM(Minor class based SVM) 방법을 제안하며, 이를 이용하여 소수 범주와 함께 과소 표본 추출된 다수 범주의 지지 벡터를 표본으로 사용한다.

가장 기본적이면서 많이 사용되는 표본 추출 기법은 무작위 추출 접근법(Random under-sampling approach)으로써 무작위 표본 추출을 통해 다수 범주와 소수 범주의 비율을 조절하게 된다(Yen and Lee, 2009). 그 외에도 거리 기반의 지표를 통한 표본 추출법(Chyi, 2003)에 대한 연구가 수행되었지만, 이 경우 표본을 추출하는데 소비하는 시간이 많고 다수 범주의 데이터가 표본으로 중복 추출될 수 있는 단점이 있다(Yen and Lee, 2009). 따라서 본 연구에서는 1) 표본 추출법을 사용하지 않는 경우, 2) 무작위 추출 접근법을 통해 다수 범주와 소수 범주의 비율을 MCSVM과 동일하게 조절한 경우, 3) 소수 범주는 전부 추출하고, 다수 범주의 데이터에서만 무작위 표본 추출을 수행하여 MCSVM과 맞춘 경우, 4) 소수 범주의 비율만큼 다수 범주 데이터를 무작위 추출한 경우 및 5) 기존 SVM을 통한 지지 벡터 추출 방법을 비교 대안으로 구성한다.

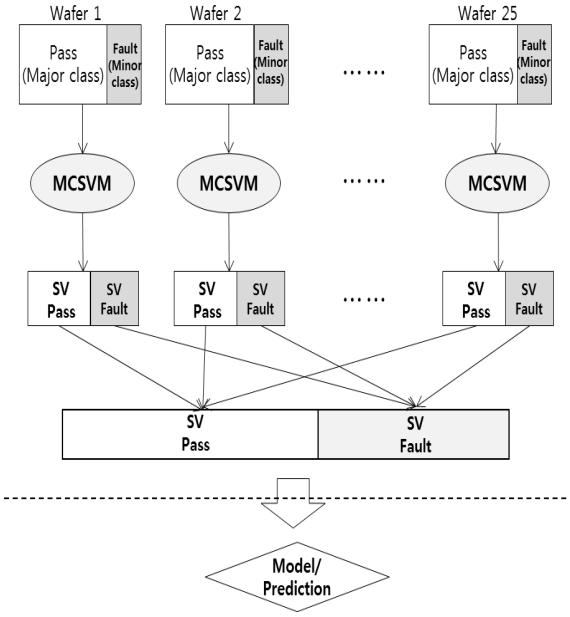


Figure 2. Framework of proposed method using MCSVM based under-sampling

본 논문의 구성은 다음과 같다. 제 2장에서는 반도체 공정에서 발생하는 데이터에 MCSVM을 적용하기 위한 절차와 과소 표본 추출의 과정을 설명하고, 기존의 SVM 알고리즘 및 MCSVM 알고리즘을 기술한다. 제 3장에서는 실험 데이터와 실험 설계, 실험 결과에 대해 설명한다. 마지막으로 제 4장에서는 본 논문에서 제안한 방법과 실험 결과를 바탕으로 결론 및 추후 연구의 내용을 제시한다.

2. MCSVM (Miner class based SVM)

MCSVM은 불균형 데이터를 SVM에 적용할 때 다수 범주로 인해 소수 범주의 경계가 침범되는 문제점을 보완하는 방법이다. 본 연구는 반도체 공정 최종 검사 단계에서 웨이퍼 단위로 발생하는 극심한 불균형 데이터에 MCSVM을 적용하여 과소 표본을 추출함으로써 수율 예측 성능의 향상을 목적으로 한다.

반도체 공정에서 발생하는 데이터는 크게 로트(Lot) 단위로 생산되며 총 25장의 웨이퍼로 구성된다. 각 웨이퍼는 일반적으로 1,700여 개의 칩으로 구성되며 이 중에서 약 20개는 불량이고 나머지는 양품이다. 따라서 반도체 웨이퍼에서 생성되는 데이터는 약 1 : 45 이상의 극심한 불균형의 특징을 가지고 있다. 따라서 대부분 양품으로 구성되어 있는 반도체 데이터를 SVM에 적용했을 때 발생하는 문제점을 해결하기 위해 소수 범주의 경계 왜곡을 방지할 수 있는 MCSVM 기반 과소 표본 추출법을 사용한다. <Figure 2>는 반도체 데이터에 적용 가능한 MCSVM 기반 과소 표본 추출법의 제안 절차이다. 웨이퍼별로 MCSVM을 적용하여 지지 벡터로 선정된 소수 범주 전체와 이에 대응되어 선정된 다수 범주의 지지 벡터를 과소 표본으로 추출한다. 이후 추출된 표본들을 하나의 데이터셋으로 구성한다. 이렇게 구성된 데이터셋을 이용해 모델을 만들고 최종 수율을 예측한다.

본 논문에서 제안하고 있는 MCSVM은 소수 범주의 데이터를 처음부터 지지 벡터로 고정하는 제약조건을 추가한 SVM이다. 미리 소수 범주를 지지 벡터로 고정하고 이에 대응되는 다수 범주의 지지 벡터들이 정해지기 때문에 기존의 불균형 데이터에서 나타나는 분류 경계에 대한 왜곡이 줄어든다. 경계선 왜곡의 감소로 인해 지지 벡터는 전체 데이터를 대표하는 표본 추출 데이터로 사용할 수 있다.

2.1 Support Vector Machine (SVM)

SVM은 입력되는 데이터를 최적의 초평면(Hyperplane)을 사용하여 두 범주 사이의 마진(Margin)을 최대화시켜 데이터를

분류하는 기계 학습 알고리즘의 하나이다(Cortes and Vapnik, 1995). 마진을 최대화시키기 위해서 식 (1), 식 (2)를 사용한다. 이때 식 (1), 식 (2)에서 나타내고 있는 y_i는 데이터 X_i에 해당하는 레이블(Label)이며, $\mathbb{W}$와 b는 각각 초평면의 가중치(Weight), 절편(Bias)을 나타낸다(Han, 2009).

$$\min\ L(\mathbb{W}) = \frac{\| \mathbb{W} \|^2}{2} \tag{1}$$

$$st. \quad y_i(\mathbb{W}^T X_i + b) \geq 1,\ \ i = 1,\ 2,\ \cdots,\ N \tag{2}$$

하지만 <Figure 3>에서 보이는 데이터는 완전하게 분류할 수 없으므로 <Figure 3>과 같이 여유 변수(Slack variable) 'ξ'를 제약조건에 추가한다.

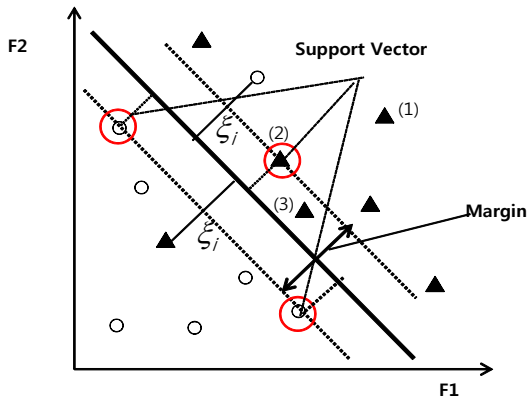
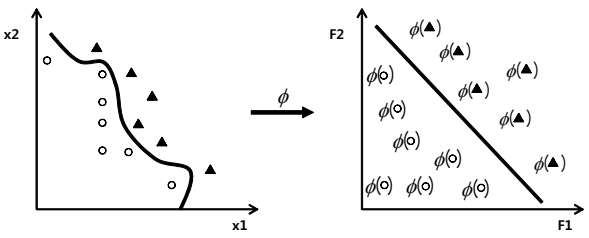


Figure 3. Support vector machine with soft margin

식 (3), 식 (4)는 식 (1)의 목적식과 식 (2)의 제약조건에서 여유 변수 'ξ'가 추가 되었으며 이것은 오분류를 허용할 수 있음을 나타낸다. C(Cost or Penalty)값은 오분류를 상충(Trade-off)하는 파라미터(Parameter)이다.

$$\min\ L(\mathbb{W},\ \xi) = \frac{1}{2} \| \mathbb{W} \|^2 + C\sum_{i=1}^{N} \xi_i \tag{3}$$

$$st. \quad \begin{cases} y_i(\mathbb{W}^T X_i + b) \geq 1 - \xi_i, & \forall_i \\ \xi_i \geq 0, & \forall_i \end{cases} \tag{4}$$



(a) Real space (b) Kernel space

Figure 4. Mapping non-linear data into kernel space

또한 <Figure 4>의 (a)와 같이 선형으로 분리할 수 없는 데이터는 그림 (b)와 같이 비선형 데이터에 커널(Kernel) 함수를 적용할 수 있다(Cristianini and Shawe-Taylor, 2000). 커널 함수는 1차원 데이터를 고차원의 공간(Kernel space)으로 사영시켜 비선형 문제를 풀 수 있다. 가장 기본적인 커널함수$K(\bullet)$는 다음 식 (5)에서 내적의 형태로 나타낼 수 있다.

$$K(\bullet) = <\Phi(X^T),\ \Phi(X)> \tag{5}$$

커널 함수는 여러 종류의 형태로 <Table 1>에서 나타나고 있지만, 그중에서도 일반적으로 가우시안 RBF(Radial basis function) 커널을 사용한다. 본 논문에서는 반도체 데이터의 특성상 오분류를 최소화하기 위해 적합한 RBF 커널식을 사용하였다(An *et al.*, 2009).

Table 1. Kernel functions

Type	Kernel functions	Parameters
RBF	$K(\bullet) = \exp(-\sigma \| X - X^T \|^2)$	σ
Laplacian	$K(\bullet) = \exp(- \| X - X^T \| / \sigma)$	σ
Hyperbolic Tangent (Sigmoid)	$K(\bullet) = \tanh(\alpha < x,\ x' > + c)$	$\alpha,\ c$
Polynomial	$K(\bullet) = (< X,\ X^T > + 1)^p$	p

식 (6)은 식 (3), 식 (4)에 커널을 적용한 후 라그랑지(Lagrange) 승수를 적용한 목적함수를 나타낸 것이다. 식 (6)은 KKT(Karush-Kuhn-Tucker) 조건을 만족해야 하며 이차계획법(Quadratic programming)으로 풀 수 있다.

$$\max\ Q(\alpha) = \sum \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(X_i,\ X_j) \tag{6}$$

$$st. \quad \sum_{i=1}^{N} \alpha_i y_i = 0, \tag{7}$$

$$0 \leq \alpha_i \leq C,\ \ i = 1,\ 2,\ \cdots,\ N \tag{8}$$

식 (7), 식 (8)은 식 (6)의 목적식에 따른 제약식이다. 이때 라그랑지 승수(Lagrange multiplier) α_i의 범위는 항상 0보다 크거나 같고 C값보다는 작거나 같다는 조건의 제약식을 만족해야 한다. <Figure 3>에서 (2)에 위치한 데이터는 마진 경계선에 위치하는 지지 벡터이며 이것의 조건 범위는 $0 < \alpha_i < C$를 만족시킨다. 만약 $\alpha_i = 0$이면 <Figure 3>의 (1)을 나타내고 있는 데이터로써 정상적으로 분류된 데이터를 나타낸다. 또한, $\alpha_i = C$이면 <Figure 3>의 (3)에서 확인할 수 있듯이 마진의 경계선 안쪽에 벡터들이 위치함을 알 수 있다(Schölkopf and Smola, 2002; Shin and Cho, 2006).

최종적으로 라그랑지 승수 α_i의 해는 초평면의 관계식에 대

입하여 계산할 수 있다. 그 결과 분류 경계 및 분류 함수(Decision function)를 구분하는 함수 $f(X)$는 식 (9), 식 (10)과 같이 나타낼 수 있다.

$$f(X) = \sum \alpha_i y_i K(X,\ X^T) + b, \tag{9}$$

$$b = \frac{i - \xi_i}{y_i} - \mathbb{W}^T X_i,\ (\text{if}\ \ y_i = 1) \tag{10}$$

하지만 극심한 불균형 데이터에 기존 SVM을 적용할 경우 <Figure 1>의 (c)처럼 다수 범주에 의해 소수 범주의 경계가 왜곡되므로 분류 성능이 저하된다(Kang and Cho, 2006). 소수 범주가 왜곡되는 현상을 방지하기 위해 본 논문은 제 2.2절의 MCSVM을 제안한다.

MCSVM은 소수 범주의 왜곡을 줄여 소수 범주의 데이터를 미리 지지 벡터로써 고정되도록 제약조건을 추가하는 방법이다. 소수 범주를 미리 지지 벡터로 고정하므로 소수 범주에 대응하는 다수 범주의 지지 벡터를 표본 추출 데이터로 활용할 수 있다.

2.2 Under Sampling Method using MCSVM

MCSVM은 불균형 데이터에서 분류 경계의 왜곡 현상을 줄이기 위해 기존 SVM 알고리즘에 소수 범주를 지지 벡터로 고정될 수 있도록 제약조건을 수정한 알고리즘이다. 지지 벡터가 되기 위해서는 α_i값이 0보다는 크고 C값보다는 작아야하므로 소수 범주를 지지 벡터로 고정하기 위해 기존 SVM의 제약조건인 식 (8)에서 라그랑지 승수 α_i를 다수 범주($y_i = -1$)와 소수 범주($y_i = 1$)인 경우로 구분하여 식 (11), 식 (12)와 같이 나타낸다.

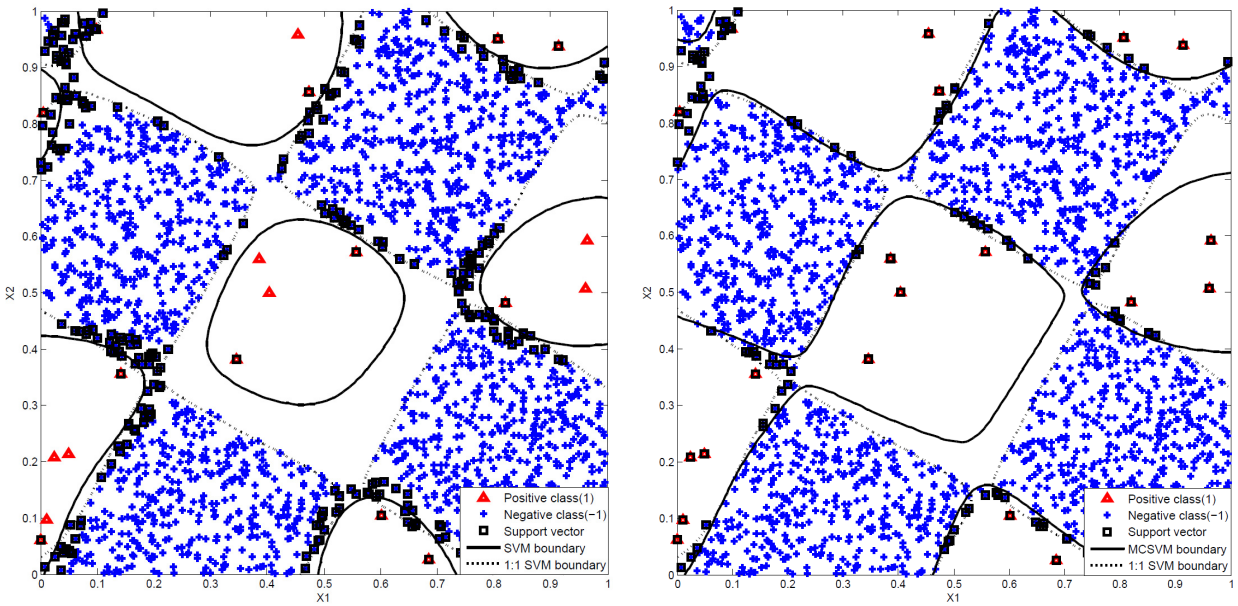
$$0 \le a_i \le C,\ \text{if}\ y_i = -1 \tag{11}$$

$$0 < a_i < C,\ \ \text{if}\ y_i = \ \ 1 \tag{12}$$

하지만 식 (12)와 같이 Unbounded 형태의 문제를 풀 때, 컴퓨터를 사용한 최적화 문제는 자원의 제약에 인한 머신 입실론(Machine epsilon) 등의 문제 때문에 제약조건을 만족하게 하는 값을 찾는 것은 쉽지 않다(Goldberg, 1991). 따라서 식 (12)의 조건을 Bounded 형태로 만들기 위해 0에 가까운 작은 값 θ를 넣어 식 (13)과 같은 형태로 수정하여 문제를 해결할 수 있다.

$$\theta \le a_i \le C - \theta,\ \ \text{if}\ y_i = \ \ 1 \tag{13}$$

본 논문에서 MCSVM의 파라미터 θ는 머신 입실론에 해당하는 아주 작은 값을 넣으면 해결이 되지만, 탐색 범위가 넓어 학습시간이 길어지는 단점이 있다. 따라서 학습시간의 속도를 증가시키기 위해 C에 따른 상대적인 범위를 설정한다. θ의 값을 10부터 10^6까지 범위를 가진 새로운 변수 T값과 이것을 C로 나눴을 때($\theta = C/T$)의 값으로 설정하여 반복 실험하였다. <Figure 5>에서는 제 1장에서 언급된 <Figure 1>의 Checker board 시뮬레이션을 통해 소수 범주와 다수 범주의 비율이 1:100인 데이터를 생성하여 MCSVM의 지지 벡터가 SVM과 어떤 차이가 있는지 나타내고 있다. 두 개의 범주에서 다수 범주의 레이블은 '−1'을 나타내며 '+'모양이다. 또한, 소수 범주는 '1'의 레이블을 가지며 '▲' 모양을 나타낸다. 얇은 점선으



(a) Support vectors on SVM (b) Support vectors on MCSVM

Figure 5. Support Vectors and boundary in checker board data set (1:100)

로 보이는 경계선은 데이터의 두 범주의 비율이 1:1일 때 SVM 학습 후 생성된 경계선을 나타낸다. 지지 벡터는 '□' 모양을 통해 표시되었으며, 실선은 각각 기존 SVM과 MCSVM을 통해 구성된 분류 경계선을 의미한다.

기존 SVM의 학습을 통해 구성된 지지 벡터와 분류 경계선이 나타나 있는 <Figure 5>의 (a)를 살펴보면 소수 범주의 데이터가 모두 지지 벡터로 잡히지 않았기 때문에 다수 범주의 데이터 중 일부가 불필요한 지지 벡터로 선정되었다. 이로 인해, 분류 경계선은 소수 범주의 영역의 경계선을 침범하는 것을 확인할 수 있다. 따라서 기존 SVM의 학습을 통해 선정된 소수 범주의 지지 벡터와 이에 대응되는 다수 범주의 지지 벡터는 표본 추출 데이터로 사용하기에 적합하지 않은 것을 확인할 수 있다.

반면 MCSVM을 이용한 <Figure 5>의 (b)에서는 소수 범주의 데이터가 모두 지지 벡터로 선정되었기 때문에 소수 범주의 영역을 대부분 확보한 상황에서 대응되는 다수 범주의 데이터만 지지 벡터로 선정되었다. 따라서 다수 범주의 데이터 중 불필요한 일부가 지지 벡터에서 제외되어 <Figure 5>의 (a)에 비해 분류 경계가 축소되지 않음을 확인할 수 있다. 또한, MCSVM을 통해 구성된 분류 경계선이 SVM보다 1:1에서 생성된 분류 경계선에 가까우므로 분류 경계의 왜곡이 감소하였다고 할 수 있다. 따라서 상대적으로 뚜렷한 경계선을 구축할 수 있는 MCSVM의 지지 벡터가 두 범주를 대표하는 표본이라고 할 수 있다.

본 연구에서는 MCSVM을 통해 다수 범주 데이터 중에서 지지 벡터로 선정되는 데이터를 과소 표본으로 추출하여 소수 범주의 데이터와 함께 두 범주를 대표하는 표본 데이터로 사용한다. 이렇게 추출된 데이터는 두 범주를 명확히 구분하기 위한 분류 경계선을 수립하고 정확하게 수율을 예측하기 위한 모델에 사용된다.

3. 실험 및 결과 분석

3.1 성능 척도

<Table 2>는 검증 데이터와 예측 데이터 사이의 관계를 표로 나타내는 혼동행렬(Confusion matrix)이다.

Table 2. Confusion matrix for evaluating performance

	Positive Prediction	Negative Prediction
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

또한 <Table 3>은 <Table 2>를 바탕으로 이루어진 성능 평가 척도를 나타내고 있다. 일반적으로 모델을 구축하여 성능을 평가할 때 TPR(True positive rate), TNR(True negative rate), ACC (Accuracy)를 사용한다. 하지만 99%의 다수 범주와 1%의 소수 범주로 구성된 불균형 데이터는 99%의 정확도로 다수 범주를 예측하는 경향이 존재한다. 따라서 한쪽으로 치우치는 극심한 불균형 데이터 문제는 정확도를 보완하는 기하평균(GM : Geometric Mean)이 성능 평가 척도로써 필요하다(Barandela *et al*., 2003).

본 논문에서는 소수 범주의 데이터를 제대로 예측한 TPR, 다수 범주를 제대로 예측한 TNR, 전체의 성능 정확도인 ACC, 다수 범주와 소수 범주의 불균형 비율을 고려한 GM을 성능 척도로 사용한다.

Table 3. Performance measure

Test Measure	
True Positive Rate (TPR)	$TPR = \frac{TP}{TP+FN}$
True Negative Rate (TNR)	$TNR = \frac{TN}{FP+TN}$
Accuracy (ACC)	$ACC = \frac{TP+TN}{TP+FP+FN+TN}$
Geometric Mean (GM)	$\sqrt{TPR \cdot TNR}$

3.2 비교 대안

본 논문에서는 MCSVM과의 비교 대안으로 총 5가지를 사용하였다. 첫 번째는 표본 추출법을 사용하지 않은 원 데이터(Raw data)를 이용하는 방법이다. 두 번째는 무작위 추출 접근법을 통해 다수 범주와 소수 범주의 비율을 MCSVM의 파라미터 조합의 시행착오법(Trial-and-error method)을 사용하여 MCSVM과 동일하게 조절하는 방법이다(RS : Random sampling). 세 번째는 소수 범주를 전부 추출한 후, 다수 범주의 데이터에서만 무작위 표본 추출을 수행하여 시행착오법으로 MCSVM과 비율을 맞추는 방법이다(MCRS : Major class random sampling). 네 번째 방법은 원 데이터에서 소수 범주의 비율만큼 다수 범주를 동등하게 1:1의 비율로 무작위 표본 추출(1:1_RS : 1:1 Random sampling)을 하는 것이다. 마지막 비교 대안은 기존 SVM을 통해 생성되는 지지 벡터들을 과소 표본 추출(SV of SVM sampling)하는 방법이다. 또한, 총 5가지 비교 대안의 성능 평가를 위해 다른 알고리즘에 비해 두 개의 범주를 나눌 때 성능이 좋은 SVM 알고리즘을 사용하려 했지만, 비교 대안에서 표본을 추출할 때 사용된 SVM을 동일하게 사용할 경우 예측 성능이 과적합(Overfitting) 될 수 있기 때문에 $\nu-SVM$을 사용하여 성능을 예측하였다.

예측 성능을 확인하기 위해 사용된 $\nu-SVM$알고리즘은 분류의 오류를 줄여가며 최적의 해를 찾는 방식이다(Chang and Lin, 2001b). 또한 $\nu-SVM$은 기존 SVM의 알고리즘의 오류를 허용하는 C의 값을 오분류율을 나타내는 ν로 바꾼 것이며,

Table 4. Data description of UCI data set

Training data set	# of dataset	# of training data		# of test data		Balance of classes
		Major class	Minor class	Major class	Minor class	
Ann-thyroid1 vs 3	6,832	3,488	93	3,178	73	40.15 : 1
Abalone19 vs 10-13	1,622	1,272	25	318	7	49.69 : 1
Abalone20 vs 8-10	1,916	1,512	21	378	5	72.69 : 1
Poker Hand8-9 vs 5	2,075	1,640	20	410	5	82 : 1
Abalone20 vs 5-10	2,682	2,125	21	531	5	103.15 : 1

파라미터 ν와 σ의 값에 따라 달라진다. 특히, ν의 값은 오류의 허용치를 나타내는 비율이기 때문에 0과 1사이의 범위를 가지며 아주 작은 값에도 민감하게 반응한다(Hsu *et al*., 2003).

3.3 UCI 데이터

본 논문에서는 반도체 데이터에 제안하는 MCSVM을 적용하기에 앞서, 불균형 데이터로 많이 사용되는 UCI 데이터셋을 사용하여 성능을 평가하였다(Akbani *et al*., 2004; Kang and Cho, 2006; Bache and Lichman, 2013). 아래의 <Table 4>는 각각의 데이터 설명을 나타내고 있다.

첫 번째 Ann-thyroid1 vs 3의 데이터셋은 원래는 총 3개의 범주로 구성되어 있지만 데이터수가 가장 적은 범주(1)와 가장 많은 범주(3)를 묶어서 다시 데이터를 구성하였다(Kang and Cho, 2006). 나머지 데이터셋은 8:2의 비율로 학습데이터와 검증데이터로 나눠 학습하였다. 또한, Abalone와 Poker데이터는 다수 범주로 구성된 데이터에서 불균형을 이루고 있는 두 개 범주의 데이터를 추출하였다. 본 논문에서 다수 범주의 레이블은 Negative(-1)를 나타내고 있으며 소수 범주는 Positive(1)로 나타낸다.

(1) 실험 설계

MCSVM은 C, σ, θ로 구성된 총 3개의 파라미터로 구성되어 있다. θ의 값은 <Table 5>에서 나타내고 있듯이 C/T를 사용하여 여러 범위에서 실험하였다. 또한, 각각의 파라미터는 모델에 적합한 값을 선택해야 하므로 시행착오법을 사용하여 파라미터를 정하였다. 여러 범위에서 파라미터로 실험한 결과 <Table 5>의 범위에서 결과값을 도출하였다. 또한, 추출된 데이터의 모델을 만들고 평가하기 위해 $\nu-SVM$을 사용하였다. $\nu-SVM$ 알고리즘 역시 RBF kernel을 사용하여 ν값(10개)과 $\sigma_{\nu-SVM}$값(12개)을 시행착오법으로 파라미터를 찾는다. 본 실험은 UCI데이터를 사용하며 MCSVM을 적용할 때 파라미터에 따라 720($10\times12\times6$)회로 실험을 하였다. 이후 반복되는 실험에 따라 생성되는 새로운 데이터셋에 $\nu-SVM$을 적용해 각각 120(10×12)회 실험을 반복하였으며 비교대안 역시 동일하게 진행하였다.

Table 5. Condition of parameters

Algorithm	Type of parameters	Condition
$MCSVM$	C	10^{-3}, 10^{-2}, 10^{-1}, 2^{-1}, 1, 5, 10, 20, 50, 100
	σ_{MBSVM}	10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 2^{-1}, 1, 5, 10, 50, 100
	T	10, 10^2, 10^3, 10^4, 10^5, 10^6
$\nu-SVM$	ν	10^{-11}, 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}
	$\sigma_{\nu-SVM}$	10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 2^{-1}, 1, 5, 10, 50, 100

(2) 실험 결과

<Table 6>은 MCSVM의 방법을 사용하여 UCI데이터로 실험한 결과이다. 본 실험은 (1)에 나와 있는 <Table 5>의 파라미터를 기반으로 시행착오법을 수행하였다. 그러나, 제 3.2절의 비교 대안으로 언급한 1:1_RS 방법과 SV of SVM sampling 방법은 실험에서 사용된 원 데이터보다 성능이 현저히 낮으므로 제외하였다. 그 이유는 소수 범주에만 맞춰서 표본 추출을 했을 때 데이터의 수가 너무 적어 성능이 저하되기 때문이다.

총 6개의 UCI 데이터에서 각각 RS, MCRS, MCSVM의 방법으로 실험한 결과, 표본 추출한 데이터의 성능이 전처리 과정을 거치지 않은 원 데이터(Raw data)보다 높은 성능을 나타낸다. 또한, 원 데이터를 $\nu-SVM$에 바로 적용했을 때, 소수 범주를 정확히 예측한 TPR과 다수 범주를 정확하게 예측한 TNR을 비교한 결과는 TNR값이 더 높다는 것을 알 수 있다. TNR의 값이 상대적으로 높은 이유는 다수 범주 데이터의 비율이 높으므로 검증 데이터를 적용했을 때 소수 범주가 아닌 다수 범주로 잘못 예측하는 경우가 많기 때문이다. 또한, 소수 범주의 데이터가 다수 범주의 데이터에 비해 현저히 적기 때문에 <Figure 5>의 (a)와 같이 경계선의 왜곡이 발생하게 된다. 따라서 경계의 왜곡이 발생했을 때 데이터 수가 많은 범주 쪽으로 분류하게 되어 TNR의 값이 높아지는 현상이 발생한다. GM의 결과값은 원 데이터를 제외한 나머지 표본 추출법을 비교하였을 때 대체로 RS, MCRS 방법보다 MCSVM을 사용한 방법에서 성능이 좋음을 알 수 있다. 특히 RS와 MCRS의 방법은 MCSVM에서 생성된 데이터셋과 같은 수로 데이터셋을 만들어 데이터셋마다 표본 추출의 개수가 다를 때 가장 좋은 결과를 나타내고 있지만, 무작위로 추출하는 것보다는 MCSVM의 방식을 사용하

Table 6. Results of UCI data set

Data set	Methods	TPR	TNR	ACC	**GM**	# of sample data
Ann-thyroid1 vs 3 (40.15 : 1)	Raw data	0.9452	0.9770	0.9763	0.9610	Major : 3,488 Minor : 93
	RS	0.9774	0.9452	0.9680	0.9612	Major : 3,487 Minor : 93
	MCRS	0.9831	0.9452	0.9720	0.9639	Major : 3,256 Minor : 93
	MCSVM	1.0000	0.9770	0.9775	**0.9884**	Major : 126 Minor : 93
Ablalone19 vs 10-13 (49.69 : 1)	Raw data	0.4286	0.7736	0.7662	0.5758	Major : 1,272 Minor : 25
	RS	0.7143	0.7390	0.7385	0.7265	Major : 1,017 Minor : 21
	MCRS	0.7143	0.7547	0.7538	0.7342	Major : 800 Minor : 25
	MCSVM	0.8571	0.6478	0.6523	**0.7452**	Major : 177 Minor : 25
Abalone20 vs 8-10 (72.69 : 1)	Raw data	0.4000	0.9868	0.9791	0.6283	Major : 1,512 Minor : 21
	RS	1.0000	0.7354	0.7389	0.8576	Major : 1,343 Minor : 21
	MCRS	0.8000	0.9259	0.9243	0.8607	Major : 163 Minor : 21
	MCSVM	0.8000	0.9497	0.9478	**0.8717**	Major : 388 Minor : 21
Poker Hand8-9 vs 5 (82 : 1)	Raw data	0.4000	0.9146	0.9084	0.6049	Major : 1,640 Minor : 20
	RS	0.6000	0.9829	0.9783	0.7680	Major : 1,420 Minor : 17
	MCRS	0.6000	0.9561	0.9518	0.7574	Major : 1,555 Minor : 20
	MCSVM	0.8000	0.7537	0.7542	**0.7765**	Major : 765 Minor : 20
Abalone20 vs 5-10 (103.15 : 1)	Raw data	0.6000	0.9718	0.9683	0.7636	Major : 2,125 Minor : 21
	RS	0.8000	0.9906	0.9888	0.8902	Major : 1,471 Minor : 10
	MCRS	1.0000	0.8023	0.8041	0.8957	Major : 2,048 Minor : 21
	MCSVM	1.0000	0.8192	0.8209	**0.9051**	Major : 209 Minor : 21

는 것이 더 효과적임을 알 수 있다. 또한, MCSVM을 적용한 방법은 지지 벡터를 미리 고정하고 거기에 대응되는 다수 지지 벡터를 표본 추출로 사용했기 때문에 전체를 대표할 수 있는 데이터가 추출되었다고 볼 수 있다.

본 논문에서 제안하고 있는 MCSVM으로 검증데이터로 실험하였을 때 생성되는 지지 벡터의 개수는 원 데이터보다 적은 수로 구성되어있다. 또한, 각각 θ의 값은 <Table 6>의 데이터 순서대로 각각 0.005, 2, 0.5, 0.05, 0.05의 값을 나타낼 때 좋은 성능을 보인다. 따라서 불균형 데이터에서 데이터 전부를 사용하는 것보다는 데이터의 과소 표본 추출 방법이 더 효과적임을 알 수 있다. 특히, 여러 방법의 과소 표본 추출법에서도 MCSVM을 사용한 경우가 가장 좋은 성능을 보이는 것을 확인할 수 있다.

3.4 반도체 공정 데이터

본 실험은 실제로 반도체 테스트 공정 중 최종 단계에서 생성된 결과 데이터를 사용하였다. 실험에 사용된 데이터는 연속형(Continuous)의 속성을 가지고 있는 총 46개의 변수가 독립변수로 사용되었으며 종속변수는 오로지 양품과 불량으로만 구성된 이진(Binary) 속성을 가진 변수로 구성되어 있다. 본 실험은 현업에 종사하고 있는 전문가의 배경지식을 바탕으로 수많은 변수 중에서 특정 변수를 선발하여 구성하였다. 선발된 변수는 분류에 가장 민감한 Fail bit count를 나타내는 3개의 변수와 웨이퍼의 좌표를 나타내는 2개의 변수를 포함하여 총 5개의 설명변수로 구성되었다. 또한, 웨이퍼의 칩이 양품 혹은 불량인지 구별하는 한 개의 종속변수를 사용한 데이터이다. 또한, 데이터

의 다수 범주의 레이블은 Negative(-1)인 양품을 나타내고 있으며 소수 범주는 Positive(1)인 불량을 나타내고 있다.

(1) 실험 설계

본 논문에서 사용한 반도체 데이터는 한 개의 로트이다. 로트에는 총 25장의 웨이퍼가 있으며 한 장의 웨이퍼는 약 1,700개의 칩으로 구성되어 있다. 즉, 실험에 사용된 전체수는 약 40,000개로 구성되어 있다. 본 실험의 진행은 <Figure 1>에서 보여주고 있는 순서로 진행하였다. 첫 번째 단계에서는 제 3.3절의 (1) 실험 설계와 동일하게 실험을 진행하였다. 각각의 웨이퍼는 극심한 불균형 데이터이기 때문에 소수 범주를 고정하여 다수 범주의 지지 벡터를 추출하는 MCSVM을 사용하여 과소 표본 추출을 수행하였다. 이때 MCSVM은 파라미터(C, σ, θ)에 따라 표본 추출 데이터가 달라지기 때문에 파라미터의 조합을 시행착오법으로 데이터를 반복적으로 추출하였다. 두 번째 단계에서는 각 웨이퍼 단위에서 지지 벡터로 추출된 다수 범주의 데이터와 소수 범주 데이터를 하나의 데이터셋으로 병합한다. 파라미터의 조합 중에서 모든 데이터가 지지 벡터로 지정되어 표본 추출이 이루어지지 않은 경우에는 모든 대안이 동일한 성능을 보이게 되므로 제외하였으며, 학습이 이루어지지 않은 경우도 제외하였다. 결과적으로 전체 데이터 중에서 일부가 지지 벡터로 표본 추출된 60개의 데이터셋을 사용하여 실험을 진행하였다. 이후 각각의 데이터셋은 제 3.3절의 (1) 실험 설계와 동일하게 $\nu-SVM$을 적용하며, 파라미터(ν, σ)에 따른 조합의 시행착오법 방법으로 실험을 수행하여 모델을 만들고 성능을 평가한다. 또한, 파라미터의 조합으로 생성된 각각의 표본 추출 데이터셋을 $\nu-SVM$에 적용하기 전에 학습데이터와 검증데이터로 나누어 실험하였다. 검증데이터는 한 장의 웨이퍼를 사용하며 나머지 24장의 웨이퍼는 학습데이터로 구성된다. 데이터는 한 번의 학습이 아닌 여러 번의 교차 학습이 필요하므로 25번의 교차 검증(CV : Cross validation) 방법인 Leave-one-out으로 실험하였으며 25번의 평균을 결과값으로 사용하였다. 결과적으로 실험 횟수는 파라미터의 조합과 Leave-one-out방식을 합쳐 180,000($60\times10\times12\times25$)번의 실험을 하였다. 또한, RS 방법과 MCRS 방법은 MCSVM의 조합으로 생성되는 데이터의 개수만큼 데이터 추출의 반복 실험을 하였으며, $\nu-SVM$ 적용 시 제안 방법과 동일한 조합으로 실험하였다.

(2) 실험 결과

<Table 7>은 MCSVM을 사용하여 과소 표본 추출한 데이터셋과 5개의 비교 대안을 학습한 결과를 나타낸다. 특히 원 데이터를 그대로 사용하였을 때는 GM의 성능이 평균 51%로 정확하게 수율을 예측하지 못함을 알 수 있다. 제 3.3절의 (2) 실험 결과와는 다르게 원 데이터에서는 소수 범주를 정확히 예측하는 TPR의 값이 더 높음을 알 수 있다. TPR이 다수 범주를 예측하는 TNR보다 성능이 높게 나온 것은 $\nu-SVM$ 학습 시, 지나치게 소수 범주에 치우쳐 분류하는 과적합이 발생했기 때문이다. 또한, 원 데이터를 사용했을 때보다 표본 추출 방법을 사용했을 때 GM의 값이 원 데이터보다는 좋음을 나타내며 표본 추출 방법이 더 효과적임을 알 수 있다.

여러 비교 대안의 GM값을 비교한 결과, MCSVM으로 과소 표본 추출한 방법이 다른 비교 대안에 비해 높은 성능을 나타내고 있다. 또한, 파라미터 θ를 추가한 MCSVM은 $C=10$, $\sigma_{MBSVM}=1$, $\theta=10^{-5}$일 때 가장 높은 성능을 나타내었다. MCSVM의 지지 벡터의 개수는 기존의 양품 데이터수인 38,723개보다 훨씬 적은 3,799개로 표본 추출이 되었으며 극소수를 차지하고 있던 불량인 데이터의 수와 많이 차이가 나지 않음을 알 수 있다. 비교 대안보다 표본의 수가 적지만 높은 성능을 나타낼 수 있는 이유는 소수 범주를 고정한 후, 소수 범주에 대응되는 다수 범주의 표본 추출 데이터가 반도체 데이터 전체에서 대표할 수 있는 표본이라고 볼 수 있기 때문이다. 비교 대안 SV of SVM sampling은 $C=50$, $\sigma_{SVM}=1$일 때 가장 좋은 성능을 보였지만 웨이퍼별로 기존 SVM의 지지 벡터만을 표본 추출 데이터로 사용하였기 때문에 MCSVM의 GM보다는 낮은 성능을 나타내고 있다. 기존 SVM에서는 극심한 불균형 데이터일 경우 다수 범주가 월등히 많기 때문에 분류 경계를 침범하는 다수 범주의 데이터가 지지 벡터로 선정되고 소수 범주의 일부가 지지 벡터로 잡히지 않는다. 따라서 불균

Table 7. Results from Sampling data set

Data set **(25-CV)**	TPR	TNR	ACC	**GM**	# of sample data
Raw data	0.6493	0.4117	0.4171	0.5111	Major class : 38,723 Minor class : 836
RS	0.5230	0.8421	0.8362	0.6582	Major class : 31,969 Minor class : 616
1:1_RS	0.5309	0.7398	0.7363	0.6161	Major class : 836 Minor class : 836
MCRS	0.6566	0.7081	0.7081	0.6696	Major class : 24,711 Minor class : 836
SV of SVM sampling	0.7008	0.6198	0.6226	0.6413	Major class : 13,129 Minor class : 836
MCSVM	0.5588	0.8894	0.8302	**0.6995**	Major class : 3,799 Minor class : 836

형 데이터에 기존 SVM을 적용할 경우 선정된 지지 벡터들은 전체를 대표할 수 있는 표본으로 사용하기에 적합하지 않다.

결과적으로 표본 추출법을 사용할 때 무작위로 데이터의 수를 줄이는 것보다는 MCSVM을 사용하여 소수 범주를 고정한 후 이에 대응되는 다수 범주를 함께 추출하여 표본 데이터로 사용하는 것이 좋은 결과를 가져온다고 할 수 있다.

4. 결론 및 추후 연구

반도체 공정은 복잡한 공정과 많은 검사항목을 거치기 때문에 데이터의 부피가 크다. 또한, 최종 검사 단계에서 수집하는 데이터는 대부분이 양품이며 극소수만 불량이 포함되는 극심한 불균형 데이터이다. 반도체의 불균형 데이터를 통한 수율 예측의 문제점을 해결하기 위해 본 논문은 MCSVM 기반의 과소 표본 추출법을 제안하였다.

MCSVM은 소수 범주 데이터를 사전에 지지 벡터로 고정한 후 학습하기 때문에 기존 SVM 학습 시 발생하는 문제점을 해결할 수 있다. 첫째, 소수 범주 데이터의 일부가 지지 벡터로 선정되지 않아 다수 범주 데이터가 불필요하게 지지 벡터로 형성되는 문제를 방지할 수 있다. 둘째, 불필요한 다수 범주의 지지 벡터 형성의 방지를 통해 소수 범주의 영역이 축소되는 경계선의 왜곡 현상을 완화할 수 있다. 마지막으로 소수 범주 데이터에 대응되는 다수 범주의 데이터만을 지지 벡터를 효과적으로 선정하기 때문에 범주를 대표하는 과소 표본으로 사용할 수 있다. 객관적인 성능 비교를 위해 제 3.3절의 UCI 데이터를 사용해 MCSVM의 우수한 성능을 확인하였다. 특히 제 3.4절과 같이 반도체 공정의 최종 검사 단계와 같이 극심한 불균형 데이터가 수집되는 경우에도 효과적이다.

현재 MCSVM의 연구는 지지 벡터로 선정되는 α_i의 값이 0보다 크고 $C-\theta$의 범위에서 많은 실험을 반복하여 최적값을 찾아 생성되는 지지 벡터의 표본을 추출하였다. 머신 입실론을 사용하는 경우보다는 적은 범위를 탐색하였지만 MCSVM에서 지지 벡터로 형성되는 α_i값들의 경향성을 파악할 수 있다면 α_i의 최적값을 효율적으로 찾을 수 있다. 또한, α_i값들의 경향성 파악을 통해 더욱 의미 있는 지지 벡터의 추출이 가능할 것이다. 마지막으로 소수 범주의 축소를 더욱 개선할 수 있는 과소 표본 추출법에 대한 연구가 필요하다. 제약조건의 수정을 통해 기존의 SVM에 비해 소수 범주의 경계가 축소되는 것을 완화하였지만, 실제 분류 경계선과는 차이가 있다. 따라서 분류 경계선을 더욱 명확히 하는 지지 벡터를 선정하는 연구를 수행한다면 예측 성능의 향상을 기대할 수 있을 것이다.

참고문헌

An, D. W., Ko, H. H., Kim, J. H., Baek, J. G., and Kim, S. S. (2009), A Yields Prediction in the Semiconductor Manufacturing Process Using Stepwise Support Vector Machine, *IE interfaces*, **22**(3), 252-262.

Akbani, R., Kwek, S., and Japkowicz, N. (2004), Applying support vector machines to imbalanced datasets, In *Machine Learning : ECML 2004*(39-50). Springer Berlin Heidelberg.

Bache, K. and Lichman, M. (2013), *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml, Irvine, CA : University of California, School of Information and Computer Science.

Baek, D. H. and Han, C. H. (2003), Application of Data mining for improving and predicting yield in wafer fabrication system, *Journal of Intelligence and Information Systems*, **9**(1), 157-177.

Barandela, R., Sánchez, J. S., Garcıa, V., and Rangel, E. (2003), Strategies for learning in class imbalance problems, *Pattern Recognition*, **36**(3), 849-851.

Chang, C. C. and Lin, C. J. (2001b), Training n-support vector classifiers : theory and algorithms, *Neural Computation*, **13**(9), 2119-2147.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2011), *SMOTE : synthetic minority over-sampling technique*, arXiv preprint arXiv : 1106.1813.

Chyi, Y.-M. (2003), *Classification analysis techniques for skewed class distribution problems*, Master thesis, Department of Information Management, National Sun Yat-Sen University.

Ciciani, B. and Iazeolla, G. (1991), A Markov chain-based yield formula for VLSI fault-tolerant chips, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, **10**(2), 252-259.

Cortes, C. and Vapnik, V. (1995), Support-vector networks, *Machine learning*, **20**(3), 273-297.

Cristianini, N. and Shawe-Taylor, J. (2000), *An introduction to support vector machines and other kernel-based learning methods*, Cambridge University press.

Crosier, R. B. (1988), Multivariate generalizations of cumulative sum quality-control schemes, *Technometrics*, **30**(3), 291-303.

Goldberg, D. (1991), What every computer scientist should know about floating-point arithmetic, *ACM Computing Surveys (CSUR)*, **23**(1), 5-48.

Han, H. Y. (2009), Introduction of Patter Recognition, *HANBIT Media*, Seoul Korea.

Hsu, C. W., Chang, C. C., and Lin, C. J. (2003), A practical guide to support vector classification.

Jang, D. Y. and Bae, S. J., (2009), Hybrid Datamining Algorithm for Monitoring Input Variables in Semiconductor Manufacturing Process, *IE Interfaces*, 563-569.

Kang, P. and Cho, S. (2006), EUS SVMs : Ensemble of under-sampled SVMs for data imbalance problems, *In Neural Information Processing* (837-846), Springer Berlin Heidelberg.

Kim, J. W., Park, J. S., Kim, J. S., Kim, S. S., and Baek, J. G. (2014), Update Cycle Detection Method of Control Limits using Control Chart Performance Evaluation Model, *Journal of the Korean Institute of Industrial Engineering*, **40**(1), 43-51.

Kim, K., Hwang, C. G., and Lee, J. G. (1998), DRAM technology perspective for gigabit era. Electron Devices, *IEEE Transactions on*, **45**(3), 598-608.

Kim, M. J. (2012), Ensemble Learning with Support Vector Machines for Bond Rating, *Journal of Intelligence and Information Systems*, **18**(2), 29-45.

Kim, M. S. and Baek, J. G. (2011), Fail Prediction of DRAM Module Outgoing Quality Assurance Inspection using Ensemble Learning Algorithm, *IE Interfaces*, **25**(2), 178-186.

Kim, S. C. (2010), A Joint Design of Rectifying Inspection Plans and

Service Capacities for Multi-Products, *Journal of the Korea Operations Research and Management Science Society*, **35**(1), 97-109.

Kim, S. E., Kang, J. H., Park, J. H., Kim, S. S., and Baek, J. G. (2012), Fault Detection of Unbalanced Cycle Signal Data Using SOM-based Feature Signal Extraction Method, *Journal of The Korea Society for Simulation*, **21**(2), 79-90.

Kymal, C. and Patiyasevi, P. (2006), Semiconductor quality initiatives : How to maintain quality in this fast-changing industry, *Quality Digest*, **26**(4), 43-48.

Li, T. S. and Huang, C. L. (2009), Defect spatial pattern recognition using a hybrid SOM-SVM approach in semiconductor manufacturing, *Expert Systems with Applications*, **36**(1), 374-385.

Schölkopf, B. and Smola, A. J. (2002), Learning with Kernels : Support Vector Machines, Regularization, Optimization and Beyond, MIT press.

Shin, H. and Cho, S. (2006), Response modeling with support vector machines, *Expert Systems with Applications*, **30**(4), 746-760.

Yan, R., Liu, Y., Jin, R., and Hauptmann, A. (2003), On predicting rare classes with SVM ensembles in scene classification. In Acoustics, Speech, and Signal Processing, 2003, *Proceedings (ICASSP '03)*, *2003 IEEE International Conference on*, **3**, III-21.

Yen, S. J. and Lee, Y. S. (2009), Cluster-based under-sampling approaches for imbalanced data distributions, *Expert Systems with Applications*, **36**(3), 5718-5727.

Wu, G. and Chang, E. Y. (2003), Adaptive feature-space conformal transformation for imbalanced-data learning, *In ICML*, 816-823.