카카오미니는 말하는 사람을 어떻게 인식할까?

최근 음성인식의 성능이 많이 향상되면서, 음성이 친숙한 인터페이스로 자리잡고 있다. 음성에는 우리가 전하고자 하는 언어적 정보뿐만 아니라, 나이, 건강 상태, 감정 상태 등의 정보도 포함되어 있다. 또한, 음성은 각 사람의 고유한 정보를 담고 있어 이를 분석하면 목소리의 차이를 구별할 수 있다. 이런 정보를 분석하고 자동화하는 방법을 화자인식이라 부른다.

화자인식은 본인 인증의 한 수단으로 사용할 수 있다. 예를 들면 휴대폰 잠금 해제, TV, 에어컨 등의 기기 제어 등에 사용할 수 있다. 또한, 음성인식 과정에서 화자의 정보를 분석하면 개인화를 통해 콘텐츠 추천, 개인화 검색 등의 결과를 같이 줄 수 있으므로 서비스의 정확도를 좀 더 향상시키는 보조 수단으로 사용할 수 있다. 이 글에서는 화자 모델링에 사용하는 여러 방법들과 화자인식 평가 방법에 대해 알아본다.

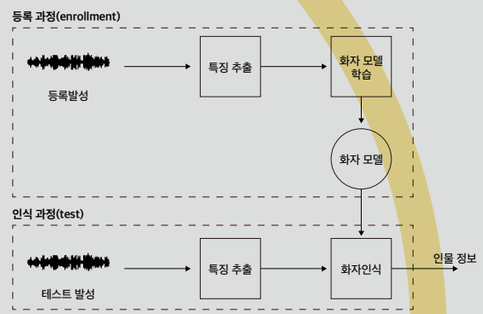
글 | 김명재 arth.mj@kakaocorp.com

SL2라는 음성 처리 회사에서 병역을 수행하고 있을 때, 음성 처리가 너무 어려워 정복해 보고 싶은 마음이 들었다. 하여 패기 있게 대학원에 진학하였으나 음성 처리는 여전히 어렵다. 어쩌다 보니 음성을 10년 넘게 다뤘지만, 아직도 배운 것보다 배워야 할 것이 많다. 항상 인공지능과 데이터 엔지니어링에 관심을 두고 있어 대가들의 움직임에 감탄하고 있다. 운 좋게 카카오에 입사하여 즐겁게 일하고 있는 개발자.

화자인식 개요

화자인식은 사람이 발성한 음성을 컴퓨터가 분석하여 음성의 인물 정보를 얻어 내는 과정을 말한다. 음성인식과 처리 과정은 유사하지만 음성인식은 발성한 음성에서 언어적 정보를 찾는 반면, 화자인식은 발성한 음성에서 인물 정보를 찾는다. 화자인식을 수행하기 위해서는 비밀번호 등록과 같이 발성한 음성을 통계적인 음향 모델(acoustic model)로 만드는 등록(enrollment)과정이 필요하며, 등록 과정에서 만들어진 음향 모델을 이용하여 인식(test) 과정에서 입력한 발성의 인물정보를 얻는다. [그림 1]은 화자인식 과정을 간단히 도식화한 그림이다.

[그림 1] 화자인식 개요도



화자인식은 얻은 정보를 분류하는 방법에 따라 화자 식별(speaker identification)[1]과 화자 확인(speaker verification)[2]으로 나눌 수 있다. 화자 식별은 등록된 사람들 중에서 발성한 사람을 찾는 과정이고, 화자 확인은 등록된 화자의 음성이 맞는지 결정하는 과정이다. 실제 화자인식기에서는 화자 식별과 화자 확인의 과정이 모두 필요하다.

화자 식별은 길이 T의 관측된 음성 특징 벡터열 $X=\{x_1,x_2,\cdots,x_T\}$가 주어졌을 때, 화자 집합 $S=\{\lambda_1,\lambda_2,\cdots,\lambda_k,\cdots,\lambda_S\}$에서 가능도(likelihood)가 가장 높은 모델 $\hat{S}$을 찾는 과정이며, 수식으로 표현하면 [수식 1]과 같다.

[수식 1]

$$\hat{S} = argmax_{1 \le k \le S} Pr(\lambda_k|X) = argmax_{1 \le k \le S} \frac{p(X \mid \lambda_k)Pr(\lambda_k)}{p(X)}$$

여기서 $Pr(\lambda_k|X)$는 음성 특징열이 주어졌을 때, 화자 모델 λ_k에 대한 가능도이며, 베이즈 룰(Bayes' rule)에 의해 우측 식으로 바꿀 수 있다. $p(X|\lambda_k)$은 화자 모델 λ_k에 대한 음성 특징의 가능도이다. $Pr(\lambda_k)$은 화자 λ_k가 등장할 사전 확률로 1/S로 모두 같다고 가정하고, $p(X)$는 모든 화자에게 동일하므로 [수식 2]와 같이 간략하게 표현할 수 있다.

[수식 2]

$$\hat{S} = argmax_{1 \le k \le S} Pr(\lambda_k|X)$$

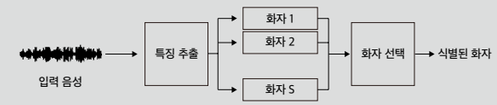
여기에 로그 함수를 사용하고 음성 특징 벡터열이 서로 독립적으로 관찰되었다고 가정하면 [수식 3]과 같이 표현할 수 있다.

[수식 3]

$$\hat{S} = argmax_{1 \le k \le S} \sum_{t=1}^{T} logP(X_t|\lambda_k)$$

[그림 2]는 화자 식별 과정을 나타낸 그림이다.

[그림 2] 화자 식별 개요도



화자 확인은 길이 T의 음성 특징 벡터열 $X=\{x_1,x_2,\cdots,x_T\}$가 관측되었을 때, 두 가지 가설 H_0와 H_1 중 한 가지를 선택하는 문제이다.

· H_0: X는 등록된 화자가 발성한 음성이다.

· H_1: X는 등록된 화자가 발성한 음성이 아니다.

이 문제를 수행하기 위해 화자 확인은 등록되지 않은 화자로 이루어진 배경 화자 모델을 필요로 한다. 화자 확인의 일반적인 접근 방법은 관측된 음성 특징 벡터열 X의 가능도 비율(likelihood ratio)을 사용하며 [수식 4]와 같이 표현할 수 있다.

[수식 4]

$$\frac{Pr(\lambda_k \mid X)}{Pr(\lambda_{UBM} \mid X)}$$

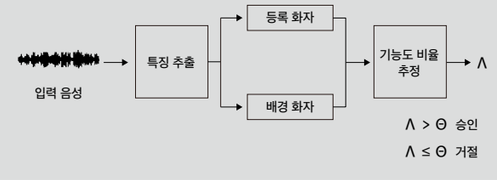
여기서 $Pr(\lambda_k|X)$는 음성 백터열 X가 주어졌을 때, 등록된 화자 모델 λ_k가 나올 확률이고, $Pr(\lambda_{UBM}|X)$는 음성 백터열 X가 주어졌을 때, 배경 화자 모델 (universal background model, UBM)λ_{UBM}가 나올 확률이다. [수식 4]에 베이즈 룰과 로그 함수를 적용하여 식을 다시 정리하면 다음과 같다.

[수식 5]

$$\Lambda(X) = log(X \mid \lambda_k) - log(X \mid \lambda_{UBM})$$

로그 유사도 비율 $\Lambda(X)$를 추정하여 기준점(threshold) θ보다 크면 등록된 화자 λ_k라 승인하고, 기준점 보다 작거나 같으면 등록된 화자가 아니라고 거절한다. [그림 3]은 화자 확인 과정을 보여준다.

[그림 3] 화자 확인 개요도

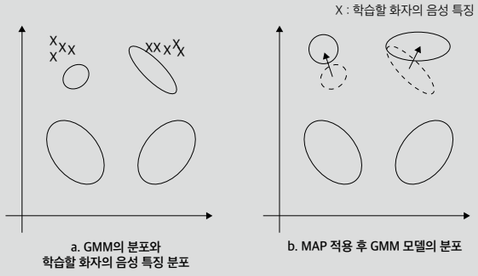


또한, 화자인식은 등록 발성 내용과 인식 발성 내용을 동일하게 제한하는 문장 종속 화자인식 방법과 인식 발성에 제한을 두지 않는 문장 독립 화자인식이 있다. 이 글에서는 화자인식에 널리 쓰이는 방법들과 평가 방법에 대해 알아본다.

화자 모델링

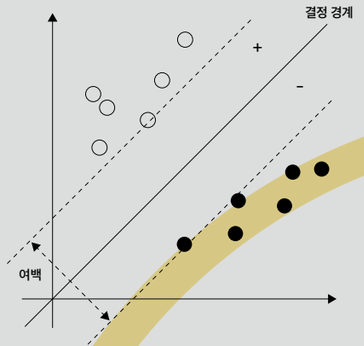
화자인식은 기본적으로 음성을 가우시안 혼합 모델(Gaussian Mixture Model, GMM)로 모델링 한다. GMM 방법은 개개인의 모델을 정교하게 만들기 위해 매우 많은 양의 음성 데이터를 필요로 한다. 그러나 수집할 수 있는 개개인의 음성 양은 제한되어 있기 때문에, GMM 방법으로는 높은 화자인식 성능을 기대하기 어려웠다. 이후, 다양한 사람으로부터 얻은 대량의 데이터로 GMM 모델을 학습하여 정교한 모델을 만들고, 개개인으로부터 얻은 소량의 등록 데이터를 최대 사후(Maximum a Posterior, MAP) 적응(adaptation) 방법을 통해 새로운 화자 모델을 만드는 GMM-UBM(Universal Background Model)[*3] 방법이 제안되었다. [그림 4]는 2차원의 GMM에서 소량의 화자 음성 특징으로 MAP 적응을 했을 때, 모델 분포의 변화를 보여 준다.

[그림 4] GMM의 분포와 MAP 적응 후의 분포



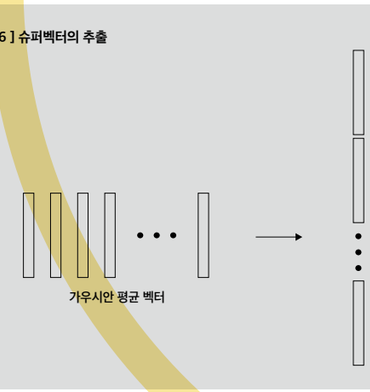
나이브 베이즈 (Naive Bayes) 기반의 분류 방법을 사용하는 GMM-UBM 방법에 SVM(Support Vector Machine)을 적용한 GMM-SVM방법이 제안되었다. [그림 5]는 2차원 공간에서 SVM의 이진 분류를 보여 준다. 실선은 부류를 결정하는 경계(hyperplane)이고, 점선에 위치하는 벡터가 서포트 벡터(support vector)이다. 서포트 벡터는 SVM의 결정 경계를 찾는 기준점이 되며, SVM은 서포트 벡터 간의 여백을 최대화하는 경계를 찾는 것을 목표로 한다.

[그림 5] SVM의 분류 예제



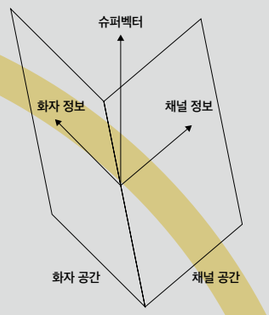
GMM-SVM의 입력으로는 슈퍼벡터(supervector)가 사용된다. 슈퍼벡터는 화자 적응된 GMM 모델의 평균 벡터를 연결하여 생성하는 하나의 매우 높은 차원의 특징 벡터이다. 슈퍼벡터는 매우 고차원의 특징이기 때문에 다루기 쉽지 않은 데다 매우 많은 메모리를 필요로 하며, 화자 정보 외에 다른 부가적인 채널 및 잡음 정보도 같이 표현된다. [그림 6]은 슈퍼벡터를 만드는 과정을 보여 준다. 예로 음성 특징이 60차원이고, GMM은 2,048개의 혼합 성분을 갖는다고 가정하면, 60 × 2,048 차원(12만 2,880차원)의 슈퍼벡터가 된다.

[그림 6] 슈퍼벡터의 추출



이와는 다른 방향으로 결합요인 분석(Joint Factor Analysis, JFA)[*4] 방법을 적용하여 슈퍼벡터에서 화자 정보(speaker factor)와 채널 정보(channel factor)를 분리하는 연구가 제안되었다. JFA 방법은 고윳값 분해(eigenvalue decomposition)에 기반하는데, 고유 분해 방법은 특정 데이터 집합으로부터 서로 수직(orthogonal)인 고유벡터(eigen vector)를 찾고, 이 고유벡터를 기저(basis)로 하는 변환 행렬을 통해 고유공간(eigenspace)으로 사영(projection)하는 방식으로 화자 정보를 얻는다. JFA에서는 변환 행렬의 정교한 추정을 위해, 많은 화자가 다양한 채널에서 녹음한 음성 데이터를 필요로 한다. 그러나 JFA는 한 화자가 다양한 환경에서 녹음한 데이터를 요구하기 때문에, 다량의 학습 코퍼스를 구축하기 어려운 문제점이 있다. [그림 7]은 JFA의 분해방법을 보여준다.

[그림 7] JFA의 벡터 분해



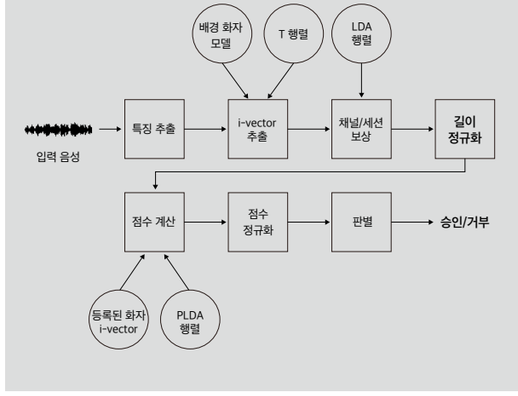
이런 방법을 개선하기 위해, 채널 특징을 제거하고, 변환 행렬(T-matrix)을 발성 단위로 처리하는 i-벡터(i-vector)[*5] 방법이 제안되었다. i-벡터 방법은 현재 화자인식 분야에서 가장 효율적인 특징 추출 방법으로 자리 잡았으며, 크게 추출(extraction), 채널/세션 보상(channel/session compensation), 길이 정규화(length normalization), 점수 계산(scoring), 점수 정규화(score normalization)로 나눌 수 있다. i-벡터 추출 방법은 앞서 소개한 JFA의 화자 정보 추출 방법과 동일하지만, 다른 점이 있다. JFA 방식은 화자 단위로 음성 데이터를 처리하는 반면, i-벡터 방식은 발성 단위로 음성 데이터를 처리한다는 점이다. 최근 i-벡터 추출과정 중, GMM posterior계산 과정을 DNN(Deep Neural Network)으로 대체한 방법도 제안되었다[*6].

채널/세션 보상은 시간이 지남에 따라 화자의 발화 상태가 조금씩 달라지는 현상과 서로 다른 마이크를 사용하여 음성 데이터를 받음으로써 생기는 왜곡, 배경 잡음 등을 감소시키기 위해 적용한다. 대표적인 채널/세션 보상 방법으로는 LDA(Linear Discriminant Analysis), NAP(Nuisance Attribute Projection) & WCCN(Within Class Covariance Normalization) 등이 있다.

i-벡터의 길이 정규화 방법은 점수 계산 방법과 큰 연관이 있다. i-벡터의 대표적인 점수 계산 방법으로 코사인 유사도(cosine similarity)와 PLDA(Probabilistic Linear Discriminant Analysis)[*7] 점수가 있다. 코사인 유사도 방법은 등록된 i-벡터와 인식하는 i-벡터의 각도를 점수화한 방법이고, PLDA 점수는 학습 발성과 인식 발성의 화자가 동일한 가정에서의 가능도와 다른 화자라 가정하는 상황에서의 가능도의 결과를 비율로 수치화 한다. PLDA의 가능도는 사전 분포(prior distribution)의 정의에 따라 HT-PLDA(Heavy-tailed PLDA)[*8]와 G-PLDA(Gaussian PLDA)[*9]로 나눌 수 있는데, HT-PLDA는 사전 분포를 Student's t 분포를 사용하며, G-PLDA에서는 Gaussian 분포를 사용한다. 코사인 유사도와 HT-PLDA는 i-벡터의 정규화를 수행하지 않으나, G-PLDA에서는 i-벡터에 길이 정규화를 수행한다. i-벡터에 길이 정규화를 수행하면 i-벡터의 분포가 가우시안 분포를 따르기 때문이다. i-벡터의 길이 정규화 방법에는 단순히 길이를 1로 만들어 주는 방법과, i-벡터의 요소(element)들의 크기를 기준으로 가우시안 정규화를 수행하는 순위 정규화(rank normalization), i-벡터를 단위 구체(unit sphere) 위에 위치하도록 변화 시키는 구면 길이 정규화(spherical length normalization)등이 있다[*10].

화자의 승인, 거부를 결정하는 기준점을 좀 더 명확히 찾기 위해 점수 정규화를 수행할 수 있다. 대표적인 점수 정규화 방법에는 Z-norm(Zero normalization), T-norm(Test normalization), ZT-norm(Z-norm + T-norm), S-norm(Symmetric normalization) 등이 있다. [그림 8]은 i-벡터/PLDA 기반의 화자 확인 과정을 보여 준다.

[그림 8] i-vector/PLDA 기반의 화자 확인 과정

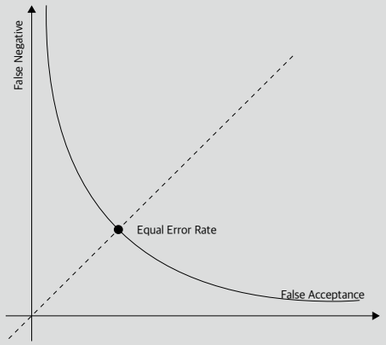


평가 방법

화자 확인의 평가 방법에는 EER(Equal Error Rate)과 minDCF(minimum Detection Cost Function)가 있다. 화자 확인은 [그림 9]에서 보듯이 특정 기준점(threshold)에 의해 오인식(false acceptance, FA)과 부정오류(false negative, FN)의 발생량이 달라지기 때문이다. 여기서 FA는 등록되지 않은 화자가 발성했지만 등록된 화자로 잘못 승인하는 경우를 말하며, FN은 등록된 화자가 발성했지만 등록된 화자가 아닌 것으로 잘못 거절한 경우를 말한다. 기준점을 높게 잡으면 FA가 적게 발생하지만 반대로 FN이 높아진다. 반대로 기준점을 낮게 잡으면 FA가 많이 발생하는 반면 FN이 적게 발생한다. 이러한 이유로 화자 확인은 EER과 minDCF

같은 평가 방법을 이용한다. EER은 FA와 FN가 동일하게 발생하는 기준값에서의 오류율을 말하며, [그림 9]는 EER이 정해지는 위치를 보여 준다.

[그림 9] Equal Error Rate



minDCF는 미국 국립표준기술연구소(National Institute of Standards Technology, NIST)*11에서 주관하는 화자인식 대회에서 사용하는 평가 방법이며, minDCF는 [수식 6]과 같다.

[수식 6]

$$C_{Det} = C_{Miss} \times P_{Miss|Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1 - P_{target})$$

오류 검출(false alarm)은 오인식(FA)와 같고, Miss는 오류 검출(FN)과 같다. [수식 6]에 사용되는 파라미터는 [표 1]과 같다.

[표 1] Detection Cost Model Parameters*12

C_{Miss}	$C_{FalseAlarm}$	P_{Target}
10	1	0.01

마치며

지금까지 고전적인 화자인식 방법과 최근에 사용하는 화자인식 방법들을 살펴보았다. 화자인식은 기존의 문제들을 조금씩 해결하는 방향으로 발전하고 있다. 최근에 화제가 된 딥러닝을 부분적으로 적용하는 시도들뿐만 아니라 end to end 방식으로 성능을 향상하려는 시도들도 많이 소개되고 있다. 그러나 화자인식의 특성상 한 화자에 대해 많은 정보를 얻기 힘들기 때문에, i-벡터 / PLDA 방법이 여전히 최고의 자리를 유지하고 있다. 현재 화자인식은 기존의 통계적인 방법과 딥러닝을 적절히 조합하는 방법으로 진화하고 있으며, 음성을 사용하는 개인화된 서비스와 융화되기를 기다리고 있다.

***1** 논문 | Reynolds, D. A. & Richard C. R. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models, IEEE transactions on speech and audio processing. ***2** 논문 | Reynolds, D. A. (1995). Speaker identification and verification using Gaussian mixture speaker models, Speech Communication. ***3** 논문 | Campbell, W. M., Strurim, D. E., Reynolds, D. A. & Solomonoff, A. (2006). SVM based speaker verification using a GMM supervector kernel and NAP variability compensation, ICASSP. ***4** 논문 | Kenny, P., Boulianne, G., Ouellet, P. & Dumouchel, P. (2005). Factor analysis simplified [speaker verification applications], ICASSP. ***5** 논문 | Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P & Ouellet P. (2011). Front-end factor analysis for speaker verification, IEEE Transactions on Audio, Speech, and Language Processing. ***6** 논문 | Snyder, D., Garcia-Romero, D. & Povey, D. (2015). Time delay deep neural network-based universal background models for speaker recognition, ASRU. ***7** 논문 | Prince S. J. & Elder J. H. (2007). Probabilistic linear discriminant analysis for inferences about identity, ICCV. ***8** 논문 | Kenny, P. (2010). Bayesian Speaker Verification with Heavy-Tailed Priors, Odyssey. ***9** 논문 | Garcia-Romero, D. & Espy-Wilson, C. (2011). Analysis of i-vector Length Normalization in Speaker Recognition Systems, Interspeech. ***10** 논문 | Shum, S., Dehak, N., Dehak, R. & Glass, J. (2010). Unsupervised Speaker Adaptation based on the Cosine Similarity for Text-Independent Speaker Verification, Odyssey. ***11** 참고 | NIST, "The NIST Year 2010 Speaker Recognition Evaluation Plan," Odyssey, 2010. ***12 참고 |** [수식 6]에 사용하는 파라미터 [표 1]은 NIST SRE(Speaker Recognition Evaluation) 평가 년도마다 조금씩 차이가 나며 [표 1]은 NIST SRE 10을 기준으로 한다.