Deep Learning for Audio

Lecture 9

Pavel Severilov

AI Masters

2024

Outline

1. Speaker verification and identification
2. Metric learning
3. Triplet loss
4. Angular softmax
5. ArcFace

Outline

Metric learning

- Task: determine how similar are two objects
- Data:
  - Supervised: labeled objects
  - Unsupervised: set of similar or dissimilar pairs
- Applications:
  - Few-shot learning
  - Biometrics
  - Face recognition (who is on the photo?)
  - Speaker verification (who is speaking?)
  - Large amount of rare classes

Speaker recognition: base pipelines

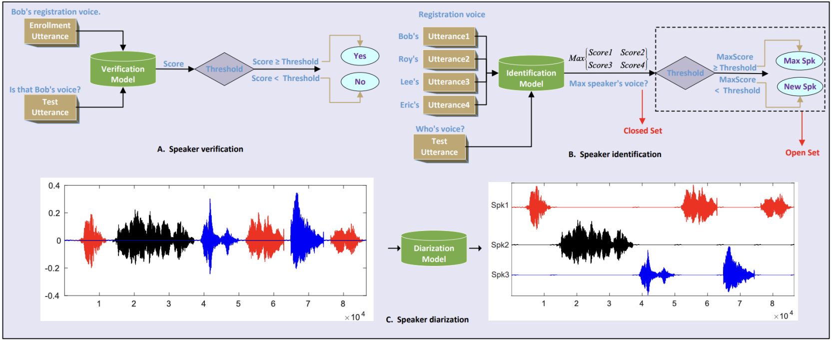
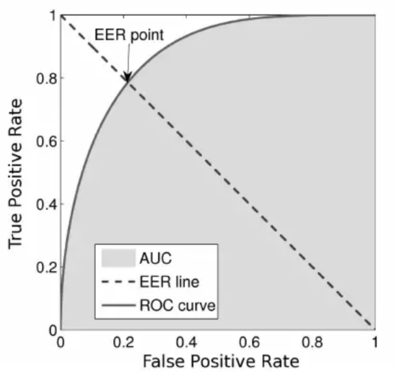


Figure: Flowcharts of speaker verification, speaker identification, and speaker diarization

Speaker verification: metrics



- model(audio1.wav, audio2.wav) $= 1.03$ - need thresholds
- False positive - true: different, prediction: same
- False negative - true: same, prediction: different
- Equal Error Rate (EER) - the point where TPR $==$ FPR for a given model

VoxCeleb2 dataset

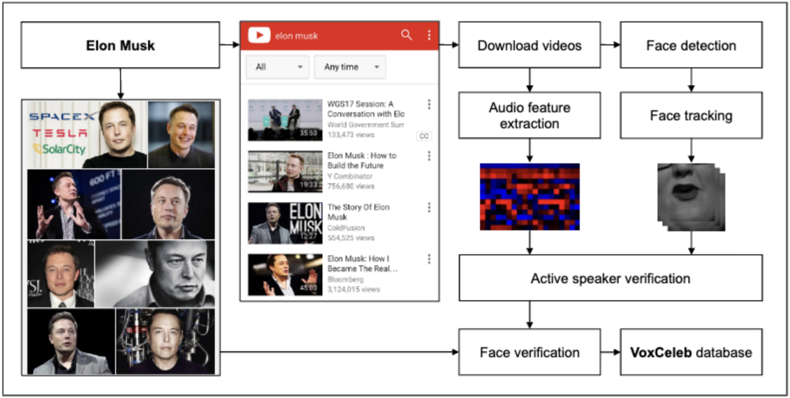
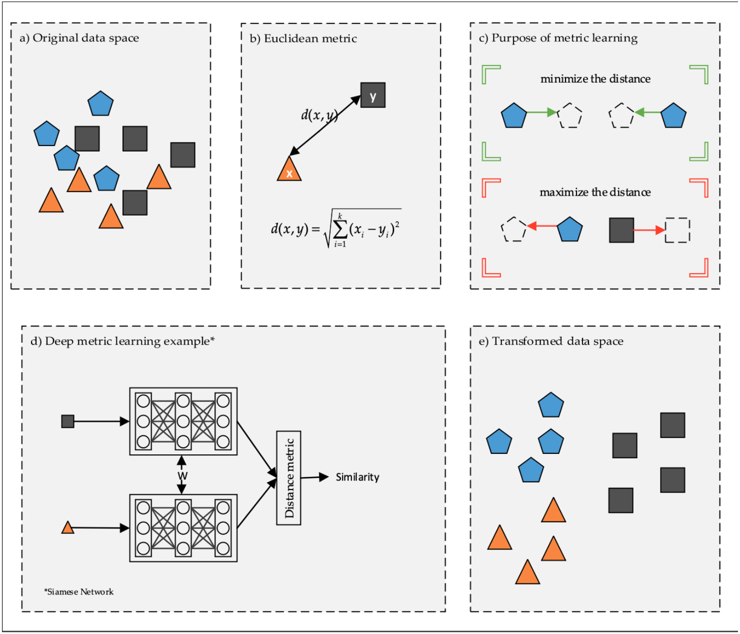


Figure: Data processing pipeline

Outline

1. Speaker verification and identification
2. Metric learning
3. Triplet loss
4. Angular softmax
5. ArcFace

Metric learning idea



Cosine loss

Idea: “attract” everything that have to be close, “spread” everything that shouldn't be close. Problems?

$$\text{loss}(x, y) = \begin{cases} 1 - \cos(x_1, x_2), & \text{if } y = 1 \\ \max(0, \cos(x_1, x_2) - \text{margin}), & \text{if } y = -1 \end{cases}$$

Contrastive loss

Why should we "spread" everything that shouldn't be close? Let's spread only the objects that are too close.

Similarity fn

$$L_{+}(x_1, x_2) = \frac{1}{4}(1 - E_{\mathrm{W}})^2$$

$$L_{-}(x_1, x_2) = \begin{cases} E_{\mathrm{W}}^2 & \text{if } E_{\mathrm{W}} < m \\ 0 & \text{otherwise} \end{cases}$$

Outline

Triplet loss

Why should we even move anything, if everything is fine?

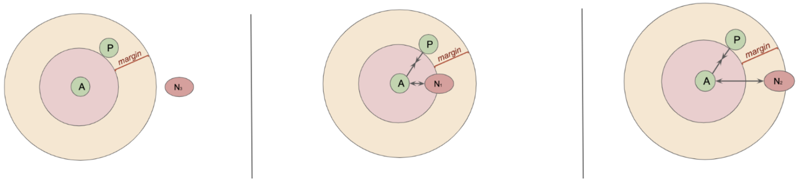


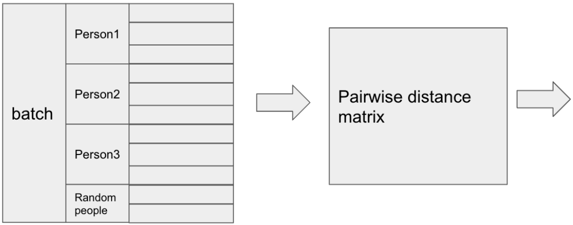
Figure: Easy negative, hard negative, semi-hard negative

$$\mathcal{L}_{\text{triplet}}\left(\mathbf{x}, \mathbf{x}^{+}, \mathbf{x}^{-}\right) =$$

$$= \sum_{\mathbf{x} \in \mathcal{X}} \max \left(0, \|f(\mathbf{x}) - f\left(\mathbf{x}^{+}\right)\|_2^2 - \|f(\mathbf{x}) - f\left(\mathbf{x}^{-}\right)\|_2^2 + \epsilon\right)$$

Triplet mining: how to sample triplets?

- Offline (compute all the embeddings on the training set, and then only select hard or semi-hard triplets)
- Online:



- Don't mine positives, just take each positive with each positive
- For each positive pair (Anchor, Positive) choose negative example so that , loss > 0

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2$$

Outline

Angular softmax

Softmax function:

$$p_1 = \frac{e^{\mathbf{W}_1^T \mathbf{x} + b_1}}{e^{\mathbf{W}_1^T \mathbf{x} + b_1} + e^{\mathbf{W}_2^T \mathbf{x} + b_2}}$$

$$p_2 = \frac{e^{\mathbf{W}_2^T \mathbf{x} + b_1}}{e^{\mathbf{W}_1^T \mathbf{x} + b_1} + e^{\mathbf{W}_2^T \mathbf{x} + b_2}}$$

- Let's remove biases, so that $b_i = 0$. Then, the decision boundary:

  $$(\mathbf{W}_1^T - \mathbf{W}_2^T)\mathbf{x} = 0 \rightarrow (||\mathbf{W}_1|| \cos(\theta_1) - ||\mathbf{W}_2|| \cos(\theta_2))||\mathbf{x}|| = 0$$

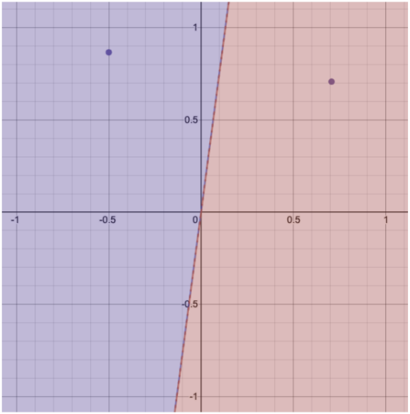
- Let's normalize weights, so that $||W_i|| = 1$

  $$(||\mathbf{W}_1|| \cos(\theta_1) - ||\mathbf{W}_2|| \cos(\theta_2))\, ||\mathbf{x}|| = 0 \rightarrow cos(\theta_1) - cos(\theta_2) = 0$$

Angular softmax

$$cos(\theta_1) - cos(\theta_2) = 0$$

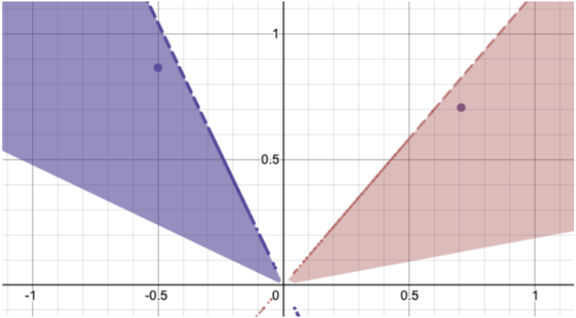
Example: decision boundary for $W_1 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right), W_2 = \left(\frac{-1}{2}, \frac{\sqrt{3}}{2}\right)$



Angular softmax

Let's introduce a more restrictive condition (m - positive integer)

- class 1, if $\cos(m\theta_1) > \cos(\theta_2)$
- class 2, if $\cos(m\theta_2) > \cos(\theta_1)$



Then the loss function:

$$L = -\log \frac{e^{\left\|\mathbf{x}^{(n)}\right\| \cos\left(m\theta_{y_n}^{(n)}\right)}}{e^{\left\|\mathbf{x}^{(n)}\right\| \cos\left(m\theta_{y_n}^{(n)}\right)} + \sum_{j \neq y_n} e^{\left\|\mathbf{x}^{(n)}\right\| \cos\left(\theta_j^{(n)}\right)}}$$

Angular softmax: meaning of m

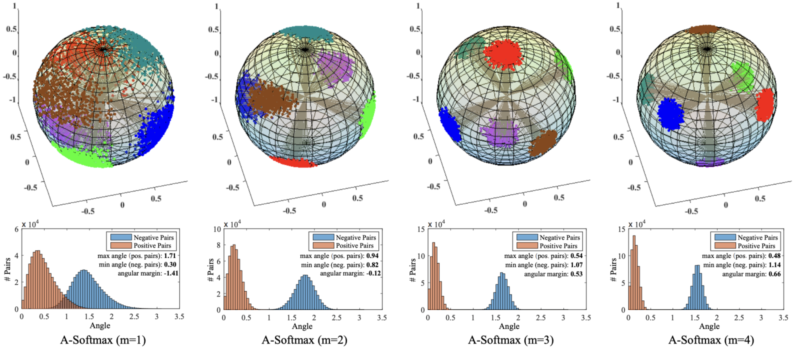


Figure: Visualization of features learned with different m.

- First row: 3D features projected on the unit sphere.
- Second row: the angle distribution of both positive pairs and negative pairs

Outline

ArcFace

Based on Angular softmax:

- Remove biases
- Normalize weights
- $+$ Normalize features x
- $+$ Additive m instead of multiplication

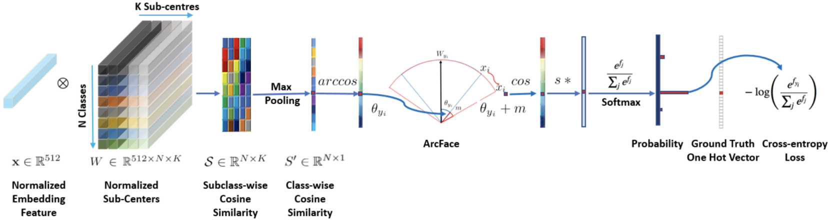
Angular softmax loss:

$$L = -\log \frac{e^{\left\|\mathbf{x}^{(n)}\right\| \cos\left(m\theta_{y_n}^{(n)}\right)}}{e^{\left\|\mathbf{x}^{(n)}\right\| \cos\left(m\theta_{y_n}^{(n)}\right)} + \sum_{j \neq y_n} e^{\left\|\mathbf{x}^{(n)}\right\| \cos\left(\theta_j^{(n)}\right)}}$$

Arcface loss:

$$L = -\log \frac{e^{s \cos\left(\theta_{y_i} + m\right)}}{e^{s \cos\left(\theta_{y_i} + m\right)} + \sum_{j=1, j \neq y_i}^{N} e^{s \cos \theta_j}}$$

ArcFace: pipeline



ArcFace: geometric interpretation

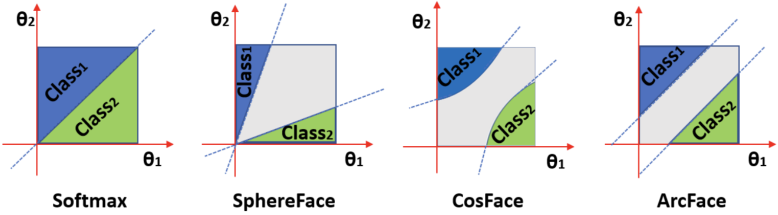


Figure: Decision margins of different loss functions under binary classification case. The dashed line represents the decision boundary, and the grey areas are the decision margins.