Ordered and Binary Speaker Embedding

Jiaying Wang

2023.04.03

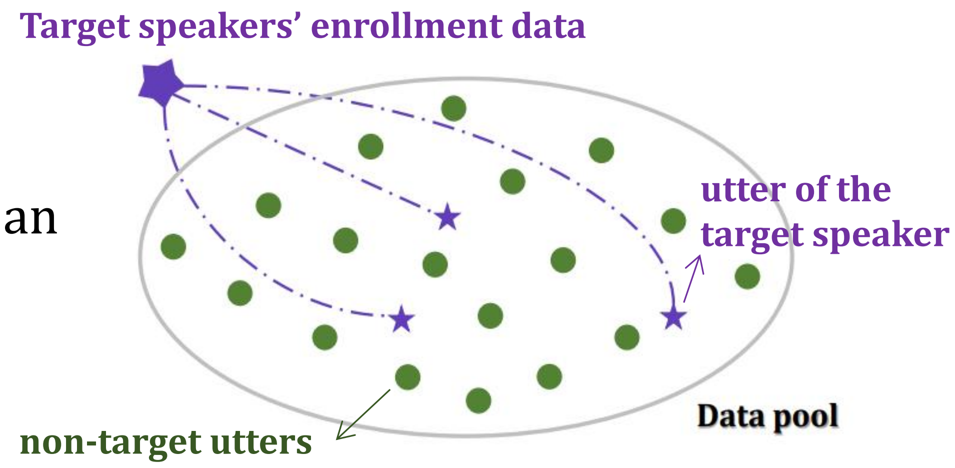
Outline

- Introduction
- Method
  - oAE
  - obAE
- Experiments
  - Baseline
  - Orderliness Test
  - Binary Test
  - Bit Test
  - Speed Test
- Conclusion

Introduction

What is speaker retrieval

- Concept
  - Speaker Retrieval (SR) task is to find out the utterances spoken by a target speaker from a large data pool, given an enrollment data of the target speaker.



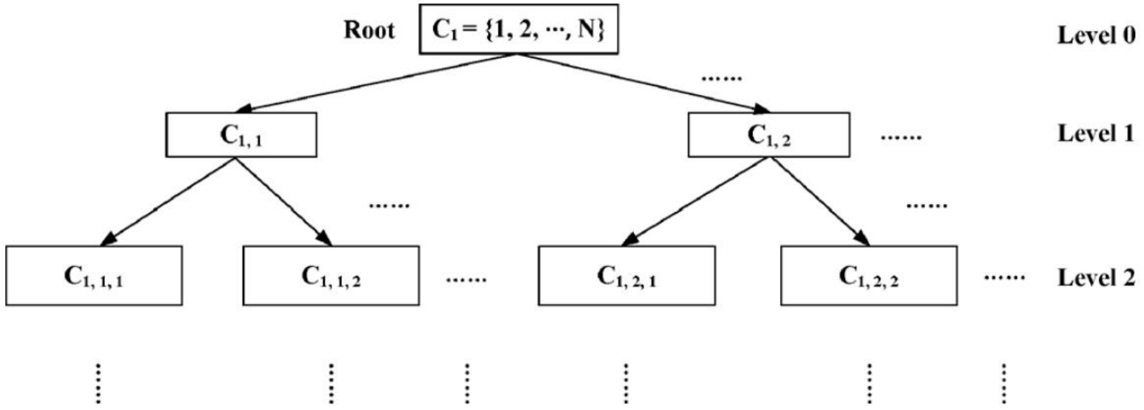
- Value of fast retrieval
  - For instance, identifying a person from 1,251 candidates using 32-dim x-vectors costs 50 ms on a 1.2 GHz CPU and with the highly optimized Scipy package. This amounts to **15.5 hours** of CPU time per query if the size of candidates is 1.4 billion, the population of China.

Introduction

Current research on fast retrieval

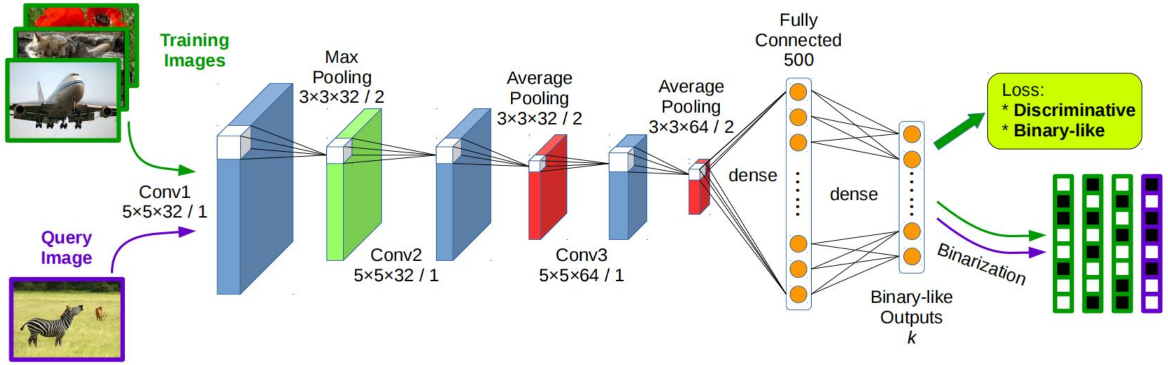
- Hierarchical clustering

  Enrolled speakers are clustered according to their similarity and multiple-level clustering forms a decision tree



- Binary code

  This approach converts dense embeddings to binary codes, and the similarity is computed as the Hamming distance



Weekness

- the depth of the tree must be carefully controlled

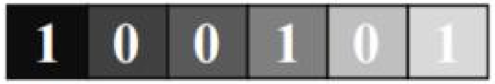
Weekness

- lose precision
- the search still needs to scan all candidates

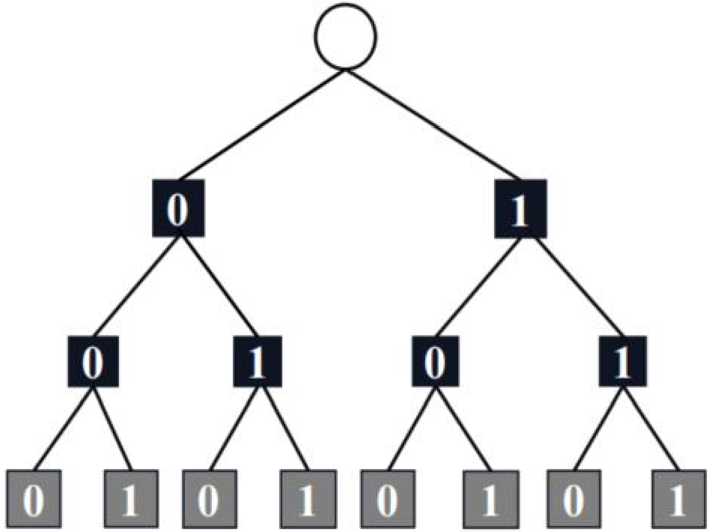
Introduction

Why ordered binary embeddings

- **ordered and binary embedding**
  - non-structual x-vector --> ordered binary (OB) code
  - combination of hierarchical clustering and binary code

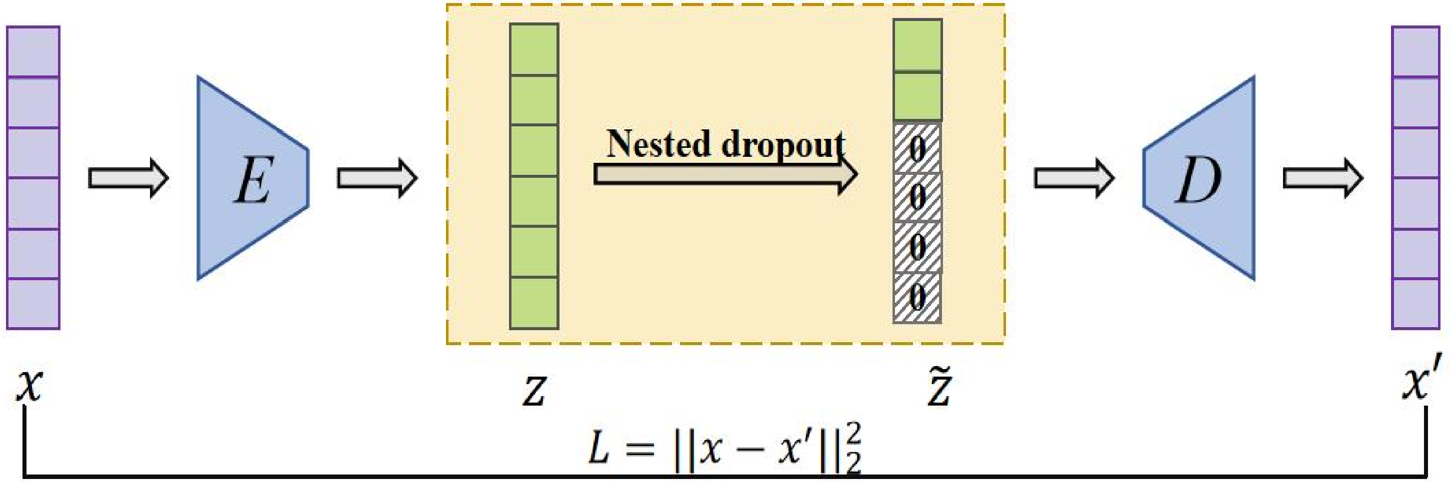


(a) OB code

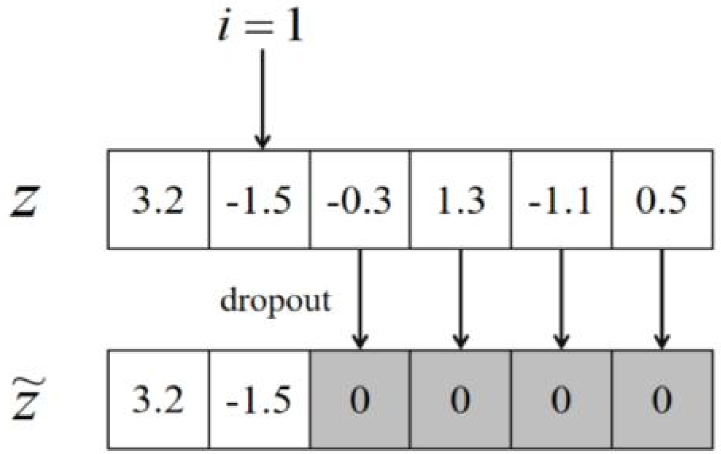


(b) Binary tree formed by OB codes

Method - **Ordered** AE (oAE)

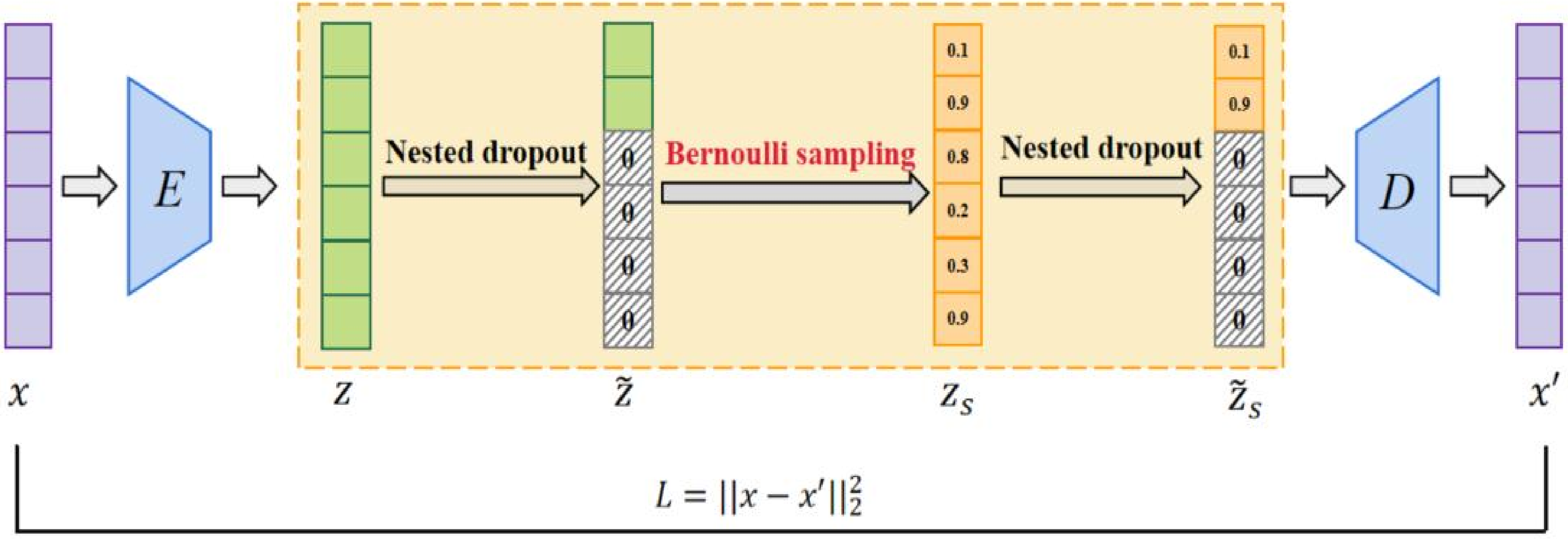


(a) oAE structure



(b) nested dropout

Method - **Ordered Binary** AE (obAE)



obAE structure

Method

Reparametric Trick in obAE

- Using reparametric trick, let u is a uniform distribution

$$z_{si} = \begin{cases} 1 & u_i \le p_i \\ 0 & u_i > p_i \end{cases}$$

- Due to this trick, graident can BP to p, which further BP to the encoder.

$$\Downarrow$$

- T stands for temperature coefficient. The smaller the value T, the higher the probability that z_s concentrates around 0 or 1.

$$z_s = \sigma\left(\frac{\log(\frac{u}{1-u}) + \log(\frac{p}{1-p})}{T}\right)$$

Experiments

- Data

Tabel 1: Data description

VoxCeleb	Train VoxCeleb2.dev	Enroll VoxCeleb1	Test VoxCeleb1
# of Spks # of Utters	5,994 1,092,009	1,251 3,753	1,251 149,763
CN-Celeb	Train CN-Celeb.T	Enroll CN-Celeb.E	Test CN-Celeb.E
# of Spks # of Utters	2,793 632,740	196 196	196 14,124

- Baseline

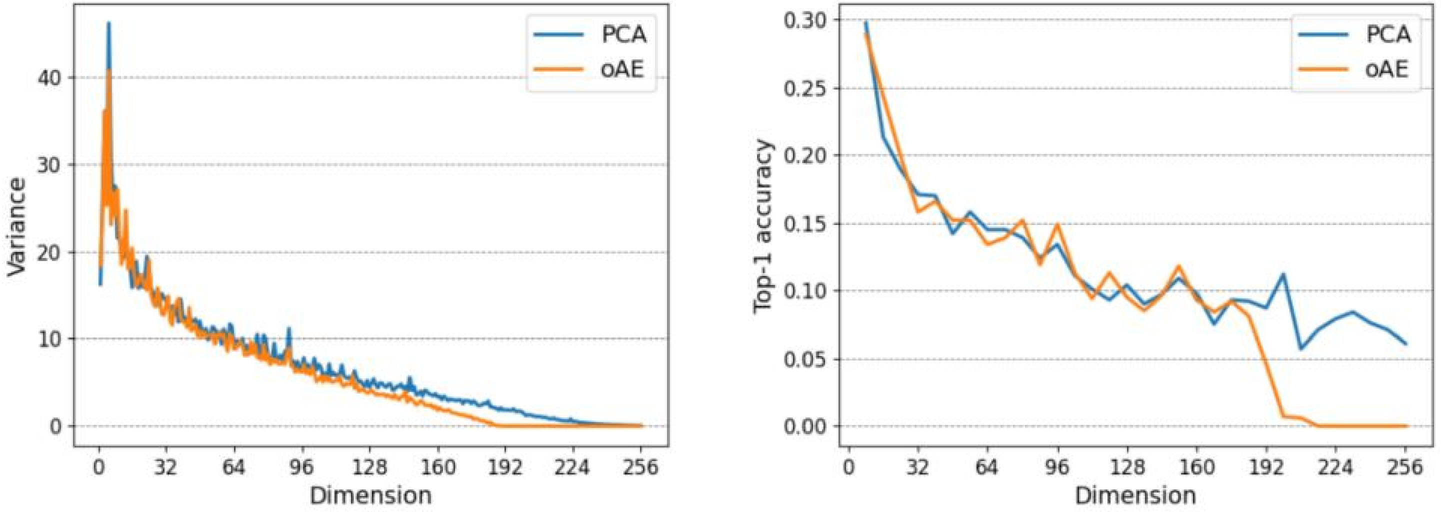
Tabel 2: Top-k accuracy with the original x-vector

	VoxCeleb1	CN-Celeb.E
Top-1	0.959	0.706
Top-3	0.984	0.800
Top-5	0.989	0.844

Experiments

- Orderliness Test

Figure 1: Orderliness test with PCA and oAE, using x-vectors in VoxCeleb1



(a) *Variance test* (b) *Performance test*

Conclusion:

- ➢ oAE learns ordered representations as PCA
- ➢ oAE performs even better than PCA on tail dimensions

Experiments

- Binary Test

Table 3: Top-k Acc on VoxCeleb1.

Bits		20	40	80	120	160	256
LSH	Top-1	0.094	0.273	0.543	0.684	0.759	0.847
	Top-3	0.176	0.412	0.689	0.808	0.865	0.924
	Top-5	0.227	0.481	0.747	0.851	0.898	0.945
PCA-LSH	Top-1	0.126	0.350	0.599	0.709	0.768	0.847
	Top-3	0.233	0.514	0.746	0.830	0.872	0.924
	Top-5	0.297	0.588	0.799	0.870	0.904	0.945
obAE	Top-1	0.182	0.440	0.681	0.779	0.813	0.824
	Top-3	0.316	0.616	0.822	0.890	0.911	0.913
	Top-5	0.392	0.690	0.870	0.923	0.939	0.939

Table 4: Top-k Acc on CN-Celeb.E.

Bits		20	40	80	120	160	256
LSH	Top-1	0.157	0.293	0.432	0.502	0.543	0.595
	Top-3	0.266	0.410	0.546	0.612	0.653	0.703
	Top-5	0.326	0.471	0.602	0.664	0.702	0.750
PCA-LSH	Top-1	0.183	0.329	0.462	0.519	0.544	0.595
	Top-3	0.323	0.478	0.588	0.634	0.655	0.701
	Top-5	0.403	0.550	0.647	0.687	0.706	0.748
obAE	Top-1	0.197	0.353	0.495	0.551	0.579	0.588
	Top-3	0.357	0.518	0.639	0.688	0.708	0.719
	Top-5	0.446	0.594	0.705	0.743	0.762	0.774

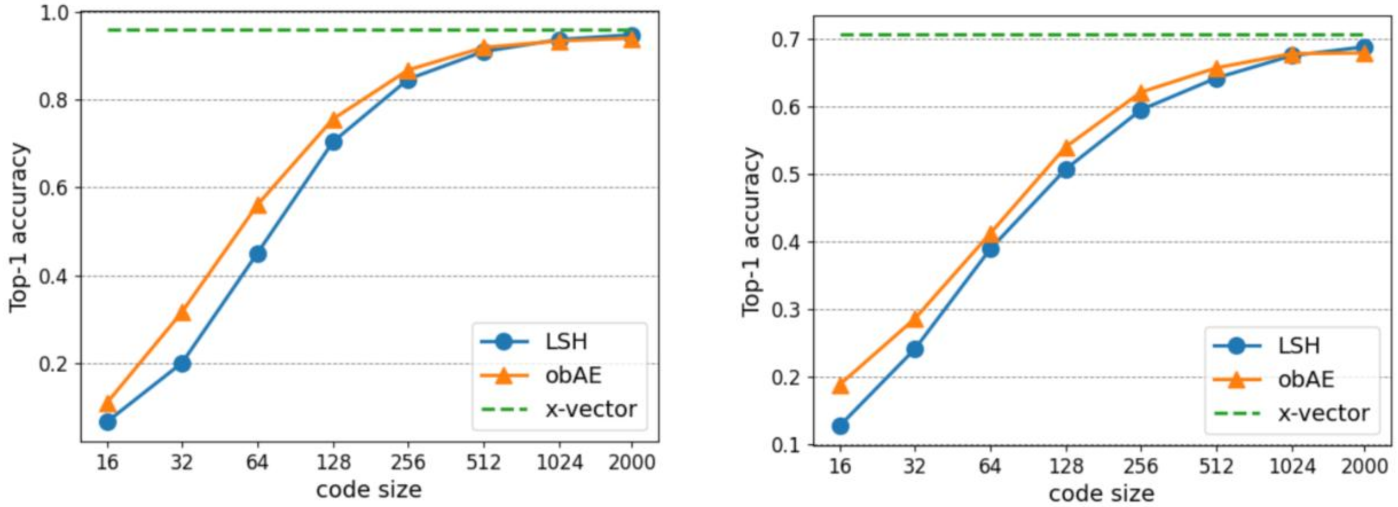
Conclusion：

- limited code: obAE shows a clear advantage over LSH and PCA-LSH
- The performance of the three codes tends to be the same

Experiments

- Bit Test

Figure 2: Bit test with LSH and obAE codes



(a) *VoxCeleb1* (b) *CN-Celeb.E*

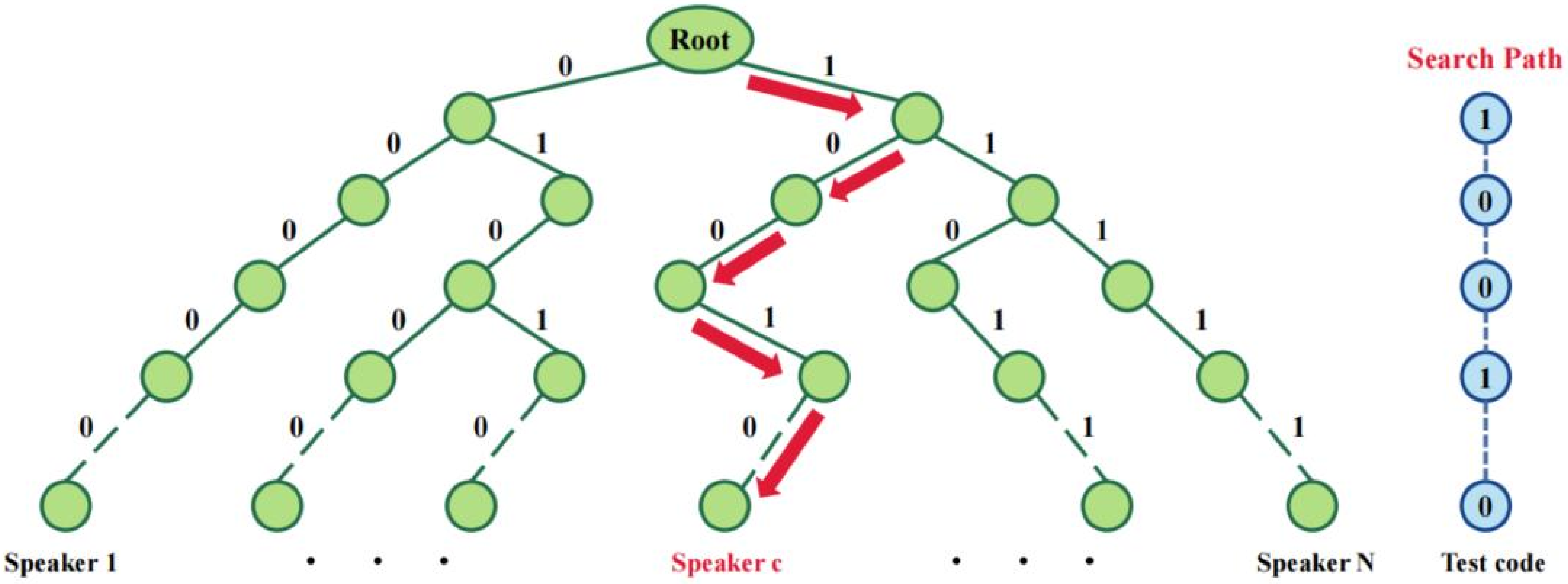
Conclusion：

- ➢ Compared to x-vector, OB codes obtain a competitive performance while code size=1,000
  - ➢ binary codes may represent the full information with less storage
- ➢ obAE can achieve better performance with limited code capacity ---> orderliness

Experiments

- Speed Test

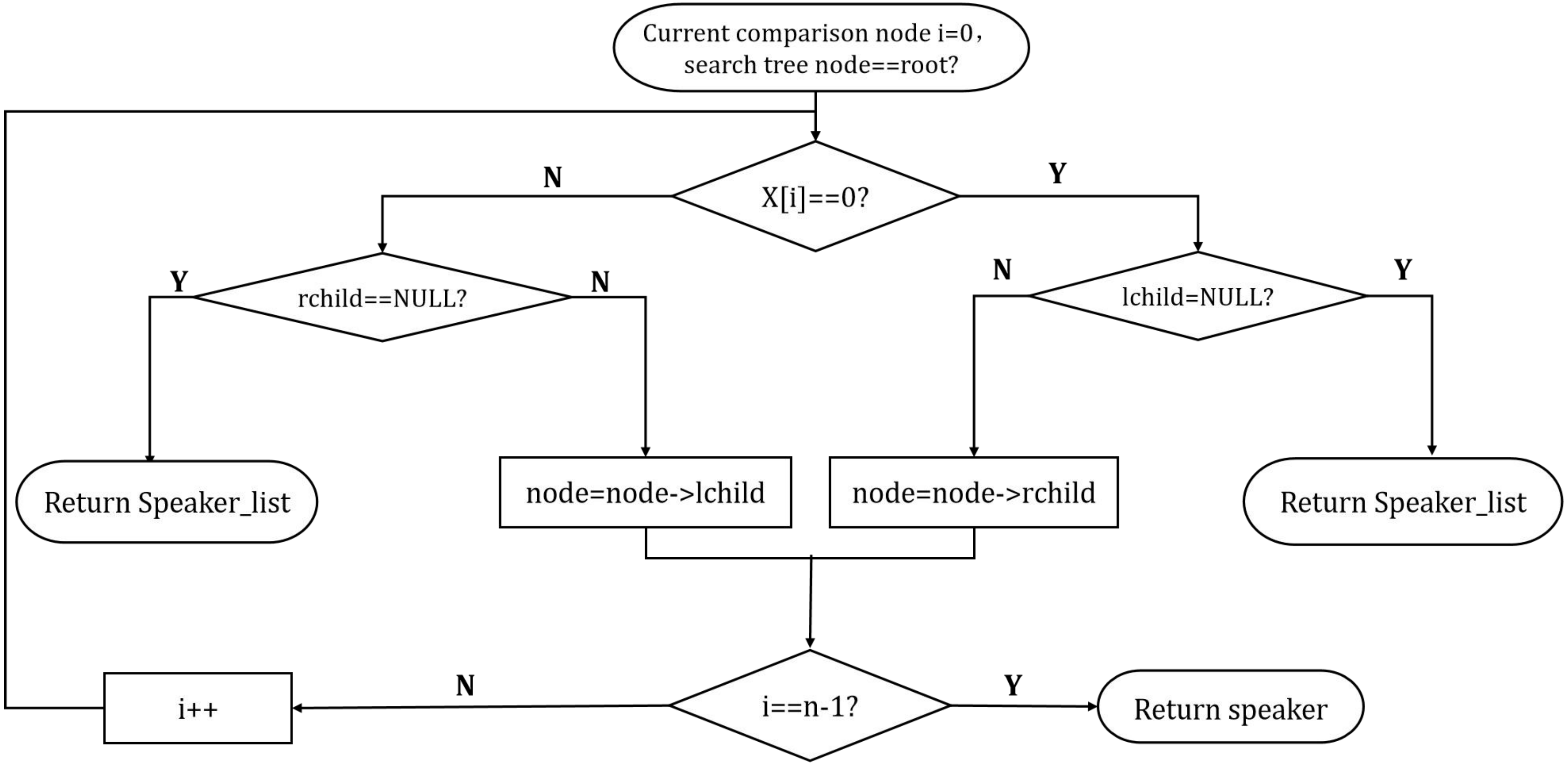
Figure 3: An illustration of the binary search tree



Experiments

- Speed Test

Figure 4: An illustration of the search pipeline



Experiments

- Speed Test

Table 5: Speed test result

Code	Distance	32 dims/bits		40 dims/bits		48 dims/bits	
		Speed	Top-1	Speed	Top-1	Speed	Top-1
Dense	Cosine	52.89	0.950	53.17	1.000	51.87	1.000
OB	Hamming	18.97	0.950	19.84	0.981	19.98	1.000
OB	Binary tree	0.04	0.950	0.05	0.981	0.07	1.000

Conclusion:

- 1,300 times faster than the linear search based on cosine distance
- 450 times faster than the linear search based on Hamming distance

Conclusion

- obAE model
  - nested dropout ---> ordered representations
  - Bernoulli sample ---> binary codes
- The orderliness, SID performance of obAE are proved
- OB codes can bring remarkable speeding up
- Discussion: reconstruction, generation, compression...