Large-Scale Content-Based Audio Retrieval from Text Queries

Rick van der Zwet
<hvdzwet@liacs.nl>

Multimedia Image Retrieval 2010

LIACS
Leiden Institute of Advanced Computer Science

Introduction

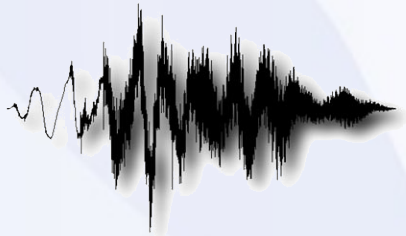
- Google powered paper
- uses free-form text queries
- searches audio-content, rather than textual meta-data
- scale to very large number of audio documents, very rich query vocabulary

Introduction::scope

- Tested using (Passive-Aggressive Model for Image Retrival) compared against:
  - Gaussian Mixture Models
  - Support Vector Machines

Introduction::Sound properties

- short sound effects
- noisier and larger collection of uses-contributes user-labeled recordings
- isolated (sound effects recordings) or combined (movie sound tracks)



Introduction::Earlier work

- few high-level categories
- Wan and Lu: features and metric evaluation
- Turnbull et al: retrieves audio documents based on text queries used training mixture models.

The learning problem

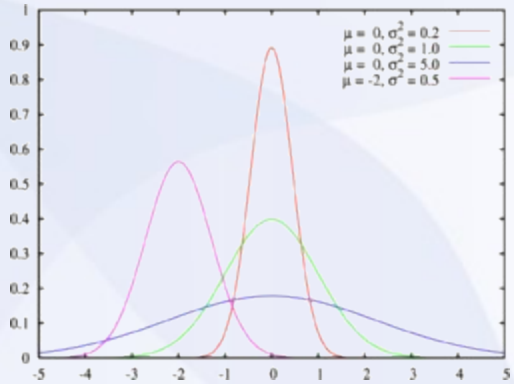
- Text query *q* -> set of audio documents *A*
- Let *R(q,A)* be the set of audio document relevant to *A*
- Optimal result should give *all* relevant entries before the *irrelevant* ones
- The score function *F(q,a)* will be used to order based on decreasing scores.

Multiple terms classification

- Bag-of-words representation
- Using a vector for to build the set.
- Sample: growling lion purring cat.
- Among the terms present in q, the terms appearing rarely in the reference corpus are more discriminant and should be assigned higher weights.

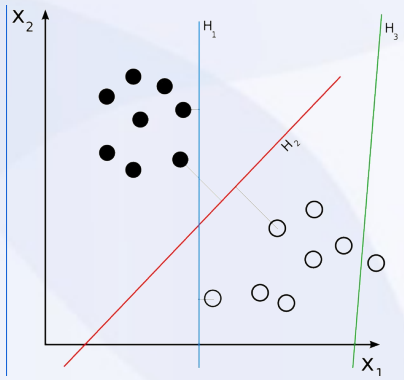
Gaussian Mixture Model

- Model probability density functions of audio documents
- Wrong idea: every frame is independent of all other frames
- Train for every term apart.



Support Vector Machines

- Discriminant function that maximizes the margin between the positive and negative examples
- Minimizing the number of miss classifications using training
- Conflicting goals
- Train for every goal apart



Passive-Aggressive Model for Image Retrieval

- Matrix of with same length of the vector
- Score will go for the dot product
- No ranking-SVM = quadratic, so does not scale
- passive-aggressive (PA) -> try to optimize with threshold for the optional value

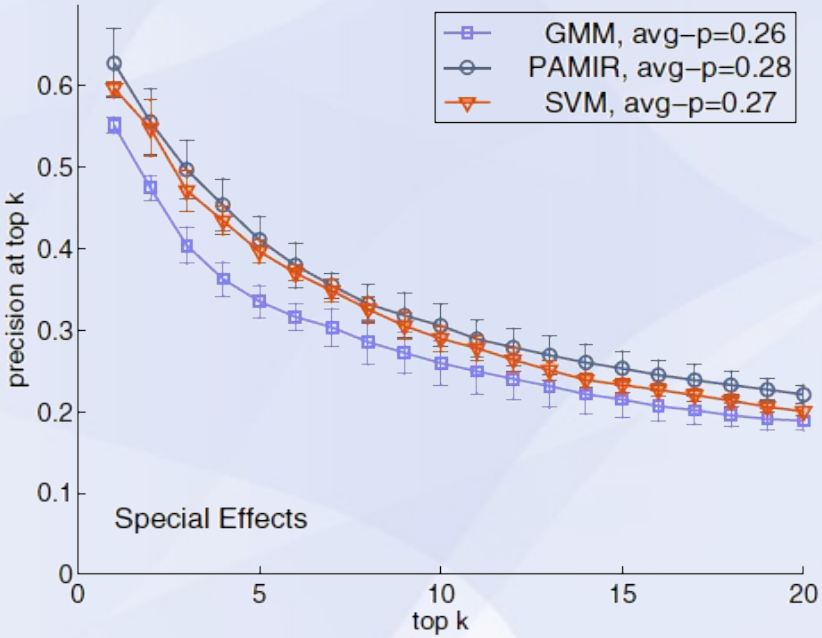
Data-set

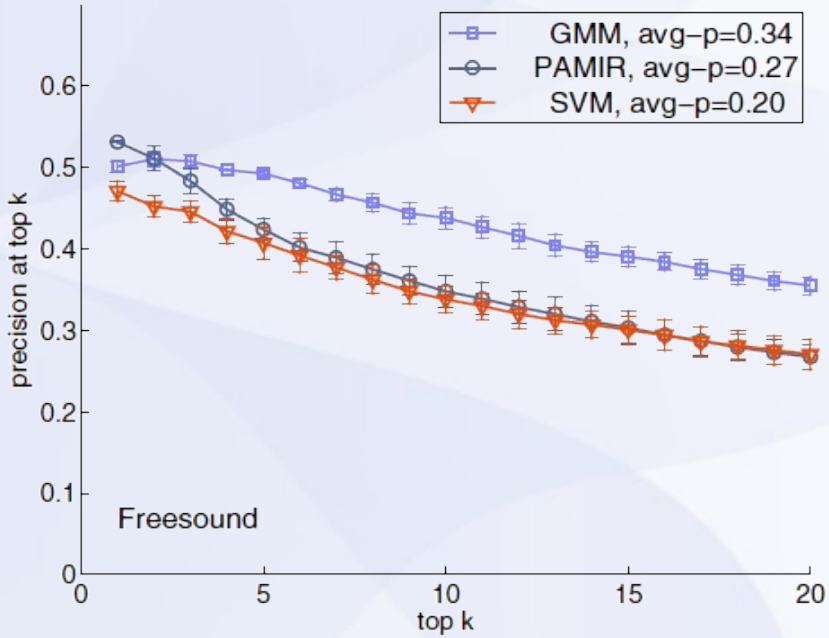
- Various sources
- By hand
- Common words in file-names
- User count being the popular factor
- Acoustic features using MatLab

Experiment

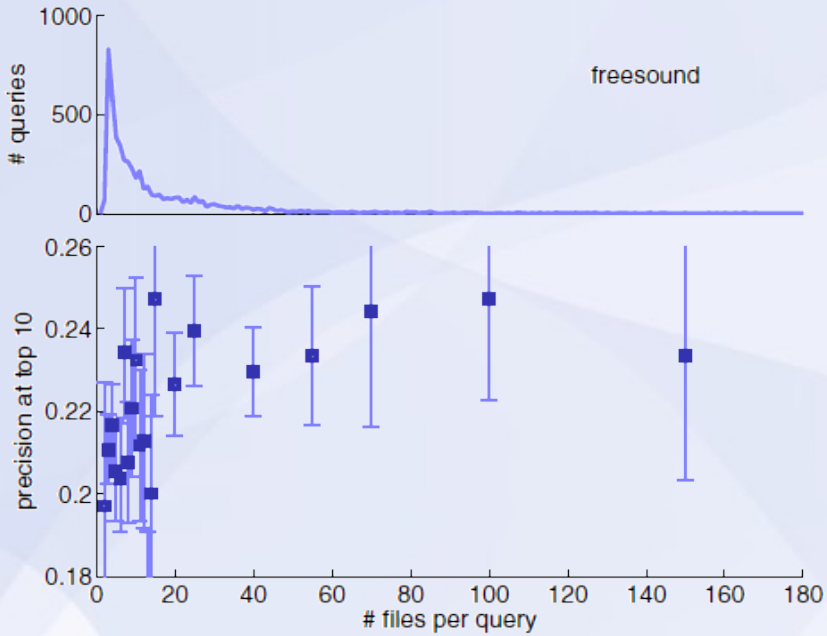
- Use 2/3 of data for training all the rest of testing
- Validate against Human query database of freesound.com
- Match if found labels will actually match the text query for a certain hit.

Results


GMM, avg–p=0.26
PAMIR, avg–p=0.28
SVM, avg–p=0.27
precision at top k
top k
Special Effects
0
0.1
0.2
0.3
0.4
0.5
0.6
0
5
10
15
20


GMM, avg–p=0.34
PAMIR, avg–p=0.27
SVM, avg–p=0.20
precision at top k
top k
Freesound
0
0.1
0.2
0.3
0.4
0.5
0.6
0
5
10
15
20

Error distribution



Speed & Scalability

- PAMIR scales best
- SVM are quadratic with regards to training examples

Table 3: Total experimental time (training+test) times, in hours, assuming a single modern CPU, for all methods and both datasets, including all feature extraction and hyper-parameter selection. File and vocabulary sizes are for a single split as in Table 1.

Data	files	terms	GMMs	SVMs	PAMIR
Freesound	15780	1392	2400 hrs	59 hrs	6 hrs
SFX	3431	239	960 hrs	5 hrs	3 hrs

Discussion

- PAMIR method is scalable
- Noisy real-world labels works fine to extend. But a better way of gathering tags will be preferred.
- Out-of-dictionary
- Dynamic matching, require further research.