TOKYO TECH Pursuing Excellence

COLLABORATIVE TEAM
for TRECVID 2010

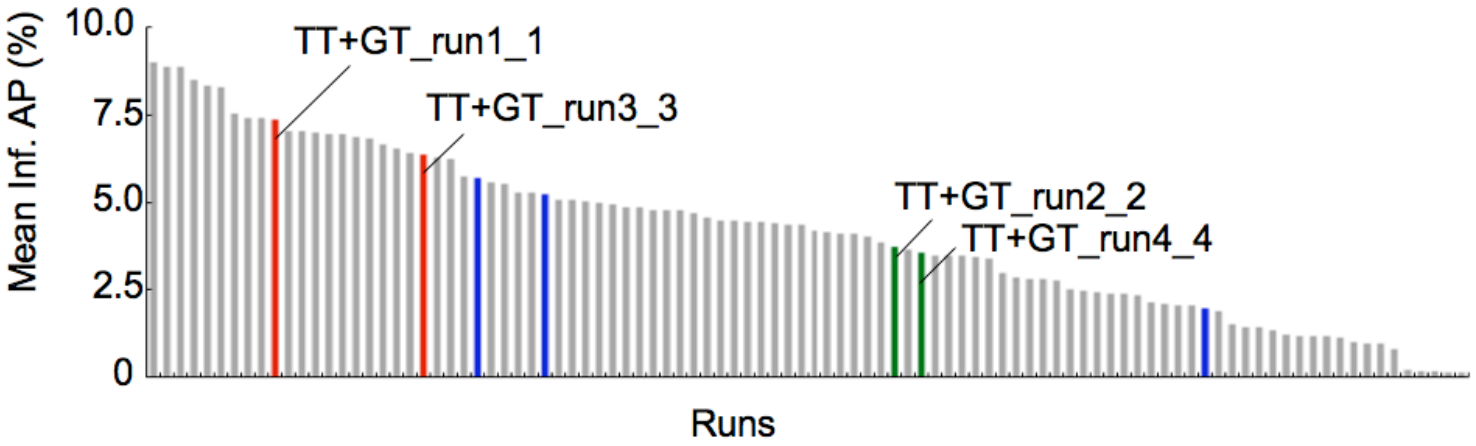
Semantic Indexing Using GMM Supervectors with MFCCs and SIFT features

Nakamasa Inoue, Toshiya Wada,
Yusuke Kamishima, Koichi Shinoda,
Department of Computer Science,
Tokyo Institute of Technology

Ilseo Kim, Byungki Byun
Chin-Hui Lee,
Department of Electrical and Computer Engineering,
Georgia Institute of Technology

Outline

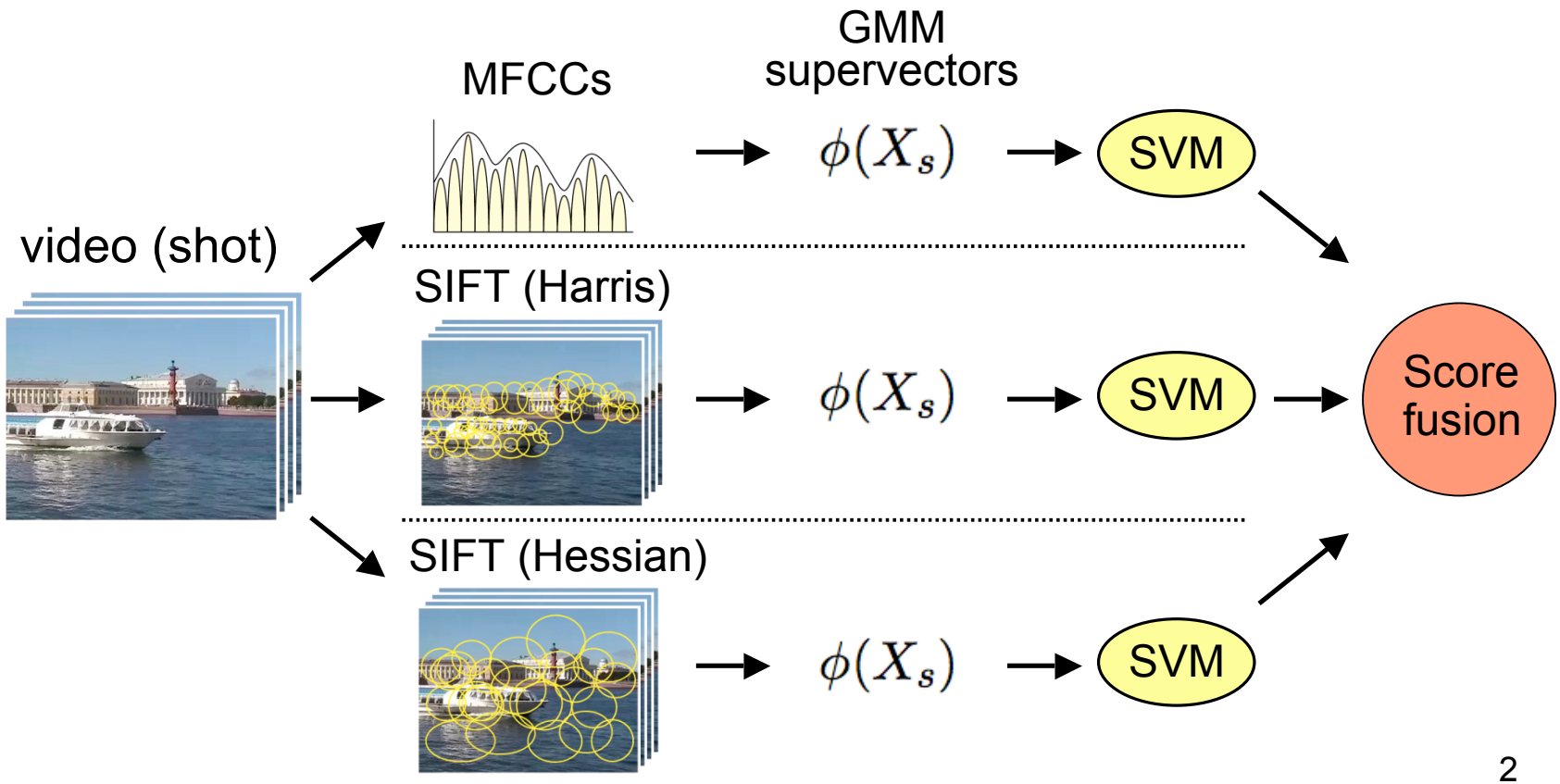
- Part 1:
  - Feature extraction: MFCCs(audio), SIFT(visual)
  - Gaussian mixture model (GMM) supervectors
- Part 2:
  - Maximal Figure of Merit (MFoM) classifier
- Best result: Mean Inf. AP = 7.36%



-- Part 1 --
GMM supervectors with MFCCs and SIFT features

System Overview

- We aim at a simple and accurate multimodal system.
  - ⇨ GMM supervectors with MFCCs and SIFT.



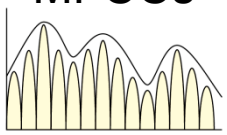
Feature Extraction

- We extract three types of audio and visual features.

video (shot)

Audio features

MFCCs

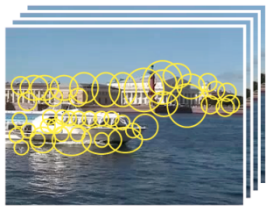


avg.
38 dim, 5,000 features per shot

MFCCs+ΔMFCCs+ΔΔMFCCs+
Δlog-power+ΔΔlog-power

Visual features

SIFT (Harris)



avg.
32 dim, 20,000 features per shot

Multiple detectors

Harris affine and Hessian affine detectors are used.

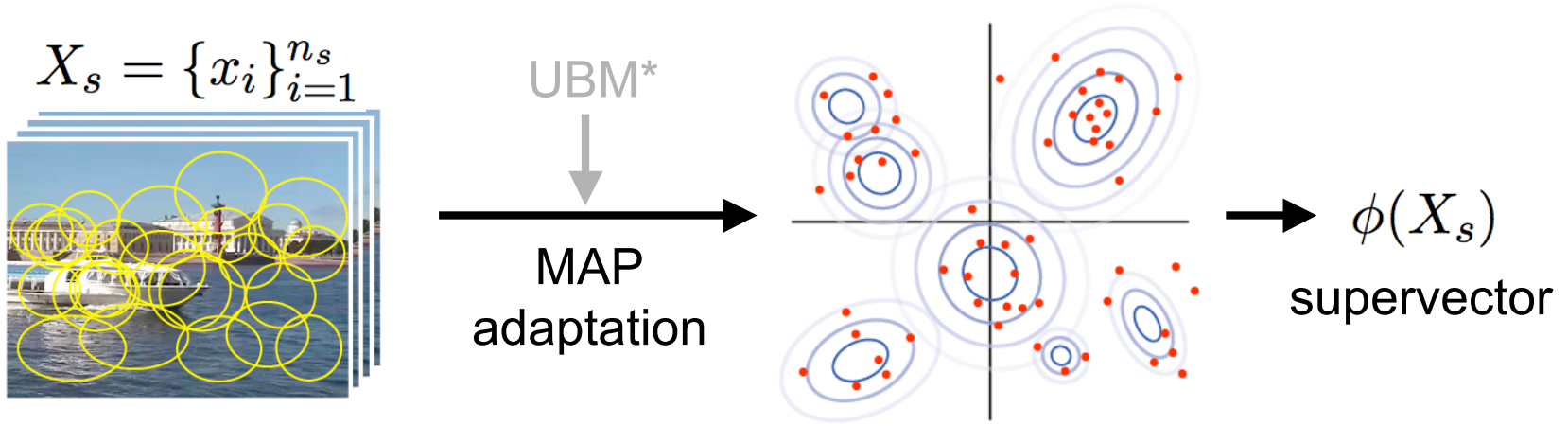
SIFT (Hessian)

Multiple frames

SIFT features are extracted from a half of image frames in a shot.

GMM Supervectors

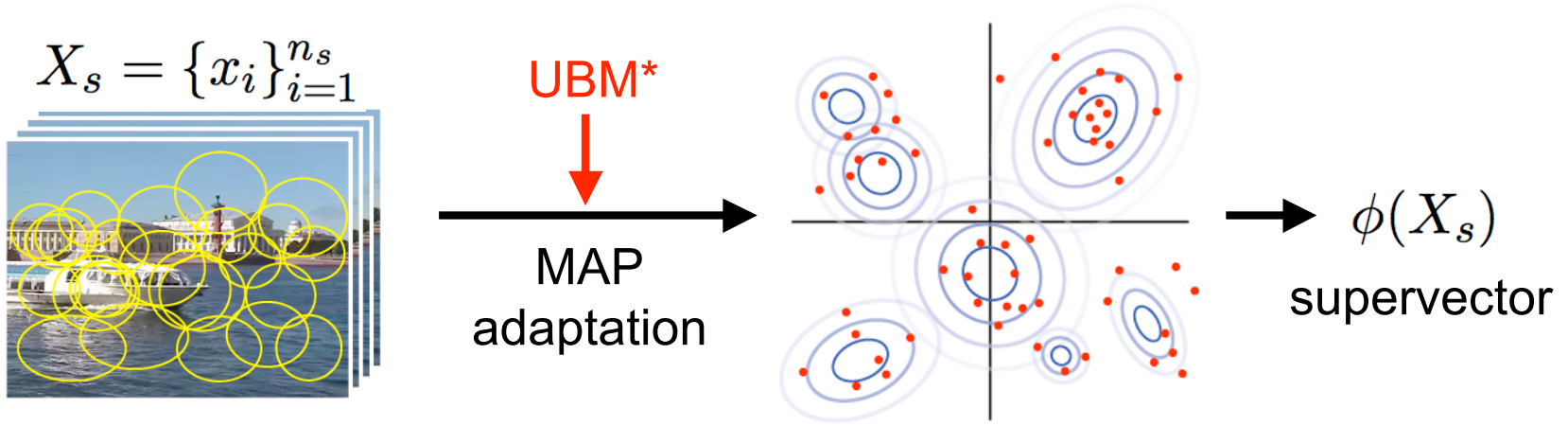
- GMM supervectors and SVMs are used for detection.
  - -- Speaker recognition (W. Campbell et al., 2006)
  - -- Event and object recognition (X. Zhou et al., 2008)
- Each shot is modeled by a GMM.



*Universal background model (UBM): a prior GMM which is estimated by using all video data.

GMM Supervectors

1. Extract a set of features $X_s = \{x_i\}_{i=1}^{n_s}$ (MFCC or SIFT).
2. Train a GMM by Maximum A Posteriori (MAP) adaptation.
3. Create a GMM supervector $\phi(X_s)$.



*Universal background model (UBM): a prior GMM which is estimated by using all video data.

GMM Supervectors (STEP2)

- Adapt mean vectors as follows:

$$\hat{\mu}_k^{(s)} = \frac{\tau \mu_k^{(U)} + \sum_{i=1}^{n_s} c_{ik} x_i}{\tau + C_k}$$

where

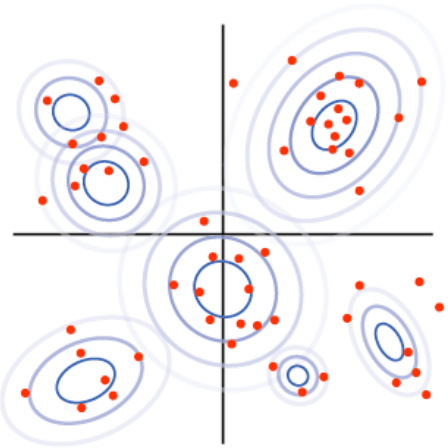
$$c_{ik} = \frac{w_k \mathcal{N}(x_i | \mu_k^{(U)}, \Sigma_k^{(U)})}{\sum_{k=1}^{K} w_k \mathcal{N}(x_i | \mu_k^{(U)}, \Sigma_k^{(U)})}, \quad C_k = \sum_{i=1}^{n_s} c_{ik}$$

Weighted sum of feature vectors at the k-th cluster

$X_s = \{x_i\}_{i=1}^{n_s}$

UBM

MAP adaptation



$\phi(X_s)$

supervector

GMM Supervectors (STEP3)

- GMM supervector: combination of mean vectors.

$$\phi(X_s) = \begin{pmatrix} \tilde{\mu}_1^{(s)} \\ \tilde{\mu}_2^{(s)} \\ \vdots \\ \tilde{\mu}_K^{(s)} \end{pmatrix}$$

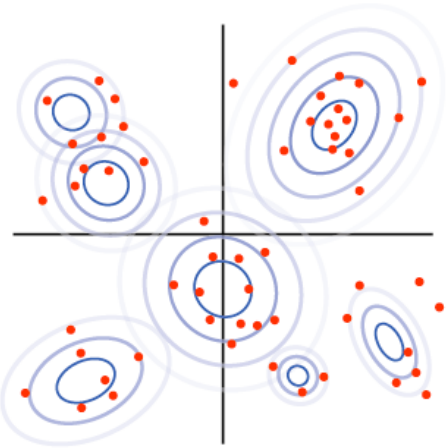
where

$$\tilde{\mu}_k^{(s)} = \underbrace{\sqrt{w_k^{(U)}}(\Sigma_k^{(U)})^{-\frac{1}{2}}}_{\text{normalized}} \underbrace{\hat{\mu}_k^{(s)}}_{\text{mean}}$$

$X_s = \{x_i\}_{i=1}^{n_s}$

UBM

MAP adaptation



$\phi(X_s)$

supervector

SVM Classification

- Train SVMs using an RBF-kernel

$$k(X_s, X_t) = \exp\left(-\gamma \|\phi(X_s) - \phi(X_t)\|_2^2\right)$$

where $\gamma = \tilde{d}^{-1}$, $\tilde{d}$: averaged distance

- Score fusion

$$f = w_{\mathrm{MFCC}} f_{\mathrm{MFCC}} + w_{\mathrm{SIFT_{har}}} f_{\mathrm{SIFT_{har}}} + w_{\mathrm{SIFT_{hes}}} f_{\mathrm{SIFT_{hes}}}$$

$\begin{cases} f_m : \text{detection score for the scheme } m \\ w_m : \text{weight coefficient for the scheme } m \end{cases}$

w_m s are optimized for each semantic concept by two-fold cross validation.

-- Experiments --

Experimental Condition

- Settings

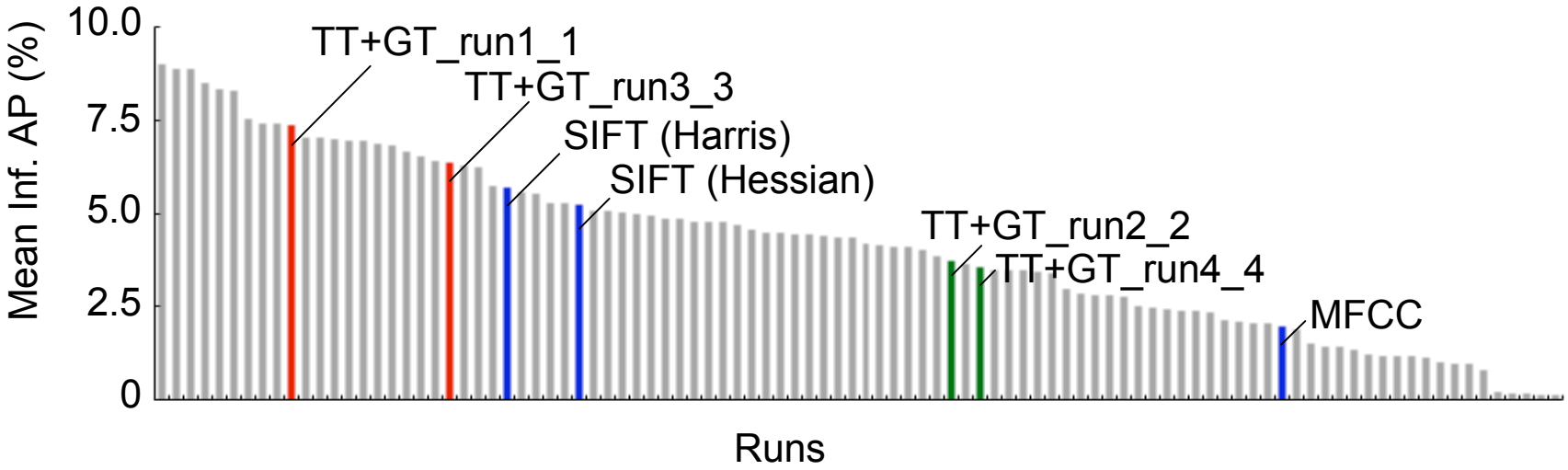
Feature	# of features per shot	Feature dimension	Vocabulary size
MFCC	5,160	38	K = 256
SIFT (Harris affine)	19,536	32 (PCA)	K = 512
SIFT (Hessian affine)	18,986	32 (PCA)	K = 512

- Submitted runs

Run ID	Feature	Classifier
TT+GT_run1_1	MFCC + SIFT (Harris+Hessian)	SVM
TT+GT_run3_3	SIFT (Harris+Hessian)	SVM
TT+GT_run2_2	LSI (Color hist.+Gabor)	MFoM
TT+GT_run4_4	SIFT (Harris)	MFoM

+ audio

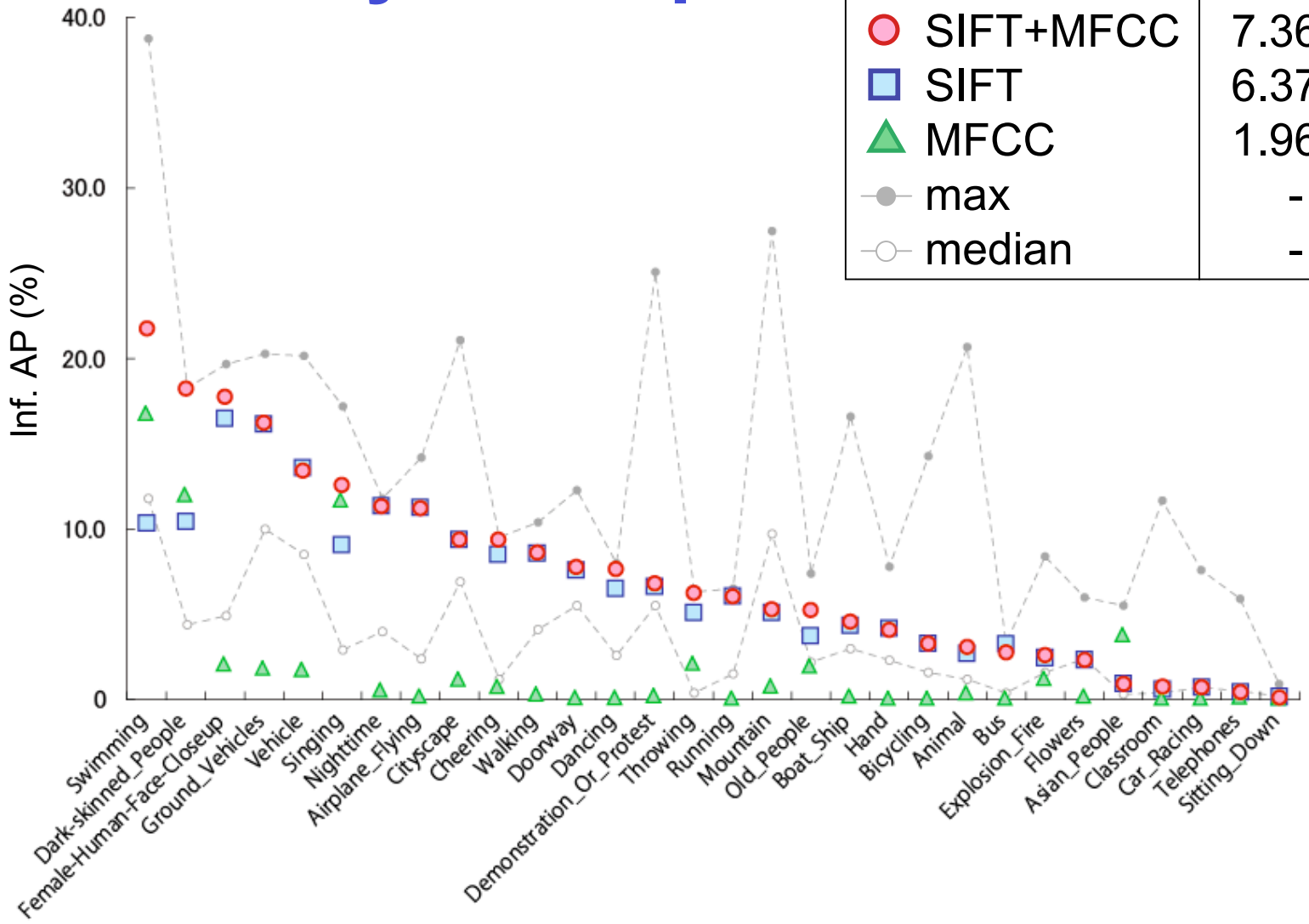
Results



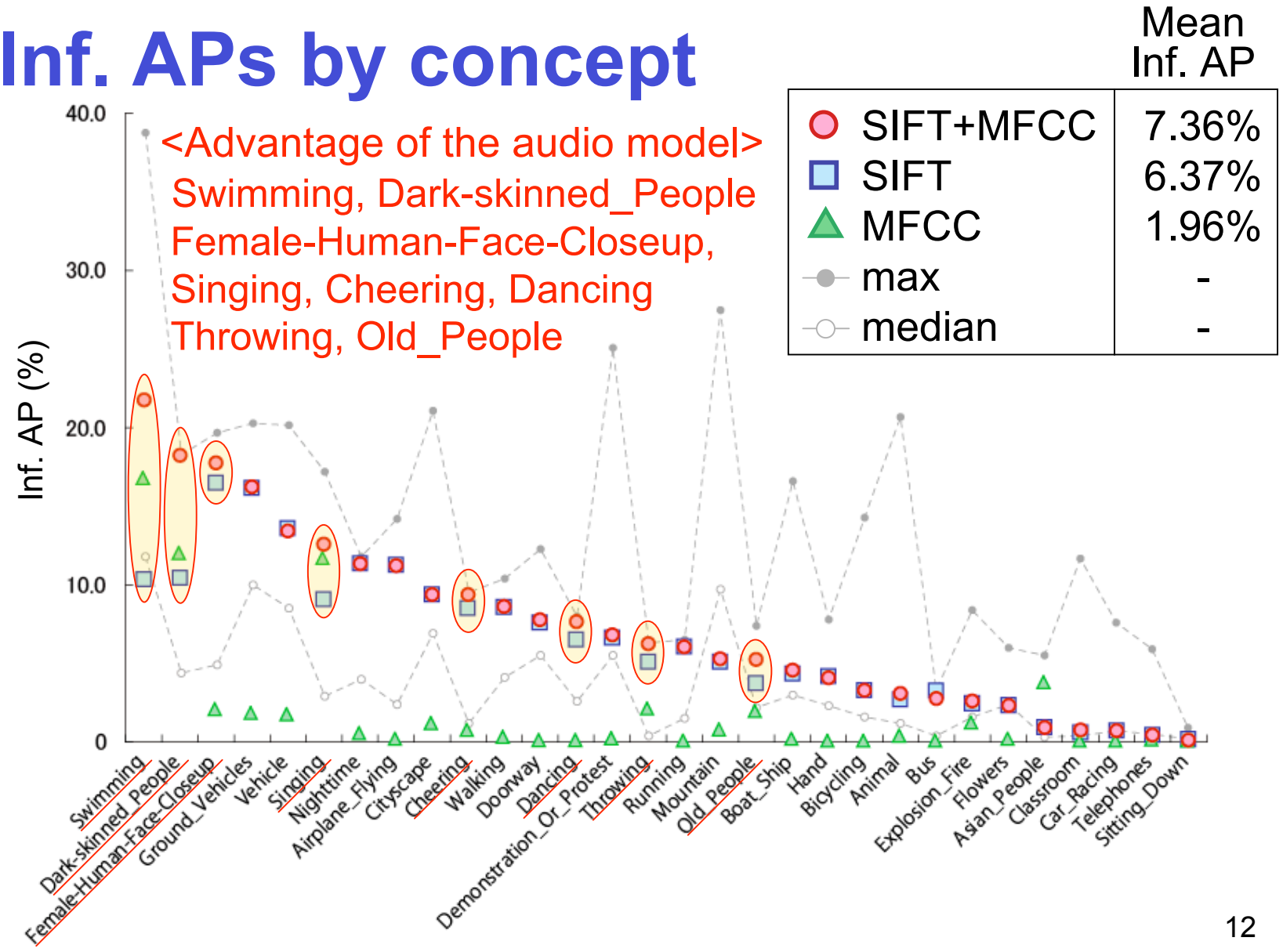
Run ID	Feature	Classifier	Mean Inf. AP
TT+GT_run1_1	MFCC + SIFT	SVM	audio 7.36%
TT+GT_run3_3	SIFT (Harris+Hessian)	SVM	6.37%
TT+GT_run2_2	LSI (Color hist.+Gabor)	MFoM	3.72%
TT+GT_run4_4	SIFT (Harris)	MFoM	3.56%

Inf. APs by concept

	Mean Inf. AP
SIFT+MFCC	7.36%
SIFT	6.37%
MFCC	1.96%
max	-
median	-



Inf. APs by concept



Conclusion (Part 1)

- Both audio and visual features are modeled effectively by the GMM supervectors.
- Effects of the audio model:

  -- Mean Inf. AP improved from 6.37% to 7.36%.

  -- Events related to human (action) can be detected.
- But APs are still low…

  10%<AP : 8 concepts (Singing, Airplane_Flying, …)

  5%~10%: 10 concepts (Cheering, Dancing, …)

  0%~5%: 12 concepts (Bus, Telephones, …)
- What is needed?

  Selection of good positives and negatives,

  Spatial and temporal localization, Other than SIFT?

-- Part 2 --
Maximal Figure of Merit Classifier

Motivation

Last year

1. LSI feature extraction & MFoM† learning optimizing F_1 measure
2. Late fusion approach

This year

1. LSI feature extraction & MFoM learning optimizing MAP measure
2. MFoM learning optimizing F_1 measure with TiTech's GMM+SIFT feature vectors (Early fusion approach)

MFoM † : Maximal-Figure-of-Merit

MFoM Learning

- Optimizing a preferred performance metric directly
  - *E.g.)* F_1

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

- Encoding concept-dependent score functions *g* into the performance metric
  - *E.g.)* FP_i *(false positive for the* i^{th} *concept)*

$$FP_i = \{1 - \sigma(d_i(X_s, \Lambda))\} \cdot I(X_s \notin C_i),$$

where σ : sigmoid function

$$d_i(X_s, \Lambda) = -g_i(X_s, \Lambda) + g_i^-(X_s, \Lambda)$$

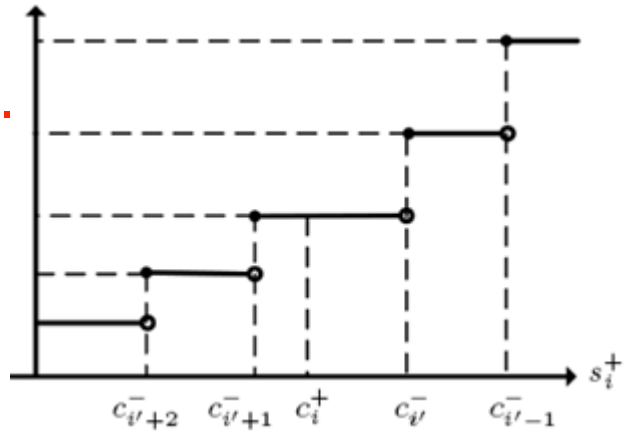
$I(\cdot)$: indicator function

AP Optimization in Linear MFoM

- Assuming AP as a function of sample scores

$$AP = f\left(s_1^+, \cdots, s_{M_p}^+, s_1^-, \cdots, s_{M_n}^-\right)$$

- With respect to an individual score, AP behaves as a staircase function.



- Using sigmoid functions, the staircase function can be approximated to a differentiable form.

- Then, the gradient of AP is calculated with a chain rule.

$$\frac{\partial AP}{\partial \omega} \approx \sum_{i=1}^{M_p} \frac{\partial \widehat{AP}}{\partial s_i^+} + \sum_{j=1}^{M_n} \frac{\partial \widehat{AP}}{\partial s_j^-}$$

→ The model parameter ω is estimated by a GPD algorithm

Kernelized MFoM Learning

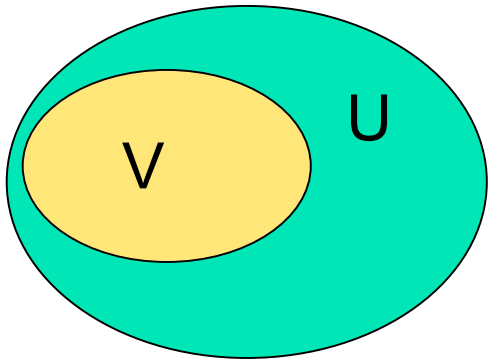
- Given a kernel matrix K, we define a score function *g*

$$g(X_s, \Lambda) = \sum_{i=1}^{N} w_i k(X_i, X_s) + b$$

# of training data samples

→ 1. The # of parameters w_i is large
2. Sparsity is no longer guaranteed!

- Subspace distance minimization



H_U : a subspace constructed from U

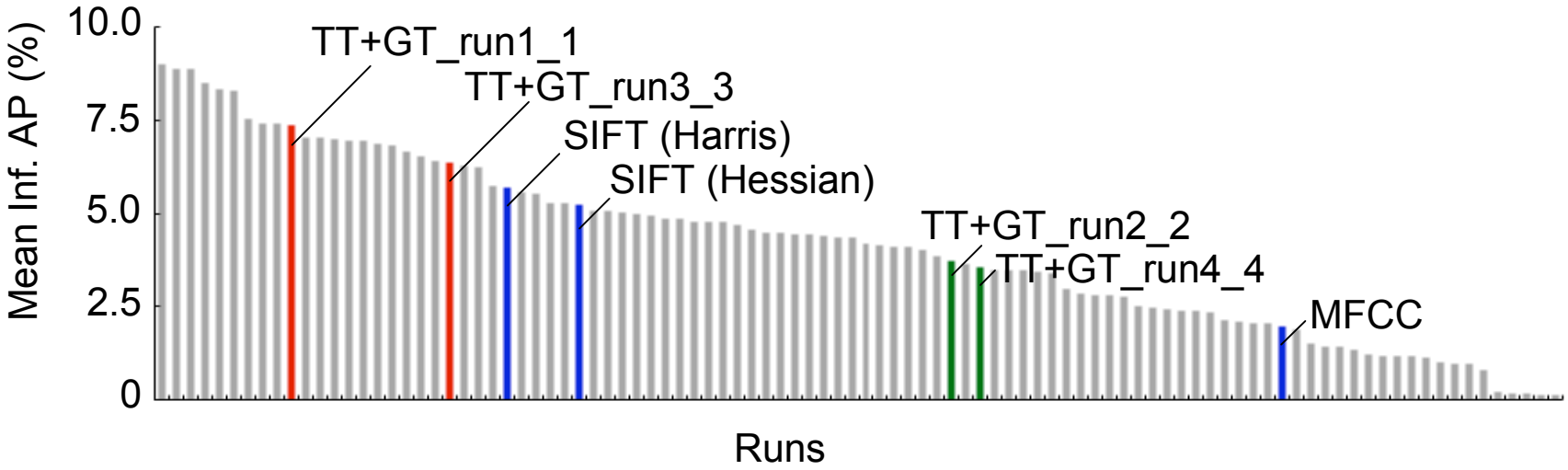
H_V : a subspace constructed from V

$$V^* = \arg\min_{V \in P} d(H_U, H_V),$$

where P is a power set of *V*

→ V can be found by the Nystrom Extension

Results



Run ID	Feature	Classifier	Mean Inf. AP
TT+GT_run1_1	MFCC + SIFT	SVM	7.36%
TT+GT_run3_3	SIFT (Harris+Hessian)	SVM	6.37%
TT+GT_run2_2	LSI (Color hist.+Gabor)	MFoM	3.72%
TT+GT_run4_4	SIFT (Harris)	MFoM	3.56%

Assessments of Run 2

- Step size problem
  - Having a difficulty to choose an appropriate step size for a GPD algorithm. -> too sensitive
  - The step sizes only for the Lite-version concepts are carefully arranged.

	Lite 20 concepts	Remaining 10 concepts
Median	2.11%	4.25%
TT+GT_run2_2	3.83%	3.66%

  - A line search algorithm is applied after the submission.
- Features are not discriminative enough.
  - Grid-based color and texture features seem not to be powerful enough to cover variations of the huge data set.

Assessments of Run 4

- Only two parameters are tuned; The rests are fixed.
  - the size of negative examples, a weight for the regularization term.
- Not-so-good initial solution
  - With an updated version, AP of 6 concepts : 3.56% -> 5.18%
  - Trade off between the size of negative examples and the amount of noise in the negative examples.
- How to determine the subset size is an open question

Future work

- Develop better feature extraction methods
- Better initial solution does matter
  - Will start from the estimated parameter vectors using other methods such as SVM.
  - Will solve the problem of selecting the size of the subset.