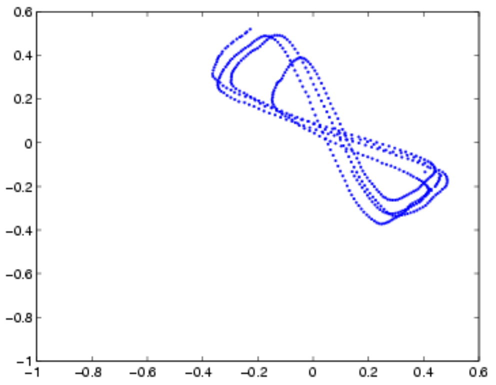
Efficient Distribution Mining and Classification

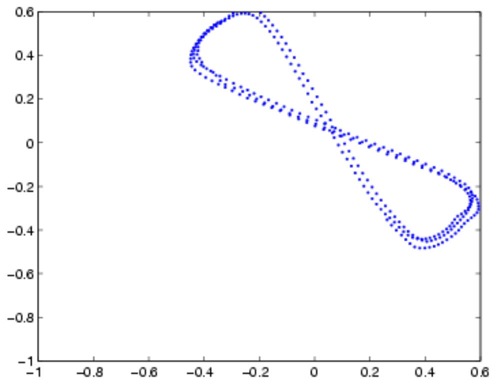
Yasushi Sakurai (NTT Communication Science Labs),
Rosalynn Chong (University of British Columbia),
Lei Li (Carnegie Mellon University),
Christos Faloutsos (Carnegie Mellon University)

Classification for Distribution Data Sets

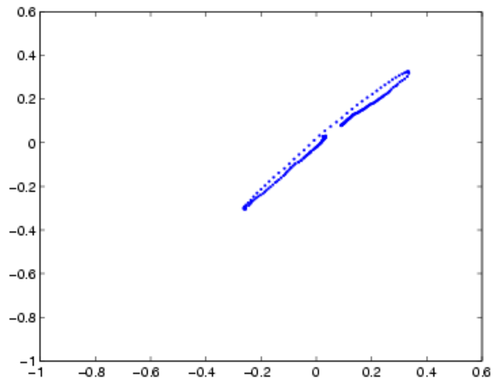
- Given *n* distributions (n multi-dimensional vector sets)
  - With a portion of them labeled and others unlabeled
- Classify unlabeled distributions into the right group
  - Ex. Distr. #1 and Distr. #2 fall into the same group



Distribution #1
(unknown)



Distribution #2
(walking)



Distribution #3
(jumping)

Scenario 1

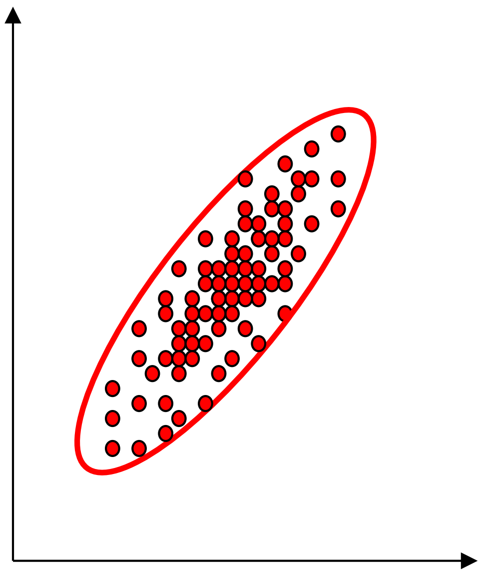
- Marketing research for e-commerce
  - Vectors:
    - orders by each customer
    - Time the customer spent browsing
    - Number of pages the customer browsed
    - Number of items the customer bought
    - Sales price
    - Number of visits by each customer
  - Distributions: customers
  - Classification: identify customer groups who carry similar traits
  - Find distribution groups to do market segmentation, rule discovery and spot anomalies
    - E.g., “Design an advertisement for each customer categories”

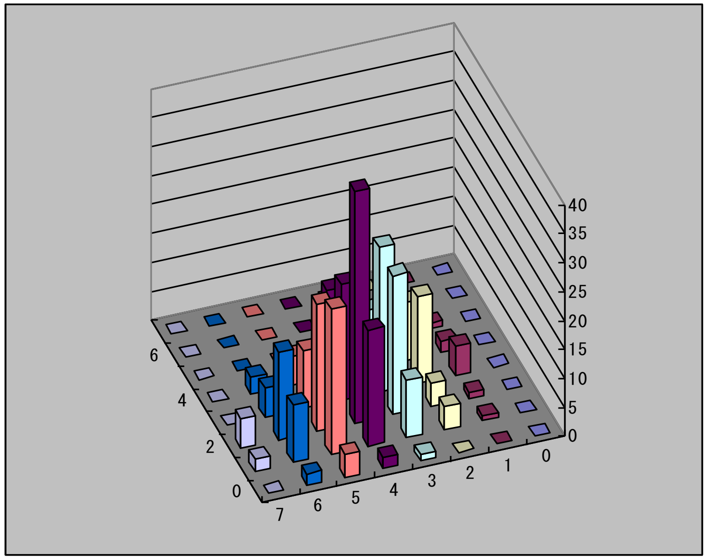
Scenario 2

- User analysis for SNS systems (e.g., blog hosting service)
  - Vectors:
    - internet habits by each participant
    - Number of blog entries for every topic
    - Length of entries for every topic
    - Number of links of entries for every topic
    - Number of hours spent online
  - Distributions: SNS participants
  - Classification: identify participant groups who have similar internet habits
  - Find distribution groups to facilitate community creation
    - E.g., "Create communities according to users' interests"

Representing Distributions

- Histograms
  - Easy to be updated incrementally
  - Used in this work
- Another option: probability density function





Background

- Kullback-Leibler divergence
  - Measures the natural distance difference from one probability distribution P to another arbitrary probability distribution Q.

$$d_{KL}(P,Q) = \int p_x \cdot \log\left(\frac{p_x}{q_x}\right) dx$$

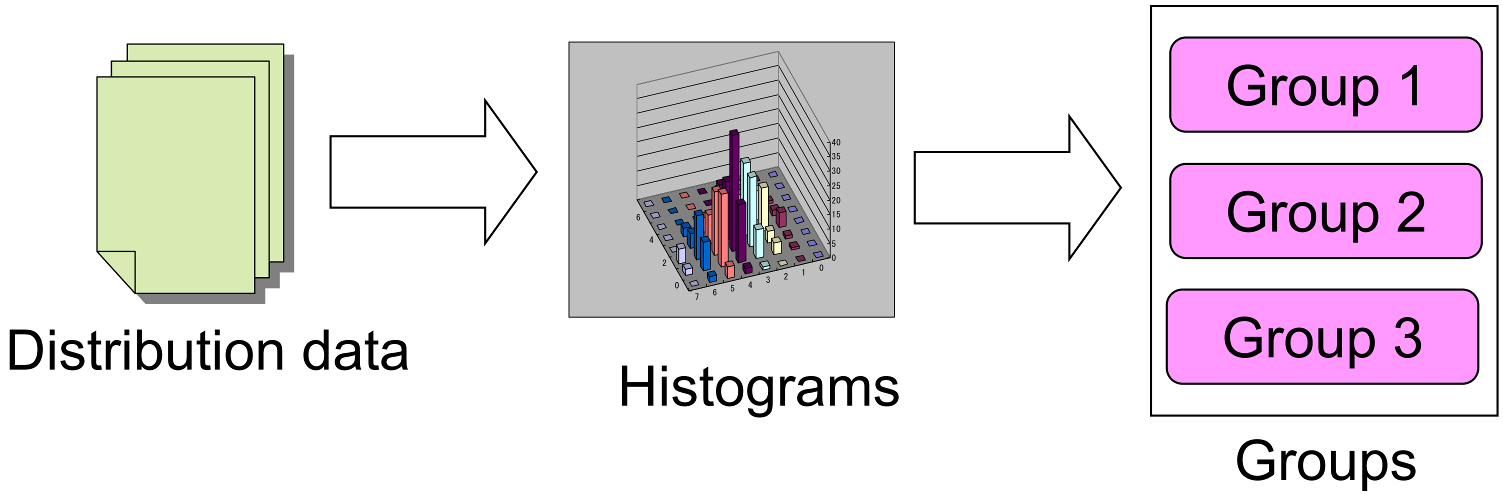
  - One undesirable property: $d_{KL}(P,Q) \neq d_{KL}(Q,P)$
- Symmetric KL-divergence

$$d_{SKL}(P,Q) = \int p_x \cdot \log\left(\frac{p_x}{q_x}\right) dx + \int q_x \cdot \log\left(\frac{q_x}{p_x}\right) dx$$

$$= \int (p_x - q_x) \cdot \log\left(\frac{p_x}{q_x}\right) dx$$

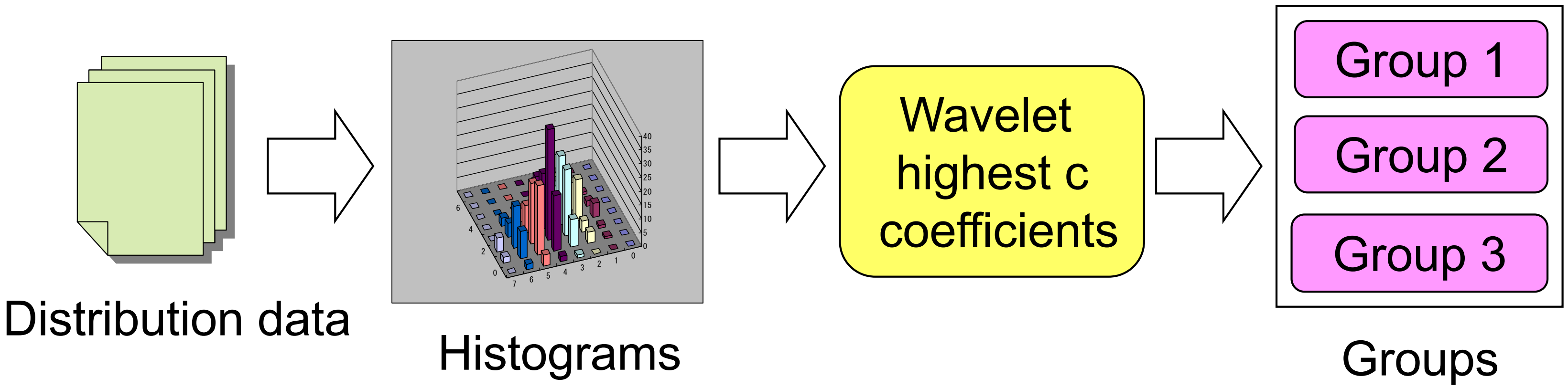
Proposed Solution

- Naïve approach
  - Create histogram for each distribution of data
  - Compute the KL divergence directly from histograms p_i and q_i
  - Use any data mining method
    - E.g., classification, clustering, outlier detection



Proposed Solution

- DualWavelet (wavelet-based approach)
  - Create histogram for each distribution of data
  - Represent each histogram p_i as wp_i and $\hat{w}p_i$ using wavelets
    - wp_i : the wavelet of p_i
    - $w\hat{p}_i$: the wavelet of log (p_i)
  - Reduce number of wavelets by selecting *c* coefficients with the highest energy ($c << m$)
  - Compute the KL divergence from the wavelets
  - Use any data mining method
    - E.g., <u>classification</u>, clustering, outlier detection



DualWavelet

- Theorem 1
  - Let

    wp_i and wq_i be the wavelet of p_i and q_i resp.

    $\hat{w}p_i$ and $\hat{w}q_i$ be the wavelet of log(p_i) and log(q_i) resp.
  - We have

m: # of bins *of a* histogram
c: # of wavelet coefficients

$$d_{SKL}(P,Q) = \sum_{i=1}^{m} (p_i - q_i) \cdot \log\left(\frac{p_i}{q_i}\right)$$

$$= \sum_{i=1}^{m} (p_i - q_i) \cdot (\log p_i - \log q_i)$$

$$= \frac{1}{2} \cdot \sum_{i=1}^{c} \begin{pmatrix} \left(wp_i - \hat{w}q_i\right)^2 + \left(wq_i - \hat{w}p_i\right)^2 \\ -\left(wp_i - \hat{w}p_i\right)^2 - \left(wq_i - \hat{w}q_i\right)^2 \end{pmatrix}$$

KL divergence can be computed from wavelets

Time Complexity

- Naïve method for the nearest neighbor classification
  - **O(*mnt*)** time
  - *n*: # of input distributions, *m*: # of grid cells
  - *t* : # of distributions in the training data set
- DualWavelet
  - Wavelet transform: **O(*mn*)**
  - Classification: **O(*nt*)**
  - Since *c* (# of wavelet coefficients we use) is a small constant value

Space Complexity

- Naïve method for the nearest neighbor classification
  - **O(*mt*)** space
  - *m*: # of grid cells
  - *t* : # of distributions in the training data set
- DualWavelet
  - Wavelet transform: **O(*m*)**
  - Classification: **O(*t*)**
  - Since *c* (# of wavelet coefficients we use) is a small constant value

GEM: Optimal grid-side selection

- Optimal granularity of histogram
  - Optimal number of segments s_{opt} provides good accuracy
  - Plus reasonable computation cost
  - Proposed normalized KL divergence (GEM criterion)

$$C_s(P,Q) = \frac{d_{SKL}(P,Q)}{H_S(P) + H_S(Q)}$$

  - Choose s_{opt} that maximizes the pairwise criteria

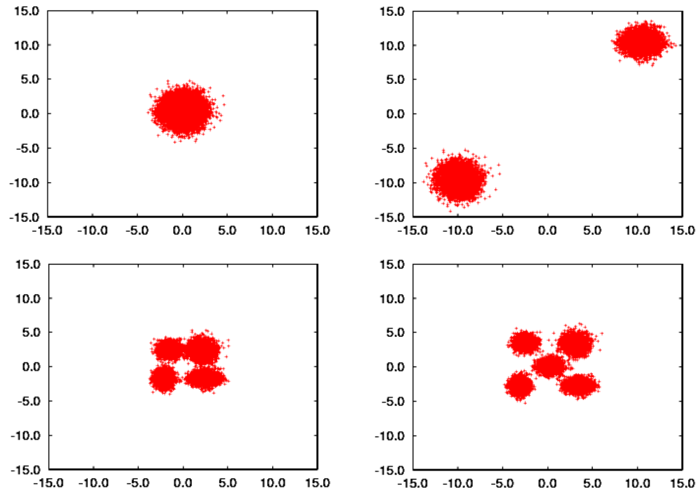
$$S_{opt}(P,Q) = \arg\max_s (C_s(P,Q))$$

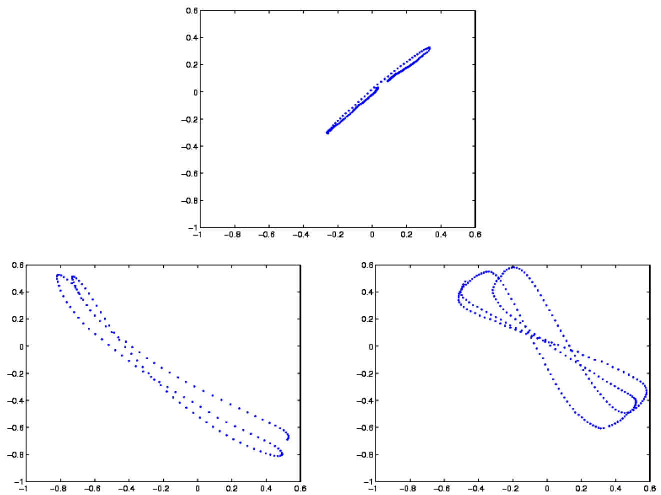
  - Obtain $s_{opt}()$ for every sampled pair, then choose the maximum

$$S_{opt} = \max_{all(P,Q)\, pairs} s_{opt}(P,Q)$$

Experiments

- Gaussians
  - n=4,000 distributions, each with 10,000 points (dimension d=3)
  - Mixture of Gaussians (1, 2, 2^d, (2^d+1))
  - Same means, but different variances for each class
- MoCap
  - n=58 real running, jumping and walking motions (d=93)
  - Each dimension corresponds to the x, y, or z position of a body joint
  - Dimensionality reduction by using SVD (d=2)

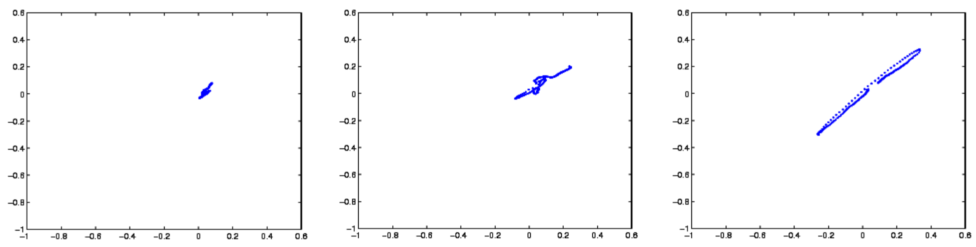




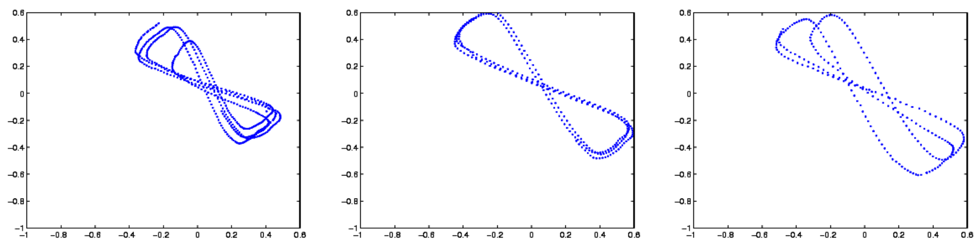
Classification (MoCap)

- Confusion matrix for classification

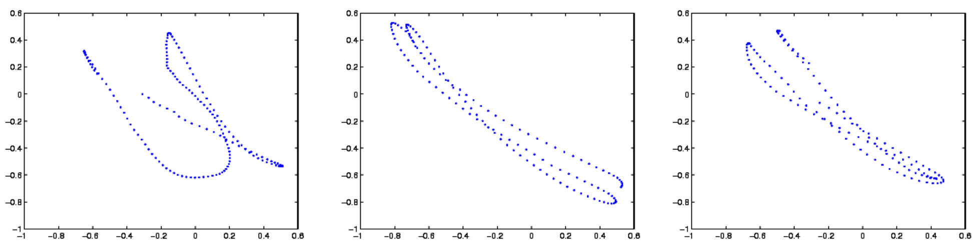
recovered / correct	J	W	R
Jumping	3	0	0
Walking	0	22	1
Running	0	1	19



Jumping



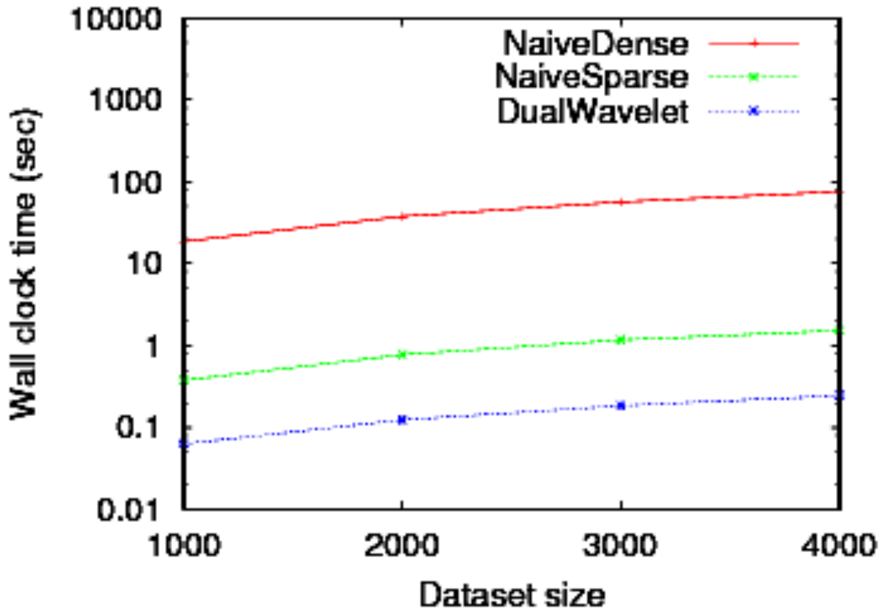
Walking



Running

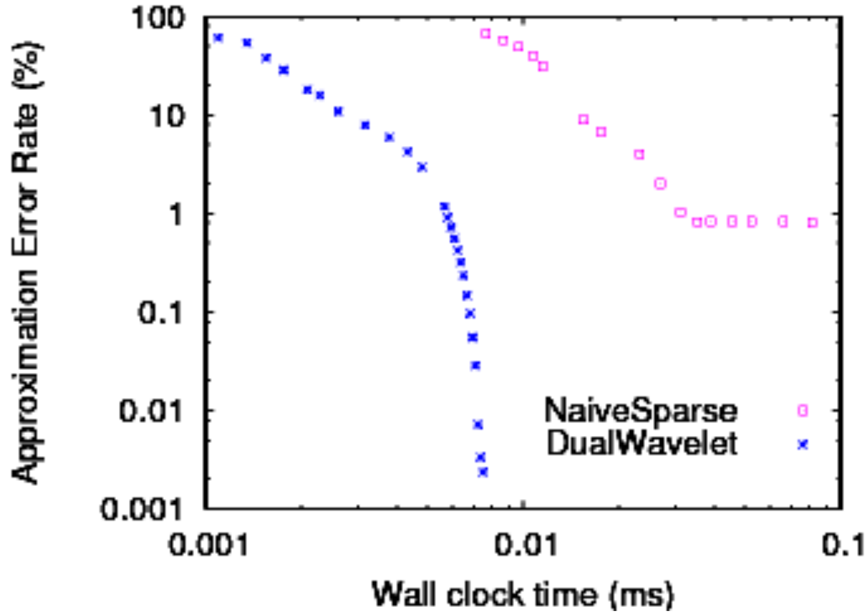
Computation Cost (Gaussians)

- NaïveDense, which uses all histogram buckets
- NaïveSparse, which uses only selected buckets (largest values)
- DualWavelet achieves a dramatic reduction in computation time



Approximation Quality

- Scatter plot of computation cost vs. approximation quality
- Trade-off between quality and cost
- DualWavelet gives significantly lower approximation error, for the same computation time



Conclusions

- Addressed the problem of distribution classification, in general, distribution mining
- Proposed a fast and effective method to solve it
  - Proposed to use wavelets on both the histograms, as well as their logarithms
  - Solution can be applied to large datasets with multi-dimensional distributions
- Experiments show that DualWavelet is significantly faster than the naïve implementation (up to 400 times)