Macro-class Selection for Hierarchical k-NN Classification of Inertial Sensor Data

Corey McCall, Kishore Reddy,
and Mubarak Shah
University of Central Florida

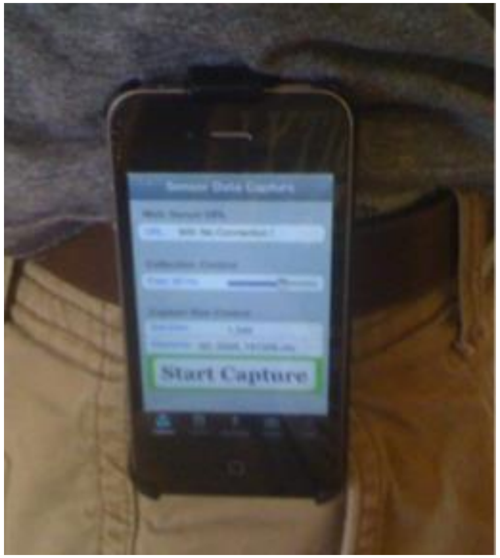
Problem

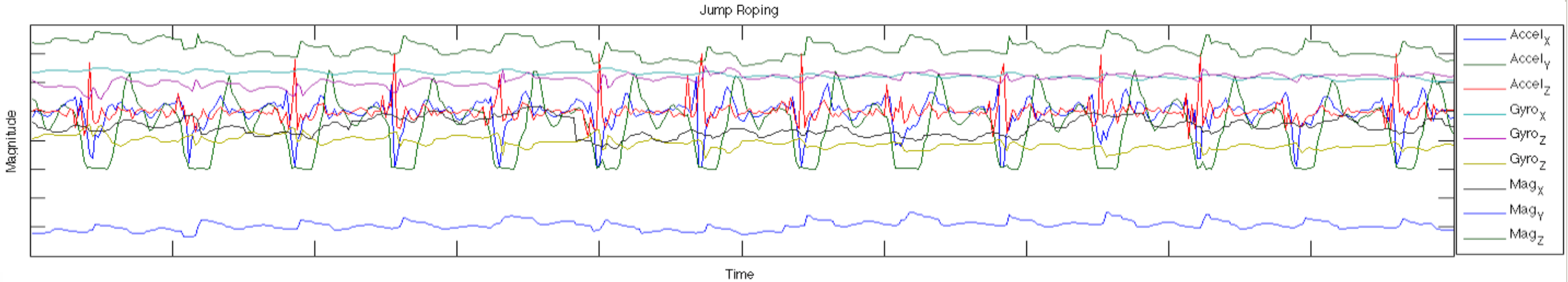
Automatic construction of a hierarchical classifier for action recognition in pervasive systems.

Datasets

Smartphone Dataset

- Exercise sequence recorded with smartphone IMU.
- 13 features across each DoF of each sensor.
- 125 candidate features per IMU.

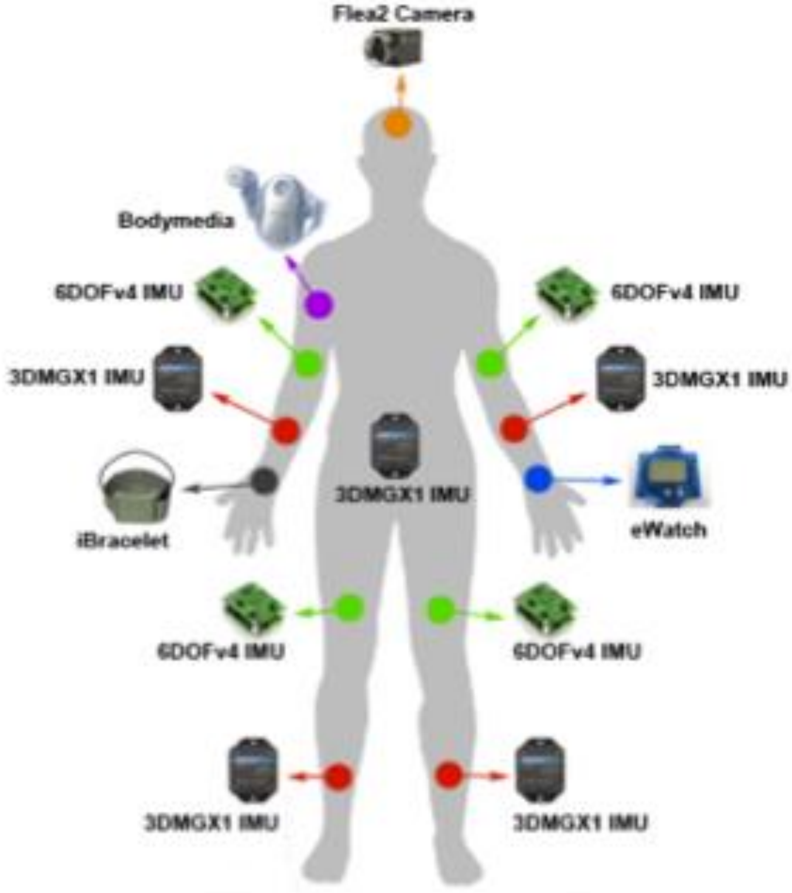




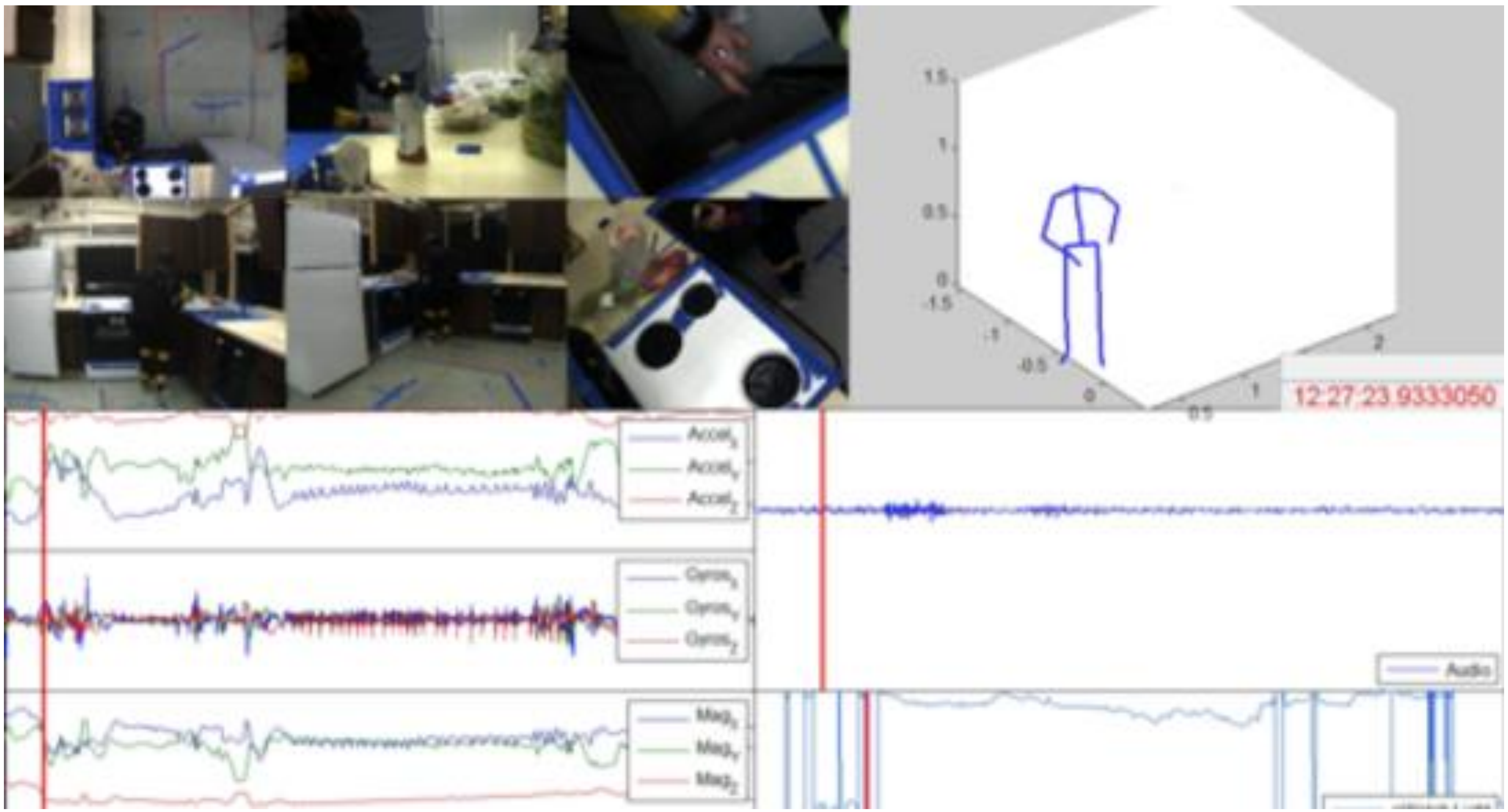
Datasets

<u>CMU-MMAC Dataset</u>

- Cooking sequences recorded with 5 IMUs.
- 525 potential features



Datasets



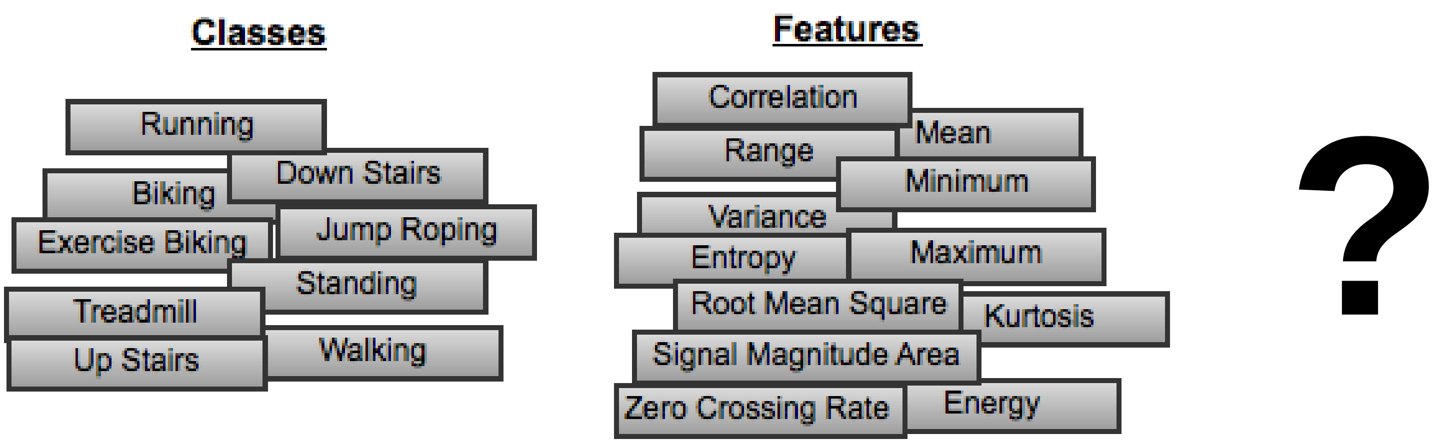
Features

Feature	Size	Formula
Mean	9	$\mu_{X^j} = \frac{1}{\ell}\sum_{i=1}^{\ell} X_i^j$
Variance	9	$\sigma_{X^j}^2 = \frac{1}{\ell}\sum_{i=1}^{\ell}(X_i^j - \mu)^2$
Minimum	9	$min_{X^j} = \text{minimum}(X_i^j)$
Maximum	9	$max_{X^j} = \text{maximum}(X_i^j)$
Range	9	$range_{X^j} = max_{X_i^j} - min_{X_i^j}$
Mean Crossing Rate	9	$mcr_{X^j} = \frac{1}{\ell-1}\sum_{i=\ell}^{N-1}\Upsilon\{(X_i^j - \mu_{X^j})(X_{i+1}^j - \mu_{X^j}) < 0\}$, where Υ is the indicator function
Root Mean Square	9	$rms_{X^j} = \sqrt{\frac{1}{\ell}\sum_{i=1}^{\ell} X_i^{j^2}}$
Skew	9	$skew_{X^j} = \frac{1}{\ell\sigma_{X^j}^3}\sum_{i=1}^{\ell}(X_i^j - \mu_{X^j})^3$

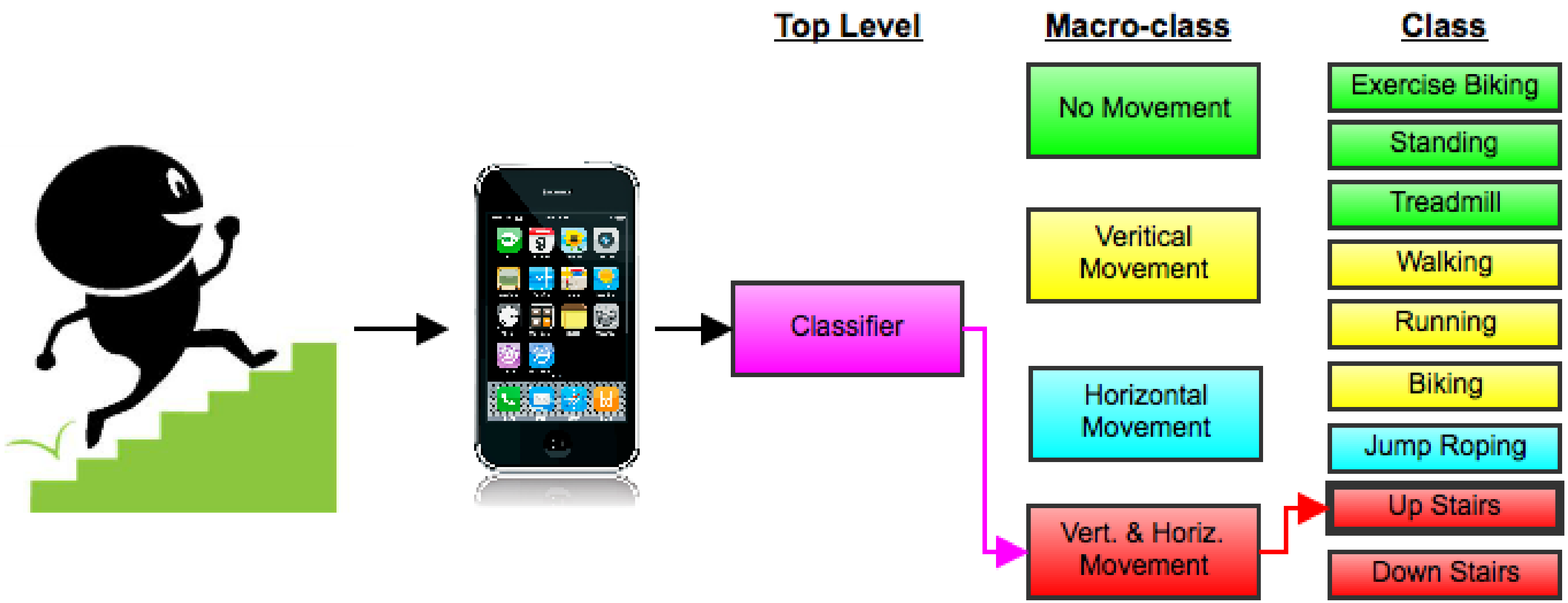
Feature	Size	Formula
Average Entropy	9	$H_{X^j} = -\frac{1}{\ell}\sum_{i=1}^{\ell} p(X_i^j)\log(p(X_i^j))$
Kurtosis	9	$kurt_{X^j} = \frac{1}{\ell\sigma_{X^j}^4}\sum_{i=1}^{\ell}(X_i^j - \mu_{X^j})^4$
Correlation	9	$corr_{X^{ab}} = \frac{1}{\ell\sigma_{X^a}\sigma_{X^b}}\sum_{i=1}^{\ell}(X_i^a - \mu_{X^a})(X_i^b - \mu_{X^b})$, for [a,b]={[1,2],[1,3],[2,3]}
Average Magnitude Area	3	$SMA_X = \frac{1}{\ell}\sum_{i=1}^{\ell}(\lvert X_i^1\rvert + \lvert X_i^2\rvert + \lvert X_i^3\rvert)$
Average Energy Expenditure	3	$EE_X = \frac{1}{\ell}\sum_{i=1}^{\ell}\sqrt{X_i^{1^2} + X_i^{2^2} + X_i^{3^2}}$

Motivation

- The difficulty and cost of classification generally increases with the number of classes.
- Goal: Hierarchical classification for accurate and efficient action recognition.
  - How to build hierarchy? (macro-class selection)
  - Which features to use at each level? (feature selection)



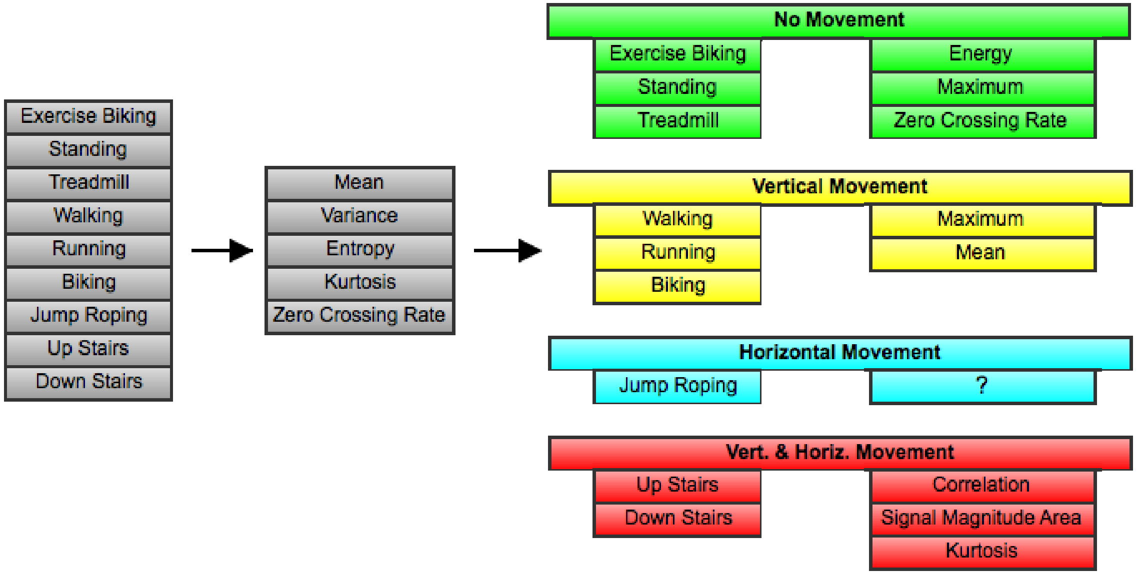
Motivation


Top Level
Macro-class
Class
Classifier
No Movement
Veritical Movement
Horizontal Movement
Vert. & Horiz. Movement
Exercise Biking
Standing
Treadmill
Walking
Running
Biking
Jump Roping
Up Stairs
Down Stairs

The Algorithm

Summary of Training Procedure

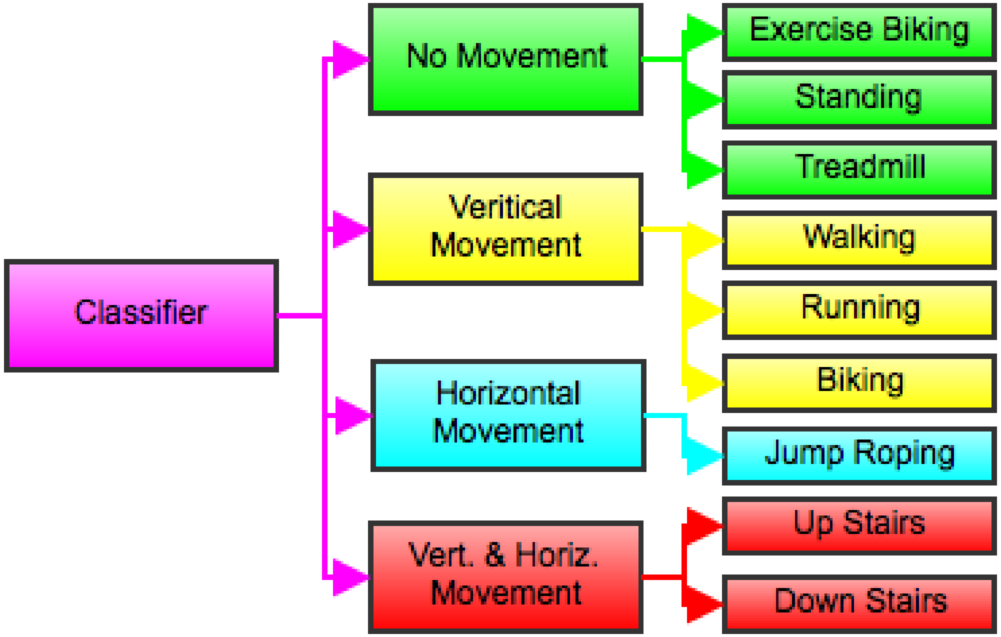
1. Select features and macro-classes for top level of hierarchy.
2. Train k-NN to classify each training sample into a macro-class.
3. Select a specialized feature set for each macro-class.
4. Train k-NN on each macro-with specialized feature set.



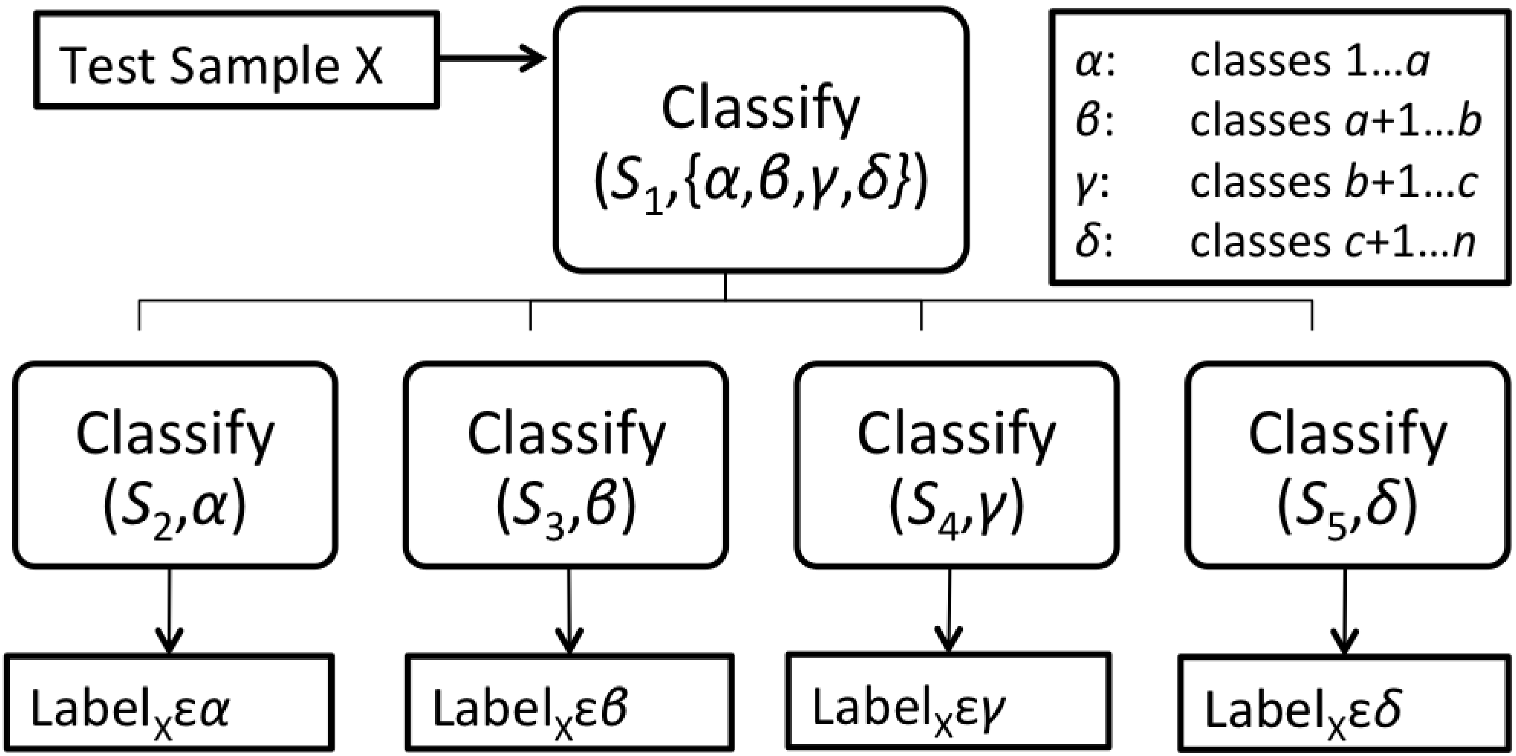
The Algorithm

Summary of Testing Procedure

1. Classify test sample into a macro-class.
2. Classify test sample on macro-class using specialized features.



The Algorithm

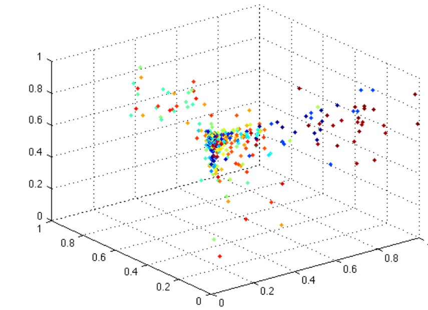


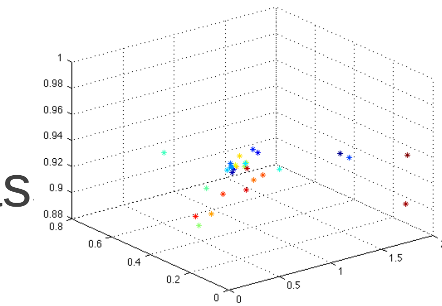
The Algorithm

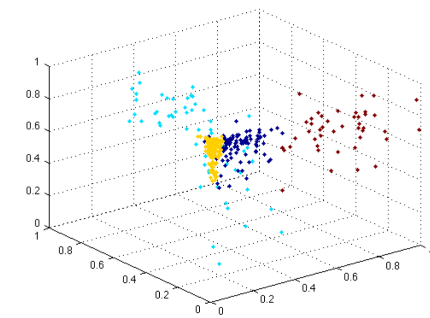
<u>A Greedy Approach based on k-NN</u>

For each potential feature set:

1. Calculate each class's mean
2. Cluster means with special k-means
3. Use 1-NN to assign class to macro-clas
4. Score feature/macro-class selection

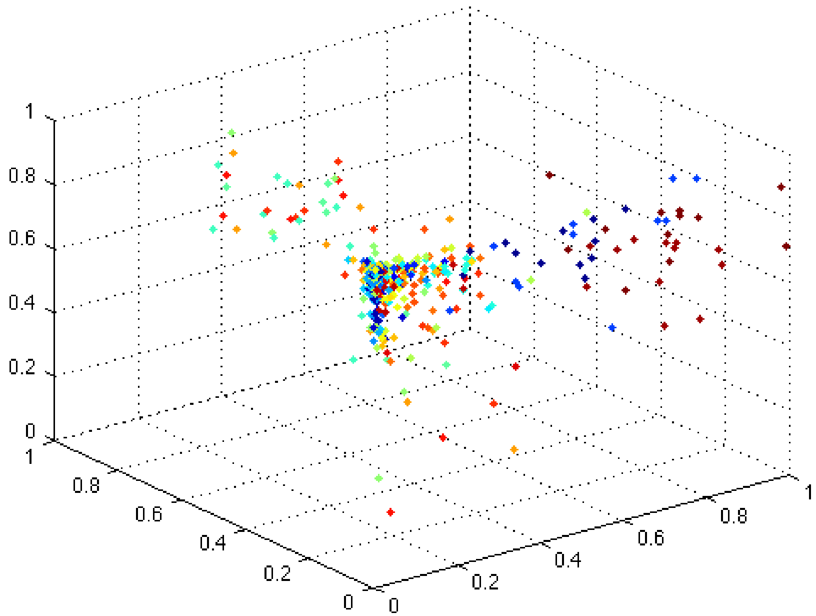




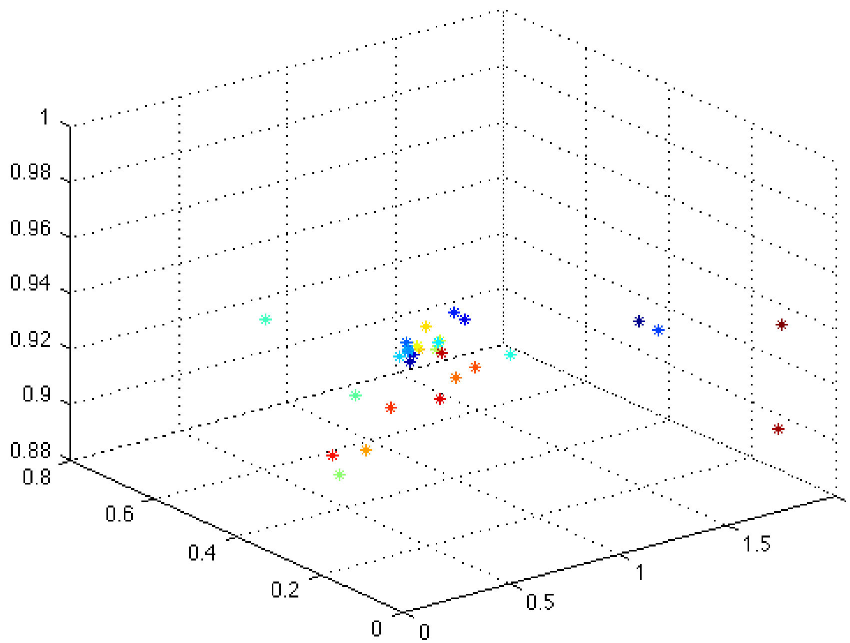


The Algorithm

Macro-class selection example



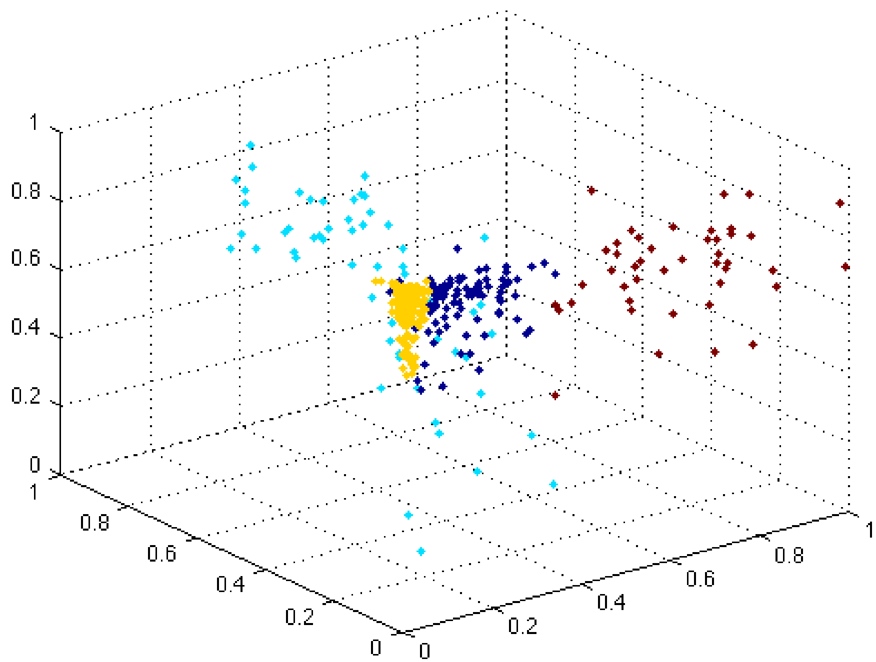
Training examples in potential feature set



Center of each class

The Algorithm

A: [4 5 12 17 22 23]
B: [13 14 15 21 24 25]
C: [2 3 7 8 9 10 11 16 18 19 20 26 27]
D: [1 6 28 29]



Training examples placed in macro-classes

Classification Results:

Classifier A	97.9%
Classifier B	86.5%
Classifier C	63.3%
Classifier D	66.0%
TOTAL:	**71.1%**
(Single Classifier):	62.2%

The Algorithm

<u>Macro-class Scoring</u>

- First, optimize for high training accuracy using θ.
- After threshold, optimize for better distribution using Φ.

$$Q = \frac{a_i^2 p_i^3}{G(l_i)^4}$$

$$F = \frac{a_i p_i^3}{G(l_i)^4}$$

The Algorithm (Details)

1. Compute candidate feature set *C* of potential features f_i
2. Scale *C* to [0,1]
3. Initialize selected feature set *S* to empty
4. Iterate while score increases or *C* is empty:

   For all f_i in *C*

   Calculate mean of each class of features in *S* U f_i

   Perform k-means on class means to get cluster centers

   Assign each data example in training set to a bin using 1-NN

   Assign each class to a bin based on popularity

   Determine a score based on accuracy threshold

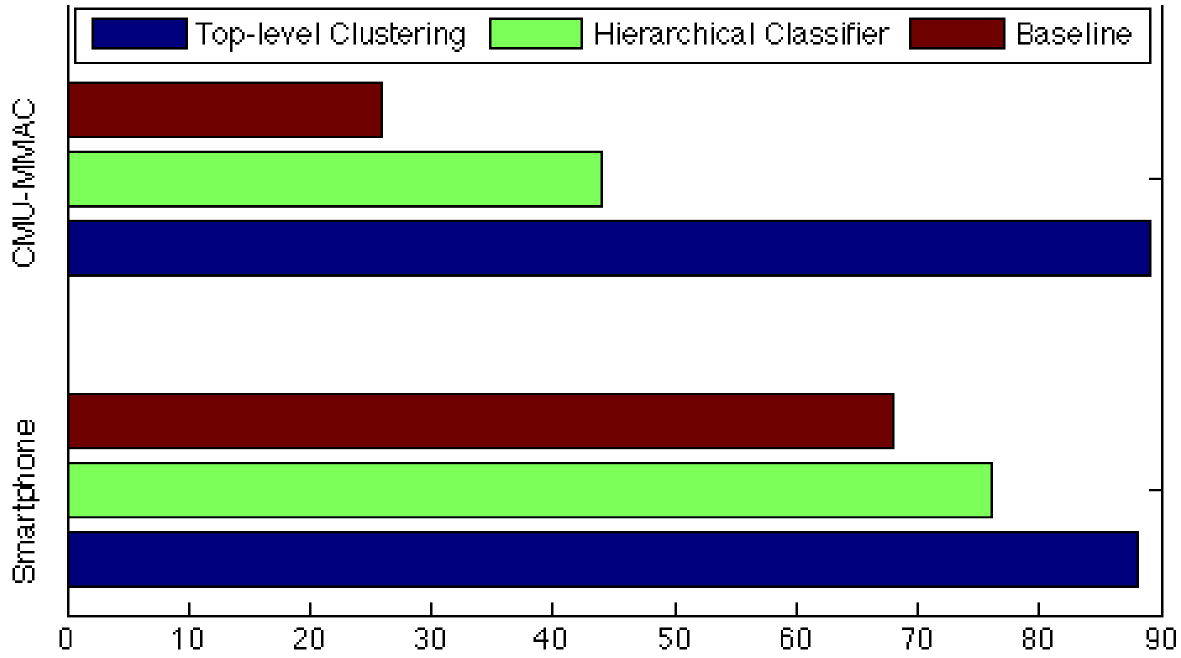
   Add feature with best score *S* and remove from *C*.

The Algorithm (Details)

Input: $\mathbf{X} \in \mathbb{R}^{M \times N}$, $\mathbf{y} \in \mathbb{R}^{M}, t, n$
Output: $S, L, C \in \mathbb{R}^{p \times |S|}$

$k = \lfloor n/2 \rfloor$
$[A, S, L, C] \leftarrow [0, \emptyset, \emptyset, \emptyset]$
while $|S| < N$ **do**
  $[a, b, \psi, l, c] \leftarrow [0, 0, 0, \emptyset, \emptyset]$
  for all $i \in \{1, \ldots, N\} \setminus S$ **do**
    $\mu \leftarrow \mathrm{CMEAN}(\mathbf{X}_{S \cup \{i\}}, \mathbf{y})$
    $\{c_i, p_i, q\} \leftarrow \mathrm{KMEANS2}(\mu, k)$
    $\{a_i, l_i\} \leftarrow \mathrm{KNN2}(\mathbf{X}_{S \cup \{i\}}, c_i, q)$
    if $A < t$ **then**
      $\psi_i \leftarrow (p_i > 1 \& \nexists 0 \in l_i \& a_i > A)?\Theta(a_i, p_i, l_i)$:0
    else
      $\psi_i \leftarrow (p_i > 1 \& \nexists 0 \in l_i \& a_i > t)?\Phi(a_i, p_i, l_i)$:0
    end if
    if $\psi_i > \psi$ **then**
      $[a, \psi, b, l, c] \leftarrow [a_i, \psi_i, i, l_i, c_i]$
    end if
  end for
  if $\psi = 0$ **then**
    break
  end if
  $[A, S, L, C] \leftarrow [a, S \cup \{b\}, l, c]$
end while

Experimental Results



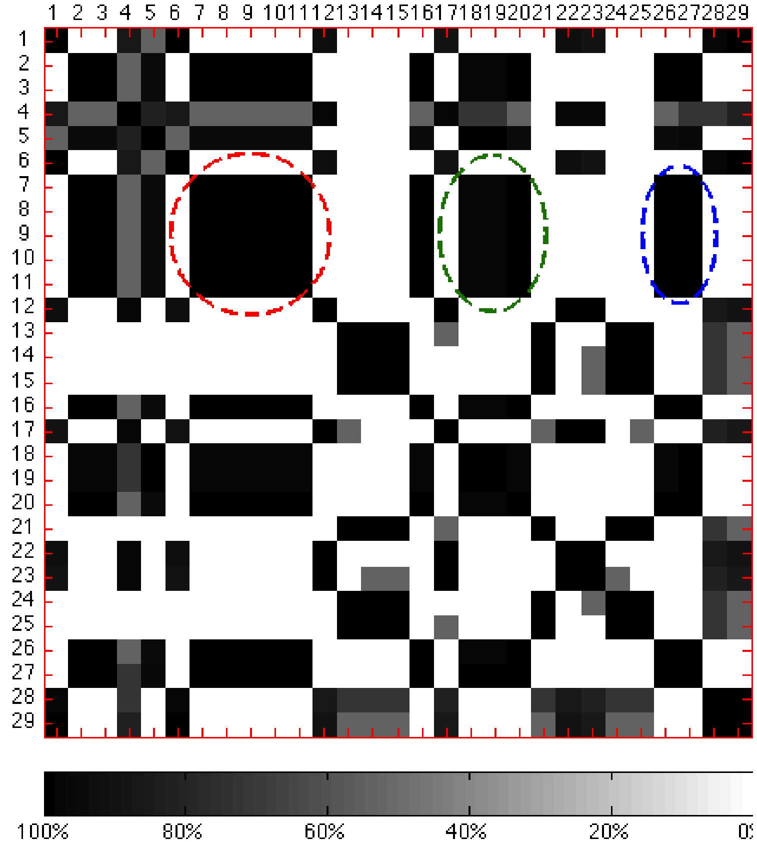
CMU-MMAC: 69% improvement

Smartphone: 12% improvement

Experimental Results

1. Close Fridge	16. Read Box
2. Crack Egg	17. Spray Pam
3. Open Box	18. Stir Bowl
4. Open Cupboard1	19. Stir Egg
5. Open Cupboard2	20.Switch On
6. Open Fridge	21. Take Pan
7. Pour Bowl in Pan	22. Take Egg
8. Pour Bag in Bowl	23. Take Fork
9. Pour Oil in Bowl	24. Take Oil
10. Pour Oil in Cup	25. Take Pam
11. Pour Water in Bowl	26. Twist Off Cap
12. Pour Water in Cup	27. Twist On Cap
13. Put Pan in Oven	28. Walk to Counter
14. Put Oil in Cupboard3	29. Walk to Fridge
15. Put Pam in Cupboard3	

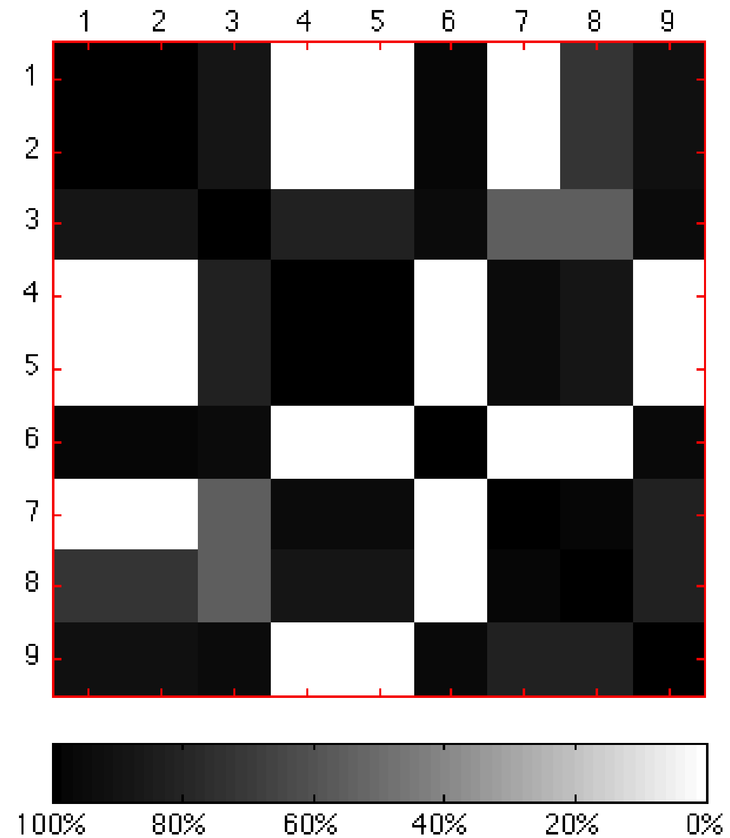
CMU-MMAC Dataset



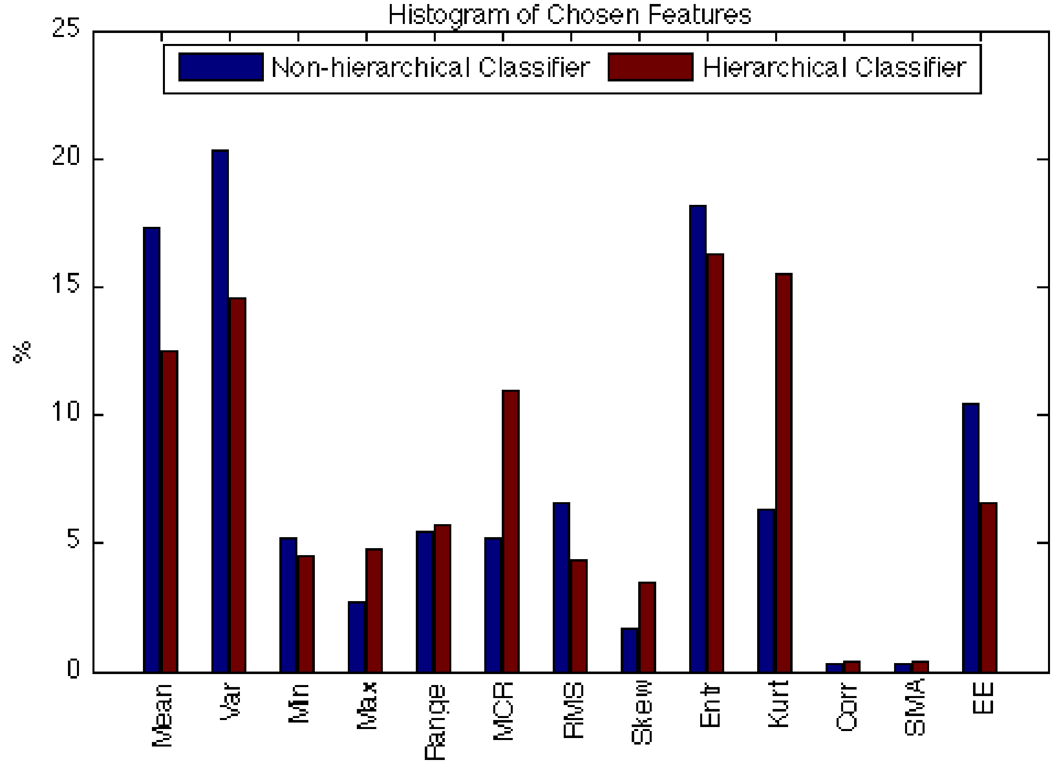
Experimental Results

1. Biking	6. Running
2. Climbing	7. Standing
3. Descending	8. Treadmill Walking
4. Exercise Biking	9. Walking
5. Jump Roping	

Smartphone Dataset



Experimental Results



Std. Dev. Non-Hierarchical: 6.8

Std. Dev. Hierarchical: 5.6

Conclusion

- Constructed hierarchical classifier shows significant improvement over baseline.
- Final results on CMU-MMAC dataset: 44%
  - Lower than (Fisher and Reddy, 2011)
  - Very High top-level clustering accuracy (~90%)
  - Large improvement because of many confusing classes.
- Final results on Smartphone dataset: 76%

Summary of Contributions

- First algorithm to automatically select features and macro-classes.
- Can be used in conjunction with more complex feature selection and classification algorithms as the top-level of hierarchy.

Smartphone dataset, UCF Computer Vision Lab, 2011:
http://vision.eecs.ucf.edu/data/SmartPhone