Compressed Sensing Meets Machine Learning - Classification of Mixture Subspace Models via Sparse Representation

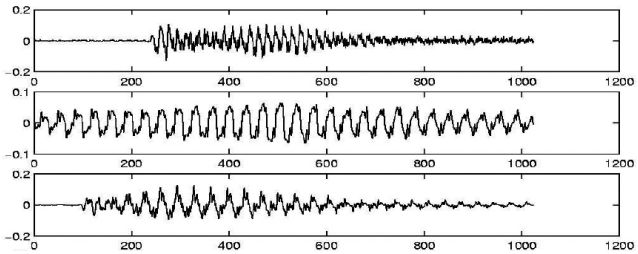
Allen Y. Yang
<yang@eecs.berkeley.edu>

Feb. 25, 2008. UC Berkeley

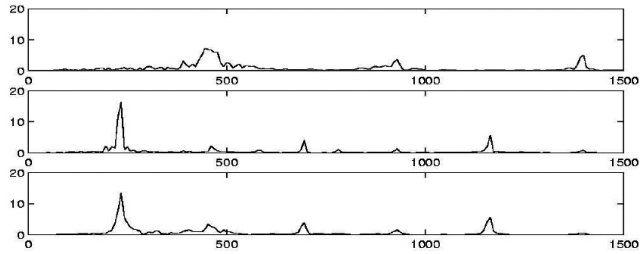
What is Sparsity

Sparsity

A signal is sparse if most of its coefficients are (approximately) zero.



(a) Harmonic functions



(b) Magnitude spectrum

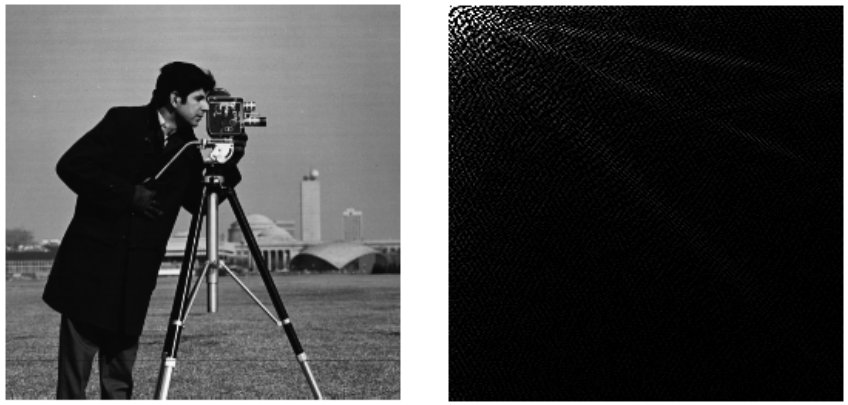
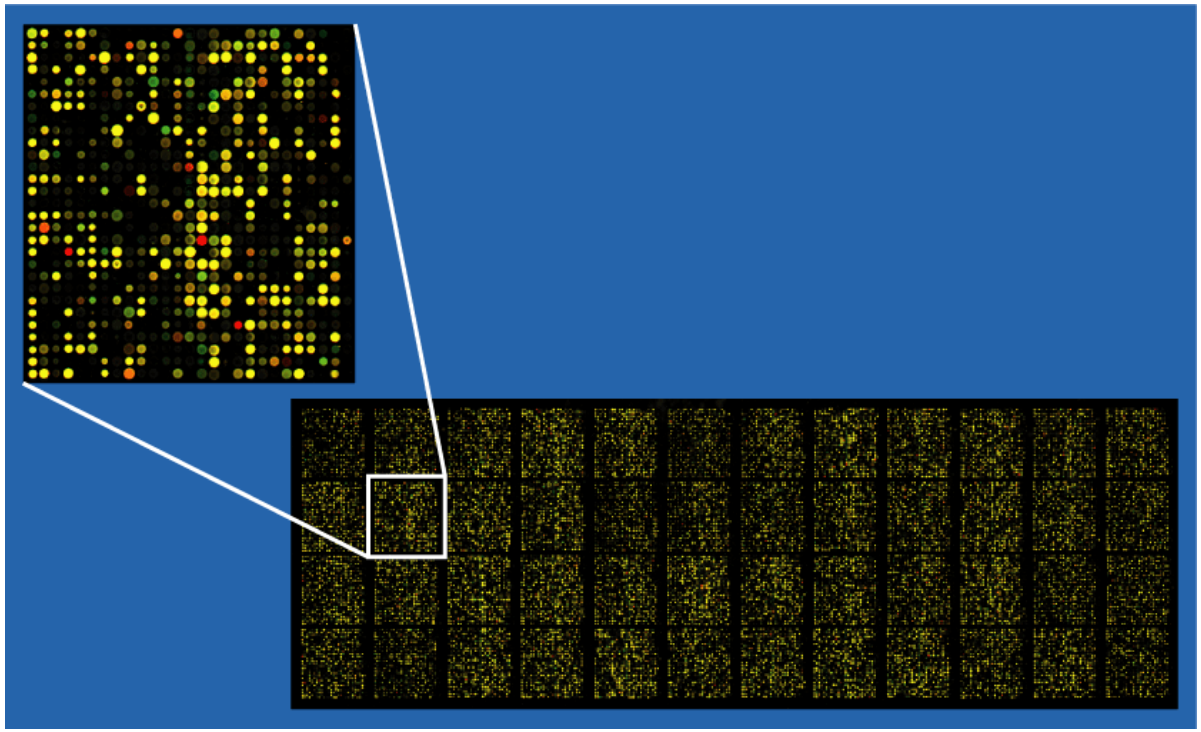


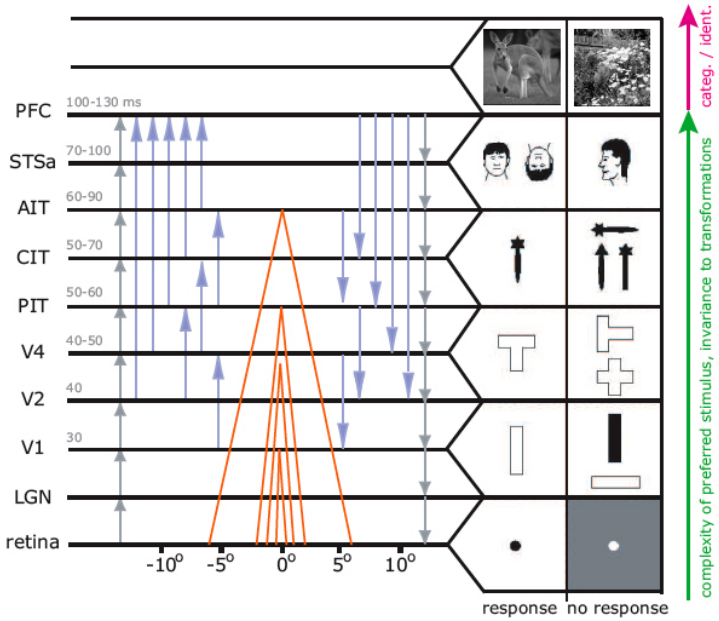
Figure: 2-D DCT transform.

Sparsity in spatial domain

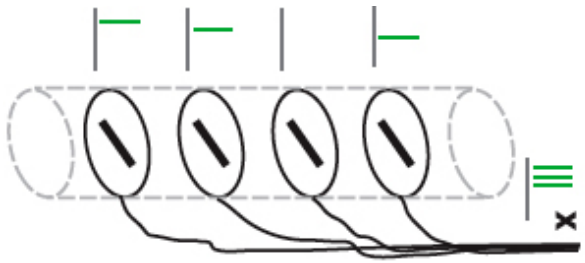
- gene microarray data [Drmanac et al. 1993]



- **Sparsity in human visual cortex** [Olshausen & Field 1997, Serre & Poggio 2006]



1. **Feed-forward**: No iterative feedback loop.
2. **Redundancy**: Average 80-200 neurons for each feature representation.
3. **Recognition**: Information exchange between stages is not about individual neurons, but rather how many neurons as a group fire together.



Sparsity and ℓ^1-Minimization

- "Black gold" age [Claerbout & Muir 1973, Taylor, Banks & McCoy 1979]

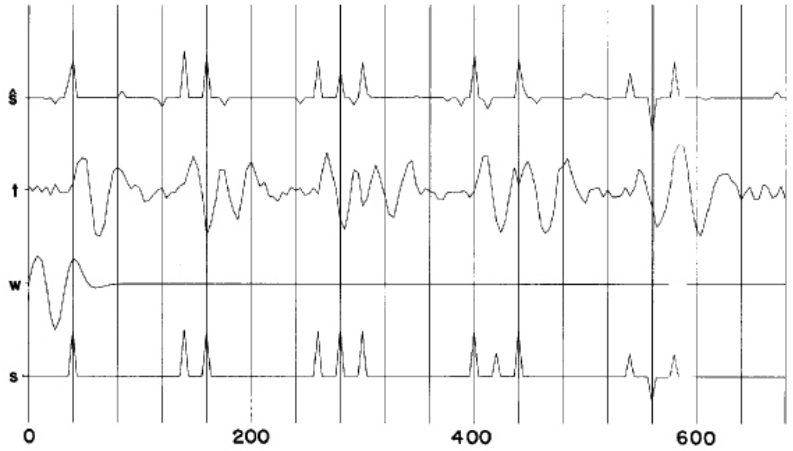
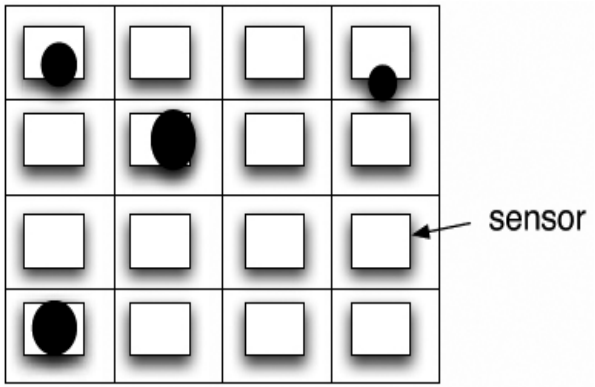


Figure: Deconvolution of spike train.

Sparse Support Estimators

- **Sparse support estimator** [Donoho 1992, Meinshausen & Buhlmann 2006, Yu 2006, Wainwright 2006, Ramchandran 2007, Gastpar 2007]



- **Basis pursuit** [Chen & Donoho 1999]: Given $\mathbf{y} = A\mathbf{x}$ and $\mathbf{x}$ unknown,

$$\mathbf{x}^* = \arg\min \|\mathbf{x}\|_1, \text{ subject to } \mathbf{y} = A\mathbf{x}$$

- **The Lasso** (least absolute shrinkage and selection operator) [Tibshirani 1996]

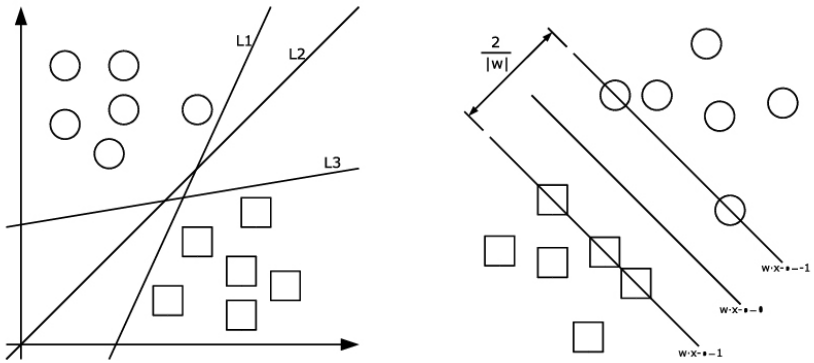
$$\mathbf{x}^* = \arg\min \|\mathbf{y} - A\mathbf{x}\|_2, \text{ subject to } \|\mathbf{x}\|_1 \leq k$$

Taking Advantage of Sparsity

What generates sparsity? (*d'après* Emmanuel Candès)

- Measure first, analyze later.
- Curse of dimensionality.

1. **Numerical analysis**: sparsity reduces cost for storage and computation.
2. **Regularization in classification**:



(a) decision boundary (b) maximal margin

Figure: Linear support vector machine (SVM)

Our Contributions

1. Classification via compressed sensing
2. Performance in face recognition
3. Extensions
   - Outlier rejection
   - Occlusion compensation
4. Distributed pattern recognition in sensor networks.

Problem Formulation in Face Recognition

1. **Notations**
   - Training: For K classes, collect training samples $\{\mathbf{v}_{1,1}, \cdots, \mathbf{v}_{1,n_1}\}, \cdots, \{\mathbf{v}_{K,1}, \cdots, \mathbf{v}_{K,n_K}\} \in \mathbb{R}^D$.
   - Test: Present a new $\mathbf{y} \in \mathbb{R}^D$, solve for label$(\mathbf{y}) \in [1, 2, \cdots, K]$.
2. Construct $\mathbb{R}^D$ sample space via **stacking**

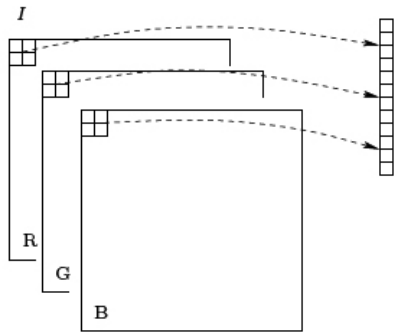
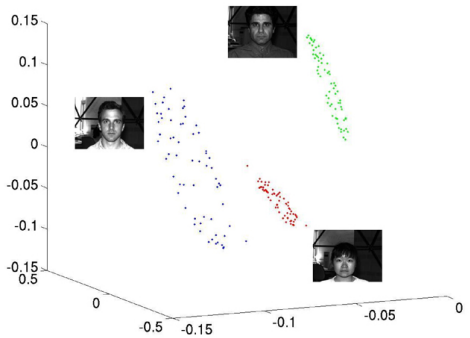


Figure: For images, assume 3-channel 640×480 image, $D = 3 \cdot 640 \cdot 480 \approx 1e6$.

3. Assume $\mathbf{y}$ belongs to Class i [Belhumeur et al. 1997, Basri & Jacobs 2003]



$$\begin{aligned} \mathbf{y} &= \alpha_{i,1}\mathbf{v}_{i,1} + \alpha_{i,2}\mathbf{v}_{i,2} + \cdots + \alpha_{i,n_1}\mathbf{v}_{i,n_i}, \\ &= A_i\alpha_i, \end{aligned}$$

where $A_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \cdots, \mathbf{v}_{i,n_i}]$.

1. Nevertheless, i is the variable we need to solve.
   Global representation:

$$\begin{aligned} \mathbf{y} &= [A_1, A_2, \cdots, A_K] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_K \end{bmatrix}, \\ &= A\mathbf{x}_0. \end{aligned}$$

2. **Over-determined** system: $A \in \mathbb{R}^{D \times n}$, where $D \gg n = n_1 + \cdots + n_K$.
   $\mathbf{x}_0$ encodes membership of $\mathbf{y}$: If $\mathbf{y}$ belongs to Subject i,

$$\mathbf{x}_0 = [0 \cdots 0 \ \alpha_i \ 0 \cdots 0]^T \in \mathbb{R}^n.$$

Problems to face

- Solving for $\mathbf{x}_0$ in $\mathbb{R}^D$ is **intractable**.
- True solution $\mathbf{x}_0$ is **sparse**: Average $\frac{1}{K}$ terms non-zero.

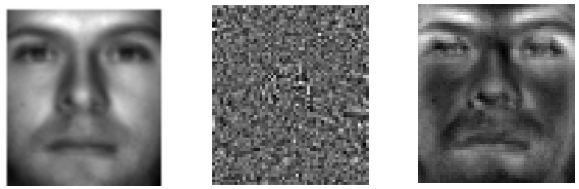
Dimensionality Reduction

1. Construct linear projection $R \in \mathbb{R}^{d \times D}$, d is the **feature dimension**, $d \ll D$.

$$\tilde{\mathbf{y}} \doteq R\mathbf{y} = RA\mathbf{x}_0 = \tilde{A}\mathbf{x}_0 \in \mathbb{R}^d.$$

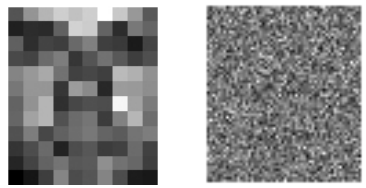
   $\tilde{A} \in \mathbb{R}^{d \times n}$, but $\mathbf{x}_0$ is unchanged.

2. Holistic features
   - Eigenfaces [Turk 1991]
   - Fisherfaces [Belhumeur 1997]
   - Laplacianfaces [He 2005]



3. Partial features

4. Unconventional features
   - Downsampled faces
   - Random projections



ℓ^0-Minimization

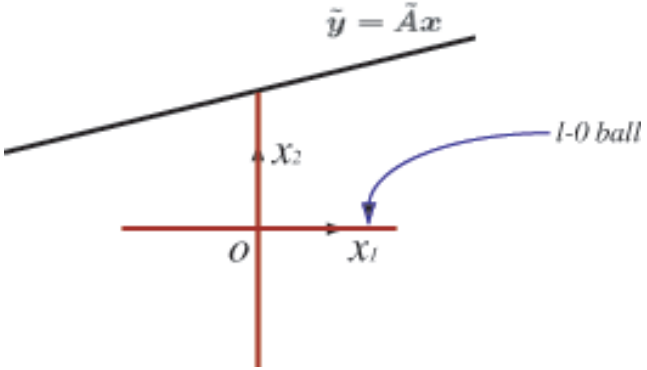
1. Solving for **sparsest** solution via ℓ^0-Minimization

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \tilde{\mathbf{y}} = \tilde{A}\mathbf{x}.$$

   $\|\cdot\|_0$ simply counts the number of nonzero terms.

2. ℓ^0-Ball

- ℓ^0-ball is not convex.
- ℓ^0-minimization is NP-hard.



ℓ^1/ℓ^0 Equivalence

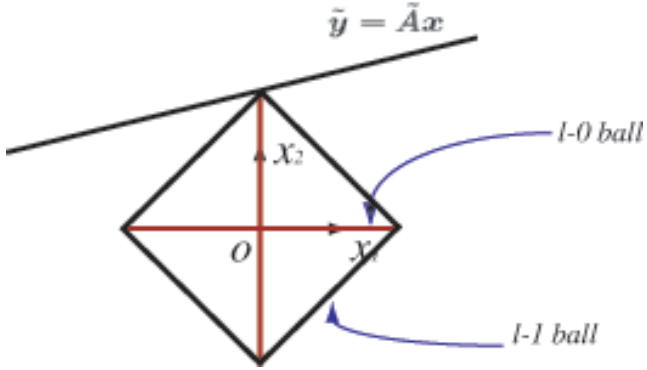
1. **Compressed sensing**: If $\mathbf{x}_0$ is *sparse enough*, ℓ^0-minimization is equivalent to

$$(P_1) \quad \min \|\mathbf{x}\|_1 \text{ s.t. } \tilde{\mathbf{y}} = \tilde{A}\mathbf{x}.$$

$\|\mathbf{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$.

2. ℓ^1-Ball

- ℓ^1-Minimization is convex.
- Solution equal to ℓ^0-minimization.



3. ℓ^1/ℓ^0 Equivalence: [Donoho 2002, 2004; Candes et al. 2004; Baraniuk 2006]
Given $\tilde{\mathbf{y}} = \tilde{A}\mathbf{x}_0$, there exists **equivalence breakdown point** (EBP) $\rho(\tilde{A})$, if $\|\mathbf{x}_0\|_0 < \rho$:
   - ℓ^1-solution is unique
   - $\mathbf{x}_1 = \mathbf{x}_0$

ℓ^1-Minimization Routines

- **Matching pursuit** [Mallat 1993]
  1. Find most correlated vector $\mathbf{v}_i$ in $\tilde{A}$ with $\mathbf{y}$: $i = \arg\max \langle \mathbf{y}, \mathbf{v}_j \rangle$.
  2. $\tilde{A} \leftarrow \tilde{A}^{\hat{i}}$, $x_i \leftarrow \langle \mathbf{y}, \mathbf{v}_i \rangle$, $\mathbf{y} \leftarrow \mathbf{y} - x_i \mathbf{v}_i$.
  3. Repeat until $\|\mathbf{y}\| < \epsilon$.
- **Basis pursuit** [Chen 1998]
  1. Assume $\mathbf{x}_0$ is m-sparse.
  2. Select m linearly independent vectors B_m in $\tilde{A}$ as a basis
  $$\mathbf{x}_m = B_m^{\dagger} \mathbf{y}.$$
  3. Repeat swapping one basis vector in B_m with another vector in $\tilde{A}$ if improve $\|\mathbf{y} - B_m \mathbf{x}_m\|$.
  4. If $\|\mathbf{y} - B_m \mathbf{x}_m\|_2 < \epsilon$, stop.
- **Quadratic solvers**: $\tilde{\mathbf{y}} = \tilde{A}\mathbf{x}_0 + \mathbf{z} \in \mathbb{R}^d$, where $\|\mathbf{z}\|_2 < \epsilon$

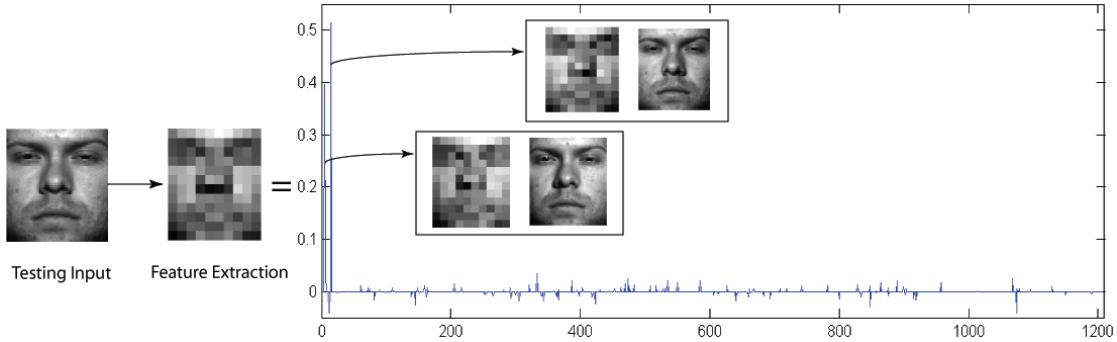
$$\mathbf{x}^* \quad = \quad \arg\min\{\|\mathbf{x}\|_1 + \lambda \|\mathbf{y} - \tilde{A}\mathbf{x}\|_2\}$$

[Lasso, Second-order cone programming]: More expensive.

Matlab Toolboxes

- ℓ^1-**Magic** by Candès at Caltech.
- **SparseLab** by Donoho at Stanford.
- **cvx** by Boyd at Stanford.

Classification

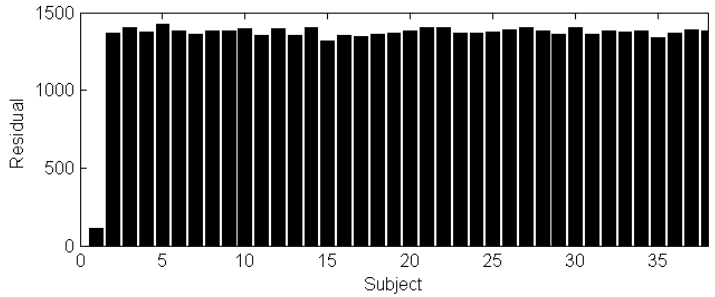


1. Project $\mathbf{x}_1$ onto face subspaces:

$$\delta_1(\mathbf{x}_1) = \begin{bmatrix} \alpha_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \delta_2(\mathbf{x}_1) = \begin{bmatrix} 0 \\ \alpha_2 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \delta_K(\mathbf{x}_1) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \alpha_K \end{bmatrix}. \quad (1)$$

2. Define residual $r_i = \|\tilde{\mathbf{y}} - \tilde{A}\delta_i(\mathbf{x}_1)\|_2$ for Subject i:

   - $\text{id}(\mathbf{y}) = \arg\min_{i=1,\ldots,K}\{r_i\}$



AR Database 100 Subjects (Illumination and Expression Variance)

Table: I. Nearest Neighbor

Dimension	30	54	130	540
Eigen [%]	68.1	74.8	79.3	80.5
Laplacian [%]	73.1	77.1	83.8	89.7
Random [%]	56.7	63.7	71.4	75
Down [%]	51.7	60.9	69.2	73.7
Fisher [%]	83.4	86.8	N/A	N/A

Table: II. Nearest Subspace

30	54	130	540
64.1	77.1	82	85.1
66	77.5	84.3	90.3
59.2	68.2	80	83.3
56.2	67.7	77	82.1
80.3	85.8	N/A	N/A

Table: III. Linear SVM

Dimension	30	54	130	540
Eigen [%]	73	84.3	89	92
Laplacian [%]	73.4	85.8	90.8	95.7
Random [%]	54.1	70.8	81.6	88.8
Down [%]	51.4	73	83.4	90.3
Fisher [%]	86.3	93.3	N/A	N/A

Table: IV. ℓ^1-Minimization

30	54	130	540
71.1	80	85.7	92
73.7	84.7	91	94.3
57.8	75.5	87.6	94.7
46.8	67	84.6	93.9
87	92.3	N/A	N/A

Berkeley University of California

Sparsity vs. **Non-sparsity**: ℓ^1 and SVM decisively outperform NN and NS.

1. Our framework seeks sparsity in representation of $\mathbf{y}$.
2. SVM seeks sparsity in decision boundaries on $A = [\mathbf{v}_1, \cdots, \mathbf{v}_n]$.
3. NN and NS do not enforce sparsity.

ℓ^1-**Minimization** vs. **SVM**: Performance of SVM depends on the choice of features.

1. Random project performs poorly with SVMs.
2. ℓ^1-Minimization guarantees performance convergence with different features.
3. At lower-dimensional space, *Fisher* features outperform.

Table: III. Linear SVM

Dimension	30	54	130	540
Eigen [%]	73	84.3	89	92
Laplacian [%]	73.4	85.8	90.8	95.7
Random [%]	54.1	70.8	81.6	88.8
Down [%]	51.4	73	83.4	90.3
Fisher [%]	86.3	93.3	N/A	N/A

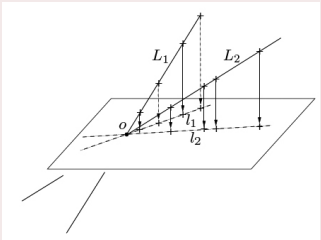
Table: IV. ℓ^1-Minimization

30	54	130	540
71.1	80	85.7	92
73.7	84.7	91	94.3
57.8	75.5	87.6	94.7
46.8	67	84.6	93.9
87	92.3	N/A	N/A

Randomfaces

Blessing of Dimensionality [Donoho 2000]

In high-dimensional data space $\mathbb{R}^D$, with **overwhelming probability**, ℓ^1/ℓ^0 equivalence holds for random projection R.



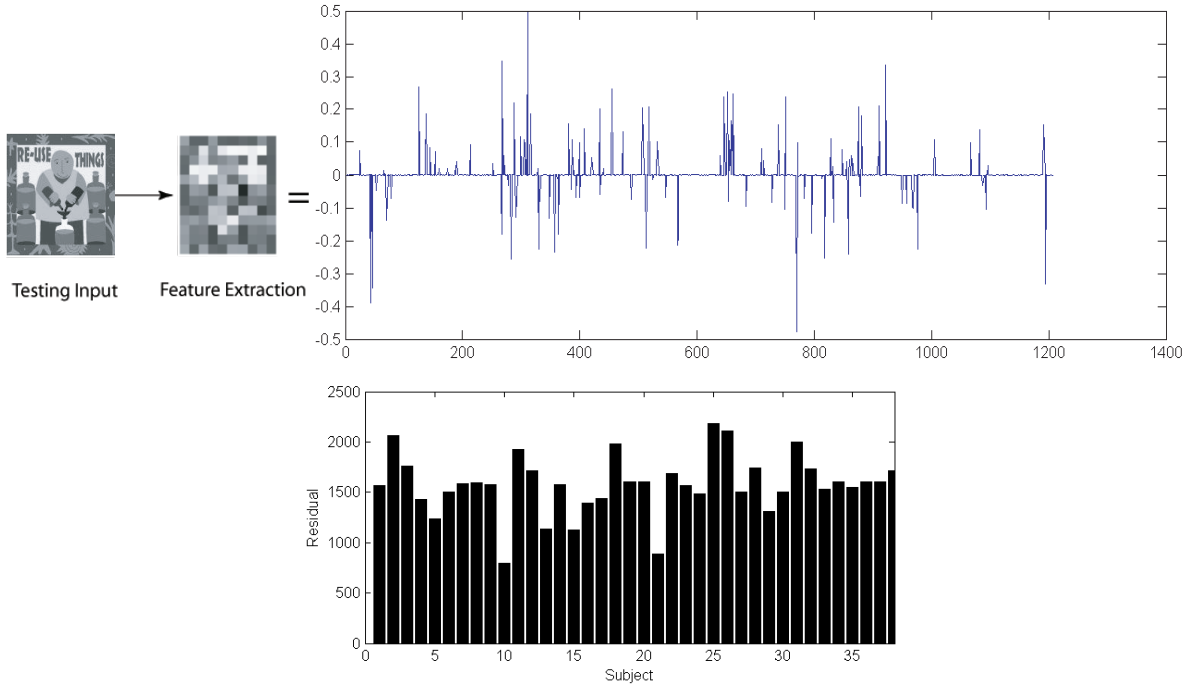
Unconventional properties:

1. Domain independent!
2. Data independent!
3. Fast to generate and compute!

Reference: Yang et al. *Feature selection in face recognition: A sparse representation perspective*. Berkeley Tech Report, 2007.

Variation: Outlier Rejection

- ℓ^1-Coefficients for invalid images

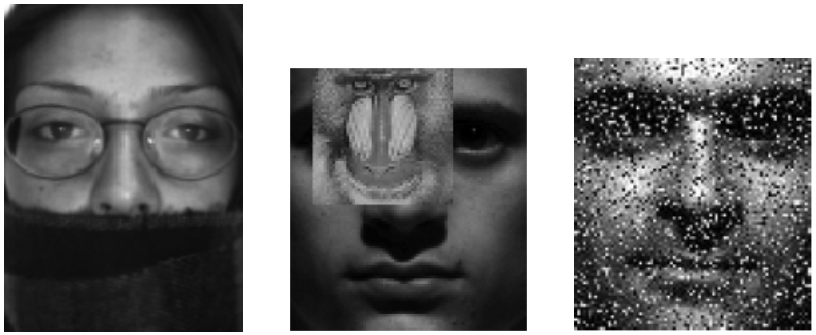


Outlier Rejection

When ℓ^1-solution is not sparse or concentrated to one subspace, the test sample is invalid.

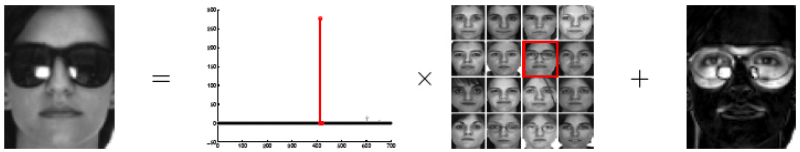
$$\text{Sparsity Concentration Index: } \text{SCI}(\mathbf{x}) \doteq \frac{K \cdot \max_i \|\delta_i(\mathbf{x})\|_1 / \|\mathbf{x}\|_1 - 1}{K - 1} \in [0, 1].$$

Variation: Occlusion Compensation



1. Sparse representation + sparse error

$$\mathbf{y} = A\mathbf{x} + \mathbf{e}$$



2. Occlusion compensation:

$$\mathbf{y} = \begin{pmatrix} A & | & I \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{e} \end{pmatrix} = B\mathbf{w}$$

Reference: Wright et al. *Robust face recognition via sparse representation*. UIUC Tech Report, 2007.

Distributed Pattern Recognition

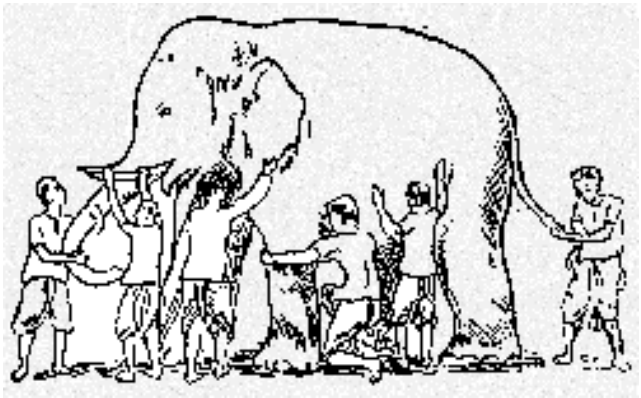


Figure: **d-Oracle**: Distributed object recognition via camera wireless networks.

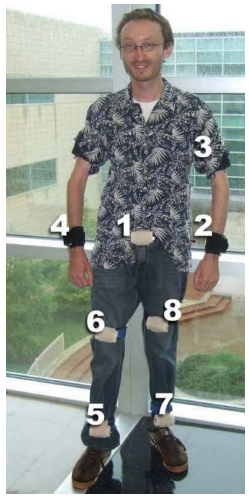
Key components:

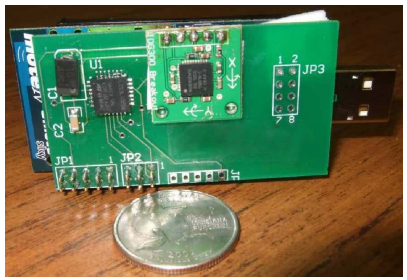
1. Each sensor only observes partial profile of the event: Demand a global classification framework.
2. Individual sensor obtains limited classification ability: Sensors become active only when certain events are locally detected.
3. The network configuration is dynamic: Global classifier needs to adapt to change of active sensors.

Problem Formulation for Distributed Action Recognition

Architecture

- 8 sensors distributed on human body.
- Location of the sensors are given and fixed.
- Each sensor carries triaxial accelerometer and biaxial gyroscope.
- Sampling frequency: 20Hz.





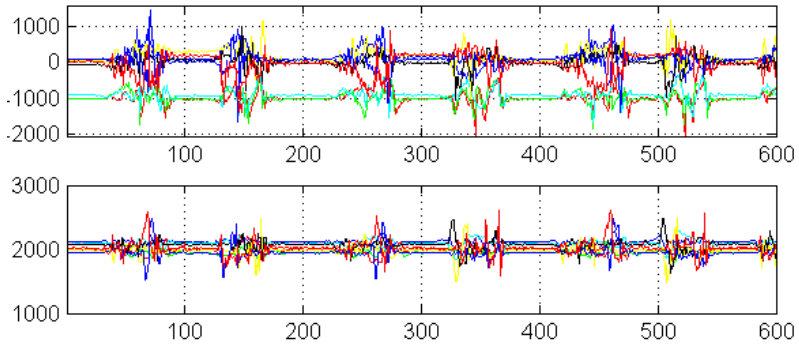


Figure: Readings from 8 x-axis accelerometers and x-axis gyroscopes for a *stand-kneel-stand* sequence.

Challenges for Distributed Action Recognition

1. Simultaneous segmentation and classification.
2. Individual sensors not sufficient to classify all human actions.
3. Simulate **sensor failure and network congestion** by different subsets of active sensors.
4. **Identity independence**: The prior examples of the subject for testing are excluded as part of training data.

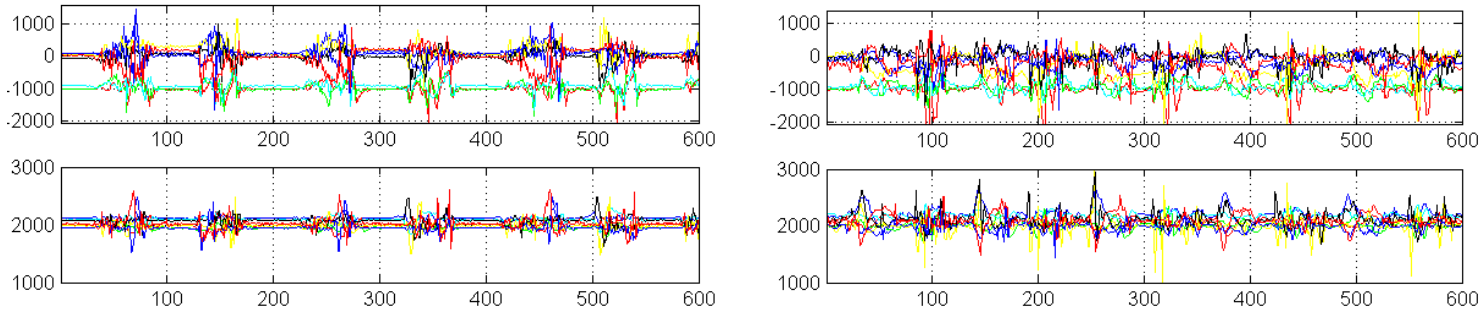
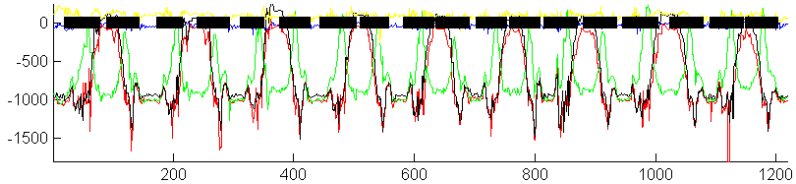


Figure: Same actions performed by two subjects.

Mixture Subspace Model for Distributed Action Recognition

1. **Training samples** are segmented manually with correct labels.



2. On each sensor node i, **normalize the vector form** (via stacking)

$$\mathbf{v}_i = [x(1), \cdots, x(h), y(1), \cdots, y(h), z(1), \cdots, z(h), \theta(1), \cdots, \theta(h), \rho(1), \cdots, \rho(h)]^T \in \mathbb{R}^{5h}$$

3. **Full body motion**

$$\text{Training sample: } \mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix} \quad \text{Test sample: } \mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} \in \mathbb{R}^{8 \cdot 5h}$$

4. **Mixture subspace model**

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} = \left(\begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_1, \cdots, \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_n \right) \mathbf{x} = A\mathbf{x}.$$

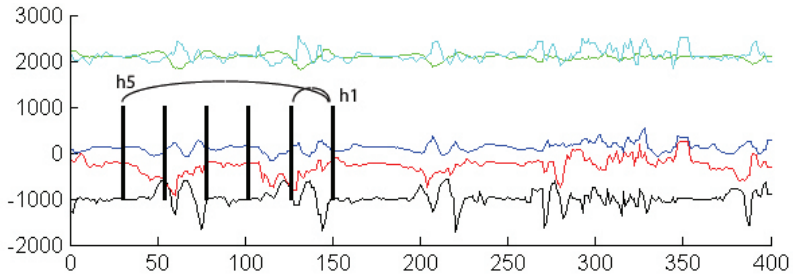
Localized Classifiers

Distributed Sparse Representation

$$\begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} = \left(\begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_1, \cdots, \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_n \right) \mathbf{x} \Leftrightarrow \begin{cases} \mathbf{y}_1 = (\mathbf{v}_{1,1}, \cdots, \mathbf{v}_{1,n})\mathbf{x} \\ \vdots \\ \mathbf{y}_8 = (\mathbf{v}_{8,1}, \cdots, \mathbf{v}_{8,n})\mathbf{x} \end{cases}$$

On each sensor node i:

1. Given a (long) test sequence at time t, apply multiple duration hypotheses: $\mathbf{y}_i \in \mathbb{R}^{5h}$.



2. Choose **Fisher features** $R_i \in \mathbb{R}^{10 \times 5h}$:

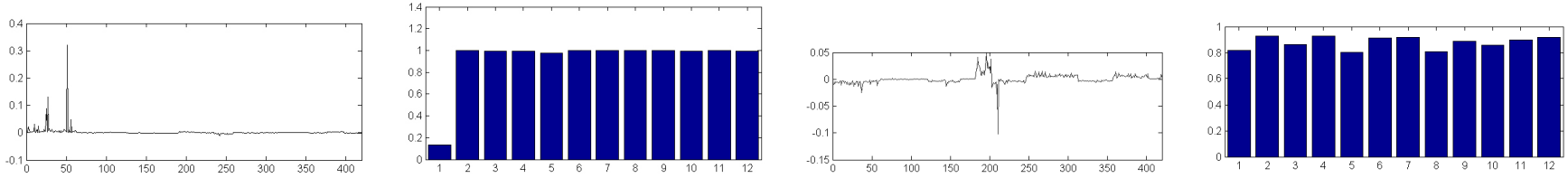
$$\tilde{\mathbf{y}}_i = R_i \mathbf{y}_i = R_i A_i \mathbf{x} = \tilde{A}_i \mathbf{x} \in \mathbb{R}^{10}$$

Localized Classifiers

1. Equivalently, **define** $R_i' = (0 \cdots R_i \cdots 0)$

$$\tilde{\mathbf{y}}_i = (0 \cdots R_i \cdots 0) \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} = (0 \cdots R_i \cdots 0) \left(\begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_1, \cdots, \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_n \right) \mathbf{x} = R_i' A \mathbf{x} \in \mathbb{R}^{10}$$

2. For all segmentation hypotheses, apply **sparsity concentration index** (SCI) threshold σ_1:



(a) valid segmentation

(b) invalid segmentation

Local Sparsity Threshold σ_1

- If $\text{SCI}(\mathbf{x}) > \sigma_1$, sensor i becomes **active** and transmits $\tilde{\mathbf{y}}_i \in \mathbb{R}^{10}$.
- $\tilde{\mathbf{y}}_i$ provides a **segmentation hypothesis** at time t and length h_i.

Adaptive Global Classifier

- Adaptive classification for a subset of active sensors (Suppose $1, \ldots, L$ at time t and h_i)

  Define **global feature matrix** $R' = \begin{pmatrix} R_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & R_L & \cdots & 0 \end{pmatrix}$:

$$\begin{pmatrix} \tilde{\mathbf{y}}_1 \\ \vdots \\ \tilde{\mathbf{y}}_L \end{pmatrix} = R' \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} = R' \begin{pmatrix} A_1 \\ \vdots \\ A_8 \end{pmatrix} \mathbf{x} = R' A \mathbf{x}$$

- Global segmentation: Regardless of L active sensors, given a global threshold σ_2,
  If SCI($\mathbf{x}$) $> \sigma_2$, accept $\mathbf{y}$ as global segmentation with label given by $\mathbf{x}$.

Distributed Classification via Compressed Sensing

1. Reformulate adaptive classification via feature matrix R':

$$\text{Local: } R' = (0 \cdots R_i \cdots 0) \quad \text{Global: } R' = \begin{pmatrix} R_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & R_L & \cdots & 0 \end{pmatrix} \Leftrightarrow R'\mathbf{y} = R'A\mathbf{x}$$

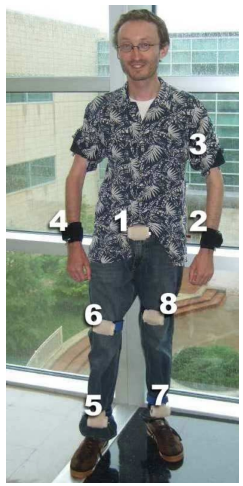
2. The representation $\mathbf{x}$ and training matrix A remain **invariant**.
3. **Segmentation, recognition, and outlier rejection are unified on $\mathbf{x}$.**

Experiment

- Algorithm only depends on two parameters: (σ_1, σ_2).
- Data communications between sensors and station are 10-D action features.

Precision vs Recall:

Sensors	2	7	2,7	1,2,7	1- 3, 7,8	1- 8
Prec [%]	89.8	94.6	94.4	92.8	94.6	98.8
Rec [%]	65	61.5	82.5	80.6	89.5	94.2



Confusion Tables:

Class (total)	1	2	3	4	5	6	7	8	9	10	11	12
1 StSi (60)	60	0	0	0	0	0	0	0	0	0	0	0
2 SiSt (60)	0	52	0	0	0	0	0	0	0	0	0	0
3 SiLi (62)	1	0	58	0	0	0	0	0	0	0	0	0
4 LiSi (62)	0	0	0	60	0	0	0	0	0	0	0	0
5 Bend (30)	1	0	0	0	29	0	0	0	0	0	0	0
6 StKn (33)	0	0	0	0	0	31	0	0	0	0	0	0
7 KnSt (30)	0	0	0	0	0	0	30	0	0	0	1	0
8 RoR (95)	0	0	0	0	0	0	0	93	0	0	0	1
9 RoL (96)	0	0	0	0	0	0	0	0	96	0	0	0
10 Jump (34)	0	0	0	0	0	0	0	0	0	31	0	0
11 Up (33)	0	0	0	0	0	0	0	0	0	0	24	0
12 Down (31)	0	0	0	0	0	0	0	0	0	0	3	26

(a) Sensor 1-8

Class (total)	1	2	3	4	5	6	7	8	9	10	11	12
1 StSi (60)	0	0	0	0	2	0	0	0	0	0	0	0
2 SiSt (60)	0	1	0	0	1	0	0	2	0	0	0	0
3 SiLi (62)	0	0	59	0	0	0	0	0	0	0	0	0
4 LiSi (62)	0	0	0	60	0	0	0	0	0	0	0	0
5 Bend (30)	4	0	0	0	9	0	0	0	0	0	0	0
6 StKn (33)	0	0	0	0	0	28	0	0	0	0	0	0
7 KnSt (30)	0	0	0	0	0	0	20	0	1	0	0	0
8 RoR (95)	0	0	0	0	1	0	0	82	0	0	0	0
9 RoL (96)	0	0	0	0	0	0	0	0	90	0	0	0
10 Jump (34)	0	0	0	0	0	0	0	0	0	22	0	0
11 Up (33)	1	0	0	0	3	0	0	1	0	0	4	0
12 Down (31)	0	0	0	0	0	0	0	0	5	0	0	10

(b) Sensor 7

Reference: Yang et al. *Distributed Segmentation and Classification of Human Actions Using a Wearable Motion Sensor Network*. Berkeley Tech Report, 2007.

Conclusion

1. **Sparsity** is important for classification of HD data.
2. A new recognition framework via **compressed sensing**.
3. In HD feature space, choosing an "optimal" feature becomes not significant.
4. **Randomfaces**, **outliers**, **occlusion**.
5. Distributed pattern recognition in body sensor networks.

Future Directions

- Distributed camera networks

- Biosensor networks in health care

Acknowledgments

Collaborators

- Berkeley: Shankar Sastry, Ruzena Bajcsy
- UIUC: Yi Ma
- UT-Dallas: Roozbeh Jafari

MATLAB Toolboxes

- ℓ^1-**Magic** by Candès at Caltech.
- **SparseLab** by Donoho at Stanford.
- **cvx** by Boyd at Stanford.

References

- *Robust face recognition via sparse representation*. Submitted to PAMI, 2008.
- *Distributed segmentation and classification of human actions using a wearable motion sensor network*. Berkeley Tech Report 2007.