A (somewhat) Unified Approach to Semi-supervised and Unsupervised Learning

Ben Recht

Center for the Mathematics of Information

Caltech

April 11, 2007

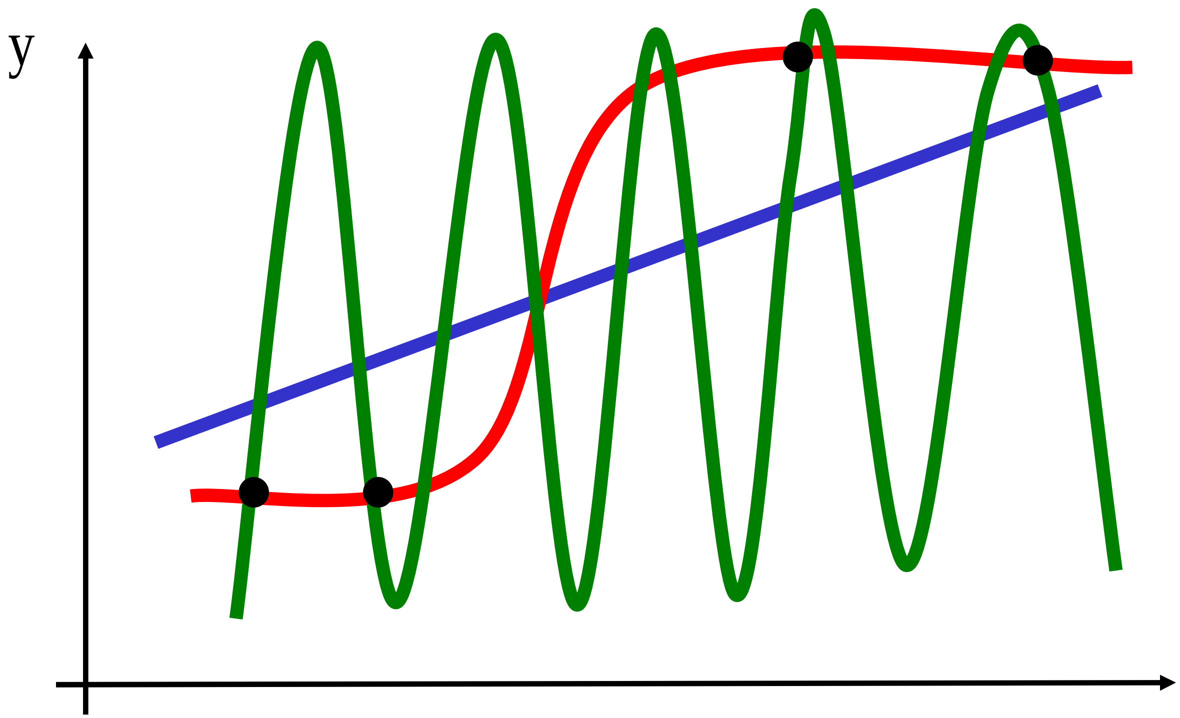
Joint work with Ali Rahimi (Intel Research)

Overview

- By abusing the standard Tikhonov regularization functional, we can derive most “kernel methods” and many new novel techniques.
- KPCA
- Semi-supervised Classification and Clustering
- Transforming Time Series with Few Examples
- Other applications (not today, sorry)
  - Kernel Learning
  - Robust SVMs and Learning with missing data
  - Constraints and Conservation Laws

Priors and "Semi-Supervision"

- **Unlabeled:** ● ● ●●● ● ● ●● ●●● ●●● ●●●



- How to leverage the unlabelled data?

Video

Representation

- Big mess of numbers for each frame

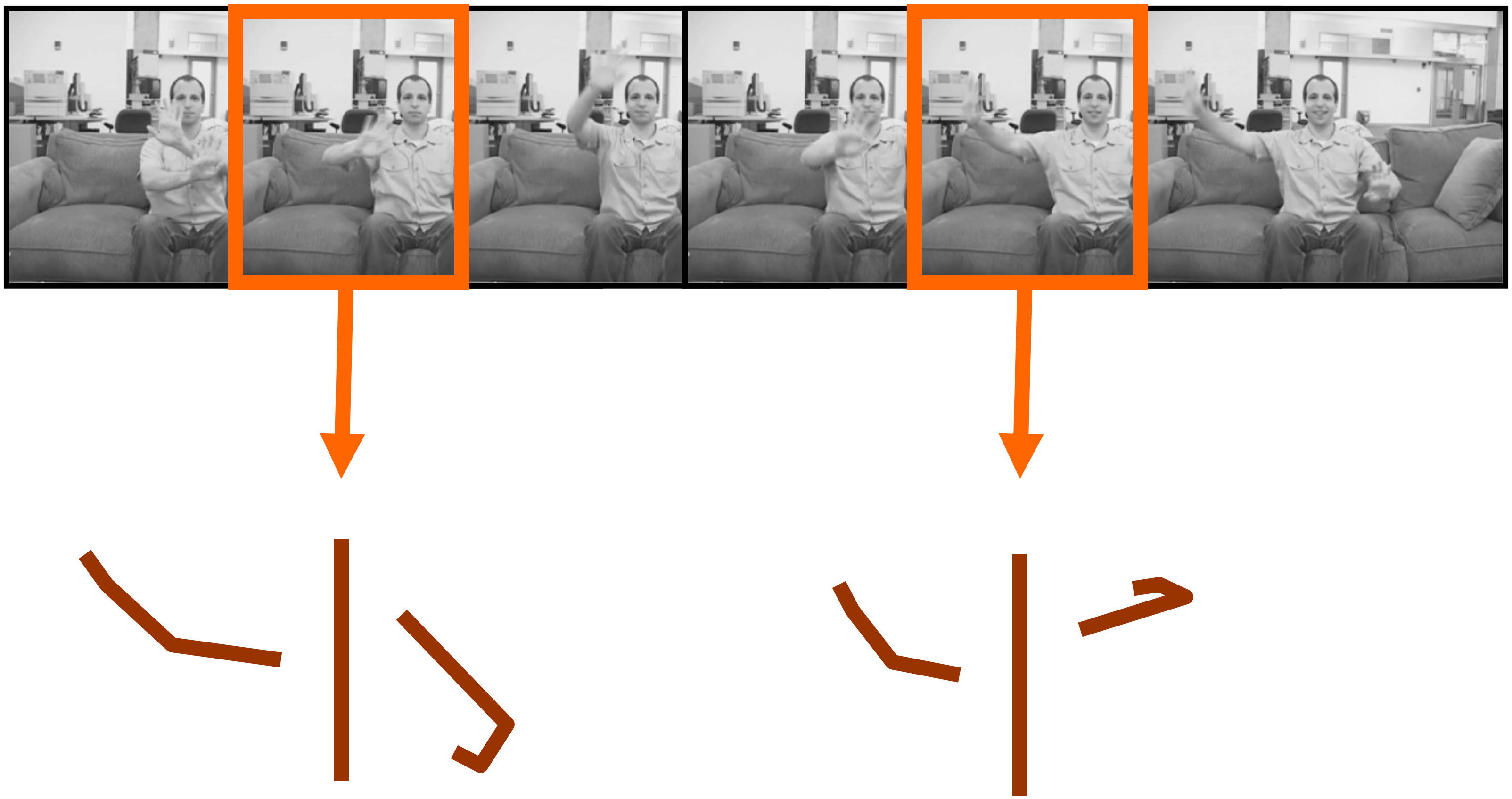
$$\rightarrow \begin{bmatrix} \vdots \\ 43 \\ 76 \\ 121 \\ 147 \\ 158 \\ 170 \\ 172 \\ 168 \\ 169 \\ 176 \\ \vdots \end{bmatrix}$$

- Raw pixels, no image processing

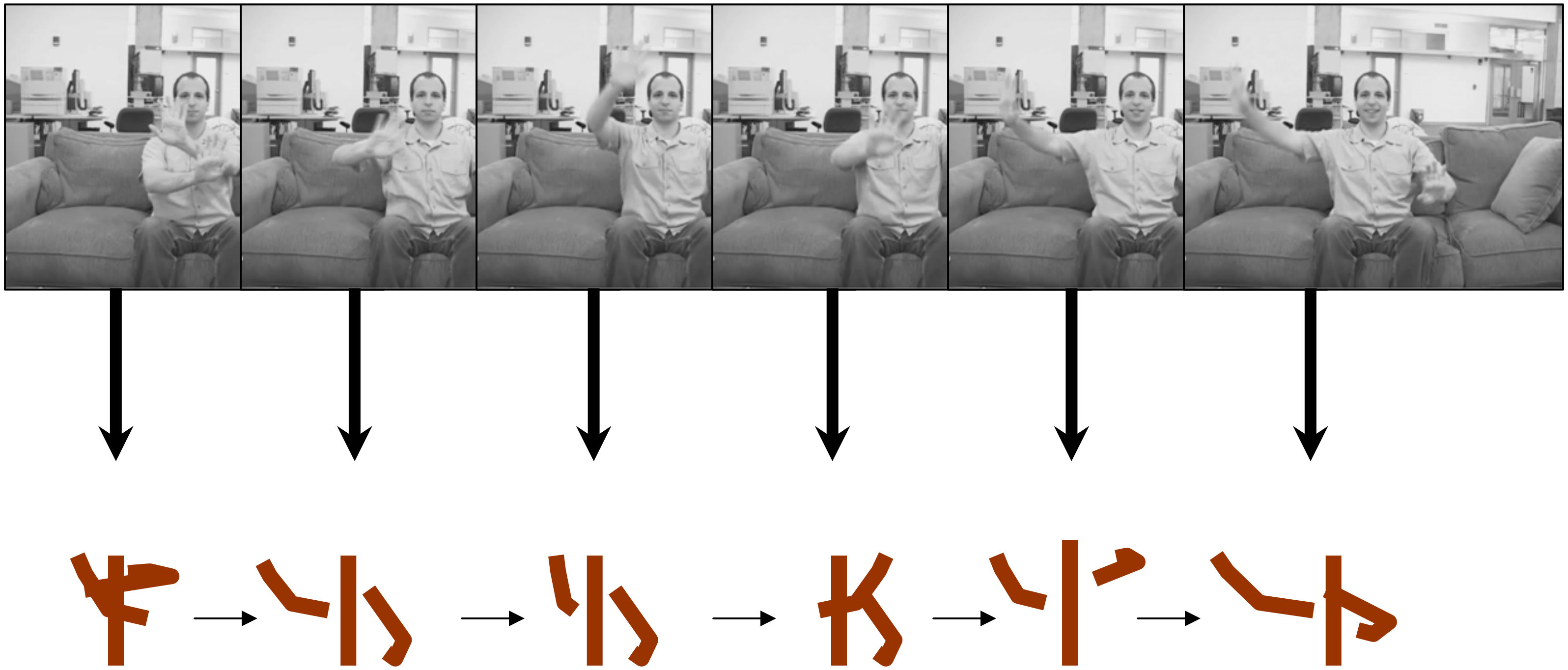
Representation

- We want to extract position of limbs

Left Hand
Left Elbow
Right Hand
Right Elbow



Annotations from user or detection algorithms



Assume that output time series is smooth.

Approach

- Look for smooth mapping from images to positions
- Annotate a subset of the frames
- Assume output obeys physical laws
- Video

Nonlinear Regression

- Let $\mathcal{H}$ be an RKHS, and consider the Tikhonov Regularization functional

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{L} V(f(\mathbf{x}_i), y_i) + \lambda \|f\|_K^2$$

- Solution: $f(\mathbf{x}) = \sum_{i=1}^{L} c_i \mathbf{k}(\mathbf{x}_i, \mathbf{x})$

Augmented Nonlinear Regression

- Suppose we add a penalty term constraining the outputs and kernel

$$\min_{f,y} \sum_{i=1}^{L} V(f(x_i), y_i) + \lambda \|f\|_K^2 + \mathcal{S}(y)$$

Search over f *and* y

Additional costs / constraints on y

- Solution: $f(\mathbf{x}) = \sum_{i=1}^{L} c_i \mathbf{k}(\mathbf{x}_i, \mathbf{x})$

A variety of learning algorithms

Constraints	**Algorithm**
None	Regression/ Classification
Outputs are binary	Clustering/ Transduction
Local geometry of the outputs	Manifold Learning/ KPCA
Output obeys linear dynamical relations	Manifolds from Video

Least-Squares Cost

- We can eliminate the function for practical purposes, recovering it from the computed y_i.

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{L} (f(\mathbf{x}_i) - y_i)^2 + \lambda \|f\|_K^2$$

- By representer theorem, we may rewrite this

$$\min_{f \in \mathcal{H}} \|\mathbf{Kc} - \mathbf{y}\|^2 + \lambda \mathbf{c}'\mathbf{Kc}$$

Least-Squares Cost

$$\min_{f \in \mathcal{H}} \|\mathbf{Kc} - \mathbf{y}\|^2 + \lambda \mathbf{c}'\mathbf{Kc}$$

- Solving for c gives $\mathbf{c} = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}$
- Plugging in this solution gives $\lambda \mathbf{y}'(\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}$
- Here **y** is the vector of all of the y_i

Multiple dimensions

- Suppose we want a vector valued function f:$\mathsf{R}^D \rightarrow \mathsf{R}^d$.
- We penalize each component individually

$$\min_{f \in \mathcal{H}} \sum_{j=1}^{d} \sum_{i=1}^{L} (f_j(\mathbf{x}_i) - y_{ji})^2 + \lambda \|f_j\|_K^2$$

- We may solve for f to find the minimum cost is given by

$$\lambda \sum_{i=1}^{d} \mathbf{y}_i'(\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}_i$$

Multiple dimensions

- Let $\mathbf{Y} = [y_{ji}]' \quad j = 1, \ldots, d \quad i = 1, \ldots, L$
- This is a d x L matrix.
- Then our optimal cost can be written succinctly as

$$\lambda \, \mathrm{Tr}(\mathbf{Y}(\mathbf{K} + \lambda \mathbf{I}_L)^{-1} \mathbf{Y}')$$

Kernel PCA

- Let f:$\mathsf{R}^D \rightarrow \mathsf{R}^d$ with D>d. Assume that the set of outputs is white and zero mean:

$$\min_{\mathbf{Y}} \mathrm{Tr}(\mathbf{Y}(\mathbf{K}+\lambda \mathbf{I}_L)^{-1}\mathbf{Y}')$$
$$\text{s.t.} \mathbf{Y}\mathbf{Y}' = \mathbf{I}_d$$
$$\mathbf{Y}\mathbf{1}_N = \mathbf{0}_d$$

- Can be solved as an eigenvalue problem. (Shoelkopf et al, '98)

Kernel Principal Components

- Solutions are the eigenvalues of K projected onto the zero-mean subspace of the RKHS.
- Since $c = (K+\lambda\ I)^{-1}\ y$, the resulting coefficients are also eigenvalues of K when the lifted data is zero-mean.
- Centering the data in feature space is often useful in unsupervised learning.
- Regularization parameter only controls the scale of each component.

Centered Kernels

- Constraining the **Y** to have zero column sum results in a hard eigenvalue problem.
- If we instead insist that $\sum_i f(x_i) = 0$, we get the ordinary eigenvalue problem

$$\min_{\mathbf{Y}} \mathrm{Tr}(\mathbf{Y}(\hat{\mathbf{K}} + \lambda \mathbf{I}_L)^{-1}\mathbf{Y}')$$
$$\text{s.t.} \mathbf{Y}\mathbf{Y}' = \mathbf{I}_d$$

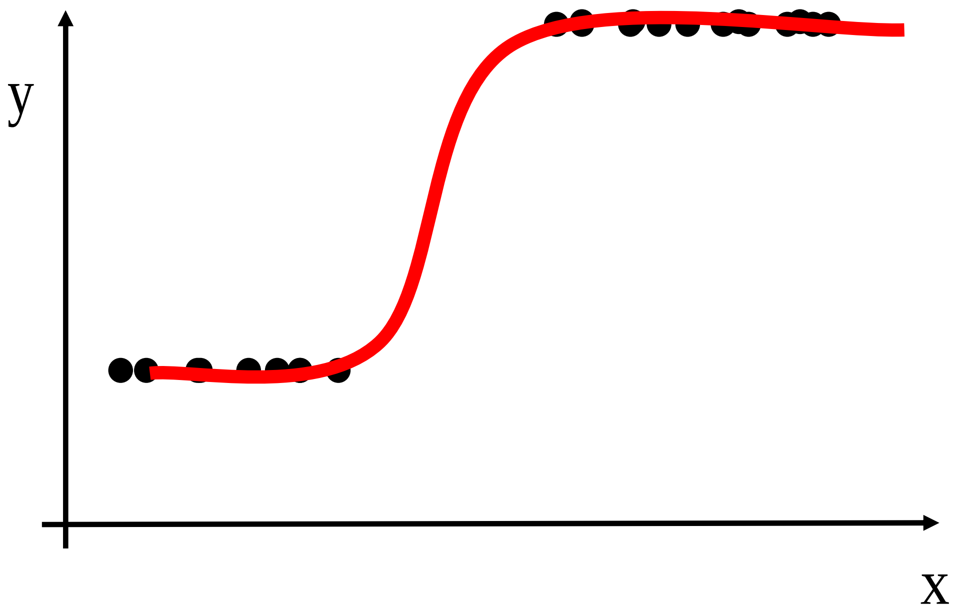
where $\hat{\mathbf{K}} = (\mathbf{I} - \mathbf{1}_N \mathbf{1}_N')\mathbf{K}(\mathbf{I} - \mathbf{1}_N \mathbf{1}_N')$

- The components are now just the eigenvalues of $\hat{\mathbf{K}}$
- You don't have to invert anything.

Clustering and Segmentation

Classification on RKHS

$$\min_{f,\mathbf{y}} \quad \sum_{i=1}^{N} V(f(\mathbf{x}_i), y_i) + \lambda \|f\|_K^2$$



- Tikhonov Regularization
- Labels set to 1 or -1
- Just choose a loss

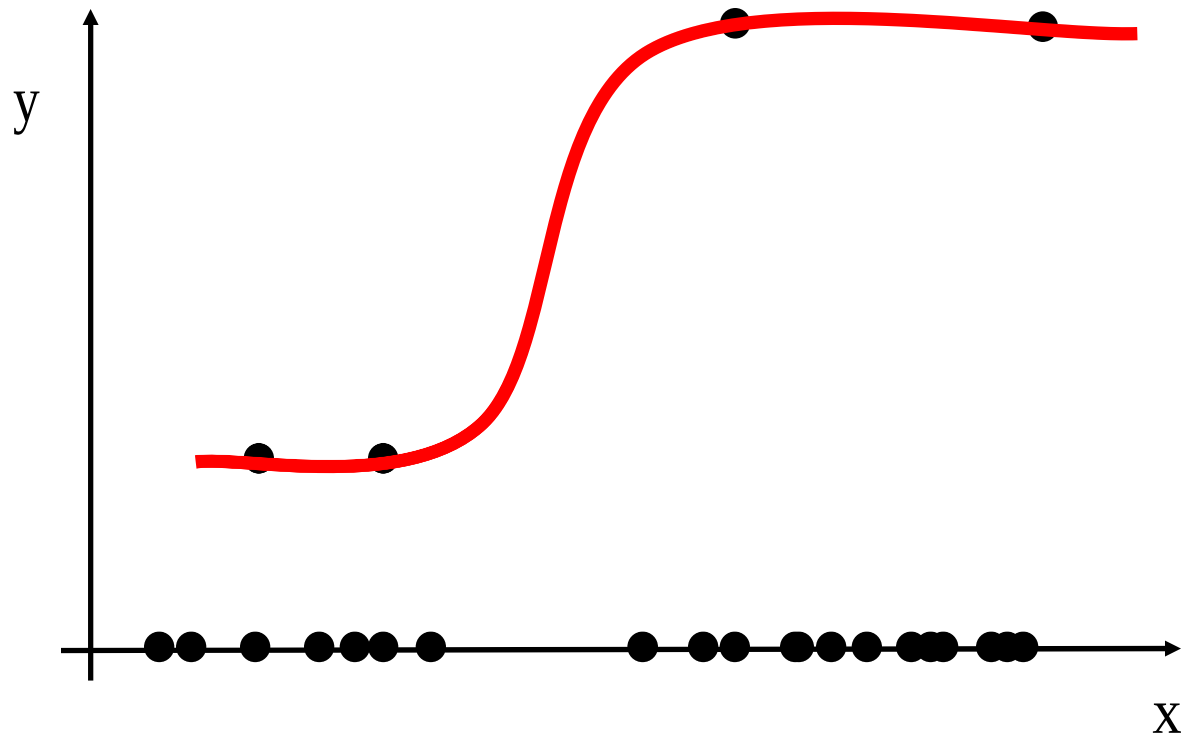
Classsification

- Example costs:

$V(f(\mathbf{x}_i), y_i)$	Classifier
$(y_i - f(\mathbf{x}_i))^2$	RBF
$\max(0, 1 - f(\mathbf{x}_i) y_i)$	SVM
$\log \text{Bin}(y_i \| \text{logit}(f(\mathbf{x}_i)))$	GPR

Transduction

- Sparsely labeled data



Taxonomy

- **Classification:** function fitting with ± 1 labels
- **Transduction:** function fitting with ± 1 labels, some of the labels withheld
- **Segmentation/Clustering:** function fitting with ± 1 labels, all of the labels withheld
- Conceptually related/algorithmically related

Alternative Approaches

- Density Estimation
  - Local minima, not well conditioned for large dimension
- Local Search for Binary Labels
  - Can’t guarantee performance
- Graph Cuts
  - Is a special case of what follows…

Transduction and Segmentation

$$\begin{aligned} \min_{\mathbf{y}} \quad & \mathbf{y}'(\hat{\mathbf{K}} + \lambda \mathbf{I})^{-1}\mathbf{y} \\ \text{s.t.} \quad & y_i^2 = 1 \end{aligned}$$

- Start with zero-meaned Tikhonov Regularization
- Force labels to be 1 or -1
- NP-Hard

Approximation 1: Eigenvalue

$$\min_{\mathbf{y}} \quad \mathbf{y}^\top(\widehat{\mathbf{K}} + \lambda\mathbf{I})^{-1}\mathbf{y}$$
$$\text{s.t.} \quad y_i^2 = 1$$

⇓ **Sum constraints**

$$\min_{\mathbf{y}} \quad \mathbf{y}^\top(\widehat{\mathbf{K}} + \lambda\mathbf{I})^{-1}\mathbf{y}$$
$$\text{s.t.} \quad \sum_{i=1}^{N} \alpha_i y_i^2 = \sum_{i=1}^{N} \alpha_i$$

- Pick $\alpha_i \geq 0$.
- Solve as Generalized Eigenvalue Problem
- Surprisingly good in practice, reasonably efficient
- Of course, how you pick the α is ad hoc
- Best α can be computed by semidefinite programming

Approximation 2: Duality

$$\min_{\mathbf{y}} \quad \mathbf{y}^\top(\hat{\mathbf{K}} + \lambda \mathbf{I})^{-1}\mathbf{y}$$
$$\text{s.t.} \quad y_i^2 = 1$$

⇓ **Dual**

$$\max_{\alpha} \quad \sum_{i=1}^{N} \alpha_i$$
$$\text{s.t.} \quad (\hat{\mathbf{K}} + \lambda \mathbf{I})^{-1} - \text{diag}(\alpha) \succeq 0$$

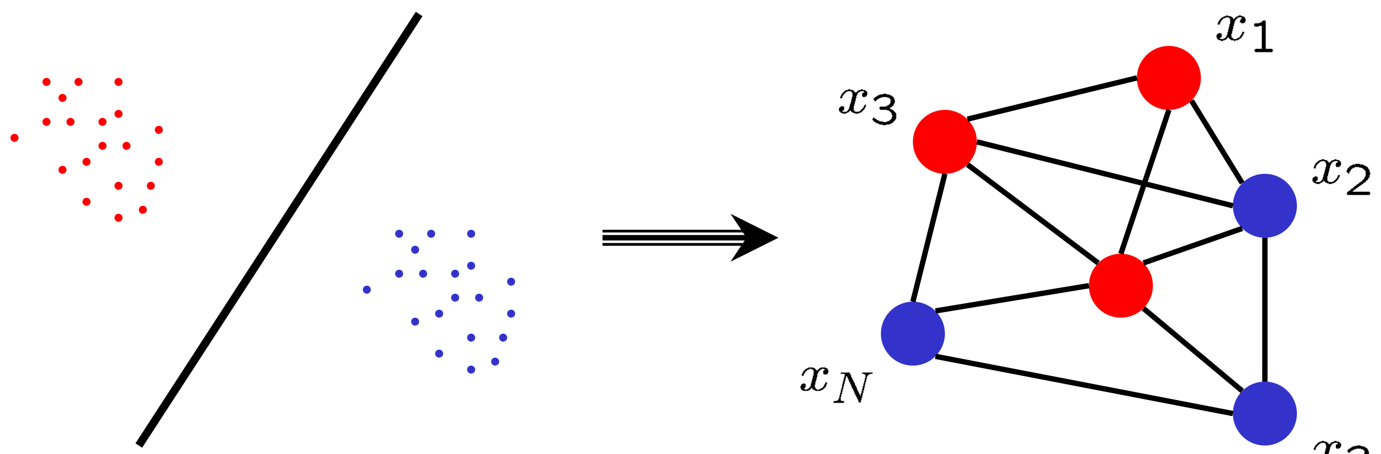
⇓ **Dual**

$$\min_{\mathbf{y}} \quad \text{Tr}((\hat{\mathbf{K}} + \lambda \mathbf{I})^{-1}\mathbf{Y})$$
$$\text{s.t.} \quad \text{diag}(Y) = 1$$
$$\mathbf{Y} \succeq 0$$

- Dual is a semidefinite program
- Randomized Algorithm of Goemans and Williamson gives you clusters.
- Compare against dual program for bounds
- Algorithms can be slow for large N

Spectral Clustering

- Freeman and Perona - Eigenvectors of adjacency matrix **K**.
- Shi and Malik – Graph Partitioning/Normalized Cuts.
- Other variants…
- **All are approximations of binary label prior!**



Normalized Cuts

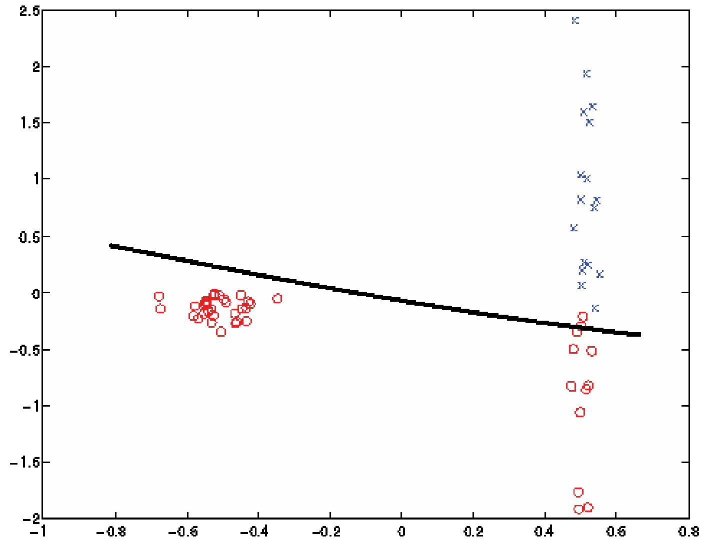
$$\begin{array}{ll} \text{min}_{\mathbf{y}} & \mathbf{y}^\top(\hat{\mathbf{K}} + \lambda \mathbf{I})^{-1}\mathbf{y} \\ \text{s.t.} & \mathbf{y}^\top \text{diag}(\alpha)\mathbf{y} = \sum_{i=1}^{N} \alpha_i \end{array}$$

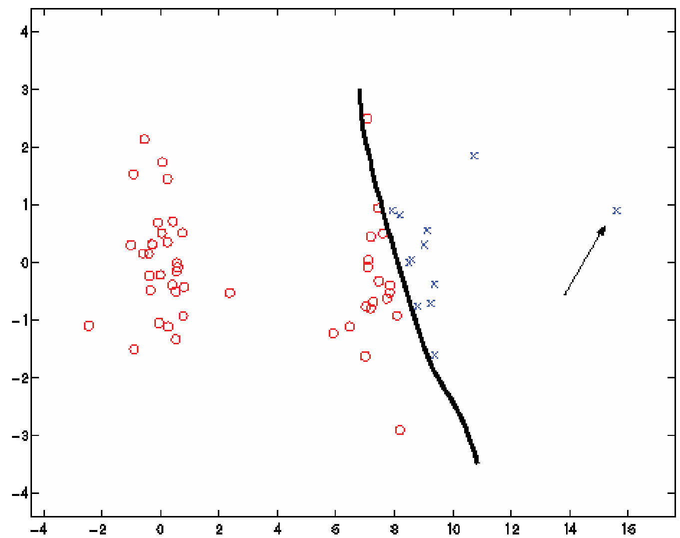
- Pick $\alpha_i = \frac{1}{\lambda + \sum_{j=1}^{N} K_{ij}}$
- Solution is second largest eigenvector of $\hat{\mathbf{D}}^{-1/2}\hat{\mathbf{K}}\hat{\mathbf{D}}^{-1/2}$
- where $\hat{\mathbf{D}} = \text{diag}(\hat{\mathbf{K}}\mathbf{1})$

Spectral Clustering sensitivity

$$\begin{array}{ll} \min_{\mathbf{y}} & \mathbf{y}^\top(\hat{\mathbf{K}} + \lambda \mathbf{I})^{-1}\mathbf{y} \\ \text{s.t.} & \mathbf{y}^\top \hat{\mathbf{D}}^{-1}\mathbf{y} = \sum_{i=1}^{N} \alpha_i \end{array}$$

- Weightings cause particular sensitivities





Solution 2: Average Gap

$$\begin{array}{ll} \min_{\mathbf{y}} & \mathbf{y}^\top(\hat{\mathbf{K}} + \lambda\mathbf{I})^{-1}\mathbf{y} \\ \text{s.t.} & \mathbf{y}^\top\mathbf{y} = N \end{array}$$

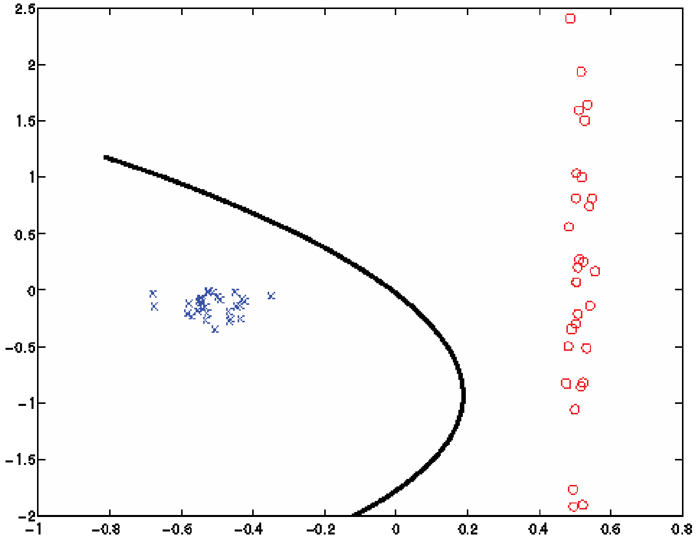
- Pick $\alpha_i = 1/N$.
- Perona-Freeman with modified kernel
- Just an Eigenvalue Problem – first KPCA component

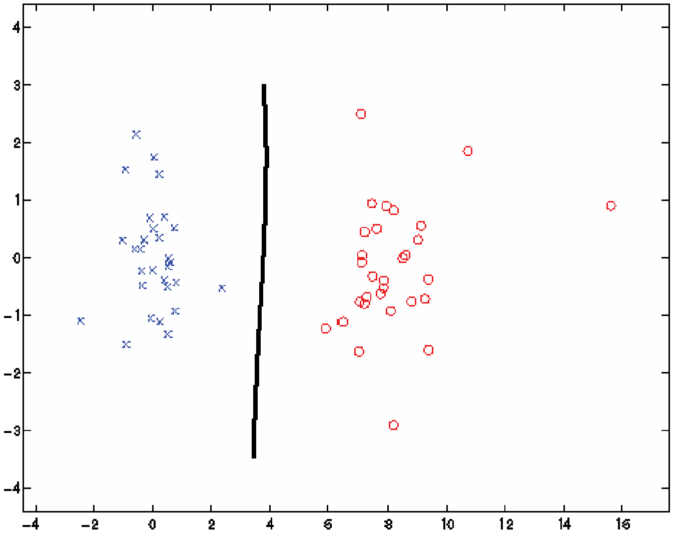
Average Gap Algorithm

- solve

$$\begin{aligned} \min_{\mathbf{y}} \quad & \mathbf{y}^\top(\hat{\mathbf{K}} + \lambda\mathbf{I})^{-1}\mathbf{y} \\ \text{s.t.} \quad & \mathbf{y}^\top\mathbf{y} = N \end{aligned}$$

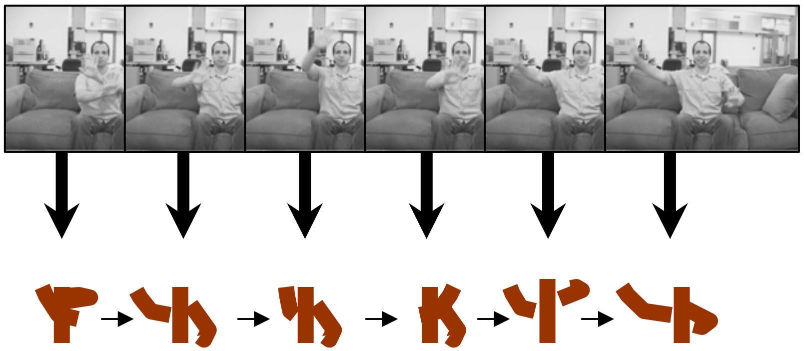
- No degree weighting

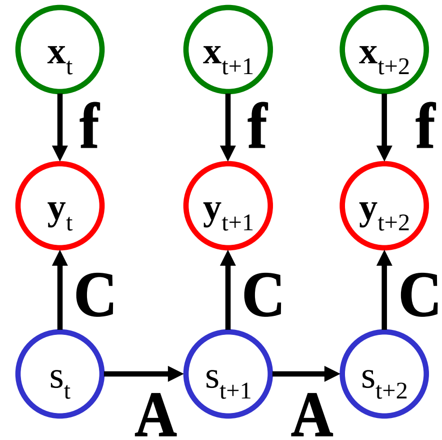


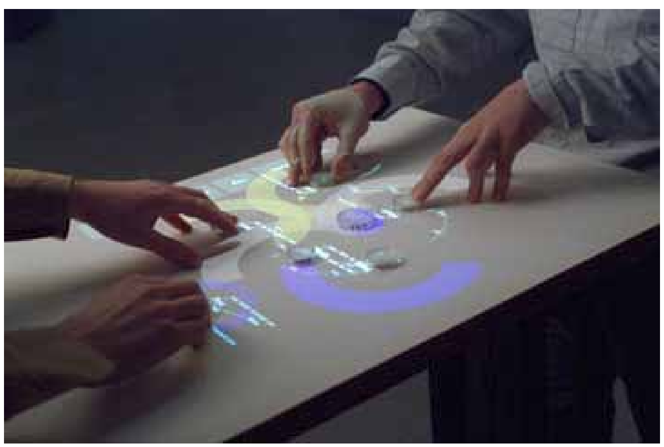


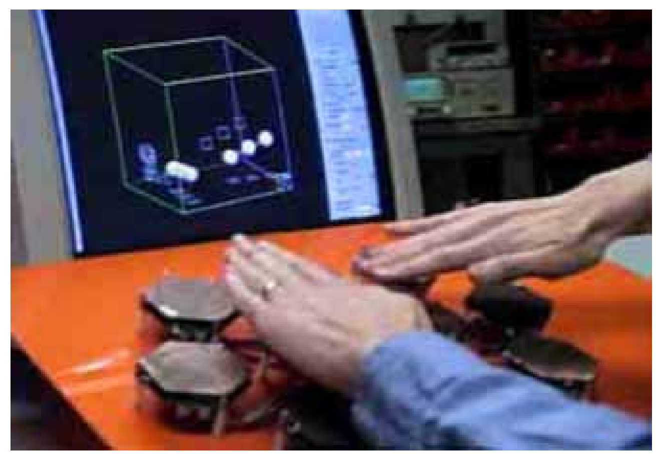
Leveraging Dynamics

Dynamics









Dynamics

$$\mathbf{s}[t+1] = \mathbf{A}\mathbf{s}[t] + \omega[t]$$
$$\mathbf{x}[t] = \mathbf{C}\mathbf{s}[t] + \nu[t]$$
$$\mathbb{E}[\omega[t]\omega[t]'] = \Lambda_\omega$$
$$\mathbb{E}[\nu[t]\nu[t]'] = \Lambda_\nu$$

Assume data is generated by an LTIG system

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

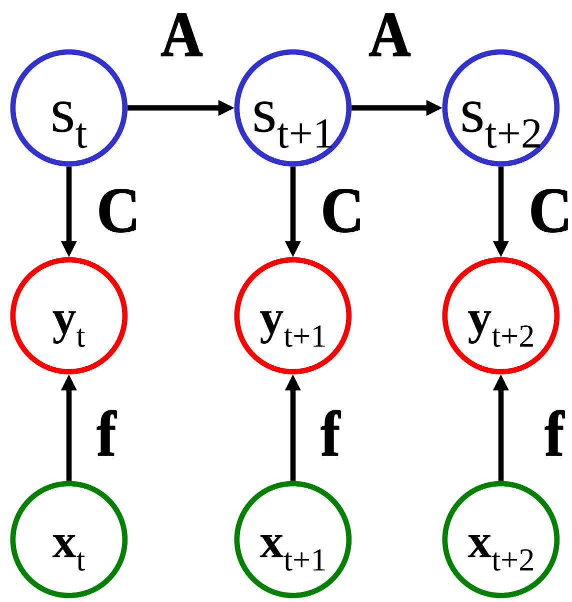
$$\mathbf{A} = \begin{bmatrix} 1 & \delta & 0 \\ 0 & 1 & \delta \\ 0 & 0 & 1 \end{bmatrix}$$

For the experiments, this model can be very dumb!

Dynamics

- Search over functions and missing data

- Assume *a priori*
  - We know (**A**,**C**)
  - **f** $\in$ RKHS is vector valued
  - Some of the $\mathbf{y}_t$ are given



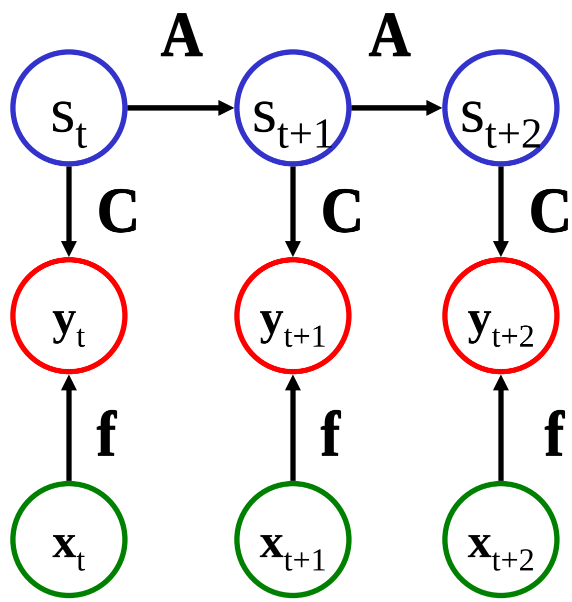
Dynamics

- Search over functions and missing data

$$\mathbf{A} = \begin{bmatrix} 1 & \delta & 0 \\ 0 & 1 & \delta \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{f}(\mathbf{x}) = \sum_{t=1}^{T} \mathbf{b}_t k(\mathbf{x}_t, \mathbf{x})$$



Optimization Problem

- Prefers outputs that evolve smoothly

$$
\begin{aligned}
\min_{\mathbf{Y},\{\mathbf{s}_t\}_{t=1..T}} \quad & \mathrm{Tr}(\mathbf{Y}(\mathbf{K}+\lambda_k \mathbf{I}_T)^{-1}\mathbf{Y}') \\
+ \quad & \lambda_d \sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{C}\mathbf{s}_t\|^2 \\
+ \quad & \lambda_d \sum_{t=2}^{T} \|\mathbf{s}_t - \mathbf{A}\mathbf{s}_{t-1}\|^2_{\Lambda_\omega} \\
\text{subject to} \quad & \mathbf{y}_\ell = \mathbf{u}_\ell
\end{aligned}
$$

Smoothness

Dynamics

Fidelity to training data

Optimization Problem

- Prefers outputs that evolve smoothly

$$\min_{\mathbf{Y}, \{\mathbf{s}_t\}_{t=1..T}} \quad \text{Tr}(\mathbf{Y}(\mathbf{K} + \lambda_k \mathbf{I}_T)^{-1}\mathbf{Y}')$$

$$+ \quad \lambda_d \sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{C}\mathbf{s}_t\|^2$$

$$+ \quad \lambda_d \sum_{t=2}^{T} \|\mathbf{s}_t - \mathbf{A}\mathbf{s}_{t-1}\|^2_{\Lambda_\omega}$$

$$\text{subject to} \quad \mathbf{y}_\ell = \mathbf{u}_\ell$$

Tikhonov Regularization

Optimization Problem

- Prefers outputs that evolve smoothly

$$\min_{\mathbf{Y},\{\mathbf{s}_t\}_{t=1..T}} \quad \text{Tr}(\mathbf{Y}(\mathbf{K}+\lambda_k \mathbf{I}_T)^{-1}\mathbf{Y}')$$

$$+ \quad \lambda_d \sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{C}\mathbf{s}_t\|^2$$

$$+ \quad \lambda_d \sum_{t=2}^{T} \|\mathbf{s}_t - \mathbf{A}\mathbf{s}_{t-1}\|^2_{\Lambda_\omega}$$

$$\text{subject to} \quad \mathbf{y}_\ell = \mathbf{u}_\ell$$

RTS Smoother (non-causal Kalman Filter)

Optimization Problem

- Semi-supervised Algorithm

$$
\begin{aligned}
\min_{\mathbf{Y},\{\mathbf{s}_t\}_{t=1..T}} \quad & \mathrm{Tr}(\mathbf{Y}(\mathbf{K}+\lambda_k \mathbf{I}_T)^{-1}\mathbf{Y}') \\
+ \quad & \lambda_d \sum_{t=1}^{T} \|\mathbf{y}_t - \mathbf{C}\mathbf{s}_t\|^2 \\
+ \quad & \lambda_d \sum_{t=2}^{T} \|\mathbf{s}_t - \mathbf{A}\mathbf{s}_{t-1}\|^2_{\Lambda_\omega} \\
\text{subject to} \quad & \mathbf{y}_\ell = \mathbf{u}_\ell
\end{aligned}
$$

Smoothness

Dynamics

Fidelity to training data

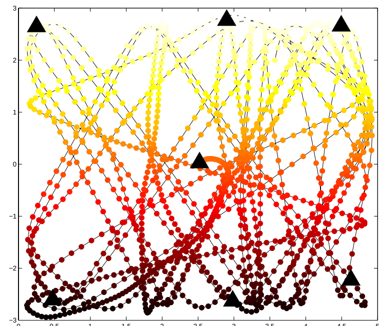
Optimization Problem

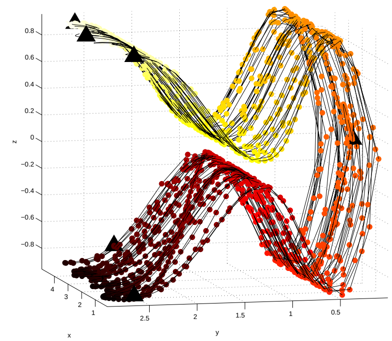
- Eliminating the state sequence by differentiation yields the following problem that may be solved by least squares

$$\begin{array}{ll} \min\limits_{\mathbf{Y}} & \mathrm{Tr}(\mathbf{Y}(\mathbf{K}+\lambda_k \mathbf{I}_T)^{-1}\mathbf{Y}') + \lambda_d \, \mathrm{Tr}(\mathbf{Y}\Omega\mathbf{Y}) \\ \text{subject to} & \mathbf{y}_\ell = \mathbf{u}_\ell \end{array}$$

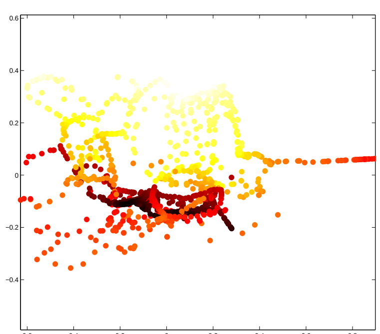
- Ω is a Toeplitz matrix that can be computed efficiently from the linear dynamics model.

Synthetic Results

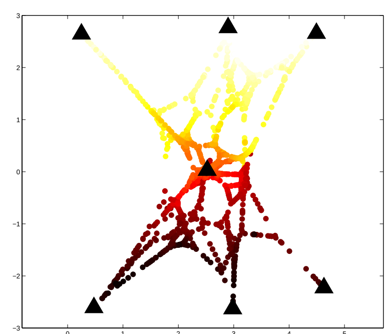




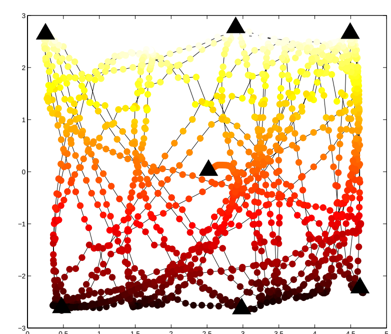
Recovered mappings:



Tennenbaum et al



Belkin/Niyogi



Rahimi/Recht

Video

Representation

- Big mess of numbers for each frame

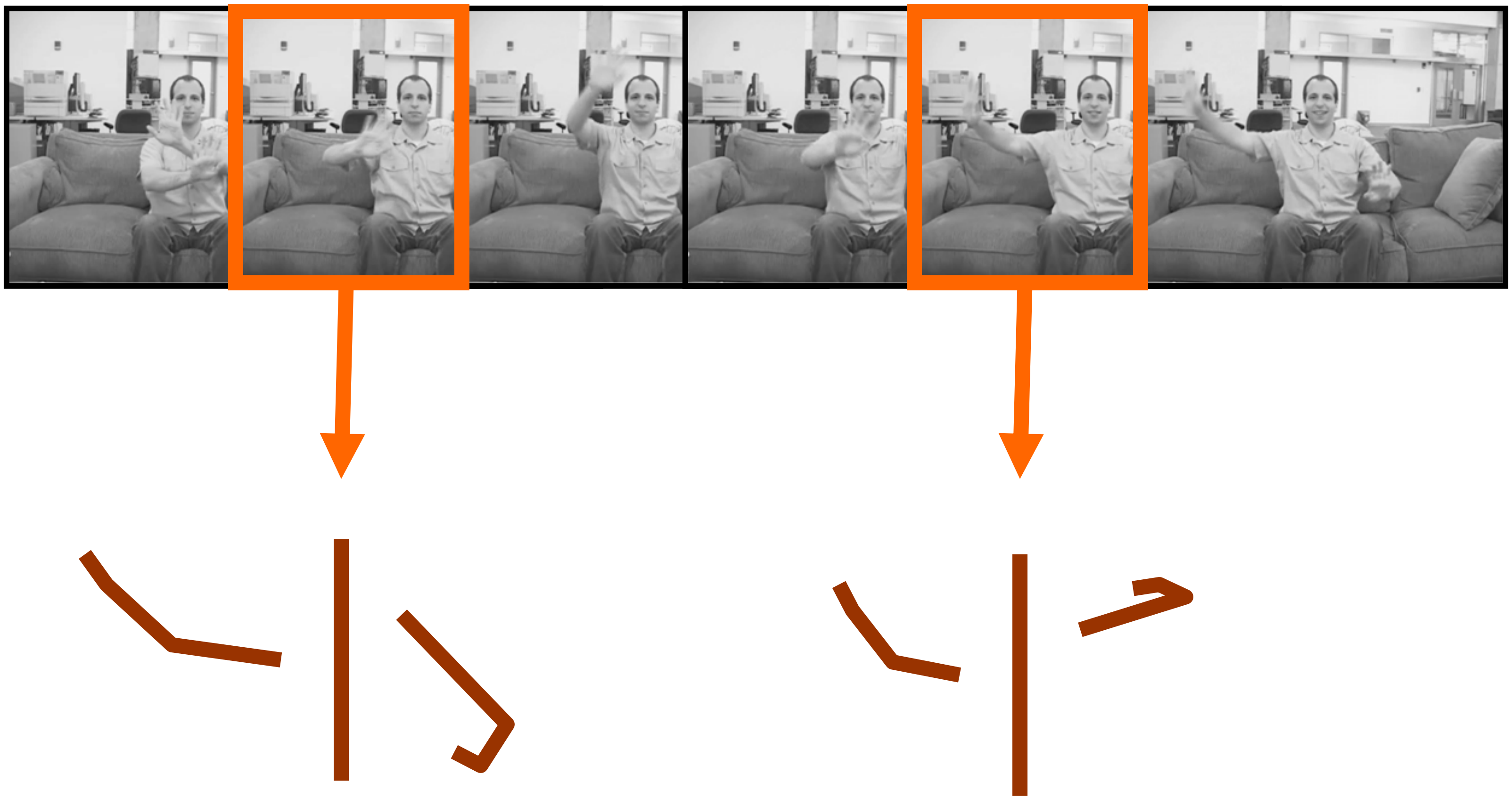
$$\rightarrow \begin{bmatrix} \vdots \\ 43 \\ 76 \\ 121 \\ 147 \\ 158 \\ 170 \\ 172 \\ 168 \\ 169 \\ 176 \\ \vdots \end{bmatrix}$$

- Raw pixels, no image processing

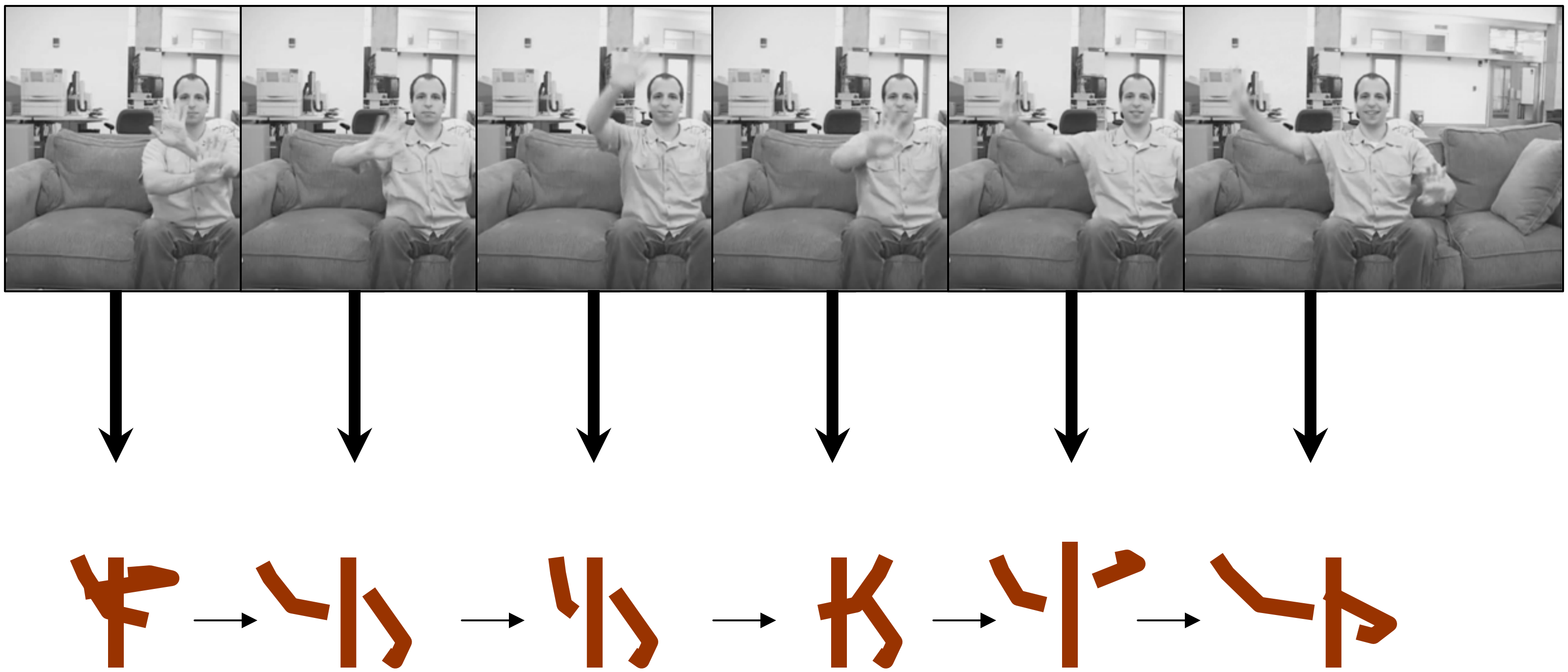
Representation

- We want to extract position of limbs

Left Hand
Left Elbow
Right Hand
Right Elbow



Annotations from user or detection algorithms



Assume that output time series is smooth.

Approach

- Look for smooth mapping from images to positions
- Annotate a subset of the frames
- Assume output obeys physical laws
- Video

References

- *Learning to Transform Time Series with a Few Examples*, Ali Rahimi, Ben Recht, in IEEE Pattern Analysis and Machine Intelligence (PAMI) (2007).
- *Clustering with Normalized Cuts is Clustering with a Hyperplane*, A. Rahimi, B. Recht, in Statistical Learning in Computer Vision (2004).
- *Convex Modeling with Priors.* Ben Recht. PhD Dissertation, MIT Media Lab (2006).