Learning to Track by Combining Flexible Discriminative Features

David Ross, Simon Osindero, and Rich Zemel

Feb 6, 2006

Overview

- Estimate continuous state variables from a sequence of observations
- Discriminative conditional model
- “Dogpile” features
  - Learn which are useful
  - Switch features on and off
- Think: CRF (but continuous + switching) or POE (but conditional)

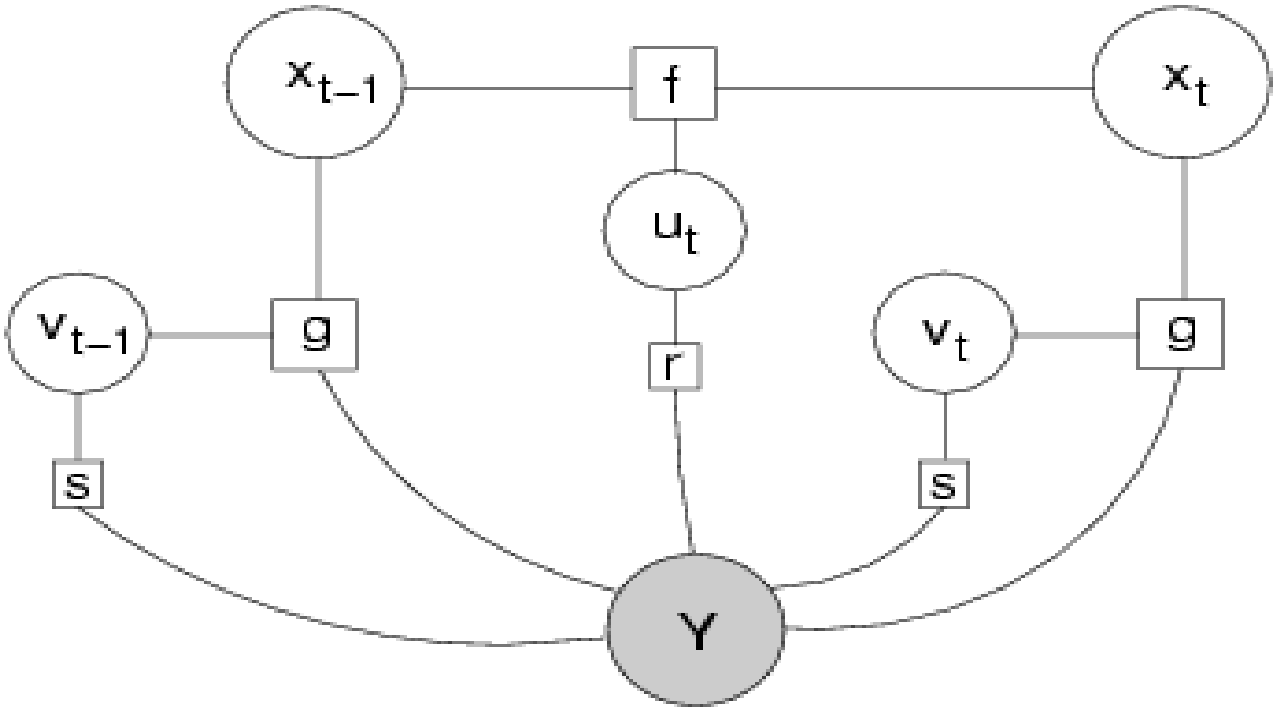
Motivation

- Standard way: generative state-space model P(Y|X) P(X) + Bayes rule
  - Assume conditional independence of observations
  - Using a single likelihood function, must model entire high-dimensional observation
- Instead: model P(X|Y) direclty
  - No C.I. -> Feature can consider any/all observations
  - Features can be discriminative
  - dogpile features, learn their relevance

Features

- Dynamics features: $f_j(\mathbf{x}_t, \mathbf{x}_{t-1})$
  - how well do two states match?
  - (non) linear dynamical models
- Observation features : $g_k(\mathbf{x}_t, \mathbf{Y}, t)$
  - is the ball at xt?
  - Any appearance model/object detector
- Robustify by switching features on and off
  - Hidden switch variables $u_{jt} \quad v_{kt}$

Probability Model



Probability Model

$$\log \sum_{\mathbf{u},\mathbf{v}} \exp\left(\sum_{t,j} f_j(\mathbf{x}_{t-1}, \mathbf{x}_t, \boldsymbol{\alpha}_j) u_{jt} + \sum_{t,k} g_k(\mathbf{x}_t, \mathbf{Y}, \boldsymbol{\beta}_k) v_{kt} \right.$$
$$\left. + \sum_{t,j} \mathcal{F}_j(\mathbf{Y}, t) u_{jt} + \sum_{t,k} \mathcal{G}_k(\mathbf{Y}, t) v_{kt} \right) - \log Z(\mathbf{Y}).$$

Features (more detail)

- Weighted distance between state and prediction

$$f_j(\mathbf{x}_{t-1}, \mathbf{x}_t) = -\frac{1}{2}\left(\mathbf{x}_t - \phi_j(\mathbf{x}_{t-1})\right)^T \boldsymbol{\alpha}_j \left(\mathbf{x}_t - \phi_j(\mathbf{x}_{t-1})\right)$$

$$g_k(\mathbf{x}_t, \mathbf{Y}) = -\frac{1}{2}\left(\mathbf{x}_t - \gamma_k(\mathbf{Y}, t)\right)^T \boldsymbol{\beta}_k \left(\mathbf{x}_t - \gamma_k(\mathbf{Y}, t)\right)$$

$$\phi_j(\mathbf{x}_{t-1}) = \mathbf{T}_j \mathbf{x}_{t-1} + \mathbf{d}_j$$

Side Information

- Extra features help decide if switches should be on or off
- Any classifier (logistic / softmax regression)

Inference

- P(X|Y) is hard
- P(X|U,V,Y) and P(U,V|X,Y) are easy
- Infer state sequence using belief propagation
- Switch probabilities:

$$\mathrm{P}(v_{kt} = 1) = \sigma\left(f_j(\mathbf{x}_{t-1}, \mathbf{x}_t) + \mathcal{F}_j(\mathbf{Y}, t)\right)$$

$$\mathrm{P}(u_{jt} = 1) = \frac{\exp(g_k(\mathbf{x}_t, \mathbf{Y}) + \mathcal{G}_k(\mathbf{Y}, t))}{\sum_{k'} \exp(g_{k'}(\mathbf{x}_t, \mathbf{Y}) + \mathcal{G}_{k'}(\mathbf{Y}, t))}$$

Learning

- Supervised training of feature weights/precisions
- Contrastive Divergence to approximate gradient
- Think: state=visible, switches=hidden
- Also refine side-information parameters

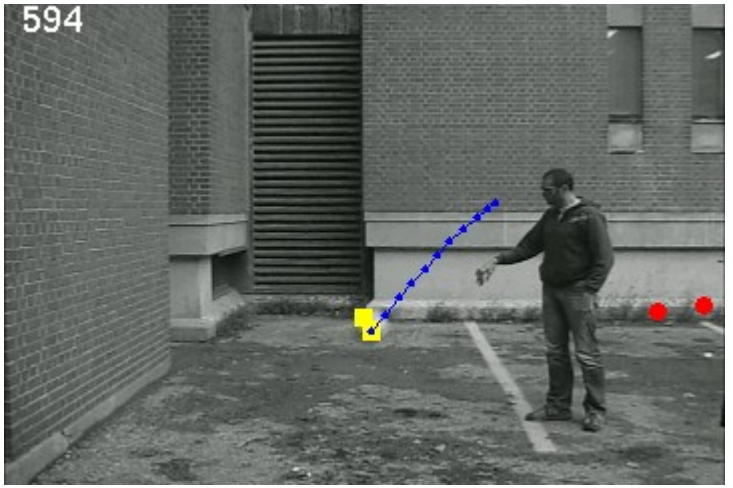
Learning - Gradients

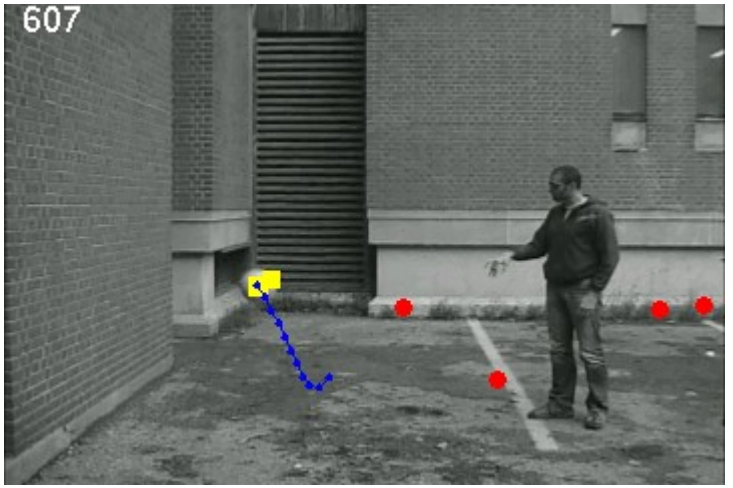
$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}_j} = E_{\mathrm{P}(\mathbf{U},\mathbf{V}|\mathbf{X},\mathbf{Y})}\Big[\sum_t (\mathbf{x}_t - \phi_j(\mathbf{x}_{t-1}))(\mathbf{x}_t - \phi_j(\mathbf{x}_{t-1}))^T\Big]$$
$$-E_{\mathrm{P}(\mathbf{X},\mathbf{U},\mathbf{V}|\mathbf{Y})}\Big[\sum_t (\mathbf{x}_t - \phi_j(\mathbf{x}_{t-1}))(\mathbf{x}_t - \phi_j(\mathbf{x}_{t-1}))^T\Big]$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}_k} = E_{\mathrm{P}(\mathbf{U},\mathbf{V}|\mathbf{X},\mathbf{Y})}\Big[\sum_t (\mathbf{x}_t - \gamma_k(\mathbf{Y},t))(\mathbf{x}_t - \gamma_k(\mathbf{Y},t))^T\Big]$$
$$-E_{\mathrm{P}(\mathbf{X},\mathbf{U},\mathbf{V}|\mathbf{Y})}\Big[\sum_t (\mathbf{x}_t - \gamma_k(\mathbf{Y},t))(\mathbf{x}_t - \gamma_k(\mathbf{Y},t))^T\Big]$$

Tracking in Video

- Combine unreliable dyn/obs features
- 6d state (position, velocity, acceleration)
- Linear dynamics features
- Observation features predict (x,y) position
- Train: first 500 frames





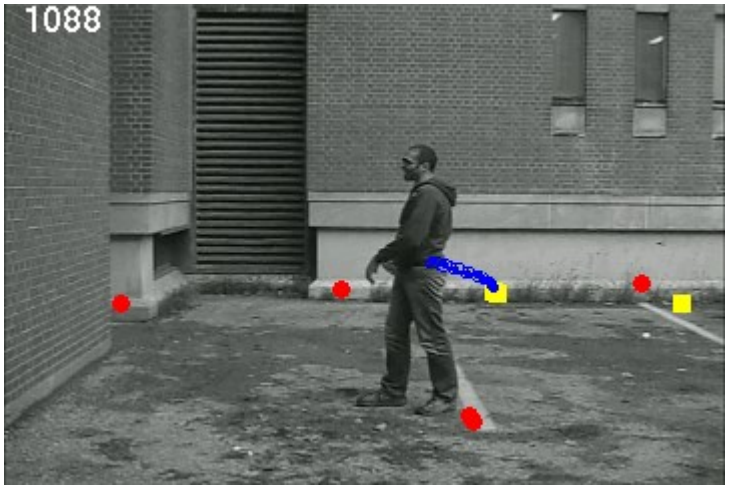
Features Used

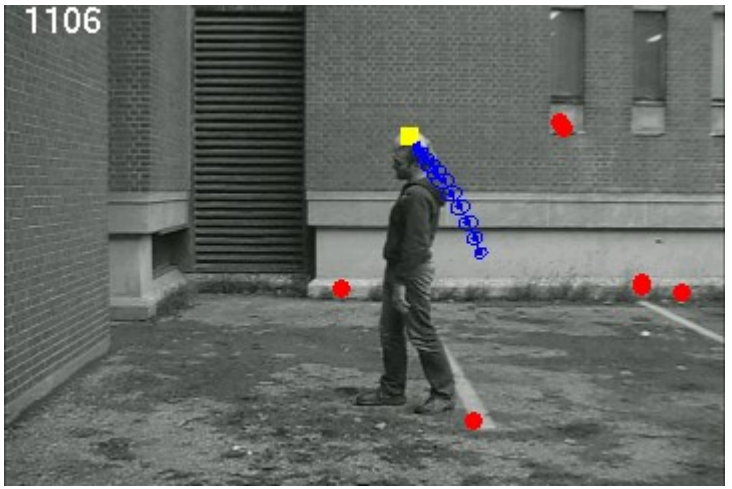
- 6 Templates: K-means, K=5 (0.3524 - 0.6314), “Last Frame” (0.3285)
- PCA, 3 components (0.8068)
- Local background subtraction (0.6932)
- 4 Linear dynamics (fly, hold, bounce:ground, bounce:wall)

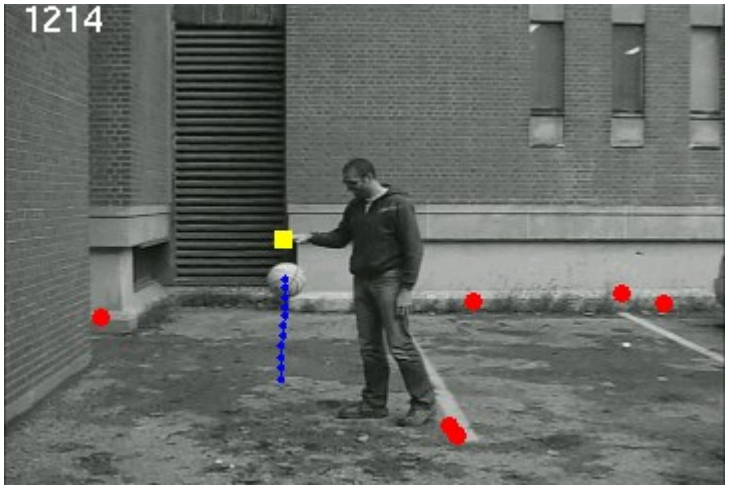
Result

- Demo #1
- loses track 4 times (but recovers):
  - 559:580 ball occluded, features fail
  - 1060:1101 ball occluded, features fail
  - 1169:1184 incorrect v's predicted, our fault
  - 1509:1551 again, incorrect v's
- Incremental PCA tracker: fails at 688

Highlights





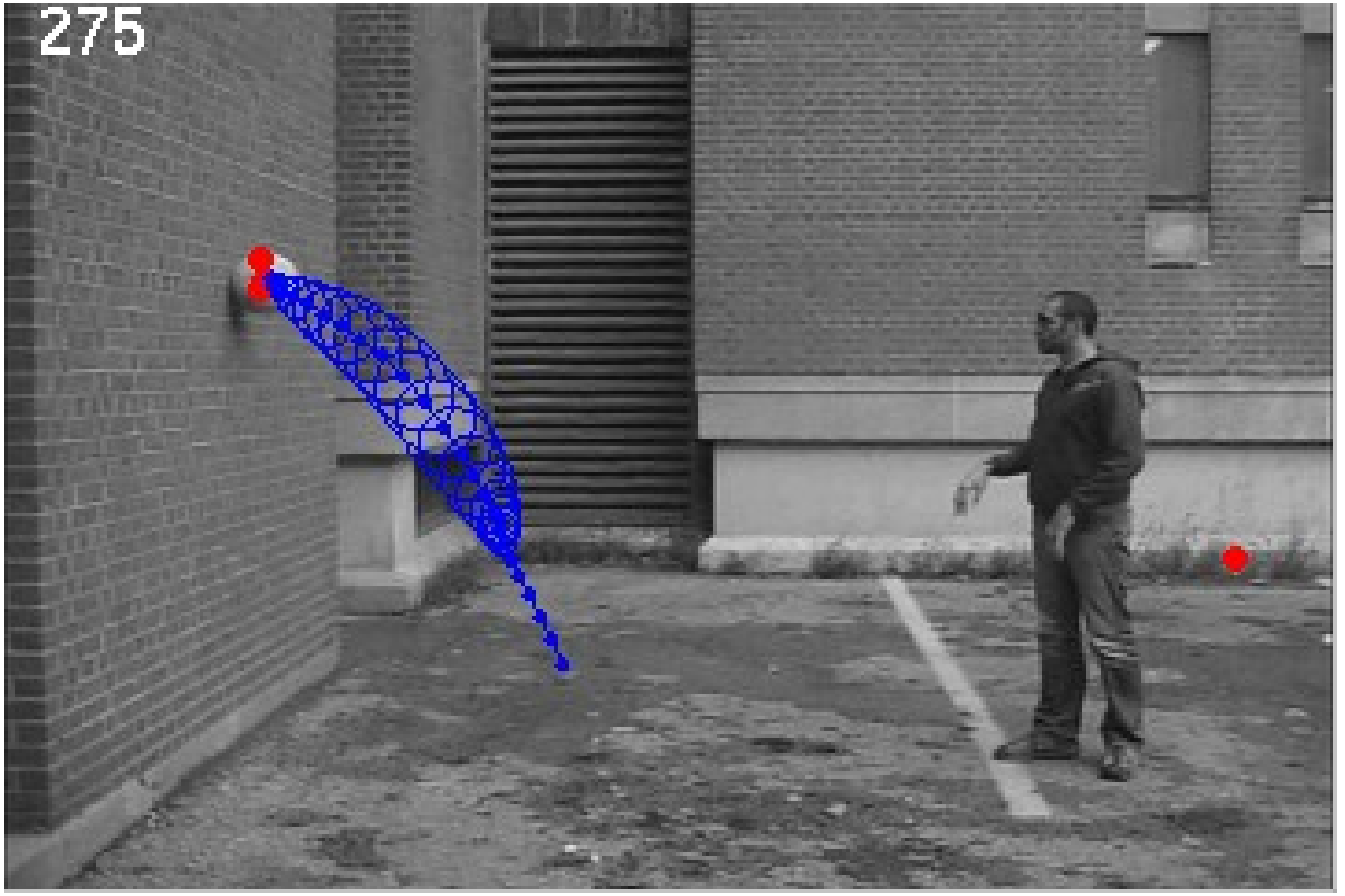


Learned Feature Weighting

```
standard deviation of 8 features (x,y)
    4.0493    5.6269 image template
    2.7972    2.7072 k-means
    2.6480    3.4817
    2.1839    2.2170
    2.6473    2.4273
    2.5850    2.5172
   32.7942    7.7448 background sub.
    1.7820    1.8025 pca
```

Missing Data

- Demo #2 (look at 255:275) 20 frames, 98 pixels



Related Work

- Conditional Random Fields
- Products of Experts (and EFH)
- Switching linear dynamical systems
- Mixed-state (continuous+discrete) particle filters
- Discriminative Trackers (Cristian Sminchisescu)
- Kalman Filter with input gating (CV:AMA)

What's Next?

- Better features (using more observations)
  - SIFT? State-of-the-art trackers?
  - Different modalities (e.g. sound)
- Non-linear dynamics
- Other data (financial time-series?)