Combining Discriminative Features to Infer Complex Trajectories

David Ross, Simon Osindero, Richard Zemel
University of Toronto, Machine Learning

ICML 2006
June 28, 2006
Inferring Complex Trajectories

- **Problem**: time series regression
  - Estimate continuous state variables from a sequence of observations
- **Running example**: Visual tracking of a target object
- **Standard approach**: Generative state-space model (Kalman filter, etc.)
  - strong likelihood, generates observations
  - weak prior, describes trajectory
Combining Discriminative Features

- Discriminative conditional model
- Model \( \text{Pr}(\text{state} | \text{observations}) \) as a (log-linear) combination of dynamics & observation features
- “Pile on” features
  - Learn which are useful
  - Switch features on and off dynamically
Discriminative Features

- Dynamics features: $f_j(x_{t-1}, x_t)$
  - how well do two states match?
  - (non) linear dynamical models

- Observation features: $g_k(x_t, Y, t)$
  - is the target at $x_t$?
  - Any appearance model/object detector
  - Can include information from the entire observation sequence

- Robustify by switching features on and off
  - Hidden switch variables $u_{jt}$, $v_{kt}$
Features & Switch Potentials

- **Weighted distance between state and prediction**

\[
f_j(x_{t-1}, x_t) = -\frac{1}{2} (x_t - \phi_j(x_{t-1}))^T \alpha_j (x_t - \phi_j(x_{t-1}))
\]

\[
g_k(x_t, Y) = -\frac{1}{2} (x_t - \gamma_k(Y, t))^T \beta_k (x_t - \gamma_k(Y, t))
\]

\[
\phi_j(x_{t-1}) = T_j x_{t-1} + d_j
\]

- **Switch Potentials**: extra features help decide if switches should be on or off

- **Any classifier** (logistic / softmax regression)
\[ P(X|Y) \propto \exp \left( \sum_{t,j} f_j(x_{t-1}, x_t)u_{j,t} + \sum_{t,k} g_k(x_t, Y)v_{k,t} \right) \]
Probability Model

\[ P(\mathbf{X}|\mathbf{Y}) \propto \exp \left( \sum_{t,j} f_j(\mathbf{x}_{t-1}, \mathbf{x}_t)u_{j,t} \right. \left. + \sum_{t,k} g_k(\mathbf{x}_t, \mathbf{Y})v_{k,t} \right. \left. + \sum_{t,j} F_j(\mathbf{Y}, t)u_{j,t} \right. \left. + \sum_{t,k} G_k(\mathbf{Y}, t)v_{k,t} \right) \]
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
- Sample switch probabilities:

\[
P(v_{kt} = 1) = \sigma \left( g_k(x_t, Y) + g_k(Y, t) \right)
\]

\[
P(u_{jt} = 1) = \frac{\exp(f_j(x_{t-1}, x_t) + F_j(Y, t))}{\sum_{j'} \exp(f_{j'}(x_{t-1}, x_t) + F_{j'}(Y, t))}
\]
Learning

- Separable into several easy sub-problems
- Learn each observation and dynamics feature separately
- Learning switch potential parameters for each feature (classification problem)
- Jointly learn feature precisions (weights), using Contrastive Divergence
Application: Tracking in Video

- Combine unreliable dyn/obs features
- 6d state (position, velocity, acceleration)
- Linear dynamics features
- Observation features predict (x,y) position
- Train: video labeled with ground truth
- expected ball position (last 10 frames)
- observation features
  - switched on
  - switched off
Features Used

- Template from one frame
- Template from K-Means
- PCA, 3 components
- Local background subtraction
- 4 Linear dynamics (fly, hold, bounce:ground, bounce:wall)
Within Sequence Generalization

- expected ball position (last 10 frames)
- observation features
  - switched on
  - switched off
How do other trackers do?

- error rate = fraction of frames where predicted state is > 5 pixels away from ground truth
- Kalman Filter \( \text{er}=0.73 \)
- Adaptive Eigentracker + particle filter fails at frame 688, can't recover \( \text{er}=0.61 \)
- THIS MODEL \( \text{er}=0.11 \)
How is it working?

standard deviation

“on the ball”
Test #2: Rolling & Bouncing
Test #3: Total Occlusion
Related Work

- Conditional Random Fields
- Switching linear dynamical systems (and particle filters)
- Discriminative Trackers (Sminchisescu)
- Unsupervised Product Models (Gehler)
Future Directions

• Expanding the range of features
  – Optic flow, SIFT, State-of-the-art trackers
  – Different modalities (e.g. sound)

• Unsupervised learning of dynamics

• Other applications (financial time-series, robot localization, any suggestions?)