Conditional Random Fields for Activity Recognition

Douglas L. Vail
Thesis Defense
April 3rd, 2008

Committee:
John Lafferty, Co-Chair
Manuela Veloso, Co-Chair
Carlos Guestrin
Dieter Fox, University of Washington
Thesis Motivation

• We want to predict activity labels for robots

[Image of a soccer field with robots scattered]
Thesis Motivation

• We want to predict activity labels for robots

• Activity labels provide a succinct description of robot actions, behaviors, and goals

• We recover activity labels by observing robots over time

• Mapping from observations to activity labels is challenging and requires
  • Temporal models
  • Rich features of the observations
The Activity Recognition Problem

- Sensor Data
  - High dimensional
  - Noisy
  - Continuous

- Labels
  - Temporal sequence
  - Discrete activities

Activity recognition is the process of mapping from observations to a label sequence
Thesis Question

How do we map from the noisy, continuous observations to a sequence of discrete activity labels?
Thesis Approach

- We track observations and labels across time with a temporal model.
- We extract relevant information by computing complex features of the observation sequence:
  - We specify a small set of relational feature templates that capture our domain knowledge.
  - We automatically expand the templates into many candidate features.
- We choose the most relevant candidate features with feature selection.
Thesis Contributions

• Introduce the use of CRFs for activity recognition in robot domains
• Empirically compare CRFs to HMMs for activity recognition
• Adapt two feature selection algorithms for use with CRFs
  • $L_1$ regularization
  • Grafting (Perkins and Theiler 2003)
• Compare against a CRF specific feature selection algorithm
  • Approximate Gain Heuristic (McCallum 2003)
• Introduce relational features for multi-robot activity recognition
• Adapt an M-estimator, originally for non-parametric log-density estimation, for fast, approximate parameter estimation in CRFs (Jeon and Lin 2006)
• Introduce a novel feature selection algorithm based on the M-estimator
Closely Related Work (Liao et al. 2005, 2007)

• Hierarchical models for human activity recognition
  • Map from GPS locations to activities, e.g., walking, sleeping, visiting a friend
  • Improved accuracy with hierarchical model structures
    • Two pass approach using CRFs
    • Relational Markov networks
  • Activity recognition with a rich set of model structures
  • We explore activity recognition with a rich set of features

• Feature selection in CRFs with virtual evidence boosting
  • Indoor human activity recognition with wearable sensors
  • An extension of McCallum’s approximate gain heuristic and LogitBoost
  • Decision stumps as weak learners in CRFs with continuous observations
  • We compute functions of the observations using a relational feature specification language
Outline

• Introduction
  • We map from observations to activity labels
    • Temporal models
    • Complex features generated from relational templates
    • Feature Selection
• Background
  • Sequential classification, CRFs, and features
    • CRFs for activity recognition
    • Feature selection in CRFs
    • Multiple robots and relational features
    • An M-estimator for fast training and feature selection
    • Conclusion
Activity Recognition as Sequential Classification

Activity recognition is the process of mapping from observations to a label sequence.

Observation vector:
\[ x_t = [r_x, r_y, r_\theta, \ldots] \]
e.g., robot positions

Observation sequence:
\[ X = \{x_1, x_2, \ldots, x_T\} \]

Labels
\[ y_t \in \text{Activities} \]
e.g., Standing, Sitting, Walking, ...

Label sequence:
\[ Y = \{y_1, y_2, \ldots, y_T\} \]
Conditional Random Fields

• Undirected graphical models that compactly represent the probability of the labels given the observations
  • Defined in terms of features and weights
  • Do not model observations
  • Assume a structure over the labels
  • Assume no structure over observations
  • Alternative to joint models such as HMMs

\[ \ell(Y|X; w) = \frac{1}{Z_X(w)} \exp \left( \mathbf{w} \cdot \bar{f}(t, y_{t-1}, y_t, X) \right) \]

weight vector

feature vector
Features in Conditional Random Fields

We use the indicator function to detect labels or combinations of labels:

\[
(a \neq b) \equiv \begin{cases} 
0 & \text{if } a \neq b \\
1 & \text{if } a = b 
\end{cases}
\]

First-order Markov transitions:

\[
f_i(t, y_{t-1}, y_t, X) = (y_{t-1} \neq \text{label}_1) \cdot (y_t \neq \text{label}_2)
\]

General state-linked and transition-linked features:

\[
f_i(t, y_{t-1}, y_t, X) = (y_t \neq \text{label}) \cdot g(t, X)
\]

\[
f_i(t, y_{t-1}, y_t, X) = (y_{t-1} \neq \text{label}) \cdot (y_t \neq \text{label}) \cdot g(t, X)
\]
Outline

• Introduction
• Background
  • We use linear-chain CRFs as a temporal model for activity recognition
  • Features are arbitrary functions of label pairs and the entire observation sequence
• CRFs for robot activity recognition
  • Empirical comparison to HMMs
• Feature selection in CRFs
• Multiple robots and relational features
• An M-estimator for fast training and feature selection
• Conclusion
Empirical Model Comparison

• Empirically compare CRFs with a traditional model for activity recognition, i.e. the hidden Markov model
  • Created a CRF with HMM-equivalent features
  • Continuous features treated as one dimensional Gaussians
• We explore
  • Ability to robustly incorporate complex features
  • Sensitivity to non-independent features
  • Accuracy of discriminative versus generative models
Robot Tag Benchmark Domain

• Three robots play a tag-like game
• One robot is “It”
  • Chases its closest neighbor
  • “Tags” its target by approaching within 4 cm
• The other players
  • Fixed behavior: navigate to random points on the field (ignoring “It”)
  • Pause in place when tagged before becoming “It”

Goal of classification: Identify “It” at each time step

Observations:

\[
x_t = [a_x, a_y, b_x, b_y, c_x, c_y]
\]

Labels:

\[
y_t \in \{a, b, c\}
\]
Robot Tag Movie

CMDragons Robot Soccer Simulator: Michael Bowling, Brett Browning, James Bruce, Stefan Zickler, and Manuela Veloso
Robot Tag Features

Robot positions

\[ f_i(t, y_t, y_{t-1}, X) = (y_t \neq \text{robot id})a_x \]

Robot velocities

\[ f_i(t, y_t, y_{t-1}, X) = (y_t \neq \text{robot id}) \text{velocity}(a) \]

Distance thresholds

\[ f_i(t, y_t, y_{t-1}, X) = (y_{t-1} \neq \text{robot id})(y_t \neq \text{robot id}) \text{dist}(a, b) \leq k \]

“Chasing”

\[ f_i(t, y_t, y_{t-1}, X) = (y_t \neq \text{robot id}) \left( \vec{off}_{a,b} \cdot \vec{vel}_a \right) \]
## Robot Tag Error Rates

<table>
<thead>
<tr>
<th>Features</th>
<th>HMM</th>
<th>CRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot Positions</td>
<td>62.9%</td>
<td>62.2%</td>
</tr>
<tr>
<td>Velocities</td>
<td>44.3%</td>
<td>29.6%</td>
</tr>
<tr>
<td>Chasing</td>
<td>34.5%</td>
<td>29.6%</td>
</tr>
<tr>
<td>Distance thresholds</td>
<td>56.5%</td>
<td>42.0%</td>
</tr>
<tr>
<td>Combining features</td>
<td>36.6%</td>
<td>1.5%</td>
</tr>
<tr>
<td>Adding irrelevant features</td>
<td>47.3%</td>
<td>6.2%</td>
</tr>
</tbody>
</table>

- Raw position observations yield low accuracy
- Adding a single type of relevant feature boosts accuracy
- Combining features:
  - Boosts CRF accuracy
  - Reduces HMM accuracy
- The discriminatively trained CRF outperforms the equivalent HMM
- Irrelevant features lower accuracy
Outline

• Introduction
• Background

• CRFs for activity recognition
  • CRFs offer compelling advantages over traditional models such as HMMs
    • Discriminatively trained
    • Robustly incorporate complex, non-independent features
  • We inject domain knowledge into our models via features

• Feature selection in CRFs
  • Introduce feature selection algorithms
  • Compare algorithm performance

• Multiple robots and relational features
• An M-estimator for fast training and feature selection
• Conclusion
Feature Selection in CRFs

• We define CRFs in terms of features and a corresponding weight for each feature
• Create a CRF that contains all of the candidate features
• Set all the weights to zero
  • Features with zero weight are inactive
  • Feature selection “turns on” the most relevant features
• Feature selection produces a sequence of weight vectors
  • The first vector contains all zeros
  • Successive vectors contain more non-zero weights (active features)
  • Each weight vector is a candidate model
  • We choose among the candidates with held out data
Mean Field Heuristic (McCallum 2003)

- Greedy forward selection with an approximate gain heuristic
  - Adds one new feature per iteration
  - Re-estimates weights via MLE after activating each feature
- Estimates the increase in the likelihood for adding feature $g$ with weight $\mu$
- Approximate gain with a pseudo-likelihood style assumption
  - Significantly faster than exact gain computation
  - Maximizes over $\mu$ with a line search to score individual features
  - Per feature line searches are computationally expensive

$$
\text{gain}(g, \mu) = \ell(Y|X; w, \mu) - \ell(Y|X; w)
$$
Grafting (Perkins and Theiler 2003)

• Greedy forward selection that scores candidate features by their gradient contribution
• We introduce grafting for feature selection in CRFs because:
  • The gradient is readily computed in CRFs
  • Grafting scales to very large numbers of candidate features
• Scoring features by their gradient contribution (versus MF heuristic)
  • Eliminates the need for one line search per feature scored
  • Less robust to local optima when choosing features

\[ A^{(n)} = A^{(n-1)} \cup \{i\}, \text{ where } i = \arg\max_{i \notin A} \left| \frac{\partial \ell(Y|X; w^{(n-1)})}{\partial w_i} \right| \]
• Each line follows one weight as features are added
• Grafting is a greedy feature selection strategy
  • Training sometimes assigns near-zero weights to selected features
  • Feature weights sometimes decrease substantially due to later additions
L₁ Regularization for Feature Selection in CRFs

\[ w^* = \arg \max_w \ell(Y|X; w) - \lambda \sum_i |w_i| \]  

- Training with an L₁ penalty produces sparse models
  - \( \lambda \) controls how strongly non-zero weights are penalized
- We introduce L₁ regularization for feature selection in CRFs because:
  - Penalized objective function is convex
  - L₁ regularization scales to very large numbers of candidate features
- We select features by training with a decreasing series of \( \lambda \) values
• $\lambda$ implicitly specifies a weight budget
  • Large $\lambda \rightarrow$ small budget
  • We spread a fixed weight budget over the features to maximize conditional likelihood
• Candidate weight vectors are vertical slices of the regularization path
Algorithm Comparison: Data Efficiency

• Robot data is a scarce resource
  • Running experiments is extremely time consuming
• We compare the three feature selection algorithms when little training data is available
  • Sample synthetic data from an HMM
    • Vary the length of the training sequence (T)
  • Perform feature selection
    • Outputs a sequence of candidate models
  • Evaluate the candidate models in terms of error rate versus number of active features
• Greedy algorithms are sparse
• Grafting severely over-fits with little data
• Grafting converges to the MF heuristic with sufficient data
• Feature selection using $L_1$ regularization and the MF heuristic produces similar error rates
Algorithm Comparison: Retraining with $L_2$ Smoothing

- Over-fitting to the training data reduces classification accuracy
- $L_1$ regularization simultaneously smoothes and selects features
- We do not smooth during greedy forward selection
  - Difficult to choose the regularization parameter for, e.g., an $L_2$ penalty
- We test if post feature selection smoothing improves accuracy
  - Choose the set of active features via feature selection
  - Select new weights by retraining under an $L_2$ penalty
- Robot tag domain with 1,200 candidate features
Generalization: Error Rates with L₂ Smoothing

<table>
<thead>
<tr>
<th>Simulated Data</th>
<th>No Smoothing</th>
<th>L₂ Smoothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>8.8%</td>
<td>3.4%</td>
</tr>
<tr>
<td>L₁</td>
<td>1.8%</td>
<td>3.2%</td>
</tr>
<tr>
<td>Grafting</td>
<td>6.2%</td>
<td>2.2%</td>
</tr>
<tr>
<td>MF</td>
<td>1.6%</td>
<td>1.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Real Robot Data</th>
<th>No Smoothing</th>
<th>L₂ Smoothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>16.1%</td>
<td>9.7%</td>
</tr>
<tr>
<td>L₁</td>
<td>5.9%</td>
<td>7.3%</td>
</tr>
<tr>
<td>Grafting</td>
<td>7.0%</td>
<td>7.0%</td>
</tr>
<tr>
<td>MF</td>
<td>7.6%</td>
<td>7.7%</td>
</tr>
</tbody>
</table>

- Feature selection reduces over-fitting
- Retraining with L₂ smoothing can improve accuracy when using grafting for feature selection
- Retraining with L₂ smoothing can harm accuracy with L₁ regularization for feature selection
  - Using the same holdout set to choose features and λ increases over-fitting
Algorithm Comparison: Generalization

- Robot data is expensive to acquire
  - Training models with simulated data has a high potential payoff
- We tested generalization across related data sets
  - We select features and weights via feature selection with data from one source (simulation or real robots)
  - Using the selected model to classify data from the second source
- Robot tag domain with 1,200 candidate features
Generalization: Error Rates Across Data Sets

<table>
<thead>
<tr>
<th>Simulation Data</th>
<th>Simulation</th>
<th>Robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>8.8%</td>
<td>26.7%</td>
</tr>
<tr>
<td>L₁</td>
<td>1.8%</td>
<td>8.2%</td>
</tr>
<tr>
<td>Grafting</td>
<td>6.2%</td>
<td>12.5%</td>
</tr>
<tr>
<td>MF</td>
<td>1.6%</td>
<td>13.5%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Robot Data</th>
<th>Robot</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>16.1%</td>
<td>23.5%</td>
</tr>
<tr>
<td>L₁</td>
<td>5.9%</td>
<td>7.1%</td>
</tr>
<tr>
<td>Grafting</td>
<td>7.0%</td>
<td>3.3%</td>
</tr>
<tr>
<td>MF</td>
<td>7.6%</td>
<td>3.0%</td>
</tr>
</tbody>
</table>

- Feature selection improves generalization
- Better generalization from real data to simulated data
- L₁ performs better moving from simulated to real data
- Greedy forward selection performs better moving from real to simulated data
  - Trade off between over-fitting and over-smoothing
  - The gap between grafting and L₁ regularization is larger on simulated data
Error Rate versus Model Size with Robot Data

- Feature selection involves complex trade-offs between
  - Classification accuracy (error rate)
  - Model sparsity (number of features)
- We compare error rate versus number of features in candidate models produced by the three algorithms
  - Robot tag domain with 1,200 candidate features
  - Synthetic data suggests
    - Greedy forward selection produces smaller models
    - $L_1$ regularization produces lower error rates
• In general, the results match our predictions from the synthetic data experiments

• Mean field heuristic significantly less sparse than grafting
  • Pseudo-likelihood style approximations perform poorly with sparse state transitions
Robot Tag Movie

CMDragons’07 Robot Soccer System: James Bruce, Mike Licitra, Stefan Zickler, and Manuela Veloso
Outline

- Introduction
- Background
- CRFs for activity recognition
- Feature selection in CRFs
  - Feature selection
    - Improves overall accuracy
    - Significantly reduces model size
  - Greedy forward selection produces extremely sparse models
  - $L_1$ regularization selects more features and produces lower error rates
- Multiple robots and relational features
  - Relational features for robot soccer
- An M-estimator for fast training and feature selection
- Conclusion
Robot Soccer

- Activity recognition in robot soccer is challenging
  - Interactions between many robots
  - Complex behaviors
  - Rapidly changing environment
- Activity recognition has a high potential payoff
  - Activity labels for opponents
- We introduce relational features to succinctly specify domain knowledge for multi-robot domains
- We demonstrate that activity recognition is feasible in the RoboCup Small Size League
  - Note: We test generalization across games rather than opponents

CMDragons’07: James Bruce, Mike Licitra, Stefan Zickler, and Manuela Veloso
The Classification Task

- Label sequence:
  - Role of a single robot
  
  \( y_t \in \{ \text{Defender, Marking, \ldots, Position for Pass} \} \)

- Observations:
  - Global positions/headings of 10 robots
  - Global position of the ball

  \( x_t = [a_x, a_y, a_\theta, \ldots, j_x, j_y, j_\theta, \text{ball}_x, \text{ball}_y] \)

- Subsampled data at 2 hz to speed training
Relational Features

- Compute quantities defined in terms of relationships between objects
  - E.g., “The distance between the ball and the robot closest to the ball”
- Dynamically select objects based on the observations
- Specified in a relational language
  - Succinct template specifications
    - ~30 for robot soccer
  - Automatic expansion into many features
    - ~92,000 for robot soccer
Predicting Robot 5’s Activity

- Properties that hint that robot 5 is “marking” robot 2:
  - Robot 2 is close to robot 5
  - Robot 2 is not the closest opponent to the ball
  - Robot 2 is the closest opponent to robot 5
A Non-Relational Feature

\[ g_2(t, X) = \text{distance}(r_5, r_2) \]

\[
\begin{align*}
((\text{distance}(r_0, \text{ball}) \leq \text{distance}(r_2, \text{ball})) \lor \\
(\text{distance}(r_1, \text{ball}) \leq \text{distance}(r_2, \text{ball})) \lor \\
(\text{distance}(r_3, \text{ball}) \leq \text{distance}(r_2, \text{ball})) \lor \\
(\text{distance}(r_4, \text{ball}) \leq \text{distance}(r_2, \text{ball})))
\end{align*}
\]

\[
\begin{align*}
((\text{distance}(r_2, r_5) \leq \text{distance}(r_0, r_5)) \land \\
(\text{distance}(r_2, r_5) \leq \text{distance}(r_1, r_5)) \land \\
(\text{distance}(r_2, r_5) \leq \text{distance}(r_3, r_5)) \land \\
(\text{distance}(r_2, r_5) \leq \text{distance}(r_4, r_5)))
\end{align*}
\]

- Robot 2 is close to robot 5
- Robot 2 is not the closest opponent to the ball
- Robot 2 is the closest opponent to robot 5
A Relational Feature

\[ g_m(t, X) = distance(\text{set-subject}, \ select-closest(\text{set-subject}, \ xor(\text{set-opponents}, \ select-closest(\text{set-ball, set-opponents}))) \]
Activity Recognition in the RoboCup Small Size League

- We demonstrate the effectiveness of relational features for accurate activity recognition in a complex robot domain
  - Interactions between 10 robots
  - We define 30 relational feature templates
  - We automatically generate 92,310 features from the templates
- We independently predict labels for each of 5 robots
  - 9 possible activity labels, e.g., position for pass, attacker, mark
  - 32 continuous observations
- Data from the CMDragons’07 team at RoboCup 2007
  - We test by predicting the roles of the CMDragons team in the final
  - We use three earlier CMDragons games for training and model selection
  - We test generalization across games rather than opponents
## Experiments

- The MF approximate gain heuristic required 45 days for training (not shown)
- Feature selection dramatically reduces model size
- Grafting produces less accurate but significantly smaller models
- Smoothing offers little benefit on top of feature selection
- Feature selection is faster than $L_2$ smoothing in the full model

<table>
<thead>
<tr>
<th>Feature Selection</th>
<th>Number of Features</th>
<th>No Smoothing</th>
<th>L2 Smoothing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Error Rate (%)</td>
<td>Time (hours)</td>
</tr>
<tr>
<td>None</td>
<td>92,310</td>
<td>15.7</td>
<td>3.1</td>
</tr>
<tr>
<td>$L_1$</td>
<td>1,823</td>
<td>10.3</td>
<td>6.9</td>
</tr>
<tr>
<td>Grafting</td>
<td>220</td>
<td>12.3</td>
<td>1.9</td>
</tr>
</tbody>
</table>
Recognizing Activities in Robot Soccer
Outline

• Introduction
• Background
• CRFs for activity recognition
• Feature selection in CRFs
• Multiple robots and relational features
  • Relational features provide a succinct means for injecting domain knowledge into our models
  • Relational features enable accurate activity recognition in complex robot domains
  • L1 regularization and grafting scale to large numbers of features
• An M-estimator for fast training and feature selection
• Conclusion
A Computationally Efficient M-estimator

• Definition: Maximum likelihood type estimator (Hubert 1981)
  • Generalization of MLE for robust statistics
• Jeon and Lin (2006) proposed an M-estimator for high dimensional log-density estimation using splines
• We adapt their M-estimator for computationally efficient parameter estimation (training) in CRFs
• Key property: Modeling $p(X,Y)$ as corrections to a base model $q_0$ eliminates the need for normalization during training

$$\hat{w} = \arg \min_w \frac{1}{n} \sum_{i=1}^{n} \exp(-w^T F(X_i, Y_i)) + w^T \mathbb{E}_{q_0}[F(X, Y)]$$
Tag Experiments with Hand Selected Features

- Activity recognition with the robot tag simulation data set
  - 210 hand selected, binary features
- M-estimator with logistic regression $q_0$ performs almost as well as the CRF
- M-estimation is significantly faster than MLE
Feature Selection with the M-estimator

- Greedy forward feature selection
  - Use the current CRF as $q_0$
  - M-estimate parameters for candidate features
  - Add the feature with the largest absolute weight
- RoboCup Small Size data
  - 92,310 features
### Feature Selection Experiments

<table>
<thead>
<tr>
<th>Feature Selection</th>
<th>Number of Features</th>
<th>Error Rate (%)</th>
<th>Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None / L(_2)</td>
<td>92,310</td>
<td>10.5</td>
<td>11.2</td>
</tr>
<tr>
<td>L(_1)</td>
<td>1,823</td>
<td>10.3</td>
<td>6.9</td>
</tr>
<tr>
<td>Grafting / L(_2)</td>
<td>220</td>
<td>12.0</td>
<td>2.5</td>
</tr>
<tr>
<td>Mean Field / L(_2)</td>
<td>483</td>
<td>21.8</td>
<td>1092</td>
</tr>
<tr>
<td>M-estimator / L(_2)</td>
<td>1,694</td>
<td>14.0</td>
<td>61.1</td>
</tr>
</tbody>
</table>

- M-estimation for feature selection
  - Requires less computation than the mean field heuristic
  - Produces a more accurate model
  - Outperformed by both grafting and L\(_1\) regularization
Outline

• Introduction
• Background
• CRFs for activity recognition
• Feature selection in CRFs
• Multiple robots and relational features
• An M-estimator for fast training and feature selection
  • M-estimation dramatically reduces training time with carefully chosen binary features
  • Feature selection with the M-estimator is a compromise between the speed of grafting and more robust selection of the MF heuristic
• Conclusion
Contributions

• Introduced the use of CRFs for activity recognition in robot domains
• Demonstrated that CRFs offer compelling advantages over joint models such as HMMs for activity recognition
• Explored feature selection in CRFs
  • Examined the trade off between sparsity and accuracy
  • Demonstrated that grafting and L1 regularization scale to large sets of candidate features
• Introduced relational features for multi-robot activity recognition
• Adapted an M-estimator fast, approximate parameter estimation in CRFs
• Introduced a new feature selection algorithm based on the M-estimator
Thanks!

• Manuela Veloso and John Lafferty
  • Sage advice and wisdom
• Carlos Guestrin and Dieter Fox
  • Helpful feedback and thoughtful questions
  • (We’ll see lots of this starting in about 30 seconds)
• James Bruce, Stefan Zickler, Michael Licitra
  • CMDragons tutorials and RoboCup data
• Brenna Argall, Sonia Chernova, Liz Crawford, Colin McMillen, Stephanie Rosenthal, Kristen Stubbs
  • Proof reading and practice talks