Machine Learning in Mobile and Sensor Networks

Qiang Yang, 杨强
http://www.cs.ust.hk/~qyang

Hong Kong University of Science and Technology

Joint Work with Dr. Rong Pan and students: Jie Yin, Xiaoyong Chai, Jeff Pan and Dou Shen Kangheng Wu
Context-Aware Computing: A Solution

- A central theme in context-aware computing is to build *predictive models of human behavior*
  - Where is the user? (location estimation)
  - What is her ultimate goal? (activity recognition)

(courtesy CMU)
Problem Domain: Wireless Environment

- A user with a mobile device walks in an indoor wireless environment (802.11b WLAN)

Where is the user? What will the user do?

Time t: (-47dB, -36dB, -62dB)
What are the learning problems?

- Learning to classify low level sequences
  - Location, actions and goals
  - Semi-supervised Learning
- Learning to segment sequences into discrete activities
- New Machine Learning Problem: Data Migration
- Semi-supervised Learning:
  - To reduce calibration efforts
- Training data migration problem
- Distributed Learning problem
Machine Learning in Pervasive Computing: A Video Demonstration

Probabilistic Goal Recognition: Architecture

ML Prob: semi-supervised classification

ML Prob: Bayesian location estimation

ML Prob: Model Migration

ML Problem: Distributed Learning

ML Prob: segmentation

Goals

Actions

Intermediate states

Behavior Model

Generalized Sensor Model

Location-based SM

Motion Pattern-based SM

observations
ML Problem: Bayesian Learning on Location

- Two phases: **offline** Training and **online** Localization

**Offline phase** – collect samples to build a mapping function \( l = f(s) \) from signal space \( S \) to location space \( L \) where \( s \in S \) and \( l \in L \)

<table>
<thead>
<tr>
<th>Loc.</th>
<th>Time</th>
<th>(AP1,AP2,AP3) dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,0)</td>
<td>1s</td>
<td>(-60,-50,-40)</td>
</tr>
<tr>
<td>(2,0)</td>
<td>2s</td>
<td>(-62,-48,-35)</td>
</tr>
<tr>
<td>.....</td>
<td>.....</td>
<td>(..., ..., ...)</td>
</tr>
<tr>
<td>(9,5)</td>
<td>9s</td>
<td>(-50,-35,-42)</td>
</tr>
</tbody>
</table>

**Online phase** – given a new signal \( s \), estimate the most likely location \( l \) from \( l = f(s) \)

- \( s^* = (-60,-49,-36) \) dB, compute \( f(s) = l \) as estimated location
Related Work in Building the Sensor Model

- Microsoft Research’s RADAR [Bahl and Padmanabhn, 2000]
  - K-Nearest-Neighbor Method
  - Offline - for each location, compute the mean signal
  - Online – estimate location with KNN and triangulation
  - Maximum Likelihood Estimation
  - Offline - for each location, train the Radio Map of each AP at each location
  - Online - apply Bayes’ rule and (user dynamics) for estimation

**Major issues**

- Radio map changes with time
  - How to adapt for other times?
  - Yin et al. [IEEE Percom 2005]
- Reduce human calibration by user unlabelled traces
  - Semi-supervised Learning
  - Chai et al. [IEEE Percom 2005]
ML Problem: Data and Model Migration

- Key idea: collect radio map once, and then adapt the radio map using reference points and a regression analysis.

- During initial time period $t_0$:
  - At each location $i$, we learn a predictive function $f_{ij}$ for the $j$th AP, based on the reference points:
    \[ S_{est}(i, j, t_0) = f_{ij}(r(t_0)) \]

- During the online phase (time period $t$):
  \[ S_{est}(i, j, t) = f_{ij}(r(t)) \]

\[ D_i(t) = \sqrt{\sum_{j=1}^{p} (s_j(t) - s_{sj}(t))^2} \]
Critical Issue: learn predictive function $f_{ij}$

- mapping between the signal-strength values received by the mobile client and the reference points.

- Two algorithms via regression analysis
  - A multiple-regression based algorithm (Linear Model LM)
    \[ s = a_0 + a_1 \cdot r_1 + a_2 \cdot r_2 + \ldots \varepsilon \]
  - A model-tree based algorithm (see result at 1.5m)
ML Problem: Semi-Supervised Learning

Total amount of calibration effort: $N_s \times N_l$
Semi-supervised Learning Framework: Using Unlabeled Traces to Improve the Radio Map

- What is a user trace
  - A sequence of signal strength measurements recorded when a user holding a wireless device navigates in the environment

<table>
<thead>
<tr>
<th>Trace #</th>
<th>Observation Sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(AP₁: -57) (AP₁: -56) (AP₁: -55) (AP₁: -52)</td>
</tr>
<tr>
<td></td>
<td>(AP₂: -33) (AP₂: -30) (AP₂: -36) (AP₂: -62)</td>
</tr>
<tr>
<td></td>
<td>(AP₂: -57) (AP₂: -41) (AP₂: -45) (AP₂: -43)</td>
</tr>
</tbody>
</table>
Modeling User Traces Using Hidden Markov Model (HMM)

- An HMM is a quintuple \( <L, O, \lambda, A, \pi> \)
  - \( L \): location-state space – \{\( l_1, l_2, \ldots, l_n \)\}
  - \( O \): observation space – \{\( o_1, o_2, \ldots, o_m \)\}
  - \( \lambda \): radio map – \{Pr(\( o_j \mid l_i \))\}
  - \( A \): location-state transition – \{Pr(\( l_j \mid l_i \))\}
  - \( \pi \): initial state distribution – \{Pr(\( l_i \))\}

- HMM model parameter \( \theta = (\lambda, A, \pi) \)
Experimental Setting

- The environment is modeled as a space of 99 locations, each representing a 1.5-meter grid cell.
- Sensor readings contain signal strength measurements from base stations.
- Sensor model construction: 100 signal samples at each location.
Reducing the calibration effort: result

Fig. 16. Improvement achieved through using an increasing number of traces ($N_s = 20, N_t = 99$)

Fig. 17. Effect of using a varying number of traces to reduce the sampling time
ML Prob: Multi-Dimensional Time-Series Classification

Objective:
- Infer what he is doing
- Recognize his ultimate goal

Actions are not directly observable

More than one goal is achieved

Sensor-based

Multiple-goal

How to recognize a user’s goals?

- **Problem**: how to ensure that goal recognition framework is robust?

- Previous Work:
  - HMM and DBN based: restricted to high-level inferences [Albrecht et al. 98] [Han & Veloso 00]
  - Sensor-based DBN: monolithic architecture but inflexible [Nguyen et al.03] [Bui 03] [Liao et al.04]

- Our Method:
  - A two-level recognition architecture
**An Example**

G1: Go-to-Print-in-Room1  G2: Go-to-Seminar-in-Room2

- **Signal vector:**
  - <58, 60, 45>
  - <56, 59, 48>

- **Probability calculations:**
  - \[ P(G1) = P(A1|G1)P(D1|A1) = 0.5 \]
  - \[ P(G2) = P(A1|G2)P(D1|A1) = 0.5 \]
An Example

G1: Go-to-Print-in-Room1  G2: Go-to-Seminar-in-Room2

\[
P(A_2|A_1, L_1) = 0.5
\]
\[
P(A_3|A_1, L_1) = 0.6
\]

\[
P(A_1) = P(A_1|G_1)P(A_3|A_1,G_1) P(D_1|A_1)P(D_3|A_3) = 0.5
\]
\[
P(A_2) = P(A_1|G_2)P(A_3|A_1,G_2) P(D_1|A_1)P(D_3|A_3) = 0.5
\]
An Example

G1: Go-to-Print-in-Room1  G2: Go-to-Seminar-in-Room2

The professor is pursuing G2.
Environment and Data Set

- 99 locations (a 1.5-meter grid cell)
- 8 out of 25 base stations
- Data for sensor model:
  - 100 samples were collected at each location
- Evaluation Data: about 600 traces (19 goals)
  - 3-fold cross-validation

99 locations
10 actions
19 goals
Evaluation Criteria

- **Efficiency**
  - The average processing time for each observation in the on-line recognition.

- **Accuracy**
  - The number of correct recognition divided by the total number of recognition.

- **Convergence rate**
  - The average number of observations, after which the recognition converges to the correct answer, over the average number of observations for those traces which converge.

<table>
<thead>
<tr>
<th>Sampling Interval</th>
<th>1s</th>
<th>1.5s</th>
<th>2s</th>
<th>2.5s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole DBN</td>
<td>89.5%</td>
<td>87.1%</td>
<td>84.2%</td>
<td>75.4%</td>
<td>71.9%</td>
</tr>
<tr>
<td>DBN + Bigram</td>
<td>90.5%</td>
<td>83.2%</td>
<td>82.1%</td>
<td>74.7%</td>
<td>72.6%</td>
</tr>
</tbody>
</table>

*Efficiency*:

- The average processing time for each observation in the on-line recognition.

*Accuracy*:

- The number of correct recognition divided by the total number of recognition.

*Convergence rate*:

- The average number of observations, after which the recognition converges to the correct answer, over the average number of observations for those traces which converge.
Sensor-Based Multiple-Goal Recognition == Time-series Classification

- Recognition based on sensory readings

<table>
<thead>
<tr>
<th>Trace #</th>
<th>Observation Sequences</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( (AP_1:57) ) ( (AP_1:56) ) ( (AP_1:55) ) ( (AP_1:52) )</td>
<td>( G_1 )</td>
</tr>
<tr>
<td></td>
<td>( (AP_2:33) ) ( (AP_2:30) ) ( (AP_2:36) ) ( (AP_2:62) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( (AP_3:51) ) ( (AP_3:62) ) ( (AP_3:56) ) ( (AP_3:47) )</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( (AP_1:62) ) ( (AP_1:39) ) ( (AP_1:46) ) ( (AP_1:11) )</td>
<td>( G_2 )</td>
</tr>
<tr>
<td></td>
<td>( (AP_2:57) ) ( (AP_2:41) ) ( (AP_2:45) ) ( (AP_2:43) )</td>
<td>( G_3 )</td>
</tr>
<tr>
<td></td>
<td>( (AP_3:55) ) ( (AP_3:32) ) ( (AP_3:43) ) ( (AP_3:27) )</td>
<td></td>
</tr>
</tbody>
</table>

- Multiple-goal in a single action sequence

![Diagram showing different types of goals](image-url)
Plan Recognition and Activity Recognition

- Two categories of approaches:
  - Consistency approaches
    - Formal theory of plan recognition [Kau87]
    - Scalable and adaptive goal recognition [Les98]
  - Probabilistic approaches
    - Hidden Markov models
    - Bayesian Net and dynamic BN

- Limitations
  - Logic-based assume actions are given, and cannot deal with uncertain signals
  - Bayesian approaches must have a winning goal; but there may be several concurrent goals
Framework of Sensor-Based Multiple-Goal Recognition

Two-level multiple-goal recognition framework
Model Instantiation and evolution

- A **default-goal** model $M_0$ is instantiated when a goal model is created at time $t$
  - $M_0$ is added into the model set $\mathbf{M}$
  - $L_t(M_0) = \pi_0 Q_0(A_t)$

- A **goal** model $M_k$ is instantiated
  - whenever $\pi_k Q_k(A_t) \geq \pi_0 Q_0(A_t)$
  - $Acc(M_k) = A_t$ and $M_k$ is added into $\mathbf{M}$
  - $L_t(M_k) = \pi_k Q_k(A_t)$
Experimental Setting

- Actions and goals:

<table>
<thead>
<tr>
<th>AID</th>
<th>Name</th>
<th>AID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A₁</td>
<td>Walk-in-HW₁</td>
<td>A₂</td>
<td>Walk-in-HW₂</td>
</tr>
<tr>
<td>A₃</td>
<td>Walk-in-HW₃</td>
<td>A₄</td>
<td>Walk-in-HW₄</td>
</tr>
<tr>
<td>A₅</td>
<td>Walk-in-HW₅</td>
<td>A₆</td>
<td>Walk-in-HW₆</td>
</tr>
<tr>
<td>A₇</td>
<td>Walk-in-HW₇</td>
<td>A₈</td>
<td>Print_R₁</td>
</tr>
<tr>
<td>A₉</td>
<td>Seminar_R₁</td>
<td>A₁₀</td>
<td>Print_R₂</td>
</tr>
<tr>
<td>A₁₁</td>
<td>Seminar_R₂</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GID</th>
<th>Name</th>
<th>GID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>G₁</td>
<td>“Print-in-Room2”</td>
<td>G₂</td>
<td>“Seminar-in-Room2”</td>
</tr>
<tr>
<td>G₃</td>
<td>“Print-in-Room1”</td>
<td>G₄</td>
<td>“Seminar-in-Room1”</td>
</tr>
<tr>
<td>G₅</td>
<td>“Go-to-Office”</td>
<td>G₆</td>
<td>“Exit-through-Entrance3”</td>
</tr>
<tr>
<td>G₇</td>
<td>“Exit-through-Entrance1”</td>
<td>G₈</td>
<td>“Exit-through-Entrance2”</td>
</tr>
</tbody>
</table>

- **eight** goals, 850 single-goal traces
- Multiple-goal traces are synthesized:
  - Segments of single-goal traces are pieced together to generate connective traces containing multiple goals.
Comparison Targets & Evaluation Criteria

- Three algorithms:
  - MG-Recognizer [Cha05]
  - SG-Recognizer [Yin04]
  - BHMM-Recognizer [Han99]

- Three criteria:
  - Recognition accuracy
  - Inference efficiency
    - Measured in terms of the number of models instantiated
  - Scalability
    - \( w.r.t. \) the number of goals modeled
    - \( w.r.t. \) the number of goals contained in a single trace
An Example

Two goals are achieved in a single trace:

\[ G_1 = \text{“Print-in-Room2”} \] and \[ G_2 = \text{“Exit-through-Entrance2”} \]
Recognition Accuracy

- **SG-Recognizer**

![Graph showing Recognition Accuracy with actions and posterior probabilities](image)

- G0 = Default
- G1 = "Print-in-Room2"
- G2 = "Exit-through-Entrance2"
Accuracy and Efficiency

- **Recognition accuracy:**

<table>
<thead>
<tr>
<th>Recognizer</th>
<th>SG-Recognizer</th>
<th>BHMM-Recognizer</th>
<th>MG-Recognizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Goal</td>
<td>97.8%</td>
<td>95.5%</td>
<td>94.6%</td>
</tr>
<tr>
<td>Multiple-Goal</td>
<td>24.5%</td>
<td>79.1%</td>
<td>91.4%</td>
</tr>
</tbody>
</table>

- **Inference Efficiency:**

<table>
<thead>
<tr>
<th>Recognizer</th>
<th>SG-Recognizer</th>
<th>BHMM-Recognizer</th>
<th>MG-Recognizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Goal</td>
<td>9</td>
<td>20.7</td>
<td>6.5 + 3.7</td>
</tr>
<tr>
<td>Multiple-Goal</td>
<td>9</td>
<td>28.7</td>
<td>6.6 + 4.8</td>
</tr>
</tbody>
</table>
ML Prob: Segmentation and Feature Selection in Multi-dimensional Time-series

- Learning a Probabilistic Segmentation Model [AAAI 2005]
- We partition an observation sequence \( Y \) into \( N_s \) segments

- Segment labels \( S = \{s_1, s_2, \ldots, s_{N_s}\} \) and segmentation points \( H = \{h_1, h_2, \ldots, h_{N_s}\} \)
ML Prob: Statistical Relational Learning: Action Model Learning

- **Input**: observed plans
  - init\(_1\), a\(_{11}\), a\(_{12}\), a\(_{13}\), …, a\(_{1n}\), goal\(_1\)
  - init\(_2\), a\(_{21}\), a\(_{22}\), a\(_{23}\), …, a\(_{2m}\), goal\(_2\)
  - …

- **Output**: action models; e.g.
  - load (x - hoist y - crate z - truck p - place)
  - pre: (at x p), (at z p), (lifting x y)
  - del: (lifting x y)
  - add: (at y p), (in y z),
    - (available x), (clear y)

- **Main Issue**:
  - Automatically guess an initial action model
  Then allow humans to edit these models

**Key contribution**:
- can learn action models even when no intermediate state observations are available
Distributed Learning In Sensor Networks


- Key Insight:
  - Robustness of learned model is key in sensor networks
    - Nodes may be added to the network or fail
    - Communication is unreliable, and link qualities change over time
  - Distributed machine learning:
    - Pieces of models are learned at different sites
    - A central model is integrated together.
Conclusions and Future Work

- Sensor and Wireless Networks provides grounds for new Machine Learning Research
  - Semi-supervised Classification on time series
  - Data and Model Migration
  - Multi-dimensional Segmentation
  - Distributed Learning of robust models
Our Other Work in 2005 (HKUST)

- See [http://www.cs.ust.hk/~qyang](http://www.cs.ust.hk/~qyang)

Research in ubiquitous computing


Research in Web mining

- KDD-CUP 2005: Champion on all three awards (see [http://webproject1.cs.ust.hk/q2c/](http://webproject1.cs.ust.hk/q2c/))

Research in case-based reasoning & Machine Learning

Accurate and Low-cost Indoor Location Estimation Using Kernels


**Problem**
A user with a mobile device walks in an indoor wireless environment (Covered by WiFi signal)

**Motivation**
- Similar signals may not necessarily be nearby locations, or vice versa
- Maximize correlation between signal and location under feature transformation

**Experiment Results**

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (Error in 3.0m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LE-KCCA</td>
<td>91.6%</td>
</tr>
<tr>
<td>SVM</td>
<td>87.8%</td>
</tr>
<tr>
<td>MLE</td>
<td>86.1%</td>
</tr>
<tr>
<td>RADAR</td>
<td>78.8%</td>
</tr>
</tbody>
</table>

**Methodology**
Kernel Canonical Correlation Analysis
- Proposed by [D.R. Hardoon et al. 2004]
- Two non-linear Canonical Vectors $W_x$ & $W_y$
  
  \[
  \begin{align*}
  W_x &= X \alpha \\
  W_y &= Y \beta
  \end{align*}
  \]
- $K$ is the kernel
  \[
  K(x, z) = \langle \Phi(x), \Phi(z) \rangle
  \]
- Maximize the correlation of projections
  \[
  \rho = \max_{\alpha, \beta} \frac{\alpha^T K_x \beta}{\sqrt{\alpha^T \alpha} \cdot \beta^T K_y \beta}.
  \]
Activity Recognition through Goal-Based Segmentation

Jie Yin, Dou Shen, Qiang Yang and Ze-Nian Li, AAAI 2005

- Application Domain: Wireless Environment

- Trace Database on Signal-Strength Readings

<table>
<thead>
<tr>
<th>Trace ID</th>
<th>Signal-strength Sequences</th>
<th>Goal ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(AP1:80) (AP2:78) (AP3:62)</td>
<td>G_1</td>
</tr>
<tr>
<td></td>
<td>(AP1:81) (AP2:77) (AP3:64)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(AP1:64) (AP2:69) (AP3:71)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(AP1:68) (AP2:84) (AP3:81)</td>
<td></td>
</tr>
</tbody>
</table>

- Goal-Based Segmentation algorithm
  - High-level goals can be recognized from low-level signal segments
  - Each segment define a motion pattern

- Probabilistic Segmentation Model

- Illustration on Sensory data

Reduce human effort in calibration for activity recognition!
Competence Driven Case-Base Mining

R. Pan, Q. Yang, J.F. Pan, L. Li, AAAI 2005

Case-Based Reasoning: Using previous cases to solve a current problem

- Step 1: learn the distribution of the original sample – using KFDA
- Step 2: Mining cases by considering the distribution and diversity

Problem: How to automatically obtain a quality case base from the raw data? What is the quality metric?

<table>
<thead>
<tr>
<th>Applicant</th>
<th>Income</th>
<th>Married</th>
<th>Cars</th>
<th>Approved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sammy</td>
<td>50K</td>
<td>n</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>Beatrice</td>
<td>50K</td>
<td>y</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>Dylan</td>
<td>80K</td>
<td>n</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>Mathew</td>
<td>30K</td>
<td>n</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>Larry</td>
<td>40K</td>
<td>n</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Basil</td>
<td>80K</td>
<td>n</td>
<td>1</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Customer Database

- Step 1: learn the distribution of the original sample – using KFDA
- Step 2: Mining cases by considering the distribution and diversity
ARMS: Learning Action Models
ICAPS 2005

- **Input:** observed plans
  - init\textsubscript{1}, a\textsubscript{11}, a\textsubscript{12}, a\textsubscript{13}, ..., a\textsubscript{1n}, goal\textsubscript{1}
  - init\textsubscript{2}, a\textsubscript{21}, a\textsubscript{22}, a\textsubscript{23}, ..., a\textsubscript{2m}, goal\textsubscript{2}
  - ...

- **Output:** action models; e.g.
  - load (x - hoist y - crate z - truck p - place)  
    - pre: (at x p), (at z p), (lifting x y)
    - del: (lifting x y)
    - add: (at y p), (in y z),
      - (available x), (clear y)

- **Main Issue:**
  - Automatically guess an initial action model
  - Then allow humans to edit these models

Winner: the first ICAPS/KE competition 2005, CA, USA
ACM KDD-CUP 2005 -- Winner
HKUST Team: D. Shen, R. Pan, J.T. Sun, J.F. Pan, K.H. Wu, J. Yin and Professor Q. Yang

- **Task**
  - Categorize 800,000 queries into 67 predefined categories;

- **Limitation**
  - “There is no restriction on what data you can/can’t use to build your models.” From http://kdd05.lac.uic.edu/kddcup.html

- **Key Characteristics:**
  - No training data
  - Meaning of Queries: ambiguous
    - A query usually contains too few words;
    - Queries often have more than one meaning.
  - Semantics of Categories: uncertain
    - Only the names of Categories, no more specification;

- **HKUST won all three awards for KDDCup 2005:**
  - Query Categorization
    - Precision Award,
  - Query Categorization
    - Performance Award
  - Query Categorization
    - Creativity Award

Phase I

Phase II

Ensemble
KDDCUP Winners Aug 2005
Thank You!

- http://www.cs.ust.hk/~qyang