802.11 + Regression = Location

John Krumm, John Platt
Microsoft Research
Poster at Machine Learning meets the User Interface workshop at NIPS 2003
Structure of Poster

- How location awareness relates to UI
- How we sense user’s location
- Machine learning approach
- Experiment in reducing calibration time
Location-aware UI applications

- Route communications to/from you
  - telephone calls, printers, video, etc.
- Give directions / tour guide
- Augmented reality
  - add clickable “hot spots” to the world
- Conference assistance
- Location sensitive web pages
- Tracking children/visitors
Privacy concerns?

Location awareness systems falls into 2 buckets:

- **Network of sensors**
  - not under your control
  - scary, Big Brother?

- **Personal mobile computing**
  - you have the sensor, you can control
  - still not perfect --- you have to interact with the world

- Networked computing in general impacts privacy
Previous sensing methods

- Before 2000, papers described specialized sensor
  - IR, ultrasound, RF, transmitters scattered around
  - measure whether you can receive/transmit
  - or, measure time delay of arrival
- Expensive, would anyone actually use it?
“RADAR” 802.11 sensing system


- Clever idea: you have an 802.11 transceiver anyway
- Use signal strength / SNR of base stations
- Estimate your WiFi card’s current location

- Two approaches
  - use building & transceiver geometry to build model
  - use gathered statistics to estimation location

- Statistical approach had lower RMS error
RADAR statistics approach

- Gather calibration data from every room
- Store signal strength + SNR for each room
- During use, measure signal strength + SNR
  - choose room which most closely matches input
- In ML terms, nearest neighbor classification
Limitations of RADAR

- Can only produce locations previously calibrated
- Calibration procedure is thus painful and slow
Our approach

- Want real-valued location outputs

\[ \text{vector of signal strengths, one per base station} \rightarrow \text{Regression} \rightarrow \text{X, Y, (Z?) location in meters} \]

- Can interpolate between calibration locations
  - not a surprise to ML people
- If base station not detected
  - Use -100dB signal strength instead
Regressor structure

- Standard Moody and Darken RBF, Gaussian kernel
  \[ x(s) = c_x + \sum_{j=0}^{M-1} \alpha_j K(s, s^*_j) \]

- Trained to minimize MSE in \( x \) and \( y \)

- Full training set
  - 27,796 measurements taken from all 137 locations
  - RBF centers are 5 k-means clusters from each location

- Test set
  - 25,457 measurements taken a few days later
Other Machine Learning Approaches

- Roos, et. al, 2002
  - Use parameterized model for $s(x, y)$
  - Use maximum likelihood to infer location from strength

- Schwaighofer, et. al, NIPS 2003
  - Use Gaussian Process Regression to model $s(x, y)$
  - Use maximum likelihood also

- We solve inverse problem directly
Experimental area

- Measured signal strengths in our building

- Took measurements from all rooms
  - will subsample measurements in software, later
Calibration program

- Simple UI for calibration:

  - Took 4 hours to gather data from every room
    - Minimizing calibration is important
Reduced number of calibration points

- Subsampled training as if calibrated fewer rooms
Results from reduced calibration

- Can reduce calibration by 2-3x, still get OK results
Conclusions

- 802.11 is a inexpensive way of finding indoor location
  - enables interesting UI
- Should use regression, rather than classification
- Can reduce calibration effort by 2-3x
  - increase RMS by 20-40%