CS144r/244r
Network Design Project
on
Secure and Intelligent Internet of Things
(personalized gesture recognition)
2/12/2014

Instructor: Professor HT Kung
Harvard School of Engineering and Applied Sciences
Announcements

• We will have no class next Monday, Feb 17, because it is a University Holiday (President's Day)

• Guest lectures
  – Tsung-Han Lin will give an overview of popular machine learning methods on Wed, Feb 19
  – Dr. Robert Cohn will discuss IoT programming on Feb 24

• Next set of questions due Tuesday night next week (Feb 18) will be posted the previous evening
  – They will cover materials in today’s lecture and readings for Feb 19’s class. For the latter, see the course website next Monday evening

• We will announce time and place for instruction labs on breadboarding techniques. These labs will be mandatory for students who do not have previous experience in this area. (This requirement will be waived for others. These students should let TFs know. Please consider volunteering to help out in the labs and get some bonus points on in classroom/lab participation)
What Is New in the News?

• “Forget Robocop, the NYPD is now experimenting with Google Glass” (2/8/2014)
  • The boys in blue at the New York Police Department have become Google Glass explorers
  • Officers could record the actions allowing those in control rooms to see live footage, while the same videos would probably be admissible in court as evidence. (Big pricy concern)

• “Instagram Makes Big Play at New York Fashion Week” (2/10/2014)
  • To encourage picture taking and posting….has mounted a massive 6 foot by 27 foot screen in at Lincoln Center. (Adding some gesture recognition would be nice)
Recap: Power-Performance and Energy-Performance Trade-offs

- Note **super-linear** trade-off curves (red)
- Where do we pay in power/energy for high performance under the same clock frequency?
  - Larger caches
  - Larger branch prediction tables
  - Faster TLB (Translation Lookaside Buffer)
  - Wider instruction and data paths
  - OOO execution
  - Larger active chip area (and higher leakage)
- These features have little to do with CISC or RISC
- A central issue with IoT devices is about making proper reduction on all these energy consuming features

Note that in Fig. 15 A8 moves to a higher location (why?)
Recap: Some Notable Wearables

• Glasses
  – Google glass and various competitors

• Smart watches in two flavors (note that Apple and Google are absent from today’s market!)
  1. Fully-fledged **standalone phones** on a wrist
     • Feature watch phone (e.g., LG GD510)
     • Smart watch phone (e.g., Neptune Pine based on Snapdragon S3)
  2. **Companion watches** for smartphones
     • E.g., Pebble, Samsung Galaxy Gear, Sony SmartWatch 2, Qualcomm Toq, Martian
     • These seem to be the current market focus (see Apple’s US patent application)

• **Smart bands**
  – Fitness tracking (e.g.,Nike FuelBand and Fitbit Force)
Recap: Motion Sensors

• Inertial sensors such as MEMS accelerometers and gyroscopes
  – Microelectromechanical systems (MEMS) contain miniaturized structures, sensors, actuators, and microelectronics
  – Google Glass uses the Invensense MPU-6050 chip which integrates a 3-axis gyroscope, 3-axis accelerometer and a Digital Motion Processor (DMP) in a small 4x4x0.9mm package

• Inertial sensors and other sensors such as image sensors and GPS play complementary roles, and can be used together (so-called sensor fusing) in consumer products
Recap: Inertial Sensors

Widely used MEMS inertial sensors

- **Accelerometers**
  - Measure **acceleration** in x, y, z directions

- **Gyroscopes** (or simply, gyros)
  - Measure **angular velocity** in yaw, pitch, and roll directions

They complement each other in sensor fusion
Today’s Topic: Personalized Gesture Recognition

- Use an **accelerometer** to recognize free-space hand movements
- Allow new **personalized gestures** to be defined with single samples
- Use **Dynamic Time Warping (DTW)** to for sample-template matching

Objective: Recognize a Gesture in a Set

- A user provides one template gesture for each of the eight vocabulary gestures offline (actually two templates are kept for each gesture; see the last slide of this lecture)

- Then online the user gives a sample gesture. The system will match the sample against the eight stored templates, and find the best match

The Nokia vocabulary of eight simple gestures:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
</tr>
</tbody>
</table>

The dot denotes the start and the arrow the end
Motivating Application Examples

1. Gesture-based user authentication
   – A small number (<10) of users share a device that can be personalized by loading user-specific data
   – After a user performs a user-specific gesture (e.g. A2 on the right) with an accelerometer, the system can use the acceleration time series to identify the user

2. Interaction with 3D mobile user interfaces using user created gestures
   – Use gesture to control a media player (e.g., Start Playing, Increase Volume, Play Next Video, Pause)

These are examples of soft authentication, which work for application scenarios where stringent security/privacy requirements are not important
Personalized Gesture Recognition: Challenges

• Data acquisition is not easy. It is difficult to get a large number of samples from users
  – Use Dynamic Time Warping (DTW) to compare an input sample series against each stored template time series directly. So just one sample is needed

• Gesture may vary among users and over time even for the same user
  – Do user-dependent recognition
  – Use template adaption to reflect recognition results (correct or incorrect) of recent input samples

• Accelerometer data sensitive to noise and tilt
  – Do “averaging” (e.g., quantization, low-pass filtering), but this could cause lagging response
  – Use complementary sensors such as gyros or Kinect
System Overview

Continuous Acceleration

Quantization

Template Library

Discrete time series of discrete values (S)

Match S with $T_1$ using DTW

Match S with $T_N$ using DTW

Select Minimum Distance

Template Adaption
System Implementation

• The Wii remote sends the acceleration data and button actions (e.g., start gesture) through Bluetooth to a PC in real time
  – The prototype detects the start of a gesture when the ‘A’ button on the Wii remote is pressed; and detects the end when the button is released

• The uWave system is implemented on a Windows PC using Visual C#, using about 300 lines of code
Technique 1: Quantization of Acceleration Data

1. The time series of acceleration is temporally compressed by an averaging window of 50ms that moves at a 30ms step.

- The acceleration data is converted into one of 33 levels. Non-linear quantization is employed because most samples are between $-g$ and $+g$ and very few go beyond $+2g$ or below $-2g$.

<table>
<thead>
<tr>
<th>Acceleration Data (a)</th>
<th>Converted Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a &gt; 2g$</td>
<td>16</td>
</tr>
<tr>
<td>$g &lt; a &lt; 2g$</td>
<td>11–15 (five levels linearly)</td>
</tr>
<tr>
<td>$0 &lt; a &lt; g$</td>
<td>1–10 (ten levels linearly)</td>
</tr>
<tr>
<td>$a = 0$</td>
<td>0</td>
</tr>
<tr>
<td>$-g &lt; a &lt; 0$</td>
<td>-1–10 (ten levels linearly)</td>
</tr>
<tr>
<td>$-2g &lt; a &lt; -g$</td>
<td>-11–15 (five levels linearly)</td>
</tr>
<tr>
<td>$a &lt; -2g$</td>
<td>-16</td>
</tr>
</tbody>
</table>

How to choose these parameters? It could be difficult.
Technique 2: Dynamic Programming (DP) ("Programming" Here Means "Planning")

- Many problems have the property that they can be solved recursively with subproblems and solution to subsubprograms can be reused by multiple subproblems.

- Dynamic programming (DP) exploits this property. It is a classical algorithm design paradigm with numerous applications in many fields.

- Consider, e.g., computing longest common subsequence (LCS) of two sequences with dynamic programming:

  \[ X = A \quad B \quad C \quad B \]
  \[ Y = B \quad D \quad C \quad A \quad B \]

DP works on segments of increasing lengths, starting from the beginning of each given sequence.
Suppose the two sequences have lengths $m$ and $n$

- Brute force: $O(2^m n)$
- DP: $O(mn)$

From the final entries of the table below we can read off the longest common subsequence: BCB

$$X = A \quad B \quad C \quad B$$
$$Y = \quad B \quad D \quad C \quad A \quad B$$
Gesture Time Series Collection

• Collect gestures corresponding to the Nokia vocabulary of eight vocabulary gestures from eight participants with the Wii remote.

• For testing purpose, from each participant, collect 70 samples for each vocabulary gesture over three weeks.

• Thus, for each participant, we have 70 length-8 data vectors with each component being a 3-dimensional time series corresponding to accelerometer data on three axes over time:

\[
\begin{bmatrix}
  s_1[1] \\
  s_2[1] \\
  \vdots \\
  s_8[1]
\end{bmatrix}, \quad
\begin{bmatrix}
  s_1[2] \\
  s_2[2] \\
  \vdots \\
  s_8[2]
\end{bmatrix}, \ldots,
\begin{bmatrix}
  s_1[i] \\
  s_2[i] \\
  \vdots \\
  s_8[i]
\end{bmatrix}, \ldots,
\begin{bmatrix}
  s_1[70] \\
  s_2[70] \\
  \vdots \\
  s_8[70]
\end{bmatrix}
\]

← vocabulary gesture 1
← vocabulary gesture 2
← vocabulary gesture 8
Performance Measurement: Methodology

Recall for each participant we have 70 data vectors:

\[
\begin{bmatrix}
    s_1[1] \\
    s_2[1] \\
    \vdots \\
    s_8[1]
\end{bmatrix}, \begin{bmatrix}
    s_1[2] \\
    s_2[2] \\
    \vdots \\
    s_8[2]
\end{bmatrix}, \ldots, \begin{bmatrix}
    s_1[i] \\
    s_2[i] \\
    \vdots \\
    s_8[i]
\end{bmatrix}, \ldots, \begin{bmatrix}
    s_1[70] \\
    s_2[70] \\
    \vdots \\
    s_8[70]
\end{bmatrix}
\]

We will perform 70 tests on these data vectors. For the \(i\)-th test, \(i = 1, \ldots, 70\), we choose

\[
\begin{bmatrix}
    s_1[i] \\
    s_2[i] \\
    \vdots \\
    s_8[i]
\end{bmatrix}
\]

as the template vector and use the remaining 69 data vectors as test vectors. DTW matching of the template vector with each test vector will yield a 8x8 confusion matrix (aka error matrix).

The final confusion matrix for a participant is the average over 70 tests.

An Illustration of Confusion Matrix:

<table>
<thead>
<tr>
<th>Actual class</th>
<th>Predicted class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>5  3  0</td>
</tr>
<tr>
<td>Dog</td>
<td>2  3  1</td>
</tr>
<tr>
<td>Rabbit</td>
<td>0  2 11</td>
</tr>
</tbody>
</table>
Dynamic Time Warping (DTW)

First $i$ items in sample time series: $S[1], S[2], \ldots, S[i])$

First $j$ items in template time series: $T[1], T[2], \ldots, T[j])$

$D_{0} = \text{Distance between } S[i] \text{ and } T[j]$

$= \sqrt{(S[i].x-T[j].x)^2 + (S[i].y-T[j].y)^2 + (S[i].z-T[j].z)^2}$

$d_1 = \text{DTW Distance between } S[1:i-1] \text{ and } T[1:j]$  

$d_2 = \text{DTW Distance between } S[1:i] \text{ and } T[1:j-1]$  

$d_3 = \text{DTW Distance between } S[1:i-1] \text{ and } T[1:j-1]$  

$D_1 = \min(d_1, d_2, d_3)$

DTW distance between $S[1:i]$ and $T[1:j]$ is $D = D_0 + D_1$

Note that three axes: $x$, $y$ and $z$
User-dependent Performance Measurement Results

### All Days

<table>
<thead>
<tr>
<th></th>
<th>↑</th>
<th>←</th>
<th>←</th>
<th>↑</th>
<th>↓</th>
<th>↑</th>
<th>↓</th>
<th>≤</th>
<th>≥</th>
<th>≤</th>
<th>≥</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>92.1</td>
<td>0.1</td>
<td>2.4</td>
<td>1.9</td>
<td>0.1</td>
<td>2.9</td>
<td>0.6</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≤</td>
<td>1.6</td>
<td>91.6</td>
<td>1.3</td>
<td>1.1</td>
<td>0.7</td>
<td>0.4</td>
<td>2.7</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>←</td>
<td>0.5</td>
<td>0</td>
<td>95.9</td>
<td>1.2</td>
<td>0.7</td>
<td>1.7</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≥</td>
<td>0.3</td>
<td>0</td>
<td>1.6</td>
<td>96.2</td>
<td>0.7</td>
<td>1.1</td>
<td>0</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≤</td>
<td>0.3</td>
<td>0</td>
<td>1.5</td>
<td>0.6</td>
<td>97.0</td>
<td>0.5</td>
<td>0</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≥</td>
<td>2.4</td>
<td>0</td>
<td>2.4</td>
<td>2.3</td>
<td>1.0</td>
<td>91.7</td>
<td>0.1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≤</td>
<td>3.4</td>
<td>1.9</td>
<td>2.6</td>
<td>1.7</td>
<td>0.4</td>
<td>0.7</td>
<td>89.2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≥</td>
<td>1.1</td>
<td>0.6</td>
<td>1.7</td>
<td>0.9</td>
<td>0.8</td>
<td>0.7</td>
<td>0</td>
<td>94.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Same Day

<table>
<thead>
<tr>
<th></th>
<th>↑</th>
<th>←</th>
<th>←</th>
<th>↑</th>
<th>↓</th>
<th>↑</th>
<th>↓</th>
<th>≤</th>
<th>≥</th>
<th>≤</th>
<th>≥</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>98.4</td>
<td>0</td>
<td>0.3</td>
<td>0.4</td>
<td>0</td>
<td>0.4</td>
<td>0.3</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≤</td>
<td>0.5</td>
<td>98.3</td>
<td>0.2</td>
<td>0</td>
<td>0.3</td>
<td>0.1</td>
<td>0.4</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>←</td>
<td>0.2</td>
<td>0</td>
<td>98.3</td>
<td>0.6</td>
<td>0.1</td>
<td>0.6</td>
<td>0.2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≥</td>
<td>0.2</td>
<td>0</td>
<td>0.3</td>
<td>98.8</td>
<td>0.3</td>
<td>0.2</td>
<td>0.2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≤</td>
<td>0.4</td>
<td>0</td>
<td>0.2</td>
<td>0.4</td>
<td>98.7</td>
<td>0.1</td>
<td>0.2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≥</td>
<td>0.7</td>
<td>0</td>
<td>0.6</td>
<td>0.5</td>
<td>0.3</td>
<td>97.7</td>
<td>0.2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≤</td>
<td>0.5</td>
<td>0.4</td>
<td>0.4</td>
<td>0.1</td>
<td>0.1</td>
<td>0.3</td>
<td>98.1</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>≥</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
<td>99.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

93.5% 98.4%

A note on the importance of user-dependent recognition: If we treat all the samples of all eight participants as from the same participant, the accuracy will decrease to 75.4% from 98.4%
Technique 3: Template Adaptation Over Days

- Keep two templates generated in two different days for each vocabulary gesture. We match a gesture input with both templates and take the smaller matching cost of the two as the matching cost between the input and vocabulary gesture.

- As the user inputs more gesture samples, updates the templates based on how old the current templates are and how well they match with new inputs.
  - **Positive Update**: If both templates for a vocabulary gesture are at least one day old and the input gesture is correctly recognized, the older one will be replaced by the newly correctly recognized input gesture.
  - **Negative Update**: We replace the older template with the input gesture when it is incorrectly recognized.

- It would be interesting to see how these updates will allow templates to converge to good gesture inputs.
## Performance Measurement Results
(All Days with Template Adaptation)

### Positive Update

<table>
<thead>
<tr>
<th></th>
<th>▲</th>
<th>▼</th>
<th>→</th>
<th>←</th>
<th>↑</th>
<th>↓</th>
<th>⊙</th>
<th>⊙</th>
</tr>
</thead>
<tbody>
<tr>
<td>▲</td>
<td>96.8</td>
<td>0</td>
<td>1.5</td>
<td>0.3</td>
<td>0</td>
<td>1.1</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>▼</td>
<td>0.7</td>
<td>96.4</td>
<td>0.5</td>
<td>0.2</td>
<td>0.2</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
</tr>
<tr>
<td>→</td>
<td>0</td>
<td>0</td>
<td>98.9</td>
<td>0.6</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>←</td>
<td>0.2</td>
<td>0</td>
<td>0.3</td>
<td>98.9</td>
<td>0.2</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>↑</td>
<td>0.2</td>
<td>0</td>
<td>0.2</td>
<td>0.1</td>
<td>99.3</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>↓</td>
<td>0.6</td>
<td>0</td>
<td>0.6</td>
<td>0.3</td>
<td>1.7</td>
<td>96.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>⊙</td>
<td>0.8</td>
<td>2.0</td>
<td>2.0</td>
<td>0.4</td>
<td>0</td>
<td>0.2</td>
<td>94.6</td>
<td>0</td>
</tr>
<tr>
<td>⊙</td>
<td>1.0</td>
<td>0.4</td>
<td>1.1</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>97.1</td>
</tr>
</tbody>
</table>

97.4%

(Achieving performance similar to same day)

### Negative Update

<table>
<thead>
<tr>
<th></th>
<th>▲</th>
<th>▼</th>
<th>→</th>
<th>←</th>
<th>↑</th>
<th>↓</th>
<th>⊙</th>
<th>⊙</th>
</tr>
</thead>
<tbody>
<tr>
<td>▲</td>
<td>97.7</td>
<td>0</td>
<td>1.2</td>
<td>0.6</td>
<td>0</td>
<td>0.6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>▼</td>
<td>0.6</td>
<td>98.6</td>
<td>0.2</td>
<td>0.1</td>
<td>0</td>
<td>0.1</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>→</td>
<td>0.1</td>
<td>0</td>
<td>99.1</td>
<td>0.4</td>
<td>0.1</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>←</td>
<td>0.1</td>
<td>0</td>
<td>0.4</td>
<td>99.0</td>
<td>0.1</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>↑</td>
<td>0.2</td>
<td>0</td>
<td>0.3</td>
<td>0.1</td>
<td>99.2</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>↓</td>
<td>0.5</td>
<td>0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.5</td>
<td>98.3</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>⊙</td>
<td>0.4</td>
<td>0.5</td>
<td>07</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>98.0</td>
<td>0</td>
</tr>
<tr>
<td>⊙</td>
<td>0.2</td>
<td>0</td>
<td>0.3</td>
<td>0.4</td>
<td>0.1</td>
<td>0.1</td>
<td>0</td>
<td>98.9</td>
</tr>
</tbody>
</table>

98.6%