Smart Player

Gina Tuazon (A53040749)
Justin Tee (A09260518)
Motivation & Related Works

Motivation
• Typical implementation of music streaming services involve:
  • Machine learning algorithms based on features of songs, ratings, likes, and dislikes, collaborative filtering
• What’s missing from today’s music recommendation services?
  • Personalization of music based on the user’s current state and mood
• Our solution:
  • Embedded systems application that automates the selection of music playlists based on the user’s heart rate

Related Works
• Pandora:
  • Music Genome Project [http://www.pandora.com/corporate/mgp.shtml]
• Bliss Buzzer:
  • Quantifying bliss with user’s heart rate [http://bliss.calit2.net]
• Google Play Store or Apple’s App Store:
  • Common music selector smart phone apps [https://play.google.com/, https://itunes.apple.com]
The Smart Player

• HW components:
  • Raspberry Pi 2
  • Gesture Sensor
  • Heart Rate Sensor
  • Arduino Uno
  • Speakers

• SW components:
  • Gesture Sensor Python API
  • Python Matlab libraries
  • Arduino IDE
  • Music Player ‘omxplayer’
The Classifier

• Gaussian Naïve Bayes & Support Vector Machines
  • Gaussian Naïve Bayes Classifier
    • Simple classifier; Bayes classifier is the maximum a posteriori (MAP) decision rule, which selects the class that maximizes the prior x likelihood
  • Support Vector Machines Classifier (SVC)
    • Linearly separable classifier, or with a kernel function, non-linear; Classifier maximizes the margin between the separating hyperplane and the nearest point to each class

• Training Data
  • Initialized with random samples from two normal (Gaussian) distributions, the first centered at the average resting heart rate (HR\text{rest}), the second at the approximate average active heart rate given age (HR\text{active})
    • Example, age = 25:
      \[
      \begin{align*}
      H_R_{\text{rest}} &= 100 \text{ bpm} - 60 \text{ bpm} + 60 \text{ bpm} = 80 \text{ bpm} \\
      H_R_{\text{active}} &= (H_R_{\text{active vigorous}} + H_R_{\text{active moderate}})/2 + H_R_{\text{active moderate}} = 134 \text{ bpm}, \quad \text{where } H_R_{\text{max}} = (220 \text{ bpm} - \text{age}) \text{ and} \\
      H_R_{\text{active moderate}} &= H_R_{\text{max}} \times \left(0.70 + 0.50 + 0.50\right)/2 \\
      H_R_{\text{active vigorous}} &= H_R_{\text{max}} \times \left(0.85 + 0.70 + 0.70\right)/2
      \end{align*}
      \]
  • Learns or personalizes to the user
    • When recommend mode is on, the Smart Player polls the HR, classifies the HR (as active or non-active), and adds this new datum to the training data
    • When recommend mode is on, the user can gesture to the Smart Player that the classification for the current HR is wrong (ie. the wrong playlist was selected), and the Smart Player instantly polls the HR, and adds this new datum
Results

• Successful implementation of real-time sensing and feedback for music recommendation

• Plots exhibit the function of the classifier
  • Initialize training data as two randomly sampled normal distributions, representing non-active and active heart rates
  • Classify new datum among 3 variants of the SVM classifier (SVC) and Gaussian Naïve Bayes; the majority vote is the classification result
    • In the 2nd plot below, the datum is classified as active by 3 classifiers, and non-active by 1 classifier
  • Update training data after a new datum arrives (polling & classification event)
    • In the 3rd plot below, the newly classified datum (circled) is added to the training data, with class label active (color red)