© 2006 Intel Corporation.

Intel Research : User Activity based Adaptive Power Management (APM)

- **Intel Research**
  - Nilesh Shah- Principal Investigator.
  - Georgios Theocharous
  - Prashant Gandhi
- **McGill U**
  - Shie Mannor
- **MIT**
  - Leslie Kaelbling
- **U of Pittsburgh**
  - Branislav Kveton
- **CMU**
  - Sajid Siddiqi
- **Harvard**
  - Chih-Han Yu
- **Santa Clara U**
  - Ogi Petrovic
- **Kentucky State U**
  - Vamsi Gudivada

© 2006 Intel Corporation.

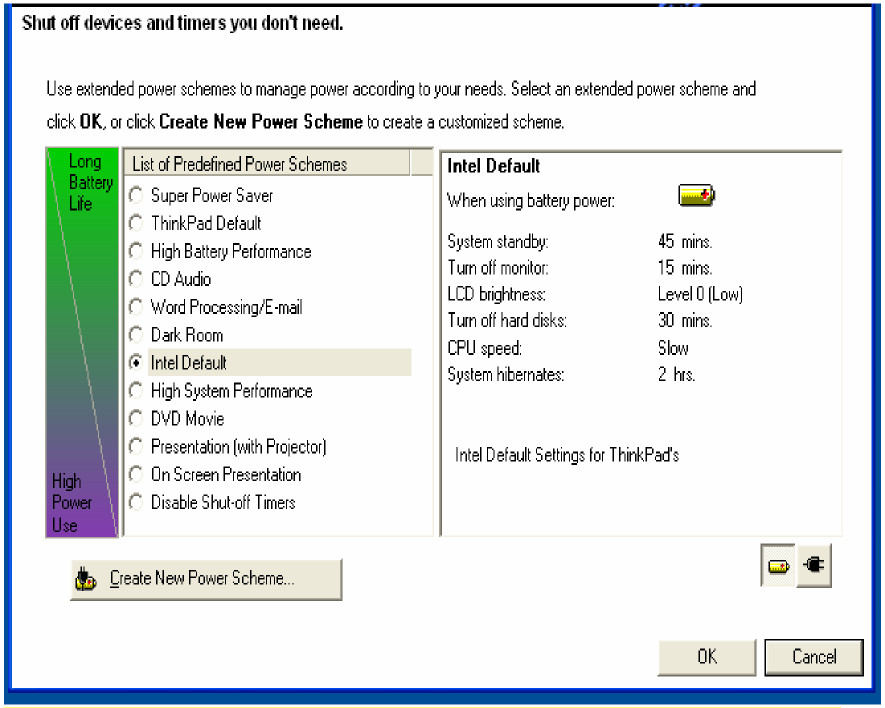
Intel Research : User Activity based Adaptive Power Management (APM)

- Agenda
  - What we want to enable.
  - Challenges.
  - Details.
  - Ideas/Feedback/Q and A.

© 2006 Intel Corporation.

What we want to enable

- Commercial products implement simple timeout based Power Management Policies.
  - Require manual interruption, waste power and annoy the user .
- We want to enable automated dynamic power management based on user-activity to maximize battery lifetime while maximizing *perceived* performance.
  - focus on mobile devices



Commercial timeout based Power Management Policies on mobile devices.

- **Recent research focused on modeling system dynamics using relatively small data corpus, rather than usage patterns, user not considered.**

© 2006 Intel Corporation.

Challenges

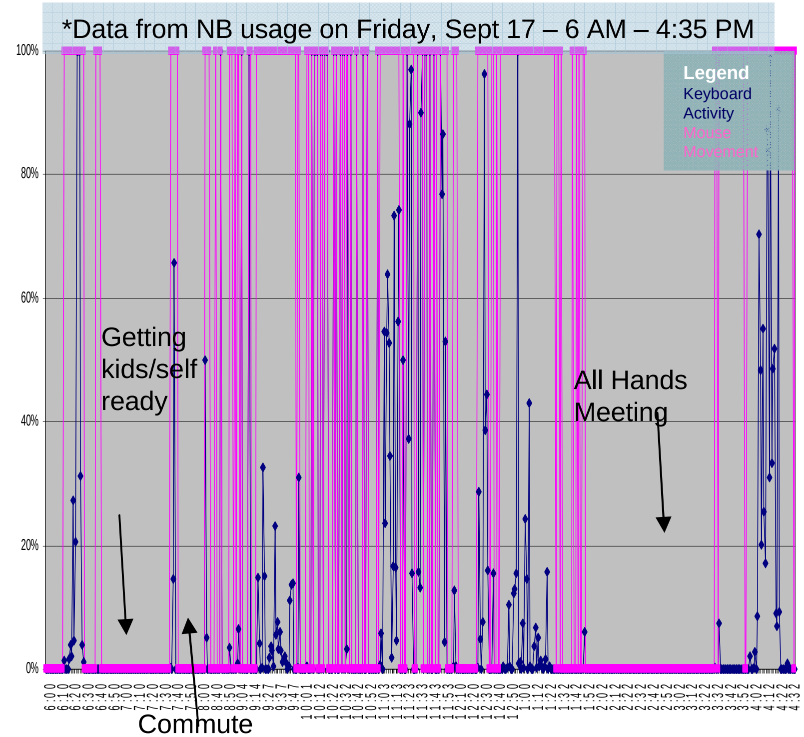
- Labeling input data given a partially observable user.
- Multi-objective optimization.
- Local and Global Learning (varying time scales, varying users).
- Data driven validation. – Non trivial problem, actions affect user behavior.

- **Develop machine learning algorithms that take up less resource than they save.**

© 2006 Intel Corporation.

Experimental setup

- **System Inputs:** Activity sensors on laptops (Software, User, Network, CPU, etc)
- **System Outputs :** 4 Binary Actions LCD, CPU, WLAN, Standby.



SENSORS

- Application
- Key counts
- Mouse clicks
- Mouse movement
- CPU utilization
- Packets/second

Machine Learning

ACTIONS

- Standby (yes/no)
- LCD (on/off)
- CPU (Maximum/slow)
- WLAN (on/off)

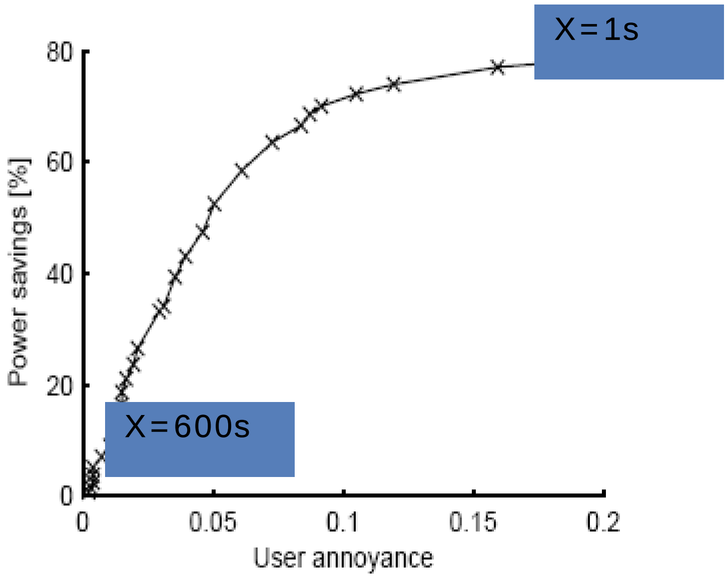
- **Ad hoc rules used to label data to provide a target policy.**

© 2006 Intel Corporation.

Optimization Objective

- Every action is assigned a cost.

- Incorrect actions incur cost, increase user annoyance.



x-Timeout value

- **Optimal operating point may be different for every user.**

© 2006 Intel Corporation.

Learning setup : Naïve approach

- Reduce problem to 4 separate binary classification problems corresponding to each of the 4 actions.
- "Oracle" policy used for labeling, benchmark.
- Current sensor space restricted to 6 dimensions.
- Predict idleness over fixed look ahead window.
- Input dimension is unrestricted.
- **Future work : Predict exact duration of idleness.**

© 2006 Intel Corporation.

Context-based classification

- **Context = Time from last activity of component (LCD, WLAN, CPU, Standby).**
  - F = mixture of expert classifiers.
  - 1{.}=Indicator function.
  - C(x) = map input space to set of contexts.
  - h_L = classifier corresponding to l-th context. (logistic regression model)
  - b_L = bias corresponding to l-th context.

- **Optimization – difficult problem.**
  - y_t = Label of training data.
  - Ao = Annoyance constraint.
  - Maximize total power savings subject to annoyance constraint. (set b_L to perform well on whole data set)

K=29

$$f(\mathrm{x}) = \sum_{\ell=1}^{k} 1\{c(\mathrm{x}) = \ell\} \mathrm{sgn}\left(h_\ell(\mathrm{x}) + b_\ell\right),$$

Greater the Threshold, less aggressive is the timeout.

Non-linear optimization: Threshold impacts Power and Annoyance.

$$\max_{b_1,\ldots,b_k} \quad \text{PowerSavings}\left(\{y_t, f(\mathrm{x}_t)\}_{t=1}^{T}\right)$$

$$\text{s.t.}: \quad \frac{1}{T}\sum_{t=1}^{T} \text{Annoyance}\left(y_t, f(\mathrm{x}_t)\right) \leq A_0$$

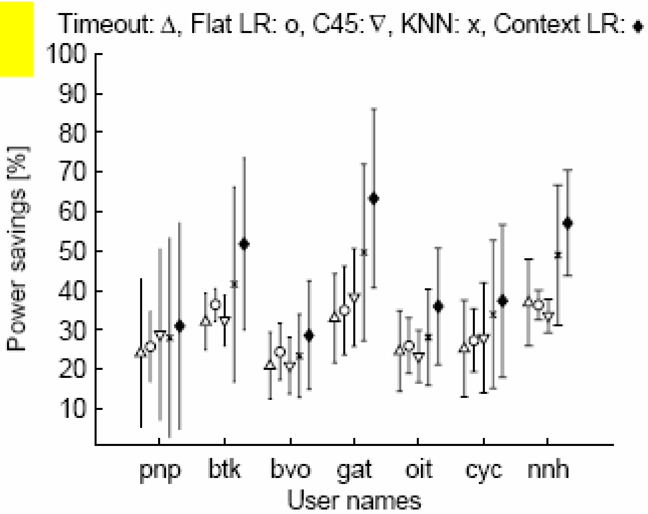
- **Interesting fact : The particular algorithm does not matter as much as problem setup. We tried KNN, C45, SVM, naïve Bayes, etc with minimal difference.**

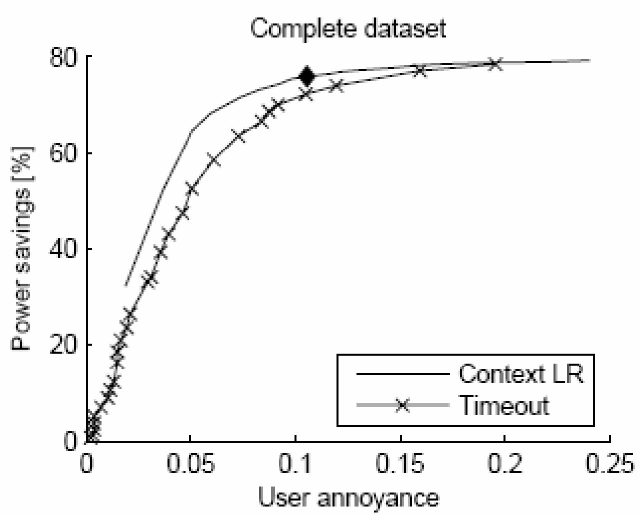
© 2006 Intel Corporation.

Current results

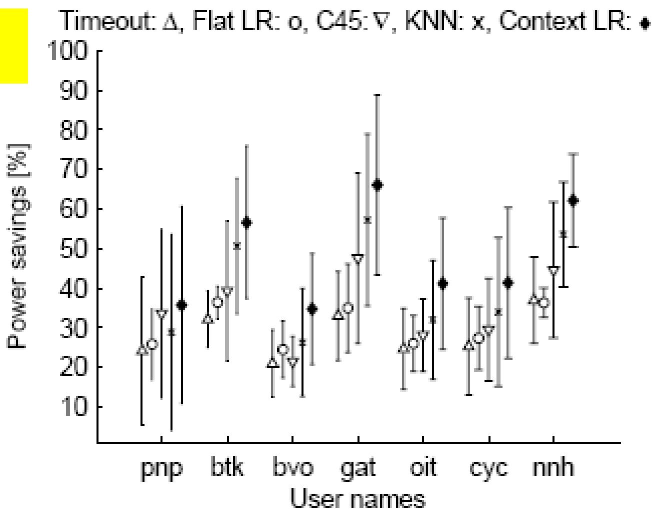
- Initial data set
  – 42 traces from 7 users, 210 usage hours collected on laptops.
  – Variance of power savings indicates need for Adaptive algorithms.

Annoyance=0.1





Annoyance=0.2

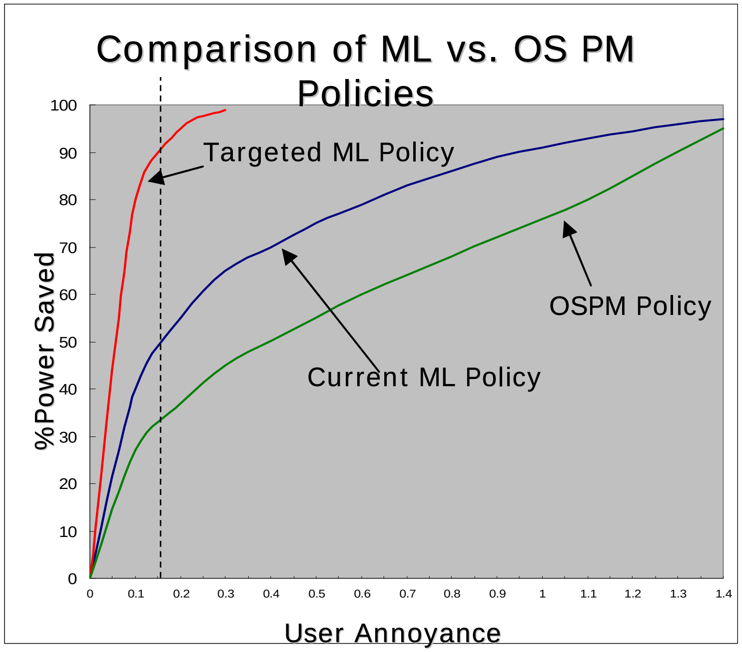


- **Context in classification of future idleness crucial, more so than choice of classifier. Mixture of classifiers utilized, each classifier suited for a particular context.**

© 2006 Intel Corporation.

Further Research Goals

- Discover modes of user behavior (contexts)
  - Segmentation and clustering techniques such as Hidden Markov Models.
- Estimate system idle durations from user activities
  - Semi-Markov Dynamic Bayesian Net models for activity-idleness relations.
- Maximize power savings with minimum annoyance level of current timeout policies
  - Algorithms to switch between timeout policies based on past user behavior
- Learn when to turn components ON before needed (pre-empt).



- Accurate and *individual* measures of annoyance.

- **Ensure results are valid real time “online” on laptops across variety of usage models and users.**

© 2006 Intel Corporation.

Ideas, Feedback, Q/A
Thank You !