Admin

- Project proposal—this Friday 10/11
  - Title
  - Andrew email addresses of participants
  - description (~500–750 words, or equivalent in pics/eqns)
    - dataset—access, contents, what do you hope to learn?
    - what is the first step? possible milestones?
    - minimal and stretch success criteria
- HW2—2 weeks from today—Mon 10/21
- Midterm—10/28 in class

Large images for handin

- Some students reported problems uploading large image files to the handin/discussion server (even if below the limit of 950k/file)
- Until we track down and fix the cause of those problems, we recommend that you avoid large-image-based handin methods
  - i.e., avoid scanned handwriting and LaTeX
  - you're welcome to ignore this advice if you really are set on handwriting or LaTeX, and we will try to support you
  - if it worked for you in HW1, it should continue to work

Projects

- Availability of an interesting data set
  - idea for what interesting things are in the data set
  - idea how to get at these things
- We are looking for interactivity
  - not just "run algorithms XYZ on data ABC," but ***interpret results*** and change course accordingly

Project ideas—ML on FAWN

- FAWN = Fast Array of Wimpy Nodes
  - handle highly multithreaded workload by throwing lots of low-energy processors at it, but great inter-node communication
- Calxeda: "Data Center Performance, Cell Phone Power"
  - one box = up to12 boards * 4 SOCs * 4 Cortex A9 cores
  - 192 high-end cell phones
  - Infiniband network
    - 100s of Gbit/s
    - ping time = 100ns (not ms!)

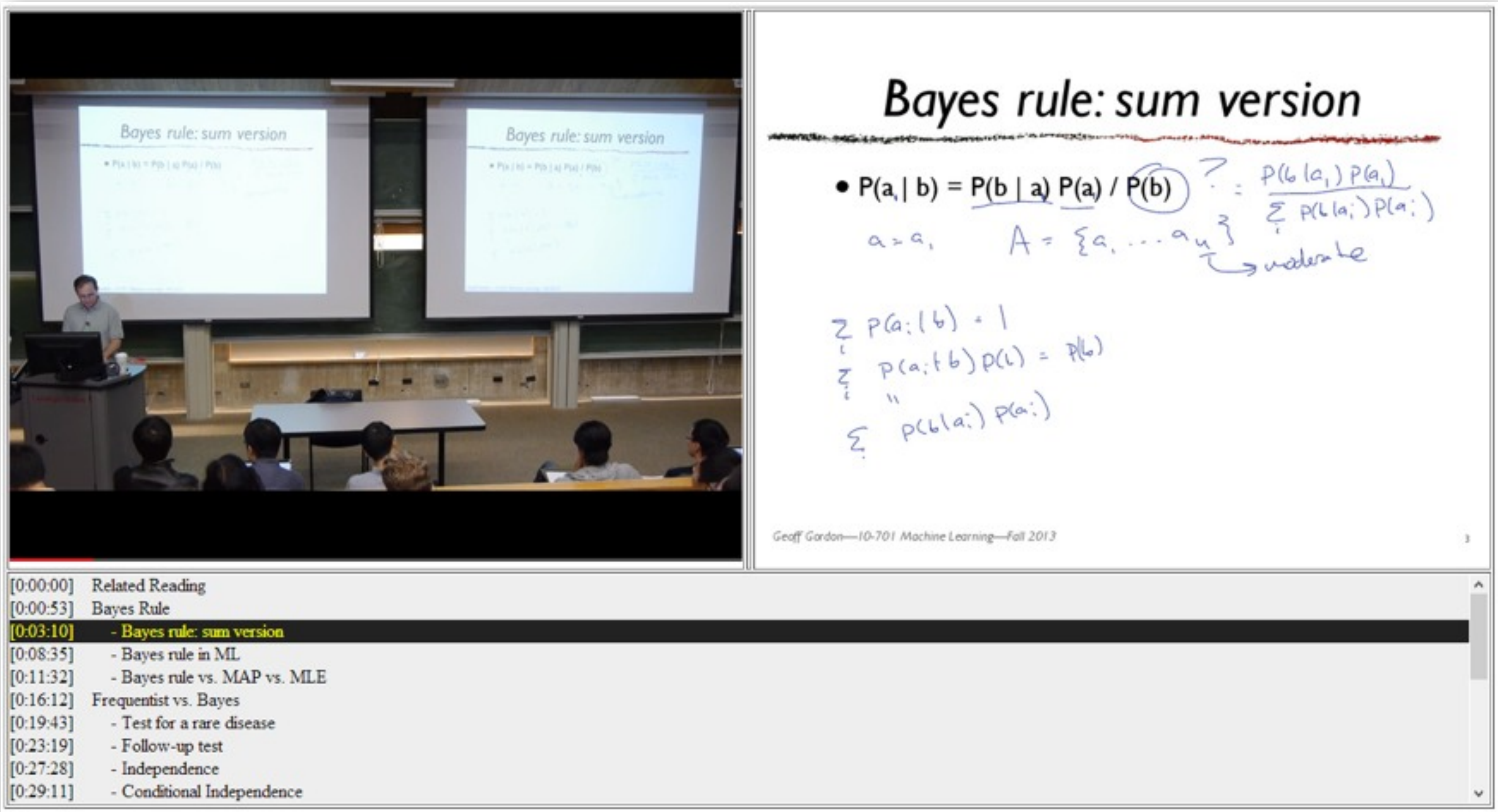
http://www.calxeda.com

Project—wearable accelerometer

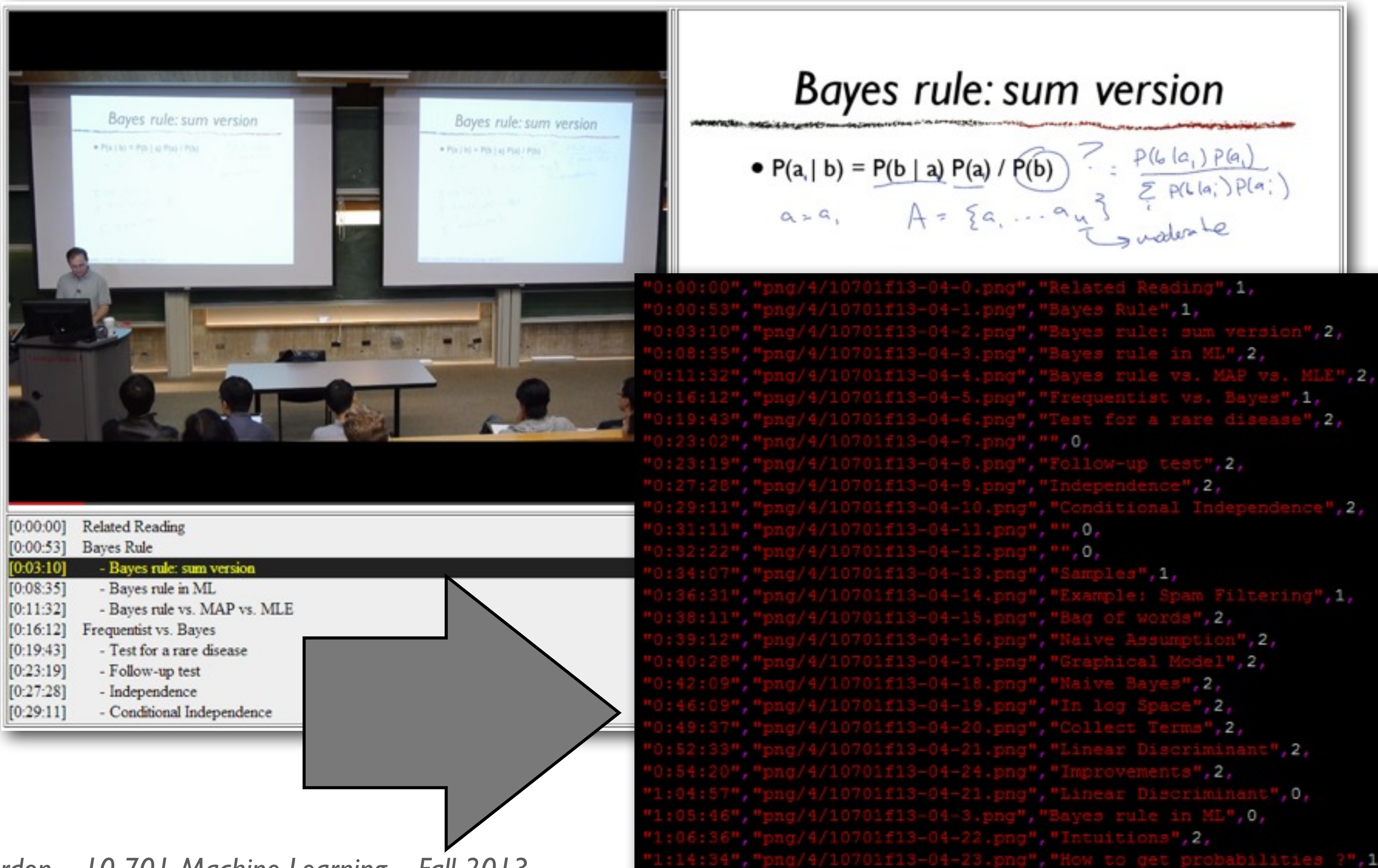
- Alex offers to buy hardware (disclaimer: may be different from picture)
- Goal: interpret data
  - segment and decompose observations into motion primitives
  - infer gait changes
  - monitor convalescing patients

http://www.bodymedia.com

Project—video annotation



Project—video annotation



Project—video annotation

- An ML project
  - Can use 3rd party toolboxes to compute features (e.g. OpenCV)—we don't care how you get them
  - Must have a learning component: use annotated lectures for training
    - ours, or scrape videolectures.net, techtalks.tv
- This is a project to satisfy a practical need
  - Your work will be used
  - We will need working, understandable code to be published as open source

Project—educational data

- Watch students interact w/ online tutoring system
- Understand what it is that they are learning, how each student is doing
- Big data set:
  - http://pslcdatashop.web.cmu.edu/KDDCup/
  - I helped run this challenge, so I have ideas about what might work…
- Goals: cluster problems by skills used, cluster students by knowledge of skills

Ed data, revisited

- Or, much smaller data but deeper learning
  - watch a student solve a problem
    - capture pen strokes as they draw diagrams or solve equations—I can provide software/HW for this
  - learn to distinguish solutions from random marks on paper, or eventually good solutions from bad ones
  - what is ***latent structure*** of a solution ("diagram grammar")

Project ideas—Kaggle

- Runs many ML competitions
  - data from StackExchange, cell phone accelerometers, solar energy, household energy consumption, flight delays, molecular activity, …
- Similar idea to challenge problems on our HWs, but less structure, and competing against the whole world
  - CMU is the hardest part of the world to compete against, so you should have no trouble…

Project ideas—Twitter

- Get a huge pile of tweets
  - http://www.ark.cs.cmu.edu/tweets/
- Build a network
- Analyze the network
- Learn something
  - topics, social groups, hot news items, political disinformation (“astroturf”), …

Others

- Loan repayment probability
- Grape vine yield
- Neural data: MEG, EEG, fMRI, spike trains
- Music: audio or MIDI
- …

Step back and take stock

- Lots of ML methods:

Step back and take stock

- Lots of ML methods:

Common threads

- Machine learning principles (MLE, Bayes, …)
- Optimization techniques (gradient, LP, …)
- Feature design (bag of words, polynomials, …)

Goal: you should be able to mix and match by turning these 3 knobs to get a good ML method for a new situation

Machine learning principles

- MLE: "a model that fits training set well (assigns it high probability) will be good on test set"

$\max_{\text{models } M} P(\text{data} \mid M)$ $\min_{M} -\ln P(\text{data} \mid M)$

- regularized MLE: "even better if model is 'simple'"

$\min_{M} -\ln P(D \mid M) + \text{pen}(M)$

- MAP: "want the most probable model given data"

$-\ln P(D \mid M) - \ln P(M)$

- Bayes: "average over all models according to their probability" $P(M \mid D) = P(D \mid M)\, P(M) / P(D)$

More principles

- Nonparametric: "future data will look like past data"

- Empirical risk minimization: "a simple model that fits our training set well (assigns it low E(loss)) will be good on our test set"

$$\min_M \sum_i \ell(x_i; M) + \text{pen}(M)$$

Examples

- linear regression (Gaussian errors) MLE, ERM
- linear regression (no error assumption) ERM
- ridge regression regularized MLE MAP
- k-nearest-neighbor — nonpar
- Naive Bayes for text classification
- Watson Nadaraya nonpar + Bayes
- Parzen windows nonpar

Selecting a principle

- Computational efficiency vs. data efficiency vs. what we're willing to assume
  - e.g., full Bayesian integration is often great for small data, but really expensive to compute
  - e.g., for huge # of examples and high-d parameter space, stochastic gradient may be the only viable option
  - e.g., if we're not willing to make strong assumptions about data distribution, suggests nonparametric or ERM
- Often wind up trying several routes
  - e.g., to see which one leads to a tractable optimization

Common thread: optimization

- Use a principle to derive an objective fn
  - hopefully convex, often not
- Select algorithm to min or max it
  - or sometimes integrate it—like optimization, but harder

Optimization techniques

- If we're lucky: set gradient to 0, solve analytically
- (Sub)gradient method
  - analyzed: –log(error) = O(# iters)   [note: bad constant]   strongly convex
- Stochastic (sub)gradient method
- Newton's method
- Linear prog., quadratic prog., SOCPs, SDPs, …
- Other: EM, APG, ADMM, …

Comparison

of techniques for minimizing a convex function

	Newton	APG	(sub)grad	stoch. (sub)grad.
convergence	*****	***	1-3 *	*
cost/iter	$$$$$	$$$\frac{1}{2}$	$$	$
assumptions	*	***$\frac{1}{2}$	3-5 *	*****

Common thread: features

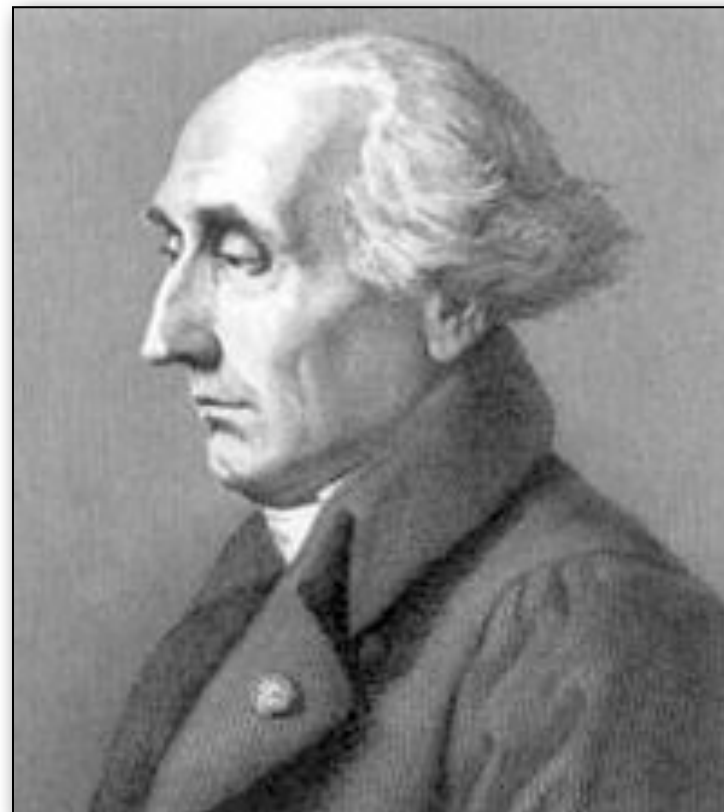
- Customer/collaborator/boss hands you SQL DB
- You need to turn it into valid input for one of these algorithms
  - discarding outliers, calculating features that encapsulate important ideas, …
- Options:
  - finite-length vector of real numbers
  - kernels: infinite feature spaces; strings, graphs, trees, etc.

Where does it all lead?

- Different principles, assumptions, optimization techniques, feature generation methods lead to different algorithms for same qualitative problem (e.g., many algos for "regression")
- Different principles can give same/similar algos
  - ridge regression as conditional MAP under Gaussian errors, or as ERM under square loss
  - many different linear classifiers: perceptron, NB, logistic regression, SVM, …

Lagrange multipliers

- Technique for turning constrained optimization problems into unconstrained ones
- Useful in general
    - but in particular, leads to a famous ML method: the support vector machine



Recall: Newton's method

- $\min_x f(x)$ → $H\Delta x + g = 0$
  - f: $R^d \to R$

$$f(x+\Delta x) \approx f(x) + f'(x)\cdot\Delta x + \Delta x^T H \Delta x / 2$$

(x)

$$\underline{\underline{f'(x)}} + \underline{\underline{H(x)}}\,\underline{\underline{\Delta x}} = 0$$

pretend const

solve

$\{q(t) \mid t \in [0,1]\}$ $x^2+y^2=1$

Equality constraints

- min f(x) s.t. p(x) = 0

$f'(x) = \lambda\, p'(x)$