CSC411
LECTURE 0 11/09/17.
- Ethan Fetaya, James Lucas, Emad Andrews
  └ OUR PROF.

BACKGROUND:
- Lin alg.
- Calculus: Part.
- Probability: Distributions, Bayes
- Statistics

http://www.cs.toronto.edu/~jlucas/teaching/csc411
ALSO piazza.

- EXTRA TEXTS:
  - Murphy: Machine Learning, a Prob. Approach.
  - Shai: Und. ML: from Th. to Algor.

UNDERGRADS:
- DO READINGS.
- READ 5 CLASSIC PAPERS
  - 5 PTS
  - HONOUR SYSTEM.
- ASSIGNMENTS:
  - 3 x 15% = 45% total
  - take Python code and extend
  - Derivations.

Midterm: 20%.
Final: 30%.

WHAT IS LEARNING?
- The main ML idea: acquiring a SKILL, or gaining KNOWLEDGE.
- ML $\neq$ AI.
  Most ML is: "I have a task, I want to SOLVE THE TASK". Very goal-oriented overall.

- CV: images are $a \times b \times 3$, big matrices. How to understand?

- Systems need to ADAPT.
- Systems need to HANDLE NOISE.

- We use TRAINING DATA to GENERALIZE. GENERALIZATION is the main idea.

- Learning systems DEVELOP OWN. PROGRAM based on EXAMPLES and TRIAL-AND-ERROR.
  - Implement an unknown fxn w/ only access to data.

Scanned by CamScanner

ML broad categories:

- SUPERVISED LEARNING: LEARN A MAPPING $x \to y$.
  Given pairs (x, y), LEARN A MAPPING $x \to y$.
  - CLASSIFICATION: categorical out
  - REGRESSION: real-valued output.

- paulownia (324)
- chrysanthemum (361)
- iris (CV)
- aloe. (ML)

- UNSUPERVISED LEARNING.
  Given pts x, find structure in the data. (dim. reduction)

- ONLINE LEARNING (more active in theory g'p)
  No training and testing
  Always learning, always predicting. (spam filtering)

- REINFORCEMENT LEARNING.
  Learn ACTIONS to MAXIMIZE FUTURE REWARDS

SUPERVISED LEARNING MATHEMATICAL SETUP:

- INPUT SPACE $\mathcal{X}$. ($\mathbb{R}^n$, images, text, sounds)
- OUTPUT SPACE $\mathcal{Y}$. ($\pm 1/-1$, $1 \dots k$, $\mathbb{R}$)
- UNKNOWN distribution D on $\mathcal{X} \times \mathcal{Y}$.
- LOSS FUNCTION: $\ell : Y \times Y \to \mathbb{R}$. e.g. 0-1 loss, sq. loss.
- Set m of i.i.d samples $(x_1, y_1) \cdots (x_m, y_m)$ sampled from distro D
  (the i.i.d ~~samples can~~ assumption can be problematic)

GOAL:
return a function (hypothesis): $h: \mathcal{X} \to Y$ that minimizes EXPECTED LOSS w/ respect to D.

$$L_D(h) = \mathbb{E}_{(x,y)\sim D}[\ell(h(x), y)].$$

But we don't know our theoretical L_D b/c D unknown.
We approximate w/ EMPIRICAL LOSS by taking mean.

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h(x_i), y_i).$$

For specific h, $L_S(h) \approx L_D(h)$.
But we might not be able to generalize. (overfitting)

CHALLENGE: Find model RICH ENOUGH to find patterns in data, but DOESNT OVERFIT by fitting random noise.
"the SWEET SPOT"

Scanned by CamScanner

ML viewpt:

- AGNOSTIC: minimiz. loss on unseen data
- DISCRIMINATIVE: Fit $P(y|x;\theta)$ by some parametric model.
- GENERATIVE: Fit $P(x,y;\theta)$ by some ~~parametric~~ m; generative model to fit $P(y|x;\theta)$.

ML WORKFLOW SKETCH:

1. Should I use ML on this problem?
   - is there a relation + pattern to detect?
   - can I solve ANALYTICALLY?
   - do I have the data?

2. Gather and organize data.
3. Preprocessing, cleaning, visualizing
4. Establish a BASE LINE (for accuracy, detection, performance, etc).
5. choose model, loss, regularization...
6. Optimization (could be simple grad. desc, could be PhD...)
7. Hyperparameter search.
8. Check performance and mistakes → go back to step 5 or 3

LINEAR REGRESSION.

REGRESSION: models continuous outputs
- Future stock prices
- Trading
- Housing prices
- Crime rates.

ASSUME SIMPLE GEOMETRY: CLOSER IS BETTER.

INGREDIENTS 2 MAKE PRED:
- Inputs (features): $\bar{x} \in \mathbb{R}^d$
- Outputs (dep. var): $y \in \mathbb{R}$.
- Training data: $(\bar{x}^1, y^1) \dots (\bar{x}^N, y^N)$.
- Model/hypothesis class:
  fam. of fxns w/ relationship btw x and y
  $f_w(x) = w_0 + w_1 x_1 + \dots + w_d x_d$ for $\bar{w} \in \mathbb{R}^{d+1}$
- Loss fxn:
  $L_2(y,\hat{y}) = (y-\hat{y})^2$, $L_1(y,\hat{y}) = |y-\hat{y}|$.
- Optimization: way to minimize loss objective
  Analytic soln, convex optimization

LINEAR MODELS are VERY SIMPLE.
IN LINEAR MODEL: LINEAR IN PARAMS NOT OUTPUTS.
Any POLYNOMIALS are a LINEAR (in w) MODEL.
LINEAR IN WEIGHTS W.

Any FIXED TRANSFORMATION: $\phi(x) \in \mathbb{R}^d$ we can run LIN. REG. w/ features $\phi(x)$.

FEATURE ENGINEERING — design GOOD FEATURES and FEED THEM TO A GOOD MODEL.
Commonly replaced w/ deep models that learn features as well, b/c features can be hard + complicated.

MOST COMMON LOSS: L_2 — prediction

$$L_2(y, \hat{y}) = (y - \hat{y})^2$$

- easy to optimize (convex, analytic soln)
- well understood
- larger mistake, harsher punishment.
- can be good (economic predictions) or bad (OUTLIERS AND NOISE)
- if lots of noise in data, can be punishing
- GOOD IF WE WANT MAX PUNISHMENT FOR BAD PREDICTION.

Optimal prediction w/r/t L_2 is CONDITIONAL MEAN $\mathbb{E}[y|x]$

Equivalent to assuming Gaussian noise

ANOTHER COMMON LOSS: L_1

$$L_1(y, \hat{y}) = |y - \hat{y}|$$

- Smoother, grad desc. works here
- Easyish to optimize (convex)
- MORE ROBUST TO OUTLIERS

Optimal prediction w/r/t L_1 is CONDITIONAL MEDIAN.
Equivalent to assuming Laplacian noise

Scanned by CamScanner

COMBINE BOTH:

HUBER LOSS:
- Close together, $\xrightarrow{\to 0}$ L_2
- Farther, L_1
- stitch smoothly.

DERIVING + ANALYZING THE OPTIMAL SOLN.

- we can include the bias by adding 1 (the w_0).
  So $\bar{x}^{(i)} = [1, x_1^{(i)} \ldots x_j^{(i)}]$, and prediction is $\bar{x}^T \bar{w}$

- Target vector:
  $$\bar{y} = [y^{(1)}, \ldots, y^N]^T$$

- Feature vectors:
  $$\bar{f}^{(j)} = [\bar{x}_j^{(1)} \ldots \bar{x}_j^N]^T$$

- Design matrix:
  $\bar{X}$, $\bar{X}_{ij} = \bar{x}_j^{(i)}$
  Each row: EXAMPLE / data pt. (inputs as rows)
  Each column: FEATURE. (each item in row corresp to a feature)
  First column is all 1s.

THM:
The OPTIMAL $\bar{w}$ wrt L_2 loss:
$$w^* = \arg\min \sum_{i=1}^N (y^{(i)} - \bar{w}^T \bar{x}^{(i)})^2, \text{ is } \bar{w}^* = (\bar{X}^T \bar{X})^{-1} \bar{X}^T \bar{y}$$

proof sketch:
- Our predictions vector are:
  $$\hat{\bar{y}} = \bar{X}\bar{w}.$$
- Total loss is:
  $$L(\bar{w}) = \|\bar{y} - \hat{\bar{y}}\|^2 = \|\bar{y} - \bar{X}\bar{w}\|^2$$
- Rewriting: (Lin alg tricks)
  $$L(\bar{w}) = \|\bar{y} - \bar{X}\bar{w}\|^2 = (\bar{y} - \bar{X}\bar{w})^T(\bar{y} - \bar{X}\bar{w}) =$$
  $$\bar{y}^T\bar{y} + \bar{w}^T\bar{X}^T\bar{X}\bar{w} - 2\bar{w}^T\bar{X}^T\bar{y}.$$
  ?
  $$\nabla L(\bar{w}^*) = 2\bar{X}^T\bar{X}w^* - 2\bar{X}^T\bar{y} = 0 \Rightarrow \bar{X}^T\bar{X}\bar{w}^* = \bar{X}^T\bar{y}.$$
  we differentiate w/ respect to $\bar{w}$!

If the features aren't lin ind, $\bar{X}^T\bar{X}$ is invertible.
(use a linear solver to "invert")
go over matrix invertibility condition

Same intuition:

$\hat{\bar{y}} = \bar{X}\bar{w}^*$ our predictions and $\bar{X}^T\bar{X}\bar{w}^* = \bar{X}^T\bar{y}$.

Residual (our mistake):

$r = \bar{y} - \hat{\bar{y}} = \bar{y} - \bar{X}\bar{w}^*$, so $\bar{X}^T r = 0$.

r as a vector is ORTHOGONAL to $\bar{X}^T$.

AKA.

r is ORTHOGONAL to $\bar{f}^{(1)} \ldots \bar{f}^{(d)}$ (zero mean)

Geometrically:

we project $\bar{y}$ to the subspace of features.

Assume features have zero mean. $\sum_j \bar{f}_j^{(i)} = 0$

Thus $[\bar{X}^T\bar{X}]_{ij} = cov(\bar{f}^{(i)}, \bar{f}^{(j)})$ and $[\bar{X}^T, \bar{y}]_j = cov(\bar{f}^{(j)}, \bar{y})$

1) Calculus involved?

2) Zero mean?

$$w_j = \frac{cov(\bar{f}^{(j)}, \bar{y})}{var(\bar{f}^{(j)})}.$$

Scanned by CamScanner