Mid-term exam info

- Mar 16 - In-class exam (in-person)
- Closed books/notes, closed electronics, 2 sided A4 hand-written (not printed) cheat sheet allowed (upload on Gradescope within 24 hrs)
- Academic integrity violations will be severely dealt with

- Hardcopy
- Grading will be curved
- 30% hard (think open book), 70% easy

Topics (everything up to and including CNNs)

(stoch) gradient descent

- Basics - Probability, Matrix/vector calculus, Optimization (convexity etc)
- Basic ML concepts – training vs test data, overfitting, generalization, ML tasks, loss metrics, optimal classifier/regressor, decision boundaries , regression fit

$P(X|Y) \cdot P(Y) = \prod_{i=1}^{d} P(X_i|Y)$ $X = \begin{bmatrix} X_1 \\ \vdots \\ X_d \end{bmatrix}$

- Distribution/Density estimation – MLE, MAP
- Classification – Bayes, Naïve Bayes, Logistic Regression, Neural Networks
- Regression – Linear Regression, Ridge, Lasso, Neural Networks $P(Y|X)$

Comparison chart (classification)

Algorithm	Generative/ Discriminative	Assumptions	Decision boundary	Loss function	Training
Bayes	Generative	Optimal: know $P(X,Y)$ → Gaussian: $P(X\|Y) \sim N(\mu_y, \Sigma_y)$, $P(Y) \sim Bel(\theta)$	Optimal - Any; Gaussian - quad or linear	0/1 loss; -ve log lon	MLE/MAP for parameters
Naïve Bayes	G	$P(X\|Y) = P(X_1 \dots X_d \mid Y) = \prod_{j=1}^{d} P(X_j \mid Y)$ ← conditional independence of features	axis-aligned	~~0/1 loss~~ -ve log loss	MLE/MAP for parameter
Logistic Regression	D	$P(Y\|X) = \frac{1}{1+\exp(-\sum_i w_i x_i)}$	linear	log -ve^conditional likelihood lon	gradient descent/ ascent
Neural Networks	D	$X \xrightarrow[NN]{} Y$	nonlinear	cross-entropy	gradient descent (stochastic)

Comparison (regression)

$E[(Y - f(X))^2]$ MSE

$f(X) = X\beta$

Algorithm	Generative/ Discriminative	Assumptions	Regression fit	Loss function	Training
Linear Regression			$\hat{\beta} = (A^T A)^{-1} A^T Y$	$O(d^3)$	$n \gg d$
Ridge					
Lasso					
Neural Networks					

$d \times d$

features

$A = \begin{bmatrix} X_1 - \\ \vdots \\ X_n \end{bmatrix}$ $n \times d$

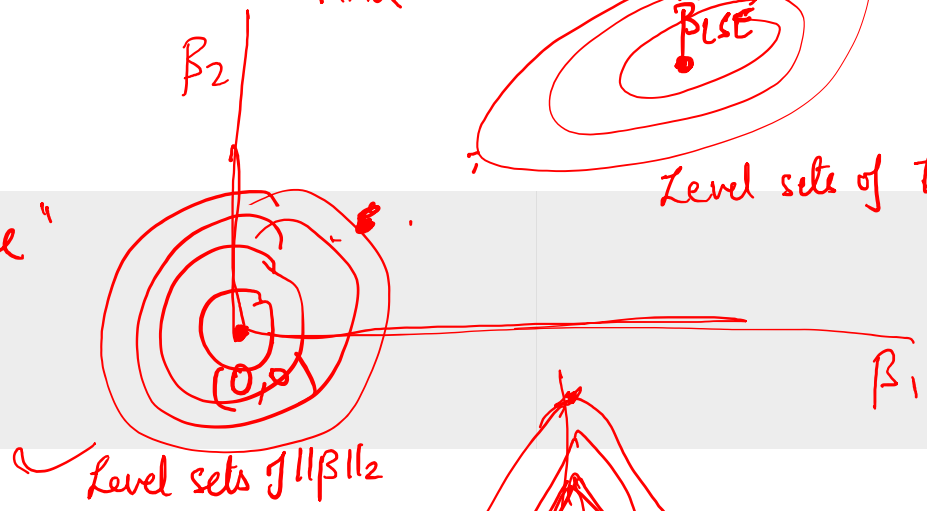
datapts n

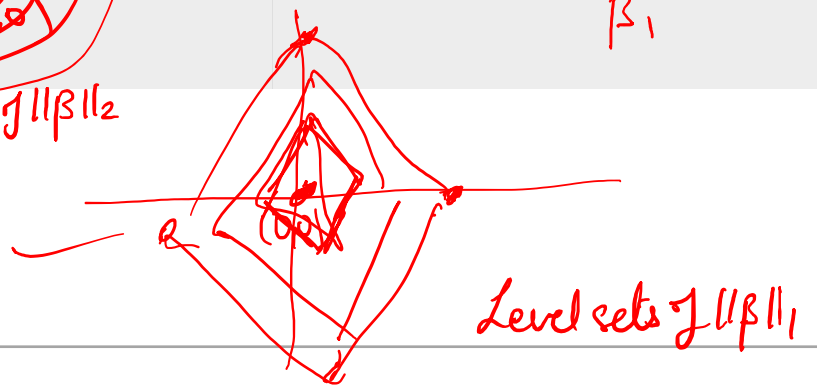
dim d

$\min_{\beta, \beta_0} \sum_i (Y_i - X_i\beta - \beta_0)^2$ LSE

$+ \lambda \|\beta\|_2$

$+ \lambda \|\beta\|_1$ "sparse"





Comparison (regression)

Algorithm	Generative/ Discriminative	Assumptions	Regression fit	Loss function	Training
Linear Regression	D (only know $X \to Y$)		linear in features	squared loss	if A^TA is full-rank, e.g. $n > d$: analytic $\hat{\beta} = (A^TA)^{-1}A^TY$; can also use gradient descent since inverse is expensive*
Ridge	D	regression coefficients are small	linear in features	l_2 penalized squared loss	$\hat{\beta} = (A^TA + \lambda I)^{-1}A^TY$ * grad descent
Lasso	D	regression coefficients are small & sparse ↳ irrelevant features	linear in features	l_1 penalized squared loss	Subgradient descent
Neural Networks	D		nonlinear	squared loss	(stochastic) gradient descent

Practice problems (basics)

- Which of the following expressions is equivalent to P(A,B|C)?

(a) $\frac{P(A,B)}{P(C)}$

(b) $\mathbf{P(A|C)P(B|A,C)}$

(c) $P(C|A,B)P(B|A)P(A)$

(d) $\frac{P(A)P(B|A)P(C|A,B)}{\sum_{a\in A} P(A=a)P(C|A=a)+\sum_{b\in B} P(B=b)P(C|B=b)}$

- The function x^TAx where x is a d-dim vector and A is a dxd matrix is
  - Convex if A is rank 1
  - Convex because its quadratic in x
  - Concave if A has all negative eigenvalues

- Stochastic Gradient descent is faster but uses more memory than regular gradient descent.

False

Practice problems (ML intro)

- When the feature space is larger, overfitting is less likely. False, more parameter to fit

- A classifier that attains 100% accuracy on the training set and 70% accuracy on test set is better than a classifier that attains 70% accuracy on the training set and 75% accuracy on test set. False. test accuracy matters for generalization

- Which of the following are supervised learning tasks:
  - Predicting the proportion of students in class who slept < 7 hours X (no label)
  - Predicting rating of a movie given movie genre ✓
  - Tagging tweets that are racially provocative ✓

Practice problems (density/distribution estimation)

- You have received a shiny new coin and want to estimate the probability θ that it will come up heads if you flip it. A priori you assume that the most probable value of θ is 0.5. You then flip the coin 3 times, and it comes up heads twice. Which will be higher, your maximum likelihood estimate (MLE) of θ, or your maximum a posteriori probability (MAP) estimate of θ?. 2/3

- The maximum likelihood estimate of model parameter α for the random variable $y \sim N(\alpha x_1 x_2^3, \sigma^2)$, where x_1 and x_2 are random variables, can be learned using linear regression on n iid samples of (x_1, x_2, y)

✗

True

$$MLE = \arg\max_{\alpha} \prod_{i=1}^{n} e^{-(y^{(i)} - \alpha x_1^{(i)} x_2^{(i)3})^2} = \arg\min_{\alpha} \sum_{i=1}^{n} (y^{(i)} - \alpha x_1^{(i)} x_2^{(i)3})^2$$

Least square linear regression with nonlinear feature $x_1 x_2^3$

Practice problems (classification)

- Which of the following classifiers can perfectly classify the following dataset

  \+ - +

  - Gaussian Naïve Bayes classifier
  - Logistic regression
  - Neural network

- Choose all that are true:
  - The ReLU is preferred to sigmoid and tanh activation functions as it doesn't saturate neurons leading to faster convergence.
  - Adding an average pooling layer to a neural network with linear activation enables it to represent nonlinear functions.

Practice problems (regression)

- Suppose you wish to predict age of a person from his/her brain scan using regression, but you only have 10 subjects and each subject is represented by the brain activity at 20,000 regions in the brain. You would prefer to use least squares regression instead of ridge regression.

- For polynomial regression, which one of these structural assumptions is the one that most affects the trade-off between underfitting and overfitting: (i) The polynomial degree (ii) Whether we learn the weights by matrix inversion (assuming data matrix is invertible) or gradient descent (iii) The assumed variance of the Gaussian noise