CS189 Midterm Review

Presented by: Arvind Rajaraman, John Ian So

Adapted from SP22 slides by: Ziye Ma, Kumar Krishna Agrawal, Gaurav Rohit Ghosal

Outline

We will be covering a few important topics, but up to Lecture 13 is in scope.
Phase 1: Classification
Phase 2: Linear Regression, Regularization
Phase 3: Bias-Variance Tradeoff

The midterm requires a solid understanding of linear algebra and multivariable calculus. For review of Linear Algebra, please check out the resources from the beginning-of-semester review sessions (on Ed).

Note: **please ask questions!** This session is supposed to be interactive!

Midterm Format

- **Coverage:** up to and including **Lecture 13** (Ridge Regression, March 7)
- **50%** of points will come from **12** multiple choice questions
  - "Select all that apply"-style questions. Mostly conceptual
  - No partial credit
- **50%** from written questions
- Try to justify each selection to yourself, think about edge cases
  - We're not trying to trick you!
- **Note : This is based on previous year's questions, subject to change.**

Classification

Number of features: d, number of points in training set: n,
number of classes: r

Known: There exist r classes, $C_1,...,C_r$ from which data can come.

Given: training feature vectors $X_1,...,X_n$ and labels $y_1,...,y_n$

The label y_i is just the class to which X_i belongs. X_i are d-dimensional vectors

Want to find: Rule that, given a feature vector X, predicts the class from which it was sampled.

How?

Paradigms of classification:

Generative: I want to learn **everything** about the data before I even try to classify.

Model and learn prior probabilities $P(y=C_i)$ and conditional distributions $f(X|y=C_i)$ from the training data
Use them to maximize posterior probabilities $P(y|X)$ or use some other rule

Ex: Gaussian Discriminative Analysis (works for any number of classes)

Discriminative: I want to learn a *few* things before I classify.

Model and learn posterior probabilities $P(y|X)$ from the training data
Use it directly or use some other rule

Ex: Logistic regression (for 2 classes), softmax regression (for multiple classes)

Paradigms of classification:

Pure decision boundary-based: I wanna classssssify

- Choose some form of the rule/boundary (model e.g. perceptron, SVM)
- Choose an appropriate loss/risk function over the given training set (Least squares, Cross-entropy)
- Choose an appropriate optimization algorithm to minimize loss and find the best parameters for the rule/model you chose.

Ex: perceptrons, SVMs

Perceptrons for binary classification (into classes +1, -1)

Model/rule we choose: If

$$X_i \cdot w \geq 0$$ predict class = 1

$$X_i \cdot w \leq 0$$ predict class = -1

Define loss and risk functions:

$$L(z, y_i) = \begin{cases} 0 & \text{if } y_i z \geq 0, \text{ and} \\ -y_i z & \text{otherwise.} \end{cases}$$

$$R(w) = \sum_{i=1}^{n} L(X_i \cdot w, y_i),$$

$$= \sum_{i \in V} -y_i X_i \cdot w$$

Find w* that minimizes this risk over the entire training set

Can use gradient descent or some other suitable algorithm

Gives some linear boundary

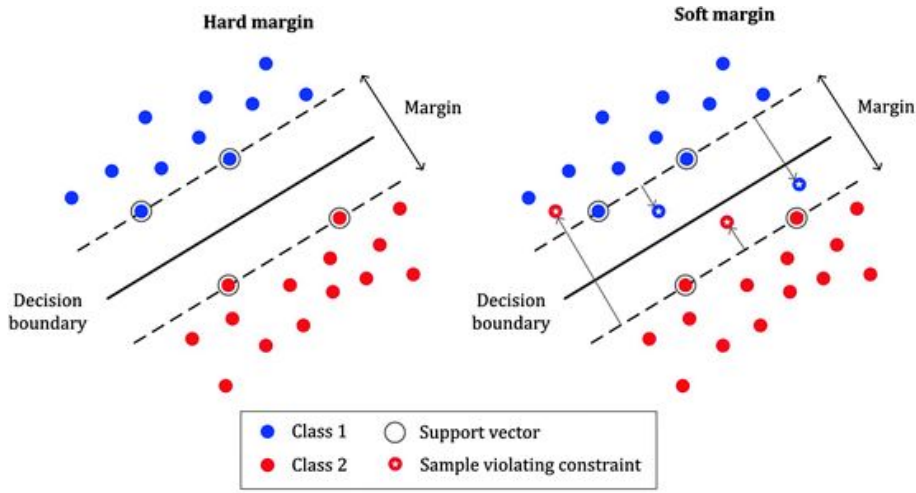
SVMs - hard margin

Want linear decision rules with **room for noise in the data.** Don't want the decision boundary to be too close to any of the training points

Model/rule we choose:

If $X_i \cdot w + \alpha \geq 0$ predict $y_i = 1$

$X_i \cdot w + \alpha \leq 0$ predict $y_i = -1$



Define loss and risk functions:

$$\text{Find } w \text{ and } \alpha \text{ that minimize } |w|^2$$
$$\text{subject to } y_i(X_i \cdot w + \alpha) \geq 1 \quad \text{for all } i \in [1, n]$$

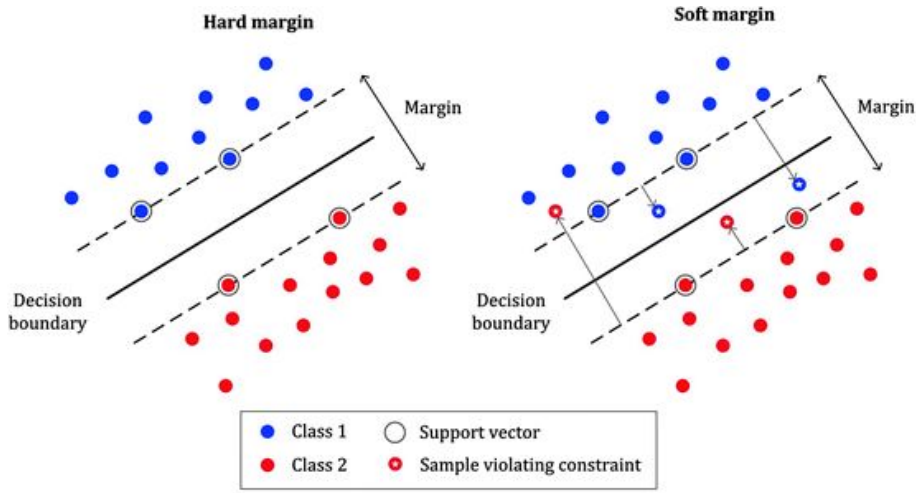
Gives best linear boundary

SVMs - soft margin

Model/rule we choose:

If $X_i \cdot w + \alpha \geq 0$ predict $y_i = 1$

$X_i \cdot w + \alpha \leq 0$ predict $y_i = -1$



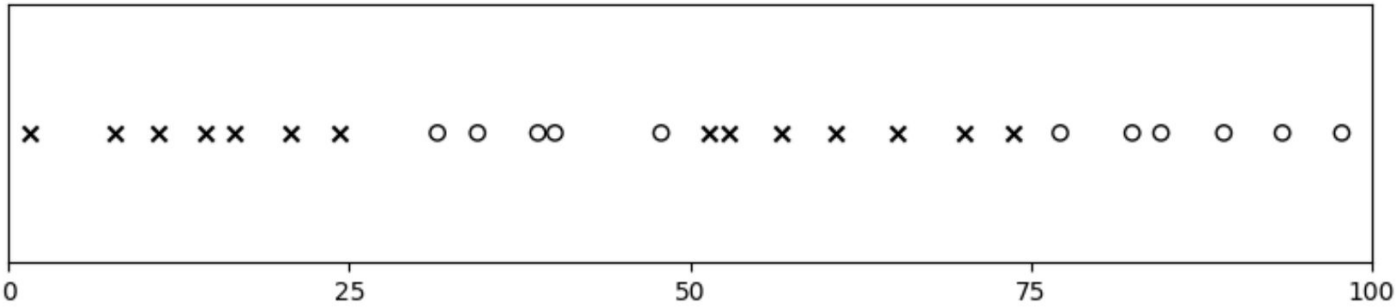
Define loss and risk functions:

$$\text{Find } w, \alpha, \text{ and } \xi_i \text{ that minimize } |w|^2 + C \sum_{i=1}^{n} \xi_i$$
$$\text{subject to} \quad y_i(X_i \cdot w + \alpha) \geq 1 - \xi_i \quad \text{for all } i \in [1, n]$$
$$\xi_i \geq 0 \quad \text{for all } i \in [1, n]$$

Gives best ‘tolerant’ linear boundary

Past exam question Spring 2019

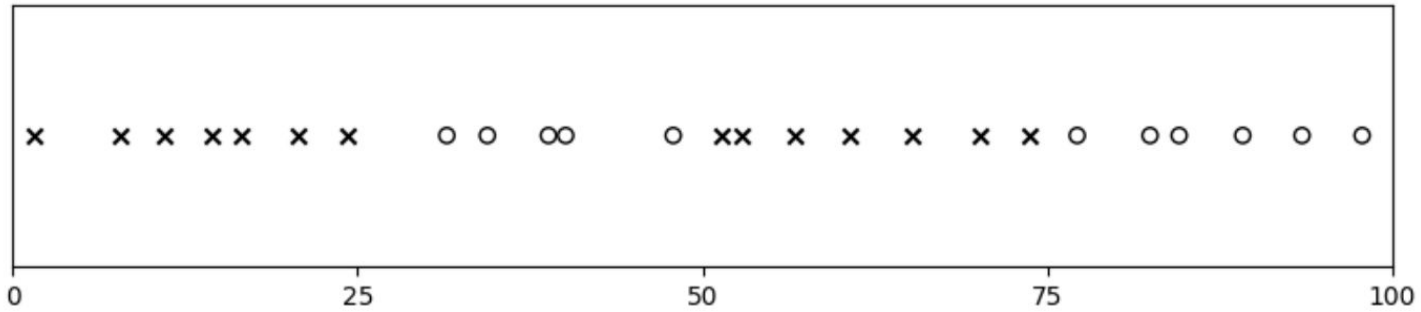
(k) [3 pts] Suppose you are given the one-dimensional data $\{x_1, x_2, \ldots, x_{25}\}$ illustrated below and you have only a **hard-margin support vector machine** (with a fictitious dimension) at your disposal. Which of the following modifications can give you 100% training accuracy?



- ○ Centering the data
- ○ Add a feature x_i^2
- ○ Add a feature that is 1 if $x \leq 50$, or -1 if $x > 50$
- ○ Add two features, x_i^2 and x_i^3

Past exam question Spring 2019

(k) [3 pts] Suppose you are given the one-dimensional data $\{x_1, x_2, \ldots, x_{25}\}$ illustrated below and you have only a **hard-margin support vector machine** (with a fictitious dimension) at your disposal. Which of the following modifications can give you 100% training accuracy?



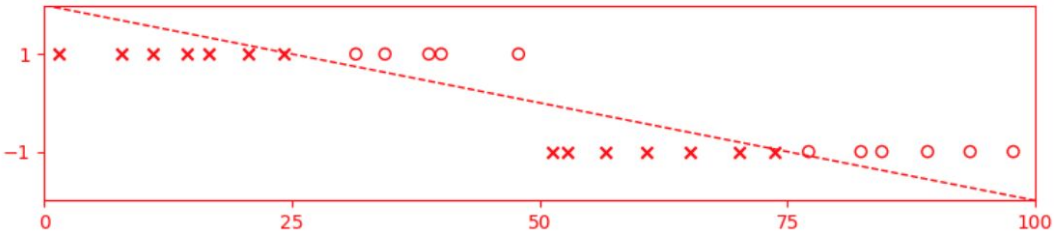
- ○ Centering the data
- ○ Add a feature x_i^2
- ● Add a feature that is 1 if $x \leq 50$, or -1 if $x > 50$
- ● Add two features, x_i^2 and x_i^3

○ The performance of SVM is shift invariant, so centering the data won't affect the result;

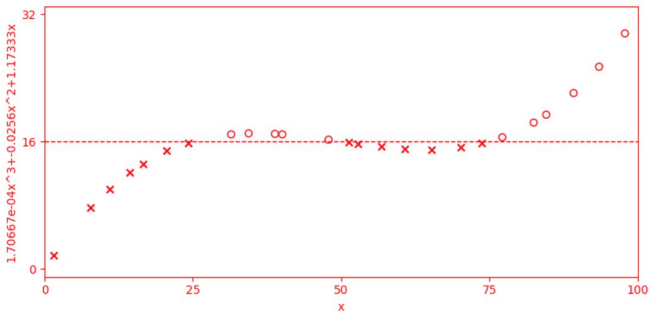
● See image below;

○ A line can separate a quadratic function into at most 3 segments and is not sufficient;

● A line can separate a cubic function into 4 segments. See image below.



Adding "1 if $x_i \leq 50$..." feature



Adding x_i^2 and x_i^3

(r) [3 pts] In the usual formulation of soft-margin SVMs, each training sample has a slack variable $\xi_i \geq 0$ and we impose a regularization cost $C \sum_i \xi_i$. Consider an alternative formulation where we impose the additional constraints $\xi_i = \xi_j$ for all i, j. How does the minimum objective value $|\mathbf{w}|^2 + C \sum_i \xi_i$ obtained by the new method compare to the one obtained by the original soft-margin SVM?

- ◯ They are always equal.
- ◯ Original SVM minimum $\geq$ new minimum.
- ◯ New minimum $\geq$ original SVM minimum.
- ◯ New minimum is sometimes larger and sometimes smaller.

(r) [3 pts] In the usual formulation of soft-margin SVMs, each training sample has a slack variable $\xi_i \geq 0$ and we impose a regularization cost $C \sum_i \xi_i$. Consider an alternative formulation where we impose the additional constraints $\xi_i = \xi_j$ for all i, j. How does the minimum objective value $|\mathbf{w}|^2 + C \sum_i \xi_i$ obtained by the new method compare to the one obtained by the original soft-margin SVM?

- ○ They are always equal.
- ● New minimum $\geq$ original SVM minimum.
- ○ Original SVM minimum $\geq$ new minimum.
- ○ New minimum is sometimes larger and sometimes smaller.

(1) [3 pts] Which the following facts about the 'C' in SVMs is (are) true?

◯ As C approaches 0, the soft margin SVM is equal to the hard margin SVM

◯ C can be negative, as long as each of the slack variables are nonnegative

◯ A larger C tends to create a larger margin

◯ None of the above

(1) [3 pts] Which the following facts about the 'C' in SVMs is (are) true?

◯ As C approaches 0, the soft margin SVM is equal to the hard margin SVM

◯ C can be negative, as long as each of the slack variables are nonnegative

◯ A larger C tends to create a larger margin

● None of the above

Why is this unselected?

A negative C would make the optimization unbounded.

I.e. you could make slack variables arbitrarily large to minimize the objective.

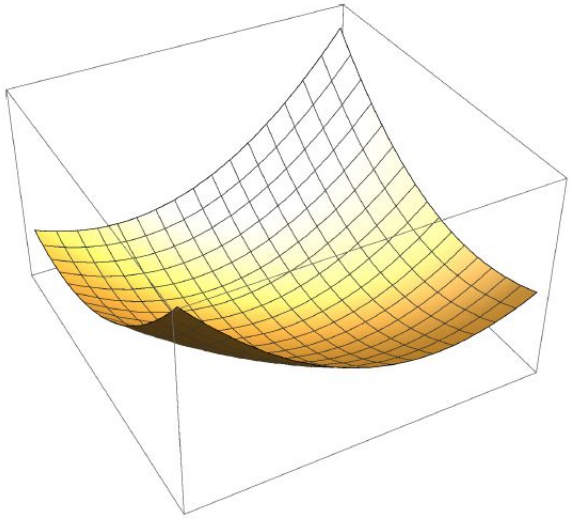
Bayesian risk minimization - Using generative models to predict

$$
\begin{aligned}
R(r) &= \mathrm{E}[L(r(X), Y)] \\
&= \sum_x \Big(L(r(x), 1)\, P(Y = 1 | X = x) + L(r(x), -1)\, P(Y = -1 | X = x) \Big)\, P(X = x)
\end{aligned}
$$

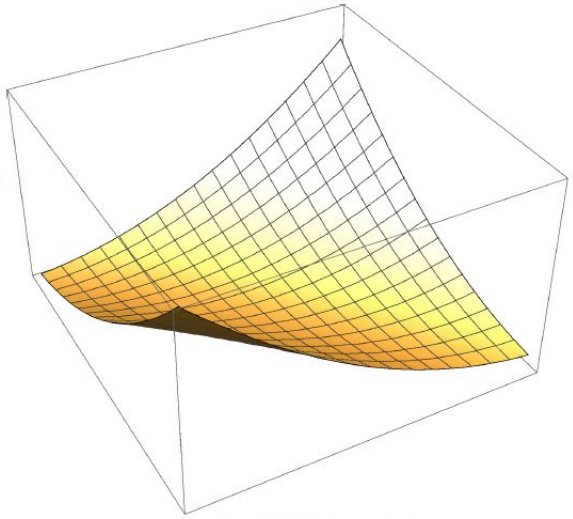
$$
\begin{aligned}
R(r) &= \mathrm{E}[L(r(X), Y)] \\
&= \sum_x \Big(L(r(x), 1)\, P(Y = 1 | X = x) + L(r(x), -1)\, P(Y = -1 | X = x) \Big)\, P(X = x)
\end{aligned}
$$

Linear algebra things to know

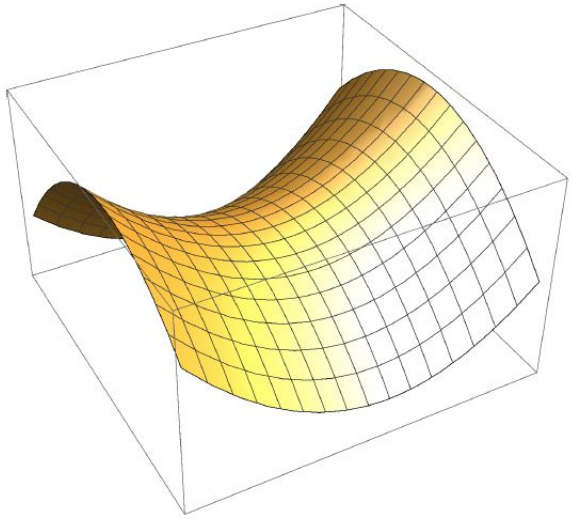
- Fundamental Theorem of Linear Algebra (row space / kernel decomposition)
- Eigendecomposition: Diagonalizable Matrices can be decomposed into $V\Lambda V^{-1}$
  - Λ is a diagonal matrix of eigenvalues, and the columns of V are eigenvectors .
- Spectral Theorem: Symmetric Matrices can be decomposed into $V\Lambda V^{T}$
  - The eigenvectors are always mutually orthogonal.
- Singular Value Decomposition: Generalization of eigendecomposition (see discussion 6; not really in scope).
- Positive semi-definiteness of A (equivalent definitions from HW)
  - $x^{T}Ax >= 0$ for all x
  - All eigenvalues are non-negative
  - $A = UU^{T}$ for some $U \in R^{NxN}$
- Intuition for…
  - spectral decomposition (rotation/reflection, scaling)
  - isocontours (ellipsoids -- always consider diagonal case first)



pos definite



pos semidefinite



indefinite

Graphs of $f(x) = x^T Ax$, where $A \in R^{2 \times 2}$

I.e. ***quadratic forms***

Multivariate Gaussians

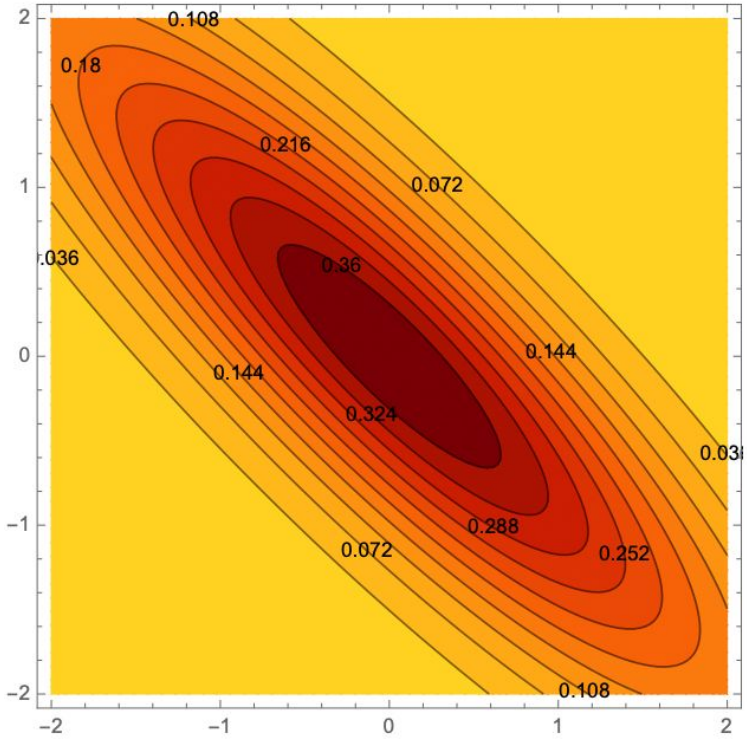
- **Isotropic**: covariance is a scalar multiple of identity:
  - Fun exercise: prove that joint PDF can be factored as product of univariate densities
- **Anisotropic:** anything else.
  - Covariance matrix must be **positive definite** (so we have a well-defined PDF with the inverse)
  - The *MLE estimate* can be positive **semi-definite** (due to linear dependence in data)
- Isocontours are quadratic forms => ellipsoids!

$$\{x \in \mathbb{R}^d : p(x) = c\}$$

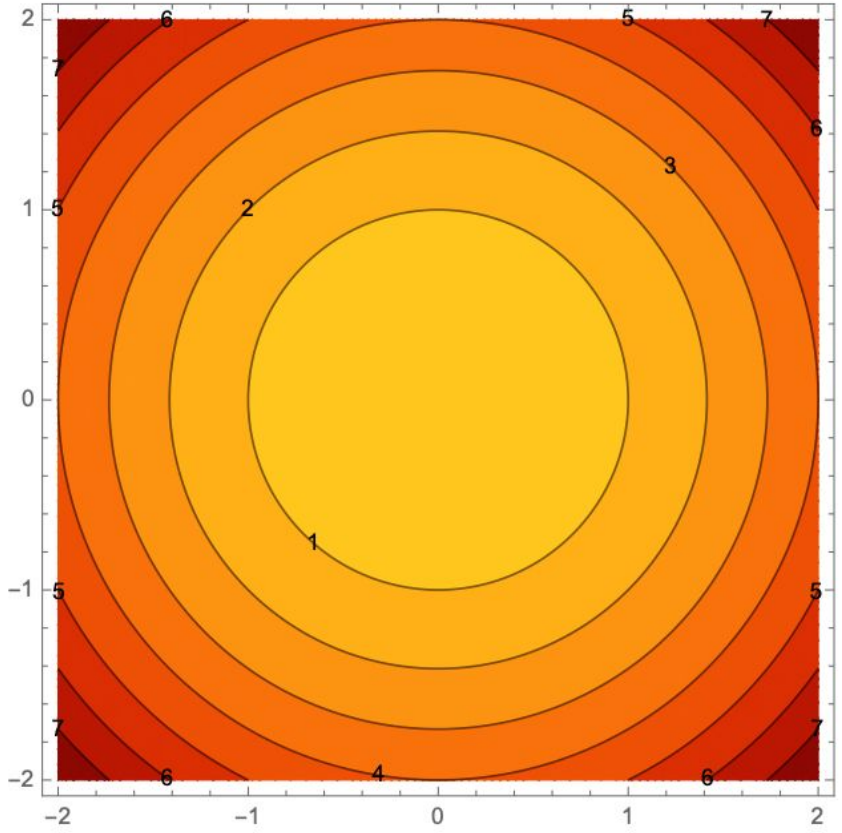
Anisotropic PDF:

$$p(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\}$$

Isocontours of a Gaussian PDF



Anisotropic



Isotropic

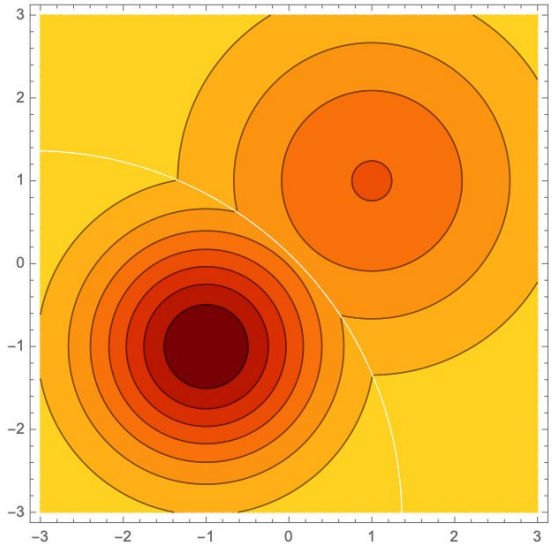
Gaussian Discriminant Analysis (LDA/QDA)

In Gaussian Discriminant Analysis, we assume the likelihood x|y is normally distributed for each class. We use maximum likelihood estimation on training data to estimate the parameters μ_i, $\boldsymbol{\Sigma}_i$ and a prior π_i for each class.

Figure: Contours of the PDFs of two 2D Gaussians.

1. Are these Isotropic or Anisotropic Gaussians?

1. Which one seems to generally have more variance? What is the consequence of this?



MAXIMUM LIKELIHOOD ESTIMATION OF PARAMETERS (Ronald Fisher, circa 1912)

[To use Gaussian discriminant analysis, we must first fit Gaussians to the sample points and estimate the class prior probabilities. We'll do priors first—they're easier, because they involve a discrete distribution. Then we'll fit the Gaussians—they're less intuitive, because they're continuous distributions.]

Let's flip biased coins! Heads with probability p; tails w/prob. $1 - p$.

10 flips, 8 heads, 2 tails. [Let me ask you a weird question.] What is the most likely value of p?

Binomial distribution: $X \sim \mathcal{B}(n, p)$

$$P[X = x] = \binom{n}{x} p^x (1-p)^{n-x} \qquad \text{[this is the probability of getting exactly } x \text{ heads in } n \text{ coin flips]}$$

Our example: $n = 10$,

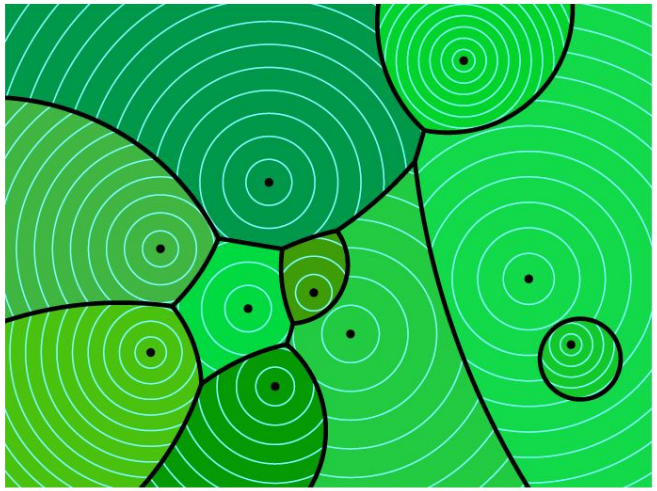
$$P[X = 8] = 45p^8 (1-p)^2 \qquad \stackrel{\text{def}}{=} \mathcal{L}(p)$$

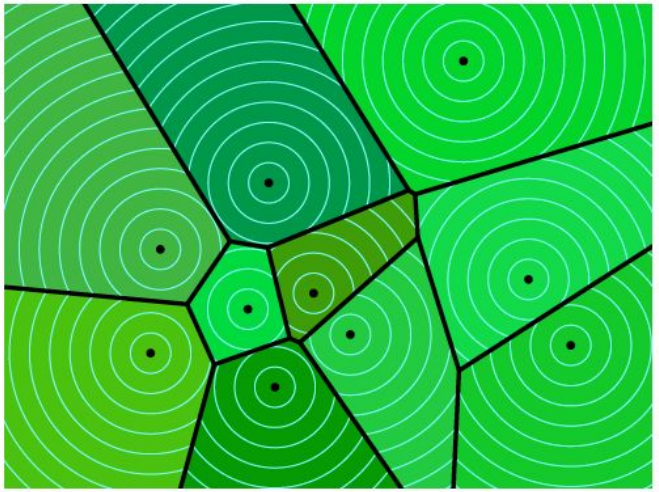
Probability of 8 heads in 10 flips:
written as a fn $\mathcal{L}(p)$ of distribution parameter(s), this is the likelihood fn.

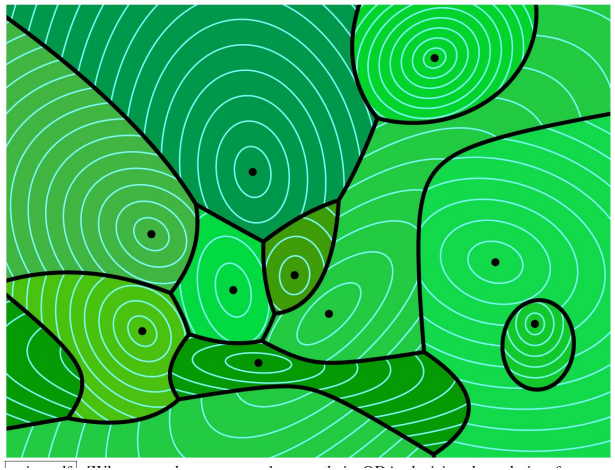
Maximum likelihood estimation (MLE): A method of estimating the parameters of a statistical model by picking the params that maximize [the likelihood function] $\mathcal{L}$.
. . . is one method of density estimation: estimating a PDF [probability density function] from data.

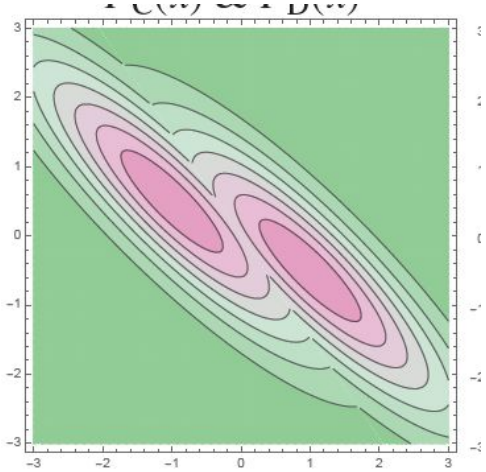
[Let's phrase it as an optimization problem.]

Find p that maximizes $\mathcal{L}(p)$.








3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Q3. [10 pts] Quadratic Discriminant Analysis

(a) [4 pts] Consider 12 labeled data points sampled from three distinct classes:

Class 0: $\begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} -2 \\ 0 \end{bmatrix}, \begin{bmatrix} 5 \\ 3 \end{bmatrix}, \begin{bmatrix} -3 \\ -5 \end{bmatrix}$ Class 1: $\begin{bmatrix} \sqrt{2} \\ \sqrt{2} \end{bmatrix}, \begin{bmatrix} -\sqrt{2} \\ \sqrt{2} \end{bmatrix}, \begin{bmatrix} 4\sqrt{2} \\ -\sqrt{2} \end{bmatrix}, \begin{bmatrix} -4\sqrt{2} \\ -\sqrt{2} \end{bmatrix}$ Class 2: $\begin{bmatrix} 3 \\ 5 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 6 \end{bmatrix}, \begin{bmatrix} 0 \\ -2 \end{bmatrix}$

For each class $C \in \{0, 1, 2\}$, compute the class sample mean μ_C, the class sample covariance matrix Σ_C, and the estimate of the prior probability π_C that a point belongs to class C. (Hint: $\mu_1 = \mu_0$ and $\Sigma_2 = \Sigma_0$.)

Q3. [10 pts] Quadratic Discriminant Analysis

(a) [4 pts] Consider 12 labeled data points sampled from three distinct classes:

Class 0: $\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}-2\\0\end{bmatrix}, \begin{bmatrix}5\\3\end{bmatrix}, \begin{bmatrix}-3\\-5\end{bmatrix}$ Class 1: $\begin{bmatrix}\sqrt{2}\\\sqrt{2}\end{bmatrix}, \begin{bmatrix}-\sqrt{2}\\\sqrt{2}\end{bmatrix}, \begin{bmatrix}4\sqrt{2}\\-\sqrt{2}\end{bmatrix}, \begin{bmatrix}-4\sqrt{2}\\-\sqrt{2}\end{bmatrix}$ Class 2: $\begin{bmatrix}3\\5\end{bmatrix}, \begin{bmatrix}1\\3\end{bmatrix}, \begin{bmatrix}8\\6\end{bmatrix}, \begin{bmatrix}0\\-2\end{bmatrix}$

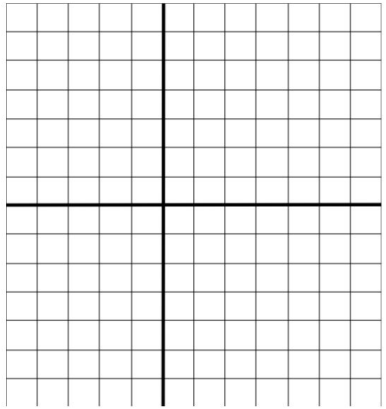
For each class $C \in \{0, 1, 2\}$, compute the class sample mean μ_C, the class sample covariance matrix Σ_C, and the estimate of the prior probability π_C that a point belongs to class C. (Hint: $\mu_1 = \mu_0$ and $\Sigma_2 = \Sigma_0$.)

Class 0: Mean is $\begin{bmatrix}0\\0\end{bmatrix}$, covariance is $\begin{bmatrix}9.5 & 7.5\\7.5 & 9.5\end{bmatrix}$, prior is $\frac{1}{3}$

Class 1: Mean is $\begin{bmatrix}0\\0\end{bmatrix}$, covariance is $\begin{bmatrix}17 & 0\\0 & 2\end{bmatrix}$, prior is $\frac{1}{3}$

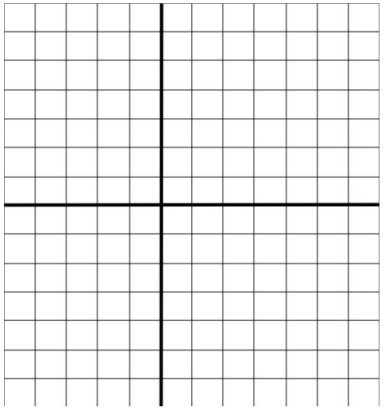
Class 2: Mean is $\begin{bmatrix}3\\3\end{bmatrix}$, covariance is $\begin{bmatrix}9.5 & 7.5\\7.5 & 9.5\end{bmatrix}$, prior is $\frac{1}{3}$

(b) [4 pts] Sketch one or more isocontours of the QDA-produced normal distribution or quadratic discriminant function (they each have the same contours) for each class. The isovalues are not important; the important aspects are the centers, axis directions, and relative axis lengths of the isocontours. Clearly label the centers of the isocontours and to which class they correspond.



(c) [2 pts] Suppose that we apply LDA to classify the data given in part (a). Why will this give a poor decision boundary?

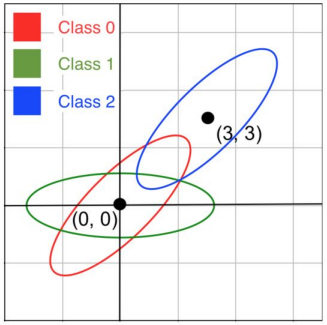
(b) [4 pts] Sketch one or more isocontours of the QDA-produced normal distribution or quadratic discriminant function (they each have the same contours) for each class. The isovalues are not important; the important aspects are the centers, axis directions, and relative axis lengths of the isocontours. Clearly label the centers of the isocontours and to which class they correspond.



The ellipses for classes 0 and 1 both need to be centered around the origin. The ellipses for class 0 should be aligned on a 45 degree rotation of the coordinate axes with more variance along the $[1, 1]$ direction than the $[1, -1]$ direction. The ellipses for class 1 should be axis aligned with more variance along the x-axis. The ellipses for class 2 must be a translation of the ellipses for class 0.

Note: If incorrect covariance matrices were calculated in the first part, full credit on this part should still be possible so long as each ellipse is centered correctly around the appropriate mean and the variance is in the appropriate directions.

(c) [2 pts] Suppose that we apply LDA to classify the data given in part (a). Why will this give a poor decision boundary?



The discriminant functions for classes 0 and 1 would have the exact same mean and covariance, so there would be no decision boundary between them.

(s) [3 pts] Suppose you have a sample in which each point has d features and comes from class C or class D. The class conditional distributions are $(X_i|y_i = C) \sim N(\mu_C, \sigma_C^2)$ and $(X_i|y_i = D) \sim N(\mu_D, \sigma_D^2)$ for unknown values $\mu_C, \mu_D \in \mathbb{R}^d$ and $\sigma_C^2, \sigma_D^2 \in \mathbb{R}$. The class priors are π_C and π_D. We use 0-1 loss.

○ If $\pi_C = \pi_D$ and $\sigma_C = \sigma_D$, then the Bayes decision rule assigns a test point z to the class whose mean is closest to z.

○ If $\pi_C = \pi_D$, then the Bayes decision rule is $r^*(z) = \text{argmin}_{A \in \{C,D\}} \left(|z - \mu_A|^2/(2\sigma_A^2) + d \ln \sigma_A \right)$

○ If $\sigma_C = \sigma_D$, then the Bayes decision boundary is always linear.

○ If $\sigma_C = \sigma_D$, then QDA will always produce a linear decision boundary when you fit it to your sample.

(s) [3 pts] Suppose you have a sample in which each point has d features and comes from class C or class D. The class conditional distributions are $(X_i|y_i = C) \sim N(\mu_C, \sigma_C^2)$ and $(X_i|y_i = D) \sim N(\mu_D, \sigma_D^2)$ for unknown values $\mu_C, \mu_D \in \mathbb{R}^d$ and $\sigma_C^2, \sigma_D^2 \in \mathbb{R}$. The class priors are π_C and π_D. We use 0-1 loss.

● If $\pi_C = \pi_D$ and $\sigma_C = \sigma_D$, then the Bayes decision rule assigns a test point z to the class whose mean is closest to z.

● If $\pi_C = \pi_D$, then the Bayes decision rule is $r^*(z) = \text{argmin}_{A \in \{C,D\}} \left(|z - \mu_A|^2 / (2\sigma_A^2) + d \ln \sigma_A \right)$

● If $\sigma_C = \sigma_D$, then the Bayes decision boundary is always linear.

○ If $\sigma_C = \sigma_D$, then QDA will always produce a linear decision boundary when you fit it to your sample.

$$\ln\left((\sqrt{2\pi})^d f_C(x) \pi_C \right) = -\frac{\|x - \mu_C\|^2}{2\sigma_C^2} - d \ln \sigma_C + \ln \pi_C$$

This is a constant, and constant affine transformations on an optimization problem does not change the optimal solution.

(f) [3 pts] The Bayes risk for a decision problem is zero when

- ◯ the class distributions $P(X|Y)$ do not overlap.
- ◯ the loss function $L(z, y)$ is symmetrical.
- ◯ the training data is linearly separable.
- ◯ the Bayes decision rule perfectly classifies the training data.

$$
\begin{aligned}
R(r) &= \mathrm{E}[L(r(X), Y)] \\
&= \sum_x \Big(L(r(x), 1)\, P(Y = 1|X = x) + L(r(x), -1)\, P(Y = -1|X = x) \Big)\, P(X = x)
\end{aligned}
$$

(f) [3 pts] The Bayes risk for a decision problem is zero when

- ● the class distributions $P(X|Y)$ do not overlap.
- ○ the training data is linearly separable.
- ○ the loss function $L(z, y)$ is symmetrical.
- ○ the Bayes decision rule perfectly classifies the training data.

$$
\begin{aligned}
R(r) &= \mathrm{E}[L(r(X), Y)] \\
&= \sum_x \Big(L(r(x), 1)\, P(Y = 1|X = x) + L(r(x), -1)\, P(Y = -1|X = x) \Big)\, P(X = x)
\end{aligned}
$$

(h) [3 pts] Gaussian discriminant analysis

- ◯ models $P(Y = y|X)$ as a Gaussian.
- ◯ models $P(Y = y|X)$ as a logistic function.
- ◯ is an example of a generative model.
- ◯ can be used to classify points without ever computing an exponential.

(h) [3 pts] Gaussian discriminant analysis

- ○ models $P(Y = y|X)$ as a Gaussian.
- ● models $P(Y = y|X)$ as a logistic function.
- ● is an example of a generative model.
- ● can be used to classify points without ever computing an exponential.

$$Q_C(x) = \ln\left((\sqrt{2\pi})^d f_C(x)\pi_C\right) = -\frac{\|x-\mu_C\|^2}{2\sigma_C^2} - d\ln\sigma_C + \ln\pi_C$$

↑ quadratic in x. ↑ normal PDF, estimates $f(X = x|Y = C)$

$$P(Y = C|X) = \frac{f(X|Y = C)\pi_C}{f(X|Y = C)\pi_C + f(X|Y = D)\pi_D}$$

recall $e^{Q_C(x)} = (\sqrt{2\pi})^d f_C(x)\pi_C$ [by definition of Q_C]

$$\begin{aligned} P(Y = C|X = x) &= \frac{e^{Q_C(x)}}{e^{Q_C(x)} + e^{Q_D(x)}} = \frac{1}{1 + e^{Q_D(x)-Q_C(x)}} \\ &= s(Q_C(x) - Q_D(x)), \quad \text{where} \end{aligned}$$

$$s(\gamma) = \frac{1}{1+e^{-\gamma}} \quad \Leftarrow \underline{\text{logistic fn}} \text{ aka } \underline{\text{sigmoid fn}}$$

[recall $Q_C - Q_D$ is the decision fn]

(m) [3 pts] In LDA/QDA, what are the effects of modifying the sample covariance matrix as $\tilde{\Sigma} = (1 - \lambda)\Sigma + \lambda I$, where $0 < \lambda < 1$?

- ◯ $\tilde{\Sigma}$ is positive definite
- ◯ $\tilde{\Sigma}$ is invertible
- ◯ Increases the eigenvalues of Σ by λ
- ◯ The isocontours of the quadratic form of $\tilde{\Sigma}$ are closer to spherical

(m) [3 pts] In LDA/QDA, what are the effects of modifying the sample covariance matrix as $\tilde{\Sigma} = (1 - \lambda)\Sigma + \lambda I$, where $0 < \lambda < 1$?

- [x] $\tilde{\Sigma}$ is positive definite
- [x] $\tilde{\Sigma}$ is invertible
- [ ] Increases the eigenvalues of Σ by λ
- [x] The isocontours of the quadratic form of $\tilde{\Sigma}$ are closer to spherical

unit norm spheres for isocontours

Covariance matrices are always at least PSD.

$$x^\intercal \tilde{\Sigma} x = x^\intercal ((1-\lambda)\Sigma + \lambda I)x$$

$$= x^\intercal (1-\lambda)\Sigma x + x^\intercal \lambda I x$$

$$= (1-\lambda)\underbrace{x^\intercal \Sigma x}_{\geq 0} + \lambda \underbrace{||x||_2^2}_{>0}$$

$$> 0 \implies \hat{\Sigma} \texttt{ is PD! } \implies \hat{\Sigma} \texttt{ is invertible}$$

Regression

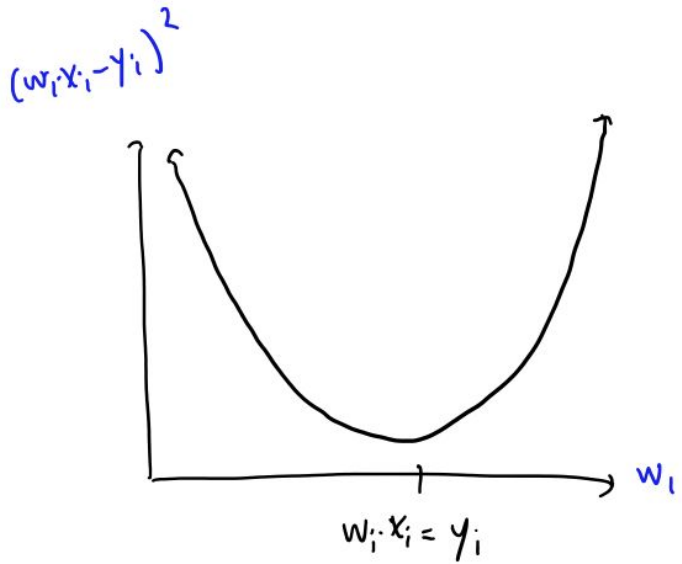
Linear Regression

Find w that minimizes $||Xw - y||^2 = RSS(w)$ ← Residual Sum of Squares

This is the sum of $(x_i^T w - y_i)^2$ over i = 1,...,n

$(x_i^T w - y_i)^2$ looks like a parabolic cylinder.

If w is 1D, it looks like a parabola.



Linear Regression

$\nabla_w \mathbf{RSS(w)} = 0$

$\rightarrow \ 2X^TXw - 2X^Ty = 0$

$\rightarrow X^TXw = X^Ty$ **[Normal Equations]**

- If X^TX is invertible, unique solution
  - colspace(X^TX) = rowspace(X)
  - datapoints must span all of w-space (R^d).
- Else, infinite number of solutions.
  - Use *Moore-Penrose* pseudoinverse (Dis6, Q1)

$\nabla^2 \mathbf{RSS(w)} = 2X^TX$

- All eigenvalues are non-negative. Notice for any v in R^d, $v^TX^TXv = (Xv)^2 \geq 0$.
- Notice 0 eigenvalue if datapoints do not span all of R^d.
- Thus, the function is **convex**.

Regularization

L1 regularization (Lasso)

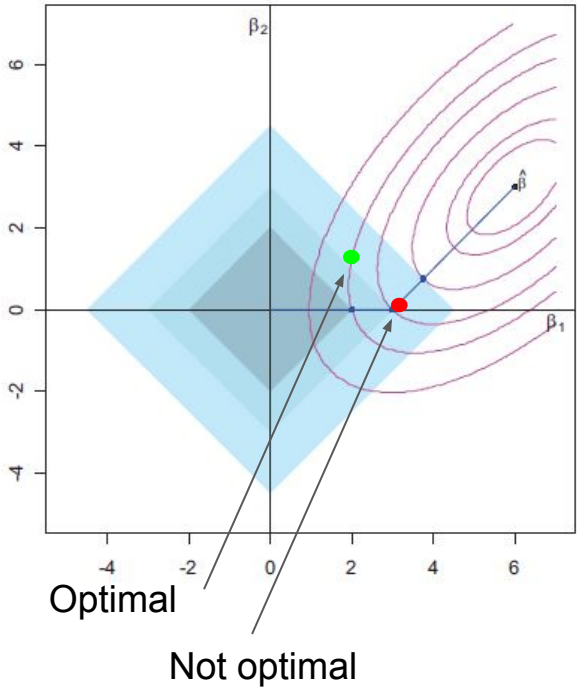
- "Shrinkage" goal: ***eliminate*** unimportant features

$$r(w) = \lambda \|w\|_1 = \sum_{i=1}^{d} |w_i|$$

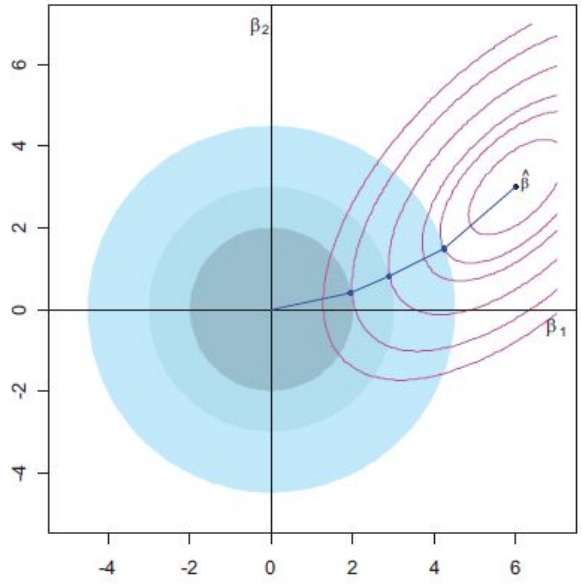
Visualizing Regularizers

Objective is now to minimize Loss + Regularization

Lasso (L1)



Ridge Regression (L2)



L2 Regularization

Recall least-squares regression can be thought of as *maximum likelihood estimation*.

Suppose $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$; then $y_i \sim \mathcal{N}(g(X_i), \sigma^2)$

Recall that log likelihood of normal PDF is

$$\ln f(y_i) = -\frac{(y_i - \mu)^2}{2\sigma^2} - \text{constant} \qquad \Leftarrow \mu = g(X_i)$$

$$\ell(g; X, y) = \ln\left(f(y_1)\, f(y_2) \cdots f(y_n)\right) = \ln f(y_1) + \ldots + \ln f(y_n) = -\frac{1}{2\sigma^2} \sum (y_i - g(X_i))^2 - \text{constant}$$

Maximizing likelihood **P(y | X,w)** = find g that minimizes least-squares loss

L2 Regularization

L2 regularization can be thought of as *maximum a posteriori* estimation where we assume a Gaussian prior over weights. We maximize **P(w | X,y)** instead of P(y | X,w)

Assign a prior probability on w': $w' \sim \mathcal{N}(0, \sigma^2)$. Apply MLE to the posterior prob.

[This prior probability says that we think weights close to zero are more likely to be correct.]

$$\text{Bayes' Theorem: posterior } f(w|X,y) = \frac{f(y|X,w) \times \text{ prior } f(w')}{f(y|X)} = \frac{\mathcal{L}(w)\, f(w')}{f(y|X)}$$

$$\text{Maximize log posterior} = \ln \mathcal{L}(w) + \ln f(w') - \text{const}$$

$$= -\text{const}\, ||Xw - y||^2 - \text{const}\, ||w'||^2 - \text{const}$$

$$\Rightarrow \text{Minimize } ||Xw - y||^2 + \lambda\, ||w'||^2$$

L2 regularization

$$\text{Gaussian prior} \quad p(w) \propto \exp\{|w|^2\}$$
$$\implies r(w) = \lambda |w|^2$$

- **Tikhonov regularization (ridge regression)**: apply this term to the least squares objective. We can still solve for a closed form solution (we can now guarantee invertibility):

$$(X^T X + \lambda I)^{-1} X^T y$$

(n) [3 pts] Given a design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and labels $\mathbf{y} \in \mathbb{R}^n$, which of the following techniques could potentially decrease the empirical risk on the training data (assuming the loss is the squared error)?

◯ Adding the feature "1" to each data point.

◯ Centering the vector $\mathbf{y}$ by subtracting the mean $\bar{y}$ from each component y_i.

◯ Adding polynomial features to each data point.

◯ Penalizing the model weights with L_2 regularization.

(n) [3 pts] Given a design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and labels $\mathbf{y} \in \mathbb{R}^n$, which of the following techniques could potentially decrease the empirical risk on the training data (assuming the loss is the squared error)?

- [x] Adding the feature "1" to each data point.
- [x] Centering the vector $\mathbf{y}$ by subtracting the mean $\bar{y}$ from each component y_i.
- [x] Adding polynomial features to each data point.
- [ ] Penalizing the model weights with L_2 regularization.

(p) [3 pts] You have a design matrix $X \in \mathbb{R}^{n \times d}$ with $d = 100{,}000$ features and and vector $y \in \mathbb{R}^n$ of binary 0-1 labels. When you fit a logistic regression model to your design matrix, your test error is much worse than your training error. You suspect that many of the features are useless and are therefore causing overfitting. What are some ways to eliminate the useless features?

- ○ Use ℓ_1 regularization.
- ○ Use ℓ_2 regularization.
- ○ Iterate over features; check if removing feature i increases validation error; remove it if not.
- ○ If the ith eigenvalue λ_i of the sample covariance matrix is 0, remove the ith feature/column.

(p) [3 pts] You have a design matrix $X \in \mathbb{R}^{n \times d}$ with $d = 100{,}000$ features and and vector $y \in \mathbb{R}^n$ of binary 0-1 labels. When you fit a logistic regression model to your design matrix, your test error is much worse than your training error. You suspect that many of the features are useless and are therefore causing overfitting. What are some ways to eliminate the useless features?

- [x] Use ℓ_1 regularization.
- [ ] Use ℓ_2 regularization.
- [x] Iterate over features; check if removing feature i increases validation error; remove it if not.
- [ ] If the ith eigenvalue λ_i of the sample covariance matrix is 0, remove the ith feature/column.

Bias/Variance

The Model

We have a data-generating distribution D (so that $x_i \sim D$), a noise distribution N (so that $\varepsilon_i \sim N$) with zero mean, and an underlying function g. [This is all theoretical, we only get samples of (x_i, y_i)]

We then model our output $y_i = g(x_i) + \varepsilon_i$ [This gives us a complete model for data]

Now, we fit a hypothesis h to our data [for example weight vector in Lin. Reg]

Observe that our training points are random variables and thus h is a random variable as well [Functions of r.vs are r.vs] $h \sim H$ [to be explicit]

The Model (2)

Finally, suppose we select a test point $z \sim D$ [not necessarily a training point] with label y.

Consider $E_h[L(h(z), y)]$

We will study this term in the bias-variance decomposition.

Why does it make sense to study this particular object?

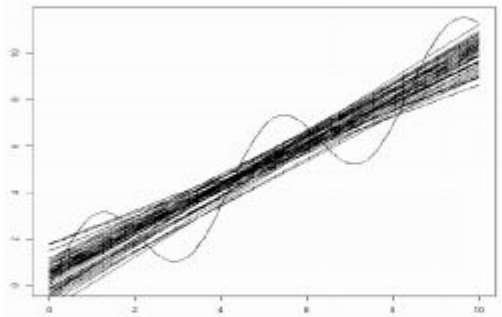
We don't get to choose the data points we get, so we want to figure out how well we do on average as we sweep through all possible data points.

We use a separate test z in order to test how well we generalize.

$$\mathrm{Bias}_D\left[\hat{f}\left(x;D\right)\right] = \mathrm{E}_D\left[\hat{f}\left(x;D\right)\right] - f(x)$$

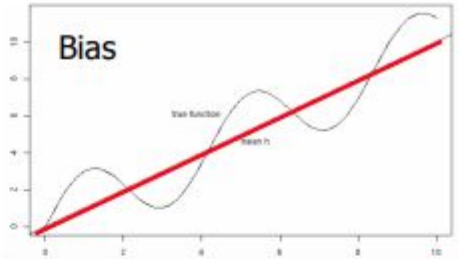
$$\mathrm{Var}_D\left[\hat{f}\left(x;D\right)\right] = \mathrm{E}_D[\hat{f}\left(x;D\right)^2] - \mathrm{E}_D[\hat{f}\left(x;D\right)]^2$$

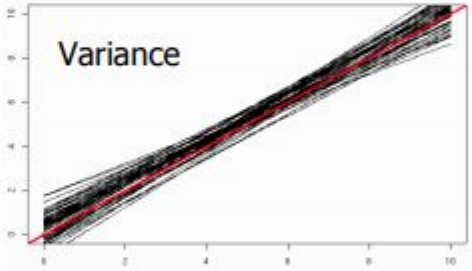
$$
\begin{aligned}
\mathrm{E}\left[(y-\hat{f})^2\right] &= \mathrm{E}\left[(f+\varepsilon-\hat{f})^2\right] \\
&= \mathrm{E}\left[(f+\varepsilon-\hat{f}+\mathrm{E}[\hat{f}]-\mathrm{E}[\hat{f}])^2\right] \\
&= \mathrm{E}\left[(f-\mathrm{E}[\hat{f}])^2\right] + \mathrm{E}[\varepsilon^2] + \mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})^2\right] + 2\,\mathrm{E}\left[(f-\mathrm{E}[\hat{f}])\varepsilon\right] + 2\,\mathrm{E}\left[\varepsilon(\mathrm{E}[\hat{f}]-\hat{f})\right] + 2\,\mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})(f-\mathrm{E}[\hat{f}])\right] \\
&= (f-\mathrm{E}[\hat{f}])^2 + \mathrm{E}[\varepsilon^2] + \mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})^2\right] + 2(f-\mathrm{E}[\hat{f}])\,\mathrm{E}[\varepsilon] + 2\,\mathrm{E}[\varepsilon]\,\mathrm{E}\left[\mathrm{E}[\hat{f}]-\hat{f}\right] + 2\,\mathrm{E}\left[\mathrm{E}[\hat{f}]-\hat{f}\right](f-\mathrm{E}[\hat{f}]) \\
&= (f-\mathrm{E}[\hat{f}])^2 + \mathrm{E}[\varepsilon^2] + \mathrm{E}\left[(\mathrm{E}[\hat{f}]-\hat{f})^2\right] \\
&= (f-\mathrm{E}[\hat{f}])^2 + \mathrm{Var}[\varepsilon] + \mathrm{Var}\left[\hat{f}\right] \\
&= \mathrm{Bias}[\hat{f}]^2 + \mathrm{Var}[\varepsilon] + \mathrm{Var}\left[\hat{f}\right] \\
&= \mathrm{Bias}[\hat{f}]^2 + \sigma^2 + \mathrm{Var}\left[\hat{f}\right]
\end{aligned}
$$

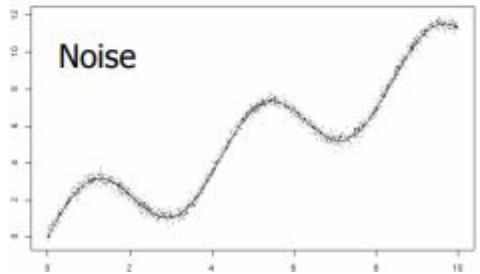


50 fits (20 examples each)

=







(j) [3 pts] Which of the following are reasons why you might adjust your model in ways that increase the bias?

- ◯ You observe high training error and high validation error
- ◯ You observe low training error and high validation error
- ◯ You have few data points
- ◯ Your data are not linearly separable

(l) [3 pts] You are performing **least-squares polynomial regression**. As the degree of your polynomials increases, which of the following is commonly seen to go down at first but then go up?

- ◯ Training error
- ◯ Variance
- ◯ Validation error
- ◯ Bias

(j) [3 pts] Which of the following are reasons why you might adjust your model in ways that increase the bias?

- ○ You observe high training error and high validation error
- ● You observe low training error and high validation error
- ● You have few data points
- ○ Your data are not linearly separable

(l) [3 pts] You are performing **least-squares polynomial regression**. As the degree of your polynomials increases, which of the following is commonly seen to go down at first but then go up?

- ○ Training error
- ○ Variance
- ● Validation error
- ○ Bias

Q4. [10 pts] Ridge Regression with One Feature

We are given a sample in which each point has only one feature. Therefore, our design matrix is a column vector, which we will write $x \in \mathbb{R}^n$ (instead of X). Consider the scalar data generation model

$$y_i = \omega x_i + e_i$$

where $x_i \in \mathbb{R}$ is point i's sole input feature, $y_i \in \mathbb{R}$ is its scalar label (a noisy measurement), $e_i \sim \mathcal{N}(0, 1)$ is standard unit-variance zero-mean Gaussian noise, and $\omega \in \mathbb{R}$ is the true, fixed linear relationship that we would like to estimate. The e_i's are independent and identically distributed random variables, and the sole source of randomness. We will treat the design vector x as fixed (not random).

Our goal is to fit a linear model and get an estimate w_λ for the true parameter ω. The ridge regression estimate for ω is

$$w_\lambda = \text{argmin}_{w \in \mathbb{R}} \left(\lambda w^2 + \sum_{i=1}^{n} (y_i - x_i w)^2 \right) \qquad \text{where } \lambda \geq 0.$$

(a) [4 pts] Express w_λ in terms of λ, S_{xx} and S_{xy}, where $S_{xx} = \sum_{i=1}^{n} x_i^2$ and $S_{xy} = \sum_{i=1}^{n} x_i y_i$.

(a) [4 pts] Express w_λ in terms of λ, S_{xx} and S_{xy}, where $S_{xx} = \sum_{i=1}^{n} x_i^2$ and $S_{xy} = \sum_{i=1}^{n} x_i y_i$.

$$\frac{\partial}{\partial w}\left(\lambda w^2 + \sum_{i=1}^{n}(y_i - x_i w)^2\right) = 2\lambda w - 2\sum_{i=1}^{n} y_i x_i + 2\sum_{i=1}^{n} x_i^2 w$$

Setting the derivative to 0, we have

$$w_\lambda = \frac{S_{xy}}{S_{xx} + \lambda}.$$

(b) [5 pts] Compute the squared bias of the ridge estimate $w_\lambda z$ at a test point $z \in \mathbb{R}$, defined to be

$$\text{bias}^2(w_\lambda, z) = (\mathbb{E}[w_\lambda z] - \omega z)^2,$$

where the expectation is taken with respect to the y_i's. Express your result in terms of ω, λ, S_{xx}, and z. (Hint: simplify the expectation first.)

(b) [5 pts] Compute the squared bias of the ridge estimate $w_\lambda z$ at a test point $z \in \mathbb{R}$, defined to be

$$\text{bias}^2(w_\lambda, z) = (\mathbb{E}[w_\lambda z] - \omega z)^2,$$

where the expectation is taken with respect to the y_i's. Express your result in terms of ω, λ, S_{xx}, and z. (Hint: simplify the expectation first.)

$$\begin{aligned}
\mathbb{E}[w_\lambda] &= \frac{\mathbb{E}[\sum_{i=1}^n x_i y_i]}{S_{xx} + \lambda} \\
&= \frac{\mathbb{E}[\sum_{i=1}^n x_i(\omega x_i + e_i)]}{S_{xx} + \lambda} \\
&= \frac{\omega S_{xx} + \sum_{i=1}^n x_i \cancel{\mathbb{E}[e_i]}}{S_{xx} + \lambda} \\
&= \frac{\omega S_{xx}}{S_{xx} + \lambda}.
\end{aligned}$$

Therefore,

$$(\mathbb{E}[w_\lambda z] - \omega z)^2 = (\mathbb{E}[w_\lambda] - \omega)^2 z^2 = \left(-\frac{\omega\lambda}{S_{xx} + \lambda}\right)^2 z^2 = \frac{\omega^2\lambda^2}{(S_{xx} + \lambda)^2} z^2.$$

(c) [1 pt] What will the bias be if we are using ordinary least squares, i.e., $\lambda = 0$?

(c) [1 pt] What will the bias be if we are using ordinary least squares, i.e., $\lambda = 0$?

The bias is zero if $\lambda = 0$.