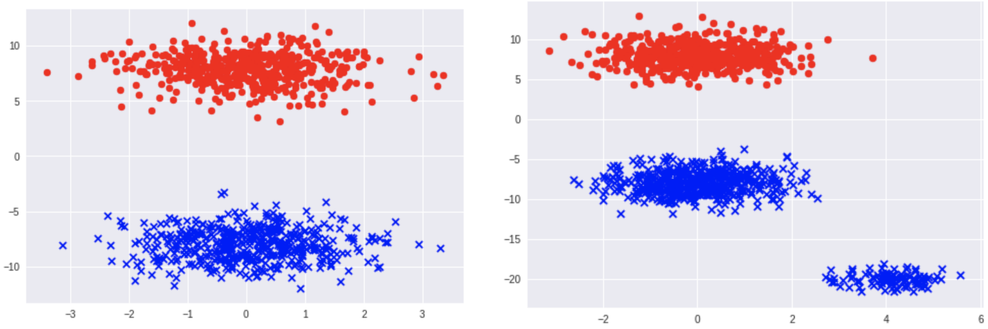
CS 189 Introduction to Machine Learning
Spring 2021 Jonathan Shewchuk

DIS7

1 Logistic Regression

In this problem, we will explore logistic regression and derive some insights.

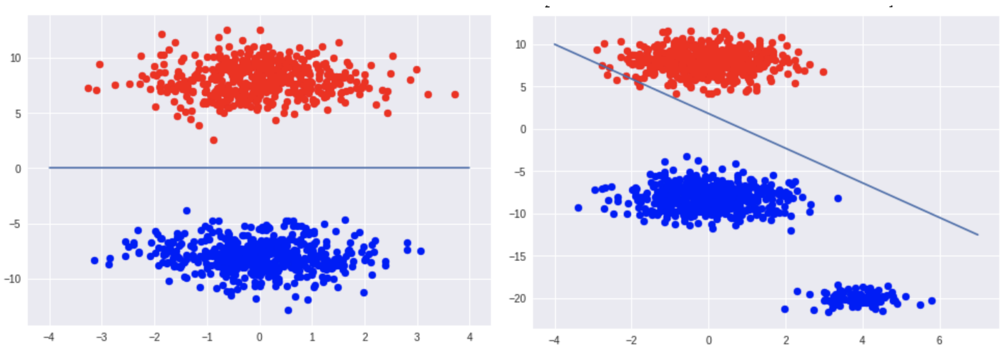
In parts (a)–(b), we will motivate the need for logistic regression. Assume you are given the following datasets, where the red circles are one class (with label −1), and the blue X's are the other class (with label +1).



(a) First, suppose we are using *least-squares linear regression* to find a decision boundary that separates the two classes, where sign($\mathbf{w}^T\mathbf{x}$) represents the classifier function. (Note that linear regression is not actually a good classification method, but we're going to briefly consider it anyway.) Draw an the decision boundary for the datasets above. Recall that the optimization problem has the form

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n}(\mathbf{w}^\top\mathbf{x}_i - y_i)^2 + \lambda\|\mathbf{w}\|_2^2.$$

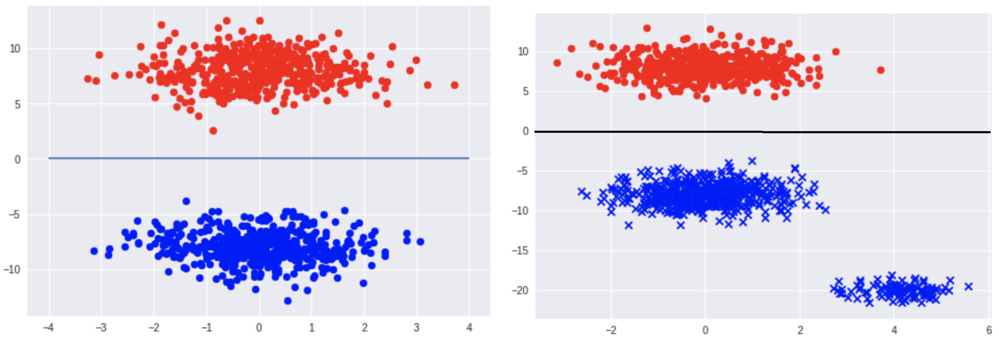
Solution:



During the optimization process, the magnitude of $\mathbf{w}^\top\mathbf{x}$ is used, but we will classify a point based on $\text{sign}(\mathbf{w}^\top\mathbf{x})$. Notice that even though the decision boundary on the first dataset would be valid for the rest, because there is a cluster of points in the bottom right corner the magnitude of the error would be higher for those points, pulling the decision boundary down.

(b) Draw the ideal decision boundary for the dataset above.

Solution:



The ideal decision boundary on the left dataset should be intuitive. Now, suppose we add a cluster of points to the left dataset to get the right dataset. Ideally we wouldn't want the decision boundary to change once we add the points in the right bottom corner, since (1) they are far away from the general mass of points, and thus outliers of the dataset, and (2) all points in that blob would still be correctly classified under the first decision boundary. Generally speaking, we don't want points that are far from the decision boundary to drastically alter it. As we can see, with *least-squares linear regression*, due to the squared-term in the objective, points that are far away have a quadratic effect on the resulting decision boundary, when ideally it would have little to no impact.

In parts (c)–(e), we will show mathematically how logistic regression tackles the issues present in least-squares linear regression, as seen above. Specifically, we will show that in logistic regression the decision boundary is less likely to be influenced by outliers of the dataset.

(c) Assume your data comes from two classes and the prior for class k is $p(y = k) = \pi_k$. Also the conditional probability distribution for each class k is Gaussian, $\mathbf{x}|(y = k) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$, that is $f_k(\mathbf{x}) = f(\mathbf{x}|y = k) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}}\exp\Big((\mathbf{x} - \boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\Big)$. Assume that $\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_1$, and $\boldsymbol{\Sigma}$ were estimated from the training data.

Show that $P(y|\mathbf{x}) = s(\mathbf{w}^\top\mathbf{x})$ is the sigmoid function, where $s(\gamma) = \frac{1}{1+e^{-\gamma}}$.

Solution: Let $Q_k(\mathbf{x}) = \ln\Big((\sqrt{2\pi})^d\pi_k f_k(\mathbf{x})\Big)$, so we get that

$$p(y = 1|\mathbf{x}) = \frac{\pi_1 f_1(\mathbf{x})}{\pi_0 f_0(\mathbf{x}) + \pi_1 f_1(\mathbf{x})} = \left(1 + \frac{\pi_0 f_0(\mathbf{x})}{\pi_1 f_1(\mathbf{x})}\right)^{-1}$$

$$= \left(1 + \frac{e^{Q_0(\mathbf{x})}}{e^{Q_1(\mathbf{x})}}\right)^{-1} = s(Q_1(\mathbf{x}) - Q_0(\mathbf{x})).$$

Now lets look at the expression $Q_1(\mathbf{x}) - Q_0(\mathbf{x})$.

$$\begin{aligned} Q_1(\mathbf{x}) - Q_0(\mathbf{x}) &= \ln\Big((\sqrt{2\pi})^d \pi_1 f_1(\mathbf{x})\Big) - \ln\Big((\sqrt{2\pi})^d \pi_0 f_0(\mathbf{x})\Big) \\ &= \ln \frac{\pi_1}{1-\pi_1} - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) \\ &= \ln \frac{\pi_1}{1-\pi_1} + \mathbf{x}^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - \frac{1}{2}\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0. \end{aligned}$$

We can write it out as

$$Q_1(\mathbf{x}) - Q_0(\mathbf{x}) = \ln \frac{\pi_1}{1-\pi_1} + \frac{1}{2}\boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \frac{1}{2}\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} = w_0 + \mathbf{w}^\top \mathbf{x}.$$

(d) In the previous part we saw that the posterior probability for each class is the sigmoid function under the LDA model assumptions. Notice that LDA is a generative model. In this part we are going to look at the discriminative model. We assume that the posterior probability has Bernoulli distribution and the probability for each class is the sigmoid function, i.e., $p(Y = y|X = \mathbf{x}; \mathbf{w}) = q^y(1-q)^{1-y}$, where $q = s(\mathbf{w}^\top \mathbf{x})$, and try to find $\mathbf{w}$ that maximizes the likelihood function. Can you find a closed form[1] maximum-likelihood estimation of $\mathbf{w}$?

Solution:

Assume that our dataset is of size n. The likelihood is

$$\mathcal{L}(\mathbf{w}) = \prod_{i=1}^n p(y = y_i|\mathbf{x}_i) = \prod_{i=1}^n q_i^{y_i}(1-q_i)^{1-y_i}.$$

Maximizing the likelihood of the training data as a function of the parameters $\mathbf{w}$, we get

$$\begin{aligned} \hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \mathcal{L}(\mathbf{w}) &= \arg\max_{\mathbf{w}} \prod_{i=1}^n q_i^{y_i}(1-q_i)^{1-y_i} \\ &= \arg\max_{\mathbf{w}} \sum_{i=1}^n y_i \ln(q_i) + (1-y_i)\ln(1-q_i) \\ &= \arg\max_{\mathbf{w}} \sum_{i=1}^n y_i \ln\left(\frac{q_i}{1-q_i}\right) + \ln(1-q_i) \end{aligned}$$

Since q_i is the sigmoid function, we get

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} -\sum_{i=1}^n y_i \mathbf{w}^\top \mathbf{x}_i - \ln\Big(1 + \exp\big\{\mathbf{w}^\top \mathbf{x}_i\big\}\Big).$$

Let $J(\mathbf{w}) = -\sum_{i=1}^n y_i \mathbf{w}^\top \mathbf{x}_i - \ln\Big(1 + \exp\{\mathbf{w}^\top \mathbf{x}_i\}\Big)$. Notice that $J(\mathbf{w})$ is convex in $\mathbf{w}$, so global minima can be found. Recall that $s'(\gamma) = s(\gamma)(1 - s(\gamma))$. Now let us take the derivative of $J(\mathbf{w})$

[1] For the purpose of this question, we define a **closed form estimation** of $\mathbf{w}$ to mean an equality $\mathbf{w} = f(\mathbf{X},\mathbf{y})$ where f is not an infinite series.

w.r.t $\mathbf{w}$:

$$\frac{\partial J}{\partial \mathbf{w}} = -\sum_{i=1}^{n} y_i \mathbf{x}_i - \frac{\exp\{\mathbf{w}^\top \mathbf{x}_i\}}{1+\exp\{\mathbf{w}^\top \mathbf{x}_i\}}\mathbf{x}_i = \sum_{i=1}^{n}(s(\mathbf{w}^\top \mathbf{x}_i) - y_i)\mathbf{x}_i = \sum_{i=1}^{n}(s_i - y_i)\mathbf{x}_i = \mathbf{X}^\top(\mathbf{s}-\mathbf{y})$$

where $s_i = s(\mathbf{w}^\top \mathbf{x}_i), \mathbf{s} = (s_1, \ldots, s_n)^\top, \mathbf{y} = (y_1, \ldots, y_n)^\top$, and $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$.

Note that we **cannot** get a closed form estimate for $\mathbf{w}$ by setting the derivative to zero. This motivates the need to use Newton's method to find an estimate of $\mathbf{w}$.

(e) In this section we use Newton's method to find the optimal solution for $\mathbf{w}$. Write out the update step of Newton method. What does this say about how logistic regression handles outliers?

Solution: In the previous section we saw that we couldn't find a closed form solution for $\hat{\mathbf{w}}$, so to solve this problem we are going to use Newton method. Newton method, is an iterative method for finding successively better approximations to the roots (or zeroes) of a real-valued function. The iterative step in Newton method for some function $f(\mathbf{x})$ is

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\nabla f(\mathbf{x}^{(k)}))^{-1} f(\mathbf{x}^{(k)}).$$

In our case we want to find the zeros of $\nabla J(\mathbf{w})$. The update step is

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - (HJ(\mathbf{w}^{(k)}))^{-1}\nabla_w J(\mathbf{w}^{(k)})$$

where

$$\nabla_w J(\mathbf{w}) = \mathbf{X}^\top(\mathbf{s}-\mathbf{y}),$$

$$HJ(\mathbf{w}) = \nabla_w^2 J(\mathbf{w}) = \nabla_w \mathbf{X}^\top(\mathbf{s}-\mathbf{y}) = \sum_{i=1}^{n} s_i(1-s_i)\mathbf{x}_i\mathbf{x}_i^\top = \mathbf{X}^\top \mathbf{\Omega}\mathbf{X},$$

$\mathbf{\Omega} = \operatorname{diag}(s_1(1-s_1), \cdots, s_n(1-s_n))$.

Finding the inverse of the Hessian in high dimensions can be an expensive operation. Instead of directly inverting the Hessian we calculate the vector $\Delta_{k+1} = \mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}$ as the solution to the system of linear equations

$$HJ(\mathbf{w}^{(k)})\Delta_{k+1} = -\nabla J(\mathbf{w}^{(k)});$$
$$\mathbf{X}^\top \mathbf{\Omega}\mathbf{X}\Delta_{k+1} = \mathbf{X}^\top(\mathbf{y}-\mathbf{s}).$$

This looks similar to weighted least squares where $\mathbf{\Omega}$ and $(\mathbf{y}-\mathbf{s})$ change per iteration. Specifically looking at points where $s(\mathbf{w}^\top \mathbf{x}_i)$ is close to 0.5, Ω_i has the highest weight and points where $s(\mathbf{w}^\top \mathbf{x}_i)$ is close to 0 or 1 has much lower weight. Points where $s(\mathbf{w}^\top \mathbf{x}_i) \approx 0.5$ are close to the decision boundary and model is least sure of these points' cluster, so they move the decision boundary most in the next iteration. Going back to part (b), this should give some intuition on why outliers on the correct side of the boundary won't affect the decision boundary as much.

2 Bias and Variance

Oftentimes, such as in linear regression, we model the data-generating process as a noisy measurement of a true underlying response,

$$y_i = g(x_i) + \epsilon_i,$$

where ϵ_i is a zero-mean random noise variable.

We use machine learning techniques to build a hypothesis model $h(x)$ which is fit to the data as an approximation of $g(x)$. We usually don't know $g(x)$, but in the experiment that generated the plots on the next pages, suppose we know $g(x)$ is a straight line,

$$g(x) = wx + b.$$

The figures on the next pages show attempts to fit 0-degree, 1-degree, and 2-degree polynomials to g using different subsets of training data.

(a) The third figure is an attempt to fit a quadratic $h(x) = ax^2 + bx + c$ when the underlying f is a line. Why does the quadratic model learn a non-zero a? Why didn't it learn straight lines?

Solution: The second-order approximation is curving to fit the noise better, because the data includes noise. In the absence of noise, all $\{x_i, y_i\}$ would lie on the same line. In that case, the second-order approximations would learn straight lines, because the data would never suggest curvature.

(b) When evaluating models, what do we mean by "bias" of a model-estimation method? Explain the differences we see in the bias for polynomials of degree 0, 1, and 2.

Solution: Bias measures how close the average hypothesis (over all possible training sets) can come to the true underlying value $g(x)$, for a fixed value of x. Low bias means that, on average (that is, on average over infinite possible datasets), the regressor $h(x)$ accurately estimates $g(x)$. The degree-0 polynomial has the most bias, because a constant is not expressive enough to learn a sloped line. The assumptions of the degree-0 polynomial are too restrictive to allow us to learn the model accurately. The degree-1 model has the lowest bias because the assumptions of the model estimator are exactly correct. The degree-2 model has low bias because it is expressive enough to capture the first model.

(c) When evaluating models, what do we mean by "variance" of a model-estimation method? Explain the differences we see in the variance for polynomials of degrees 0, 1, and 2.

Solution: Variance measures the variance of the hypothesis (over all possible training sets), for a fixed value of x. A low variance means that the prediction does not change much as the training set varies. An unbiased method (bias = 0) could have large variance. The degree-0 polynomial has the least variance. It doesn't change as much from data subset to data subset because it always ignores x and guesses the average value of y in the training data. The degree-2 polynomial had the most variance because it will heavily curve to fit noise if it trains on a noisier data subset.

(d) We can decompose the least squares risk function into bias and variance as shown in lecture.

$$\begin{aligned}\mathbb{E}[(h(x)-y)^2] &= \mathbb{E}[h(x)^2] + \mathbb{E}[y^2] - 2\mathbb{E}[y(h(x)] \\ &= \mathrm{Var}(h(x)) + \mathbb{E}[h(x)]^2 + \mathrm{Var}(y) + \mathbb{E}[y]^2 - 2\mathbb{E}[y]\mathbb{E}[h(x)] \\ &= (\mathbb{E}[h(x)] - \mathbb{E}[y])^2 + \mathrm{Var}(h(x)) + \mathrm{Var}(y) \\ &= \underbrace{(\mathbb{E}[h(x)-g(x)])^2}_{\text{bias squared of method}} + \underbrace{\mathrm{Var}(h(x))}_{\text{variance of method}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{irreducible error}}\end{aligned}$$

We can decompose the error this way over the entire dataset, or we can decompose an individual point's error into these three components.

Now, observe the last figure. Why is the variance larger for points near the left and right extremes, and smaller for points in the middle?

Solution: Interpolation (estimating within the region of your data) is more stable and reliable than extrapolation (making predictions outside of the main distribution of your training data). Interpolation is more robust because a large deviation from ground truth in the middle of the data range will cause lots of misclassifications, but a large deviation at one of the extremes of the distribution would cause relatively few misclassifications.

(e) Why is our estimate of the bias not zero for the degree-1 and degree-2 models? Would it be zero if we generated an infinite number of datasets?

Solution: Bias describes an expectation over all possible datasets, but we are estimating the bias using a finite number of datasets. Our measure of the bias would be zero for the 1-degree model if we used an infinite number of datasets. In the case of the degree-2 model, the curvature would also approach zero as the number of datasets tends to infinity, but more slowly.

(f) How are bias and variance related to overfitting and underfitting?

Solution: High variance models are prone to overfitting. They are more prone to fit the noise of the data rather than the underlying distribution because their assumptions are too weak.

High bias models are prone to underfitting, because their assumptions are too restrictive and inexpressive to fit the underlying distribution well.

(g) Does training error provide a measure of bias, variance, or both? How about validation and test error?

Solution: Training error only provides a measure of bias. For example, training error will fool you if you use a 100-degree polynomial to fit 80 training points on a line. The model will heavily overfit and the training error will be zero, but the model will fail to generalize because of the variance of your model-estimation method. Validation and test error measure generalization ability, so they measure both bias and variance.

(h) How can we interpret the bias-variance trade-off in hard- and soft-margin SVM? Recall that the soft margin SVM objective is

$$\min \|w\|^2 + C\sum_i \xi_i \quad \text{subject to} \quad y_i(x_i^\top w + \alpha) \geq 1 - \xi_i; \quad \xi_i \geq 0.$$

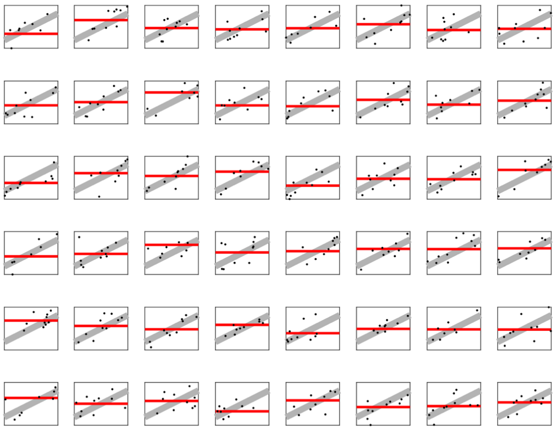
Solution: Hard-margin SVM is essentially soft-margin SVM with the C parameter tending to infinity. Increasing the C parameter shrinks the margin, weighs misclassified points more heavily, and results in more support vectors. One misclassified data point is allowed to have a larger impact on the learned model. Thus, increasing C increases the variance of the model. However, reducing C close to 0 results in underfitting—the margin grows larger because we are allowed to misclassify everything and we have little gravity towards learning the true model—which corresponds to high bias.

(i) How can we interpret the bias-variance trade-off in LDA and QDA?

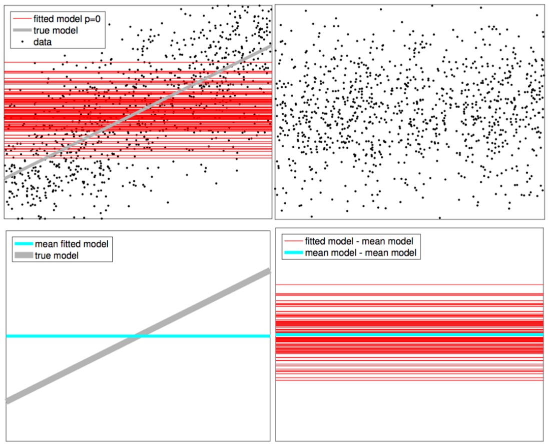
Solution: LDA makes a strong assumption that provides regularity: that all class covariance matrices are equal. The model learned by LDA varies less over possible datasets, but it risks underfitting data where class covariance matrices really are unequal. Thus, QDA has more variance and usually (but not always) less bias. For example, if the decision boundary is linear and LDA has zero bias, then QDA can't have less bias than that.

The figures on the left show many different models fit on subsets of training data for degrees $p = 0, 1, 2$. The figures on the right, the top left shows all learned models on top of the true model and data. The top right shows the noise of each data point, or the residual after subtracting $y - g(x)$. The bottom left shows the average learned model on top of the true model, and the figure on the bottom right shows all learned models on top of the average learned model.

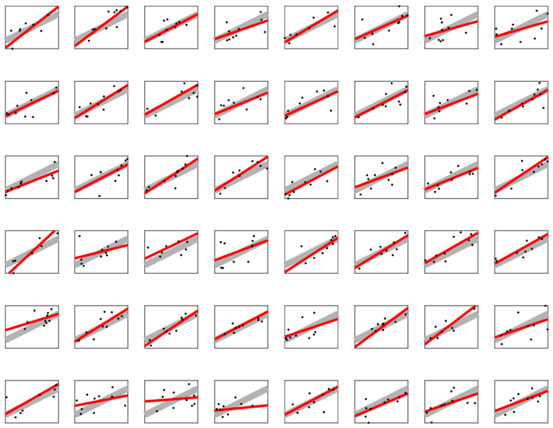
Fitting A Model over Multiple Datasets: $p = 0$



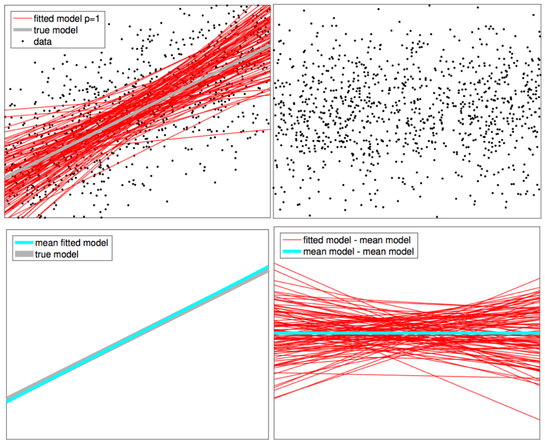
Bias and Variance in Model Selection: $p = 0$



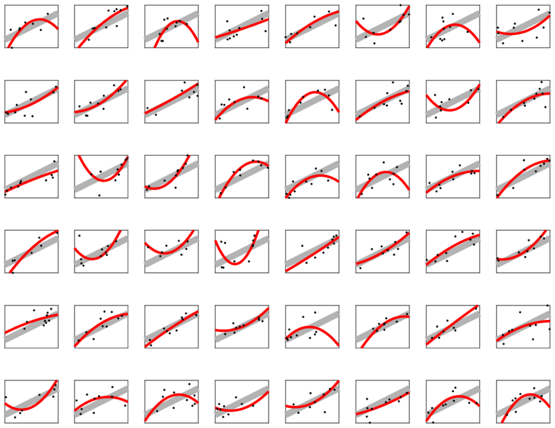
Fitting A Model over Multiple Datasets: $p = 1$



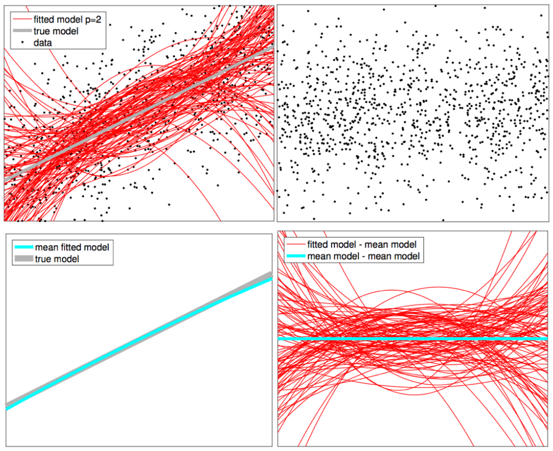
Bias and Variance in Model Selection: $p = 1$



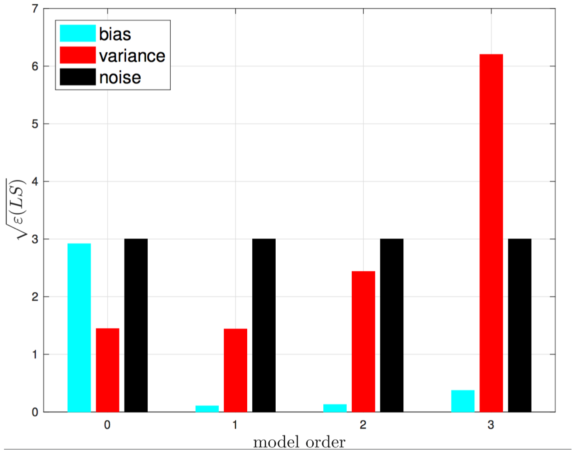
Fitting A Model over Multiple Datasets: $p = 2$



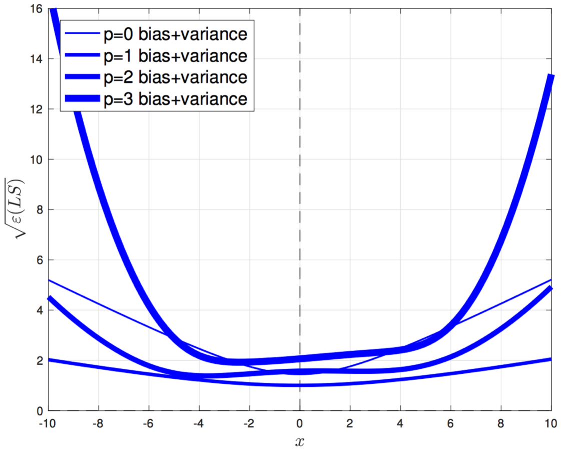
Bias and Variance in Model Selection: $p = 2$



Bias and Variance: Underfitting vs. Overfitting



Variation of Prediction Error with Model Order



3 Logistic Posterior with Different Variances

In Discussion 3, we proved that under an exponential class-conditional distribution, the posterior could be written in the form of a sigmoid that was linear over x. In this problem, we show that the posterior of a univariate QDA problem can also be written in the form of a sigmoid, but now it is *quadratic* over x. Consider the case when the class conditionals are Gaussian, but have different variances, i.e.,

$$\begin{aligned}(X|Y=i) &\sim \mathcal{N}(\mu_i, \sigma_i^2), \quad \text{where } i \in \{0,1\} \\ Y &\sim \text{Bernoulli}(\pi)\end{aligned}$$

1. Show that the posterior distribution of the class label given X is also a logistic function, but with a quadratic argument in X. That is, show that $P(Y=1|X=x)$ is of the form $1/(1+e^{-h(x)})$, where $h(x) = ax^2 + bx + c$ is quadratic in x.
2. Assuming 0-1 loss, what will the decision boundary look like (i.e., describe what the posterior probability plot looks like)?
3. Now suppose that we are dealing with an asymmetric loss function. Describe how this changes the decision boundary, if at all.

Solution:

1. We are solving for $\mathbf{P}(Y=1|x)$. By Bayes' Theorem,

$$\begin{aligned}\mathbf{P}(Y=1|x) &= \frac{f(x|Y=1)\mathbf{P}(Y=1)}{f(x|Y=1)\mathbf{P}(Y=1) + f(x|Y=0)\mathbf{P}(Y=0)} \\ &= \frac{1}{1 + \frac{\mathbf{P}(Y=0)f(x|Y=0)}{\mathbf{P}(Y=1)f(x|Y=1)}} \\ &= \frac{1}{1 + \frac{\sigma_1}{\sigma_0}\frac{1-\pi}{\pi}\exp\left(\frac{(x-\mu_1)^2}{2\sigma_1^2} - \frac{(x-\mu_0)^2}{2\sigma_0^2}\right)}.\end{aligned}$$

   Looking at the last expression, we have the subexpression

$$\frac{\sigma_1}{\sigma_0}\frac{1-\pi}{\pi}\exp\left(\frac{(x-\mu_1)^2}{2\sigma_1^2} - \frac{(x-\mu_0)^2}{2\sigma_0^2}\right)$$

$$= \exp\left[\left(\frac{1}{2\sigma_1^2} - \frac{1}{2\sigma_0^2}\right)x^2 + \left(\frac{\mu_0}{\sigma_0^2} - \frac{\mu_1}{\sigma_1^2}\right)x + \left(\frac{\mu_1^2}{\sigma_1^2} - \frac{\mu_0^2}{\sigma_0^2} + \ln\left(\frac{\sigma_1}{\sigma_0}\frac{1-\pi}{\pi}\right)\right)\right].$$

   Now we see that we have a logistic function $1/(1+e^{-h(x)})$, where $h(x) = ax^2 + bx + c$ is a quadratic function for certain values of a, b, c. Note that the special case of $\sigma_1 = \sigma_0$ gives a linear function in x (LDA).

2. Since we are assuming 0-1 loss, we use the optimal classifier $r^*(x) = 1$ when $\mathbf{P}(Y = 1|x) > \mathbf{P}(Y = 0|x)$ (equivalently, in alternate notation, when $Q_C(x) - Q_D(x) > 0$, where class C = 1 and class D = 0). Thus, the decision boundary can be found when $\mathbf{P}(Y = 1|x) = \mathbf{P}(Y = 0|x) = \frac{1}{2}$. This happens when $h(x) = 0$. Solving for the roots of $h(x)$ results in 2 values where this equality holds. One can convince herself/himself that in the plot of posterior probability graph, the horizontal (x) axis will be split into three regions: we classify the two outer regions as one class, and the middle one as another class. The choice of which class to classify in the outer regions depends on the values of σ_1 and σ_2.

3. Under the Bayes optimal classifier, we use $r^*(x) = 1$ when $L(0, 1)\mathbf{P}(Y = 1|x) > L(1, 0)\mathbf{P}(Y = 0|x)$. As before, the decision boundary can be found where $L(0, 1)\mathbf{P}(Y = 1|x) = L(1, 0)\mathbf{P}(Y = 0|x)$. This happens when $h(x) = \alpha$, where α can be any real number depending on the loss function L. Note that since α is no longer guaranteed to be 0, there could be anywhere from 0 to 2 solutions, and thus 0 to 2 points x that represent the decision boundary.

4 Multivariate Gaussians: A Review

Consider a two dimensional random variable $Z \in \mathbb{R}^2$. In order for the random variable to be jointly Gaussian, a necessary and sufficient condition is that

- Z_1 and Z_2 are each marginally Gaussian, and
- $Z_1|Z_2 = z$ is Gaussian and $Z_2|Z_1 = z$ is Gaussian.

Recall that the PDF of a multivariate Gaussian is $f(z) = \frac{1}{\sqrt{(2\pi)^k|\Sigma|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$.

(a) Let X_1 and X_2 be i.i.d. standard normal random variables. Let U be a discrete random variable such that $P(U = -1) = P(U = 1) = \frac{1}{2}$, independent of everything else. First, verify if the conditions of the first characterization hold for the following random variables (i.e., they are each marginally Gaussian, and their conditional probabilities are also Gaussian). Second, calculate the covariance matrix Σ_Z.

(a) $Z_1 = X_1$ and $Z_2 = X_2$.

(b) $Z_1 = X_1$ and $Z_2 = -X_1$.

(c) $Z_1 = X_1$ and $Z_2 = UX_1$.

Solution: Before diving into the solution, recall that the covariance matrix of a vector random variable X with mean (vector) μ is $\Sigma = \mathbb{E}[(X-\mu)(X-\mu)^\top]$. In other words, entry i, j of the covariance matrix is the covariance between the random variables X_i and X_j, i.e., $\Sigma_{ij} = \text{cov}(X_i, X_j) = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]$.

(a) Z_1 and Z_2 are i.i.d. standard Gaussian, and so $Z_1|(Z_2 = z) \sim N(0, 1)$. Also, $Z_2|(Z_1 = z) \sim N(0, 1)$. Hence, the random variables are jointly Gaussian. We also have $\Sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

(b) We have $Z_1 \sim N(0, 1)$ and $Z_2 \sim N(0, 1)$ marginally. However, we have $Z_1|(Z_2 = z) \sim N(-z, 0)$, which is a degenerate Gaussian (in this case, it is the scalar $-z$). The other conditional distribution is identical. Hence, the random variables are jointly Gaussian. The covariance matrix is $\Sigma_Z = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$.

(c) As before, we have $Z_1 \sim N(0, 1)$ and $Z_2 \sim N(0, 1)$ marginally. To see this, write

$$\begin{aligned} f(Z_2 = z_2) &= f(Z_2 = z_2|U = 1)P(U = 1) + f(Z_2 = z_2|U = -1)P(U = -1) \\ &= \frac{1}{2}f(X_1 = z_2|U = 1) + \frac{1}{2}f(X_1 = -z_2|U = -1) \\ &= \frac{1}{2}f(X_1 = z_2) + \frac{1}{2}f(X_1 = z_2) \\ &= f(X_1 = z_2) \end{aligned}$$

However, the random variable $(Z_2|Z_1 = z)$ is uniformly distributed on $\{-z, z\}$, and is therefore not Gaussian, so the random variables are therefore **not** jointly Gaussian. The covariance matrix is $\Sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ since Z_1, Z_2 are uncorrelated (but not independent).

(b) Use the example above to show that two Gaussian random variables can be uncorrelated, but not independent. On the other hand, show that two uncorrelated, jointly Gaussian random variables are independent.

Solution:

Recall that two random variables X and Y are said to be *uncorrelated* if $\mathrm{cov}(U,V) = 0$, and are *independent* if and only if $P(X = x, Y = y) = P(X = x)P(Y = y), \forall x, y$.

The last example in the previous part shows uncorrelated Gaussians that are not independent. In order to show that jointly Gaussian RVs (with individual variances σ_1^2 and σ_2^2) that are uncorrelated are also independent, assume without loss of generality that the RVs have zero mean, and notice that one can write the joint PDF as

$$\begin{aligned} f_Z(z_1,z_2) &= \frac{1}{(2\pi)\det\left(\Sigma_Z^{1/2}\right)} \exp\left(-\frac{1}{2}\begin{bmatrix} z_1 & z_2 \end{bmatrix} (\Sigma_Z)^{-1} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right) \\ &= \frac{1}{\sqrt{2\pi\sigma_1^2}} \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2\sigma_1^2} z_1^2\right) \exp\left(-\frac{1}{2\sigma_1^2} z_2^2\right) \\ &= f_{Z_1}(z_1) f_{Z_2}(z_2). \end{aligned}$$

The decomposition follows since Σ_Z is a diagonal matrix when the random variables are uncorrelated. Since we have expressed the joint PDF as a product of the individual PDFs, the random variables are independent.

(c) Let $Z = VX$, where $V \in \mathbb{R}^{2\times 2}$, $Z, X \in \mathbb{R}^2$, and $X \sim N(0, I)$. What is the covariance matrix Σ_Z? Is this also true for a random variable other than Gaussian?

Solution: The covariance matrix of a random vector Z (by definition) is given by $\mathbb{E}(Z - \mathbb{E}[Z])(Z - \mathbb{E}[Z])^\top$. Since the mean $\mathbb{E}[Z]$ is 0, we may write $\Sigma_Z = \mathbb{E}[VXX^\top V^\top] = V\mathbb{E}[XX^\top]V^\top = VV^\top$. This follows by linearity of expectation applied to vector random variables (write it out to convince yourself!)

Yes, this relation is also true for other distributions, since we didn't use the Gaussian assumption in this proof.

(d) Use the above setup to show that $X_1 + X_2$ and $X_1 - X_2$ are independent. Give another example pair of linear combinations that are independent.

Solution: By our previous arguments, it is sufficient to show that these are uncorrelated. Calculating the covariance matrix, we have $\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$, which is diagonal. Any linear combination $Z = VX$ with $VV^\top = D$ for a diagonal matrix D results in uncorrelated random variables.

5 Gradient Descent and Convexity

The smoothed version of the hinge loss function[2] with parameter t is

$$f(y) = \begin{cases} \frac{1}{2} - ty & \text{if } ty \le 0, \\ \frac{1}{2}(1-ty)^2 & \text{if } 0 < ty < 1, \\ 0 & \text{if } 1 \le ty. \end{cases}$$

Define $L(w) = \frac{1}{n}\sum_{i=1}^{n} f(w^\top x_i - y_i)$. Given sample points $x_1, x_2, \dots, x_n$ and labels $y_1, y_2, \dots, y_n$, we define the optimization problem

$$\min_w \frac{1}{n}\sum_{i=1}^{n} f(w^\top x_i - y_i)$$

1. Is $L(w)$ convex?
2. Write out the gradient descent update equation.
3. Write out the stochastic gradient descent update equation.

Solution:

1. Define $g(y) = \frac{\partial f}{\partial y}$.

$$g(y) = \begin{cases} -t & \text{if } ty \le 0, \\ -t(1-ty) & \text{if } 0 < ty < 1, \\ 0 & \text{if } 1 \le ty. \end{cases}$$

   Taking the second derivative,

$$\frac{\partial g}{\partial y} = \begin{cases} 0 & \text{if } ty \le 0, \\ t^2 & \text{if } 0 < ty < 1, \\ 0 & \text{if } 1 \le ty. \end{cases}$$

   This implies that the second derivative is always greater than or equal to 0, so the function f is convex. Since the function L is the sum of convect functions, $L(w)$ is convex.

2. Using the chain rule,

$$\frac{\partial f(w^\top x - y)}{\partial w} = g(w^\top x - y)\frac{\partial(w^\top x - y)}{w} = g(w^\top x - y)x.$$

   Hence the gradient descent update rule is

$$w = w - \alpha\frac{1}{N}\sum_{i=1}^{N} g(w^\top x_i - y_i)x_i.$$

[2] This function is not required knowledge.

3. Pick an index i uniformly at random from $\{1, ..., n\}$.

   The stochastic gradient descent update rule is

$$w = w - \alpha g(w^\top x_i - y_i)x_i.$$