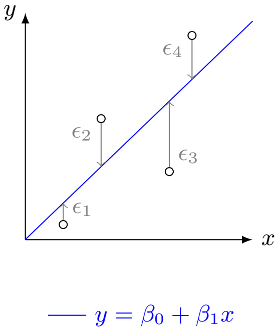
Exam 2 Review

Spring 2021

Announcements

- Exam 2 will be "take home": 80 minutes from when you **begin** the exam.
- The exam *must* be completed by 5pm Central time on Friday 3/26.
- I will have additional office hours on Thursday, 8:30–9:20am for last-minute questions.

RMSE of a linear model

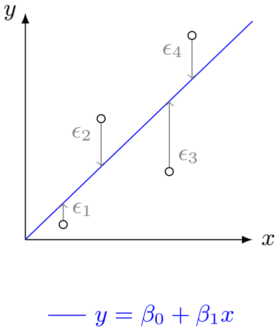


$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\mathrm{pred}} - y_i^{\mathrm{true}}\right)^2}$$

- The RMSE measures the uncertainty in the model's predictions. Predictions should be reported as $y^{\mathrm{pred}} \pm$ RMSE.
- The 95% confidence interval of a model's predictions are

$$[y^{\mathrm{pred}} - 2\,\mathrm{RMSE},\, y^{\mathrm{pred}} + 2\,\mathrm{RMSE}]$$

RMSE of a linear model



$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\text{pred}} - y_i^{\text{true}}\right)^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2}$$

- The RMSE measures the uncertainty in the model's predictions. Predictions should be reported as $y^{\text{pred}} \pm$ RMSE.
- The 95% confidence interval of a model's predictions are

$$[y^{\text{pred}} - 2\,\text{RMSE},\; y^{\text{pred}} + 2\,\text{RMSE}]$$

0-, 1-, and 2-norm regularization

$$\boldsymbol{\beta} = \begin{pmatrix} 0 \\ 1.2 \\ -0.3 \\ 0 \\ 1 \end{pmatrix}$$

$$\|\boldsymbol{\beta}\|_0 = 0 + 1 + 1 + 0 + 1 = 3 \quad \text{(3 nonzero entries)}$$

$$\|\boldsymbol{\beta}\|_1 = |0| + |1.2| + |-0.3| + |0| + |1| = 2.5$$

$$\|\boldsymbol{\beta}\|_2 = \sqrt{0^2 + 1.2^2 + (-0.3)^2 + 0^2 + 1^2} = 2.53$$

0-, 1-, and 2-norm regularization

$$\boldsymbol{\beta} = \begin{pmatrix} 0 \\ 1.2 \\ -0.3 \\ 0 \\ 1 \end{pmatrix}$$

$$\|\boldsymbol{\beta}\|_0 = 0 + 1 + 1 + 0 + 1 = 3 \quad \text{(3 nonzero entries)}$$

$$\|\boldsymbol{\beta}\|_1 = |0| + |1.2| + |-0.3| + |0| + |1| = 2.5$$

$$\|\boldsymbol{\beta}\|_2 = \sqrt{0^2 + 1.2^2 + (-0.3)^2 + 0^2 + 1^2} = 2.53$$

$$\min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}) \quad \xrightarrow{\text{regularization}} \quad \min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_k$$

0-, 1-, and 2-norm regularization

$$\boldsymbol{\beta} = \begin{pmatrix} 0 \\ 1.2 \\ -0.3 \\ 0 \\ 1 \end{pmatrix}$$

$$\|\boldsymbol{\beta}\|_0 = 0 + 1 + 1 + 0 + 1 = 3 \quad \text{(3 nonzero entries)}$$
$$\|\boldsymbol{\beta}\|_1 = |0| + |1.2| + |-0.3| + |0| + |1| = 2.5$$
$$\|\boldsymbol{\beta}\|_2 = \sqrt{0^2 + 1.2^2 + (-0.3)^2 + 0^2 + 1^2} = 2.53$$

$$\min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}) \quad \xrightarrow{\text{regularization}} \quad \min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_k$$

Regularization	**Computation**	**Sparsity**	**Unique?**
0-norm	Hard (combinatorial)	Sparse	No
1-norm	Easier (discontinuous derivative)	Mostly sparse	No
2-norm	Easy	Dense	Yes
Elastic Net (1&2-norm)	Easier (like 1-norm)	Some sparsity	Yes

Systems of linear inequalities

- A system of linear inequalities has the form $\mathbf{Ax} \leq \mathbf{b}$.
- This includes inequalities of the form $\mathbf{Ax} \geq \mathbf{b}$, since these can be transformed to $-\mathbf{Ax} \leq -\mathbf{b}$.
- Systems of inequalities often have infinitely many solutions. We have not discussed how to solve these systems (we let MATLAB solve the SVM problem).
- Also, note that we haven't discussed *strict* inequalities ($\mathbf{Ax} < \mathbf{b}$). These are much harder to solve since the solution set is open.
- Are the solution sets to linear inequalities always convex?

Systems of linear inequalities

- A system of linear inequalities has the form $\mathbf{A}\mathbf{x} \leq \mathbf{b}$.
- This includes inequalities of the form $\mathbf{A}\mathbf{x} \geq \mathbf{b}$, since these can be transformed to $-\mathbf{A}\mathbf{x} \leq -\mathbf{b}$.
- Systems of inequalities often have infinitely many solutions. We have not discussed how to solve these systems (we let MATLAB solve the SVM problem).
- Also, note that we haven't discussed *strict* inequalities ($\mathbf{A}\mathbf{x} < \mathbf{b}$). These are much harder to solve since the solution set is open.
- Are the solution sets to linear inequalities always convex?

Proof. Assume $\mathbf{x}_1$ and $\mathbf{x}_2$ are solutions to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$. Then

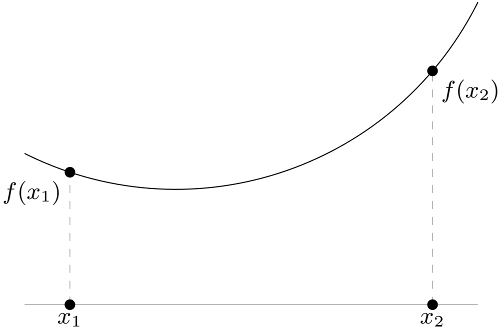
$$\begin{aligned}
\mathbf{A}(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) &= \lambda\mathbf{A}\mathbf{x}_1 + (1-\lambda)\mathbf{A}\mathbf{x}_2 \\
&\leq \lambda\mathbf{b} + (1-\lambda)\mathbf{b} \\
&= \mathbf{b}
\end{aligned}$$

Since all points on the line connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ are also solutions, the solution space must be convex.

Convex functions

A function f is convex if and only if

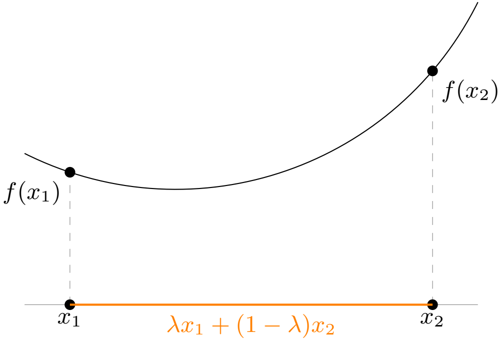
$$f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2), \quad \lambda \in [0,1]$$



Convex functions

A function f is convex if and only if

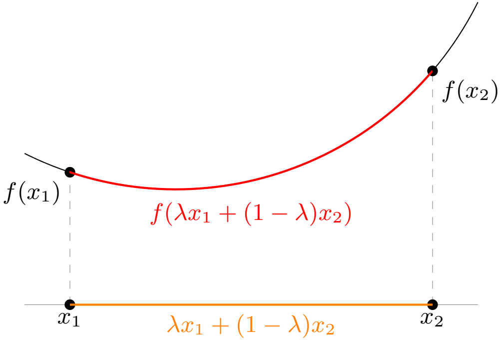
$$f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2), \quad \lambda \in [0,1]$$



Convex functions

A function f is convex if and only if

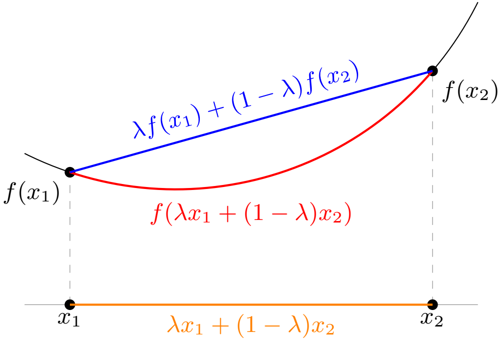
$$\color{red}{f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2)} \leq \color{blue}{\lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2)}, \quad \lambda \in [0,1]$$



Convex functions

A function f is convex if and only if

$$f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2), \quad \lambda \in [0,1]$$

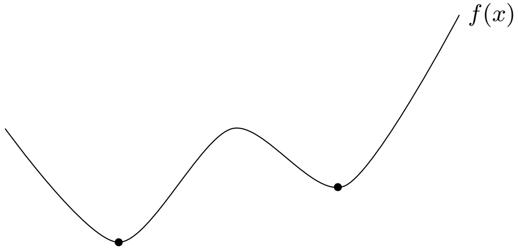


Convex functions with convex domains have only global minima.

A function f is convex if and only if

$$f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2), \quad \lambda \in [0,1]$$

Here is a function with local minima. However, it isn't convex.

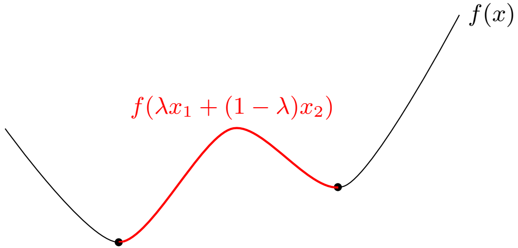


Convex functions with convex domains have only global minima.

A function f is convex if and only if

$$f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2), \quad \lambda \in [0,1]$$

Here is a function with local minima. However, it isn't convex.

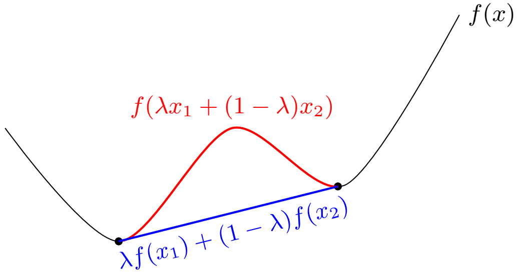


Convex functions with convex domains have only global minima.

A function f is convex if and only if

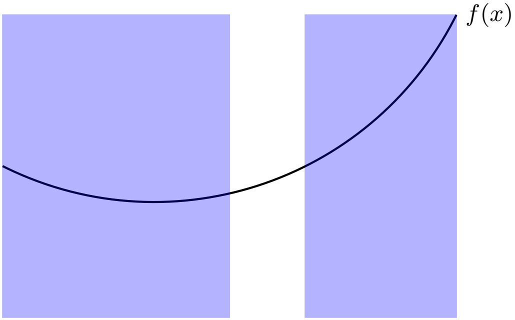
$$f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2), \quad \lambda \in [0,1]$$

Here is a function with local minima. However, it isn't convex.



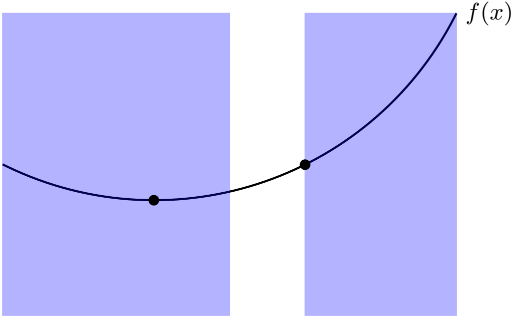
Why does the domain need to be convex?

Imagine minimizing a convex function over a discontinuous (non-convex) domain (shaded blue).



Why does the domain need to be convex?

Imagine minimizing a convex function over a discontinuous (non-convex) domain (shaded blue).



The boundaries create a second local (but not global) minimum of the convex function.

The Jacobian Matrix

Consider a multivariate function $\mathbf{g}(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ g_3(\mathbf{x}) \end{pmatrix}$

The Jacobian matrix of partial derivatives is

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \frac{\partial g_2}{\partial x_3} \\ \frac{\partial g_3}{\partial x_1} & \frac{\partial g_3}{\partial x_2} & \frac{\partial g_3}{\partial x_3} \end{pmatrix}$$

The Jacobian Matrix

Consider a multivariate function $\mathbf{g}(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ g_3(\mathbf{x}) \end{pmatrix}$

The Jacobian matrix of partial derivatives is

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} \\ \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \frac{\partial g_2}{\partial x_3} \\ \\ \frac{\partial g_3}{\partial x_1} & \frac{\partial g_3}{\partial x_2} & \frac{\partial g_3}{\partial x_3} \end{pmatrix}$$

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} x_1x_2x_3 \\ x_2^2 - x_1x_3 \\ 2x_1^3 \end{pmatrix} \quad \Rightarrow \quad \mathbf{J}(\mathbf{x}) = \begin{pmatrix} x_2x_3 & x_1x_3 & x_1x_2 \\ -x_3 & 2x_2 & -x_1 \\ 6x_1^2 & 0 & 0 \end{pmatrix}$$

The pseudoinverse

Imagine a linear system $\mathbf{A}\mathbf{x} = \mathbf{y}$. If $\mathbf{A}$ is square and full rank, then it has a true inverse $\mathbf{A}^{-1}$. We can use the inverse to solve for $\mathbf{x}$:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$

The pseudoinverse

Imagine a linear system $\mathbf{A}\mathbf{x} = \mathbf{y}$. If $\mathbf{A}$ is square and full rank, then it has a true inverse $\mathbf{A}^{-1}$. We can use the inverse to solve for $\mathbf{x}$:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$

What if $\mathbf{A}$ is not square or not full rank? It still has a pseudoinverse $\mathbf{A}^{+}$ that we can use to solve for $\mathbf{x}$:

$$\mathbf{x} = \mathbf{A}^{+}\mathbf{y}$$

The pseudoinverse

Imagine a linear system $\mathbf{Ax} = \mathbf{y}$. If $\mathbf{A}$ is square and full rank, then it has a true inverse $\mathbf{A}^{-1}$. We can use the inverse to solve for $\mathbf{x}$:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$

What if $\mathbf{A}$ is not square or not full rank? It still has a pseudoinverse $\mathbf{A}^{+}$ that we can use to solve for $\mathbf{x}$:

$$\mathbf{x} = \mathbf{A}^{+}\mathbf{y}$$

If $\mathbf{A}$ is not full rank there are often infinitely many solutions to the linear system. Solving with the pseudoinverse gives the *least squares* solution, i.e. the solution that minimizes the elementwise squared difference between $\mathbf{Ax}$ and $\mathbf{y}$. The least squares solution is ideal for building linear models.

The pseudoinverse

Imagine a linear system $\mathbf{Ax} = \mathbf{y}$. If $\mathbf{A}$ is square and full rank, then it has a true inverse $\mathbf{A}^{-1}$. We can use the inverse to solve for $\mathbf{x}$:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$

What if $\mathbf{A}$ is not square or not full rank? It still has a pseudoinverse $\mathbf{A}^{+}$ that we can use to solve for $\mathbf{x}$:

$$\mathbf{x} = \mathbf{A}^{+}\mathbf{y}$$

If $\mathbf{A}$ is not full rank there are often infinitely many solutions to the linear system. Solving with the pseudoinverse gives the *least squares* solution, i.e. the solution that minimizes the elementwise squared difference between $\mathbf{Ax}$ and $\mathbf{y}$. The least squares solution is ideal for building linear models.

Final note: If the matrix $\mathbf{A}^{\mathsf{T}}\mathbf{A}$ has full column rank, then $\mathbf{A}^{+} = (\mathbf{A}^{\mathsf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}$. Otherwise, we find the pseudoinverse using the Singular Value Decomposition (as we will see in Part III).

Interpreting the output of `fitlm`

```
model2 =
Linear regression model:
    y ~ 1 + x + x^2

Estimated Coefficients:
                   Estimate      SE        tStat        pValue
                   ________    _______    ________    __________

    (Intercept)    0.33485      8.0944    0.041369       0.96709
    x               1.3816      2.3069     0.59887       0.55065
    x^2             1.0595     0.14057       7.537    2.5514e-11


Number of observations: 100, Error degrees of freedom: 97
Root Mean Squared Error: 20.9
R-squared: 0.932,  Adjusted R-Squared 0.931
F-statistic vs. constant model: 667, p-value = 2.1e-57
```

- **Estimate**: The estimated values of the parameter (β_i).
- **SE**: The standard error of the estimate. Roughly, if $\beta \pm 2$ SE includes 0, the parameter is not significant, but we prefer judgements based on the p-value of a t-test (below).
- **tStat**: The t-statistic used to calculate the p-value. Not directly interpretable.
- **pValue**: The probability that a nonzero parameter estimate of this size could have occurred randomly. If $p < 0.05$, we say the parameter is significantly nonzero.

Why predict $\log(\text{odds})$ in logistic regression?

Remember that for logistic regression we use a linear model to predict the $\log(\text{odds})$, i.e.

$$\log(\text{odds}(y = 1)) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

Why predict $\log(\text{odds})$ in logistic regression?

Remember that for logistic regression we use a linear model to predict the $\log(\text{odds})$, i.e.

$$\log(\text{odds}(y = 1)) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

Why log odds? Two reasons:

- The odds of an event is always non-negative, and we have no way of forcing our linear model to only make non-negative predictions. However, the log odds can take any value, with negative values indicating odds < 1.
- When we pass the log odds through the logistic link function, the output becomes the probability $P(y = 1)$. This is easier to interpret and can be compared with binary data during model fitting.

Deriving estimators

1. The goal of model fitting is to find parameters that minimize the *loss*.
2. For example, the quadratic loss for a model is

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i^{\mathsf{pred}}(\boldsymbol{\beta}) - y_i^{\mathsf{true}} \right)^2$$

Deriving estimators

1. The goal of model fitting is to find parameters that minimize the *loss*.
2. For example, the quadratic loss for a model is

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i^{\text{pred}}(\boldsymbol{\beta}) - y_i^{\text{true}} \right)^2$$

3. We substitute the model in for y^{pred} and minimize, usually by finding roots of the gradient with respect to $\boldsymbol{\beta}$.

Newton's Method vs. Gradient descent

We use Newton's method to find the roots of a multivariate function $\mathbf{g}(\mathbf{x})$.

1. Compute $\mathbf{J}(\mathbf{x})$, the Jacobian of $\mathbf{g}$.
2. Start with an initial guess $\mathbf{x}^{(0)}$.
3. Iterate:

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \mathbf{J}^{-1}(\mathbf{x}^{(0)})\mathbf{g}(\mathbf{x}^{(0)}).$$

4. Repeat until

$$\mathbf{g}(\mathbf{x}^{(k)}) \approx 0$$

or

$$\mathbf{x}^{(k)} \approx \mathbf{x}^{(k-1)}.$$

We use gradient descent to minimize a scalar-valued function $f(\mathbf{x})$.

1. Comptute $\mathbf{g}(\mathbf{x})$, the gradient of f.
2. Start with an initial guess $\mathbf{x}^{(0)}$.
3. Iterate:

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \alpha\mathbf{g}(\mathbf{x}^{(0)}).$$

4. Repeat until

$$f(\mathbf{x}^{(k)}) \approx f(\mathbf{x}^{(k-1)})$$

or

$$\mathbf{x}^{(k)} \approx \mathbf{x}^{(k-1)}.$$