CS6375: Machine Learning

Gautam Kunapuli

Linear Regression

THE UNIVERSITY OF TEXAS AT DALLAS
Erik Jonsson School of Engineering and Computer Science

Example: Pattern Analysis in 17th Century Astronomy

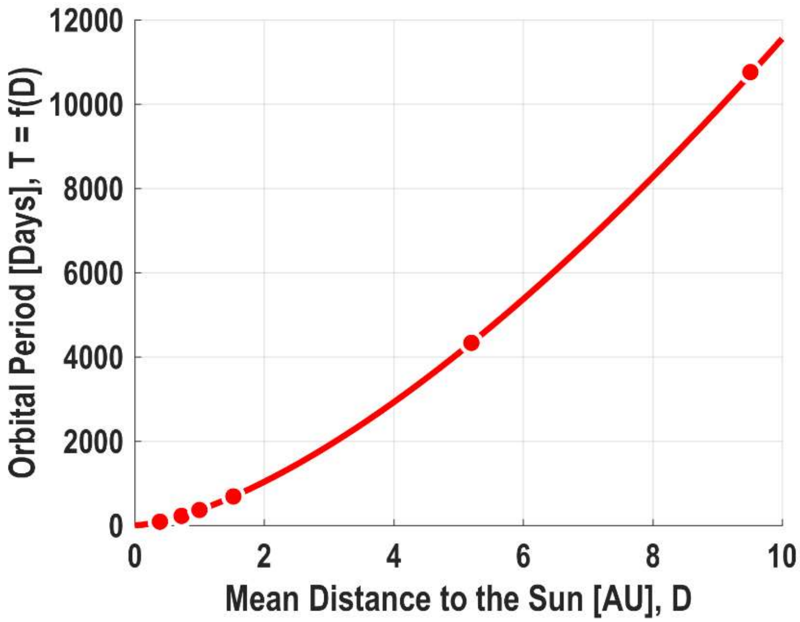
*The German astronomer **Johannes Kepler** published his laws of planetary motion in 1609 & 1619, and discovered them by **analyzing** the astronomical observations of **Tycho Brahe***

Planet	Mean dist. to sun [AU], D	Orbital Period [days], T	D^3/T^2
Mercury	0.389	87.77	7.64
Venus	0.724	224.70	7.52
Earth	1	365.25	7.50
Mars	1.524	686.95	7.50
Jupiter	5.2	4332.62	7.49
Saturn	9.510	10759.2	7.43

Kepler's 3rd Law: the **square of the orbital period** of a planet (T^2) is **proportional** to the **cube of the semi-major axis** of its orbit (D^3).

$$T = f(D) = cD^{\frac{3}{2}}$$

Example: Pattern Analysis in 17th Century Astronomy



Planet	Mean dist. to sun [AU], D	Orbital Period [days], T	D^3/T^2
Mercury	0.389	87.77	7.64
Venus	0.724	224.70	7.52
Earth	1	365.25	7.50
Mars	1.524	686.95	7.50
Jupiter	5.2	4332.62	7.49
Saturn	9.510	10759.2	7.43
Uranus	19.191	?	~7.50
Neptune	30.069	?	~7.50

Kepler's 3rd Law is an example of a ***model*** *that relates data (D) to labels (T)*

Kepler's 3rd Law: the **square of the orbital period** of a planet (T^2) is **proportional** to the **cube of the semi-major axis** of its orbit (D^3).

$$T = f(D) = cD^{\frac{3}{2}}$$

This is an example of a ***supervised machine-learning*** *problem, where* ***labels*** *(T) are available for learning the* ***model****. This is, in fact, a (non-linear)* ***regression problem****.*

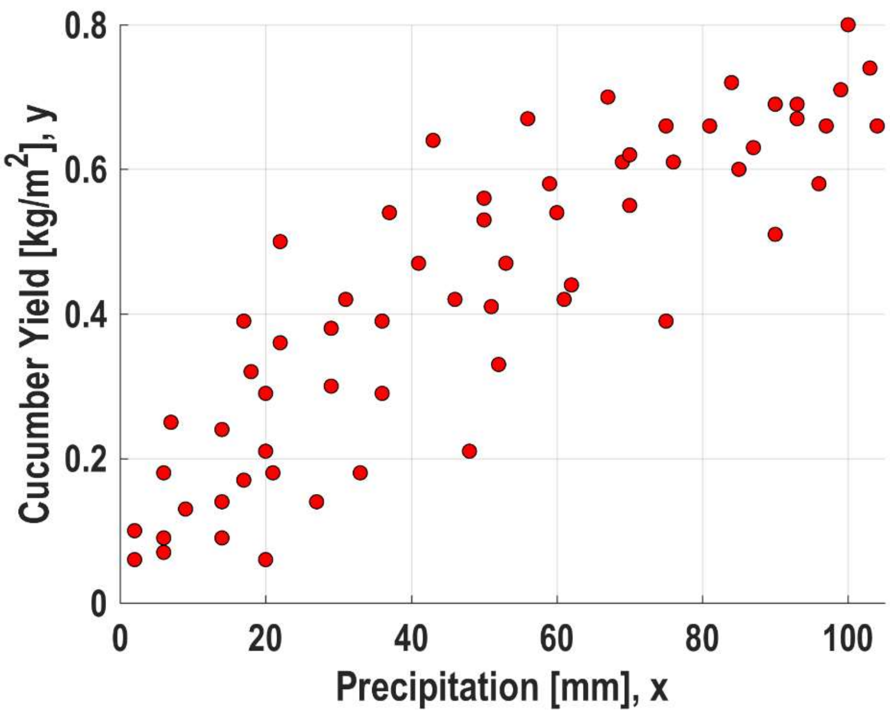
If we know the ***model****, we can use it to* ***predict*** *the orbital periods of newly-discovered planets. This property of machine-learning models is called* ***generalization****.*

Univariate Linear Regression

Problem Setup: Given data (x_i) and real-valued labels (y_i), find the best model that **fits current data** <u>**and**</u> **predicts future data**

Example: Develop a model to predict produce yield depending on the precipitation this year.

Here, the independent variable (training data) is precipitation (x_i) and the dependent variable (label) is yield (y_i).



*First, we select the **hypothesis class**, which is the set of allowable functions to model the relationship between data (x_i) and labels (y_i):*

$$y = f(x)$$

Our hypothesis class is the space of all **univariate linear functions**, $y = f(x) = w \cdot x + b$

*the model is **univariate** because there is only one independent (training) variable, x*

*the model is **linear** because the highest allowed degree is x^1. Higher-order models will be nonlinear, for example, the quadratic hypothesis class:*

$$y = u \cdot x^2 + w \cdot x + b$$

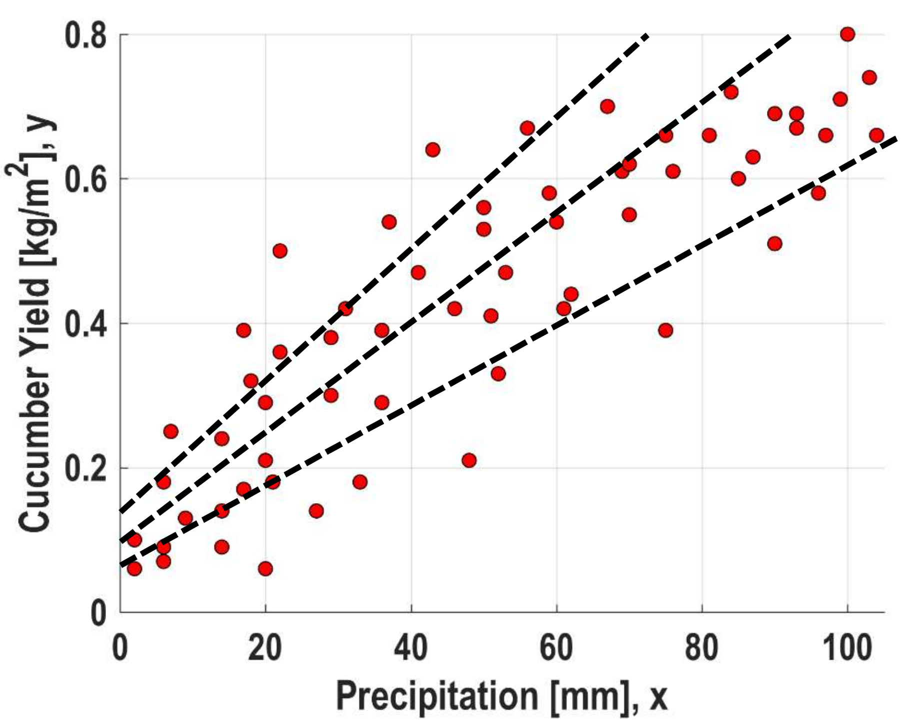
*The goal is to learn the **parameters** w and b that **best fit** the training data.*

Univariate Linear Regression

Problem Setup: Given data (x_i) and real-valued labels (y_i), find the best model that **fits current data <u>and</u> predicts future data**

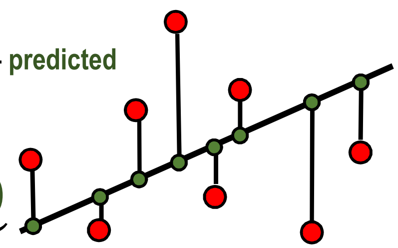
Example: Develop a model to predict produce yield depending on the precipitation this year.

Here, the independent variable (training data) is precipitation (x_i) and the dependent variable (label) is yield (y_i).



*There are **infinitely many functions in the hypothesis class**: $y = w \cdot x + b$ that can model the data. To identify the best, we must measure the **quality of fit**.*

$$e_i = y_i - \underbrace{(w \cdot x_i + b)}_{f(x_i)}$$



The quality of fit can be measured using a **loss function** that depends on the **error** between the **true** and **predicted** labels

*In linear regression, we measure fit using the squared loss over the error, that is, we use a **squared loss function**, $\frac{1}{2} e_i^2$*

$$L(f(x_i), y_i) = \frac{1}{2}\left(y_i - (w \cdot x_i + b)\right)^2$$

Formulating and Solving Linear Regression

Problem Formulation: the **best model minimizes** the **average squared loss** across all the data; that is, find the **best parameters** w and b such that their predictions **minimize the average squared loss**.

Problem: Given n training examples (x_i, y_i), $i = 1, \ldots, n$, find the best model (w, b) by solving

$$\underset{w,b}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (w \cdot x_i + b)\right)^2$$

This is an **(unconstrained) optimization problem** in the variables (w, b). The **optimal solution** will be our model.

- **Solution Approach 1:** Take derivatives and solve analytically. This leads to a **closed-form solution**.

  Note that closed-form solutions are **not always directly computable**.

Formulating and Solving Linear Regression

Problem Formulation: the **best model minimizes** the **average squared loss** across all the data; that is, find the **best parameters** w and b such that their predictions **minimize the average squared loss**.

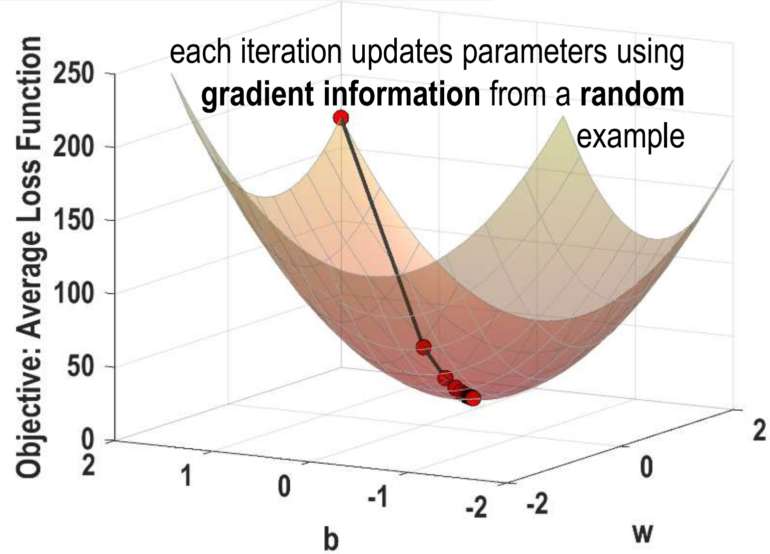
Problem: Given n training examples (x_i, y_i), $i = 1, \dots, n$, find the best model (w, b) by solving

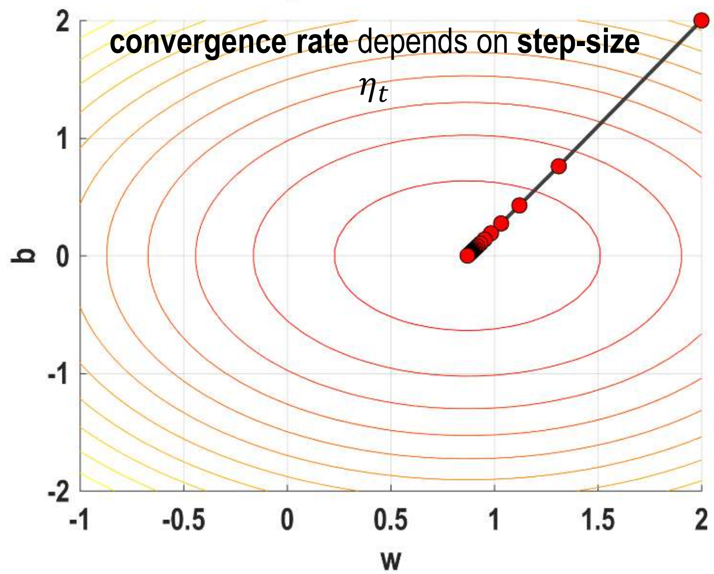
$$\underset{w,b}{\text{minimize}} \;\; \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (w \cdot x_i + b)\right)^2$$

This is an **(unconstrained) optimization problem** in the variables (w, b). The **optimal solution** will be our model.

- **Solution Approach 2:** Solve using optimization techniques, e.g., **gradient descent**.

Initalize: $w = w_0$,$b = b_0$,$t = 0$
Iterate until convergence
Compute updates:
$$w_{t+1} = w_t - \eta_t \nabla_w L(f(x), y)$$
$$b_{t+1} = b_t - \eta_t \nabla_b L(f(x), y)$$
Check for convergence
Continue to next iteration: $t = t + 1$





Multivariate Linear Regression

Problem Setup: Given data ($\boldsymbol{x}_i$) and real-valued labels (y_i), find the best model that **fits current data and predicts future data**

Example: Develop a model to predict produce yield depending on **multiple factors** such as precipitation, average manure usage, temperature, plant spacing, and relative humidity.

Here, the independent variables (training data) are denoted $\boldsymbol{x}_i$ and the dependent variable (label) is yield (y_i).

Our hypothesis class is the space of all **multivariate linear functions**, $y = f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$

*the model is **multivariate** because there are many independent (training) variables*

*the model is still **linear** because the highest allowed degree is x^1 in each dimension of $\boldsymbol{x}$*

*the **intercept** can be absorbed into the inner-product by augmenting the data $\widehat{\boldsymbol{x}} = [\boldsymbol{x}, 1]$ and by augmenting the weights $\widehat{\boldsymbol{w}} = [\boldsymbol{w}, b]$ such that $\widehat{\boldsymbol{w}}^T\widehat{\boldsymbol{x}} = \boldsymbol{w}^T\boldsymbol{x} + b \cdot 1$*

Precip. [mm]	Manure [kg/m²]	Temperat. [°C]	Spacing [m]	Humid. [%]	Yield [kg/m²]
22	1.5	33.1	1.0	32.5	**0.36**
11	0.75	27.9	1.5	45.0	**0.09**
94	0.85	28.5	1.0	78.0	**0.67**
62	3.0	22.6	2.0	55.0	**0.44**
84	4.25	35.4	1.0	68.5	**0.72**
14	1.25	34.4	0.75	72.0	**0.24**
104	2.75	19.3	0.5	37.5	**0.33**

*each row $\boldsymbol{x}_i^T$ corresponds to a multi-dimensional training example, represented as a **column vector**, $\boldsymbol{x}_i$*

the goal is to predict the label, y_i, as a function of the multiple factors

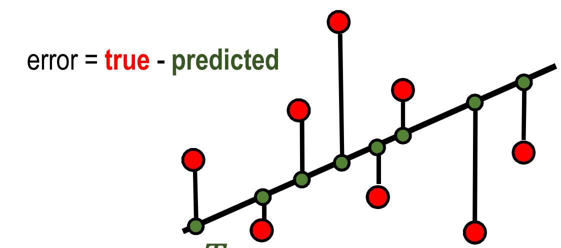
Multivariate Linear Regression

Problem Setup: Given data $(\boldsymbol{x}_i)$ and real-valued labels (y_i), find the best model that **fits current data and predicts future data**

Example: Develop a model to predict produce yield depending on **multiple factors** such as precipitation, average manure usage, temperature, plant spacing, and relative humidity.

Here, the independent variables (training data) are denoted $\boldsymbol{x}_i$ and the dependent variable (label) is yield (y_i).

*The **loss function** is still the squared loss, $\frac{1}{2}e_i^2$, though the error is measured in d-dimensional space via the **inner-product** $\boldsymbol{w}^T\boldsymbol{x}_i$*



$$e_i = y_i - \boldsymbol{w}^T\boldsymbol{x}_i$$

$$L(f(x_i), y_i) = \frac{1}{2}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2$$

Precip. [mm]	Manure [kg/m²]	Temp. [°C]	Spacing [m]	Humid. [%]	Yield [kg/m²]
22	1.5	33.1	1.0	32.5	**0.36**
11	0.75	27.9	1.5	45.0	**0.09**
94	0.85	28.5	1.0	78.0	**0.67**
62	3.0	22.6	2.0	55.0	**0.44**
84	4.25	35.4	1.0	68.5	**0.72**
14	1.25	34.4	0.75	72.0	**0.24**
104	2.75	19.3	0.5	37.5	**0.33**

*All the training examples are collected into a **matrix of training data** X, where each row is a training example*

$$X = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_i^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix}$$

*Note the **transpose** to denote that multivariate training examples (which are column vectors) are transposed to rows in the data matrix*

Multivariate Linear Regression

Problem Setup: Given data $(\boldsymbol{x}_i)$ and real-valued labels (y_i), find the best model that **fits current data** <u>**and**</u> **predicts future data**

Problem: Given n training examples $(\boldsymbol{x}_i, y_i)$, $i = 1, \dots, n$, find the best model $\boldsymbol{w}$ by solving

$$\underset{\boldsymbol{w}}{\text{minimize}} \ \ \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2$$

This expression can be written more compactly in **vector notation**

$$\underset{\boldsymbol{w}}{\text{minimize}} \ \ \frac{1}{n}(\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w})$$

and fully **expanded** into:

$$\underset{\boldsymbol{w}}{\text{minimize}} \ \ \frac{1}{n}(\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T X\boldsymbol{w} + \boldsymbol{w}^T X^T X\boldsymbol{w})$$

Precip. [mm]	Manure [kg/m²]	Temp. [°C]	Spacing [m]	Humid. [%]	Yield [kg/m²]
22	1.5	33.1	1.0	32.5	**0.36**
11	0.75	27.9	1.5	45.0	**0.09**
94	0.85	28.5	1.0	78.0	**0.67**
62	3.0	22.6	2.0	55.0	**0.44**
84	4.25	35.4	1.0	68.5	**0.72**
14	1.25	34.4	0.75	72.0	**0.24**
104	2.75	19.3	0.5	37.5	**0.33**

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix}$$

$$X = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_i^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix}$$

All the training examples are collected into a ***matrix of training data*** *X, where each row is a training example*

Note the ***transpose*** *to denote that multivariate training examples (which are column vectors) are transposed to rows in the data matrix*

Multivariate Linear Regression

Problem Setup: Given data $(\boldsymbol{x}_i)$ and real-valued labels (y_i), find the best model that **fits current data <u>and</u> predicts future data**

Problem: Given n training examples $(\boldsymbol{x}_i, y_i)$, $i = 1, \dots, n$, find the best model $\boldsymbol{w}$ by solving

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \frac{1}{n}(\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T X\boldsymbol{w} + \boldsymbol{w}^T X^T X\boldsymbol{w})$$

The solution to this problem is the **ordinary least squares estimator**

$$w = (X^T X)^{-1} X^T \boldsymbol{y}$$

solution depends on the inverse of the ***covariance matrix*** *$C = X^T X$, which can be* ***ill-conditioned***

unique closed-form solution*, provided that number of data points (n) exceeds data dimension (d)*

$(X^T X)^{-1} X^T = X^+$ is called the **pseudo-inverse**

Ridge Regression

Problem Setup: Given data ($\boldsymbol{x}_i$) and real-valued labels (y_i), find the best model that **fits current data and predicts future data**

Problem: Given n training examples $(\boldsymbol{x}_i, y_i)$, $i = 1, \dots, n$, find the most robust model $\boldsymbol{w}$ by solving (for $\lambda > 0$)

$$\underset{\boldsymbol{w}}{\text{minimize}} \ \frac{1}{n}(\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w}) + \lambda \boldsymbol{w}^T\boldsymbol{w}$$

$\boldsymbol{w}^T\boldsymbol{w}$ is a **regularization term** that is used to overcome ill-conditioning, $\lambda > 0$ is the **regularization parameter**, which is **tunable**

The solution to this problem is the **regularized least squares estimator**

$$w = (X^TX + \lambda I_d)^{-1}X\boldsymbol{y}$$

for $\lambda > 0$, *inverse is can always be computed, algorithm more* ***robust***

Exercise: *Derive the regularized least squares estimator from the optimization formulation for Ridge Regression.*

Ridge Regression and the Bias-Variance Tradeoff

Problem Setup: Given data $(\boldsymbol{x}_i)$ and real-valued labels (y_i), find the best model that **fits current data and predicts future data**

Problem: Given n training examples $(\boldsymbol{x}_i, y_i)$, $i = 1, \dots, n$, find the most robust model $\boldsymbol{w}$ by solving (for $\lambda > 0$)

$$\underset{\boldsymbol{w}}{\text{minimize}} \;\; \frac{1}{n}(\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w}) + \lambda \boldsymbol{w}^T\boldsymbol{w}$$

$\boldsymbol{w}^T\boldsymbol{w}$ is a **regularization term** that is used to overcome ill-conditioning, $\lambda > 0$ is the **regularization parameter**, which is **tunable**

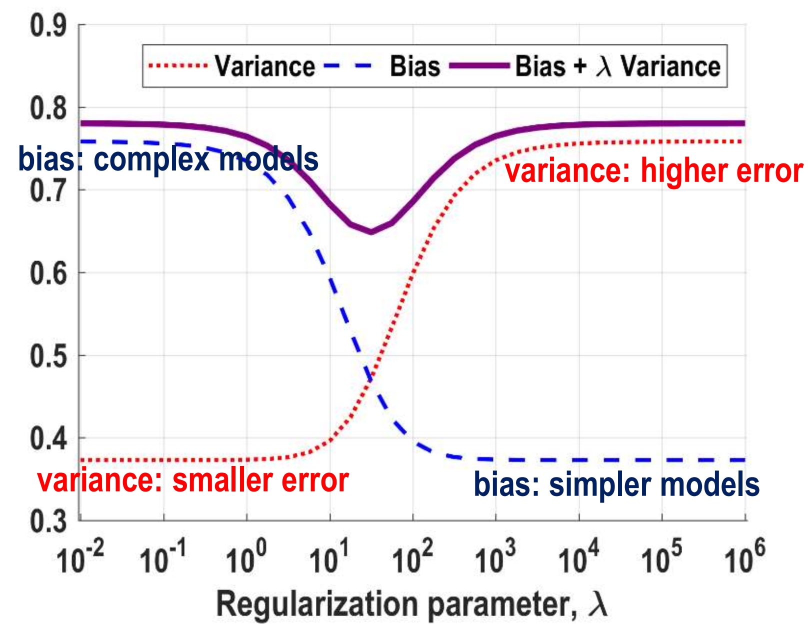
The solution to this problem is the **ordinary least squares estimator**

$$w = (X^TX + \lambda I_d)^{-1}X\boldsymbol{y}$$

for $\lambda > 0$, *inverse is can always be computed, algorithm more* ***robust***

$\lambda > 0$ can be **tuned** to train different models with different behaviors:

- λ controls the **amount of regularization**
- as $\lambda \downarrow 0$, the model focuses on **minimizing error** (**variance**) and **overfits** the data
  - when the model is too complex and trivially fits the data (i.e., too many parameters)
  - when the data is not enough to estimate the parameters
  - model captures the noise (or the chance)
- as $\lambda \uparrow \infty$, the model focuses on **shrinking the coefficients** $\boldsymbol{w}$ (**bias**) and **underfits** the data

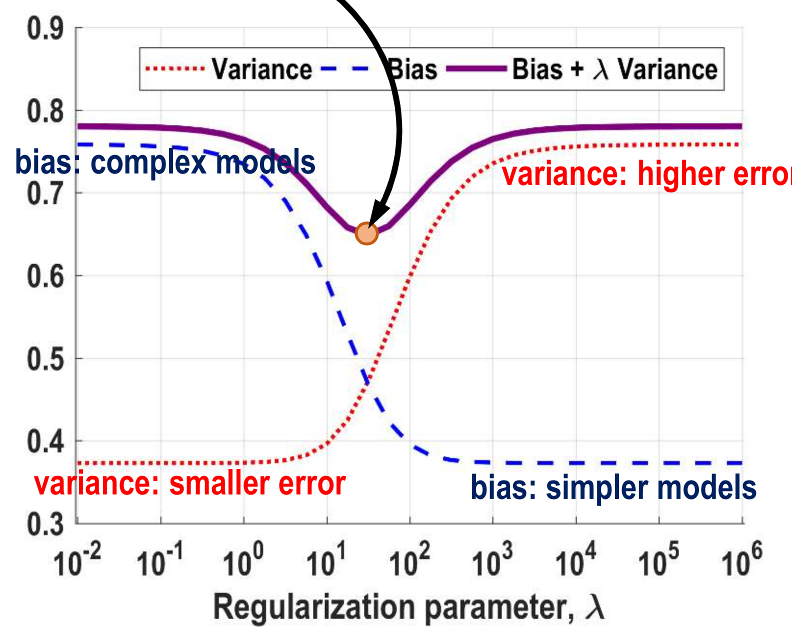


Bias-Variance Tradeoff

The best model is the one that **generalizes well**, that is, the best model **trades-off effectively between bias and variance** and can be expected to perform well on **future data**.

$\lambda > 0$ can be **tuned** to train different models with different behaviors:

- λ controls the **amount of regularization**
- as $\lambda \downarrow 0$, the model focuses on **minimizing error** (**variance**) and **overfits** the data
- as $\lambda \uparrow \infty$, the model focuses on **shrinking the coefficients w** (**bias**) and **underfits** the data



All machine-learning algorithms will exhibit this **bias-variance tradeoff**; selecting the **best model parameters** is an **important practical aspect** of machine-learning.