L7. Machine Learning (Regression)

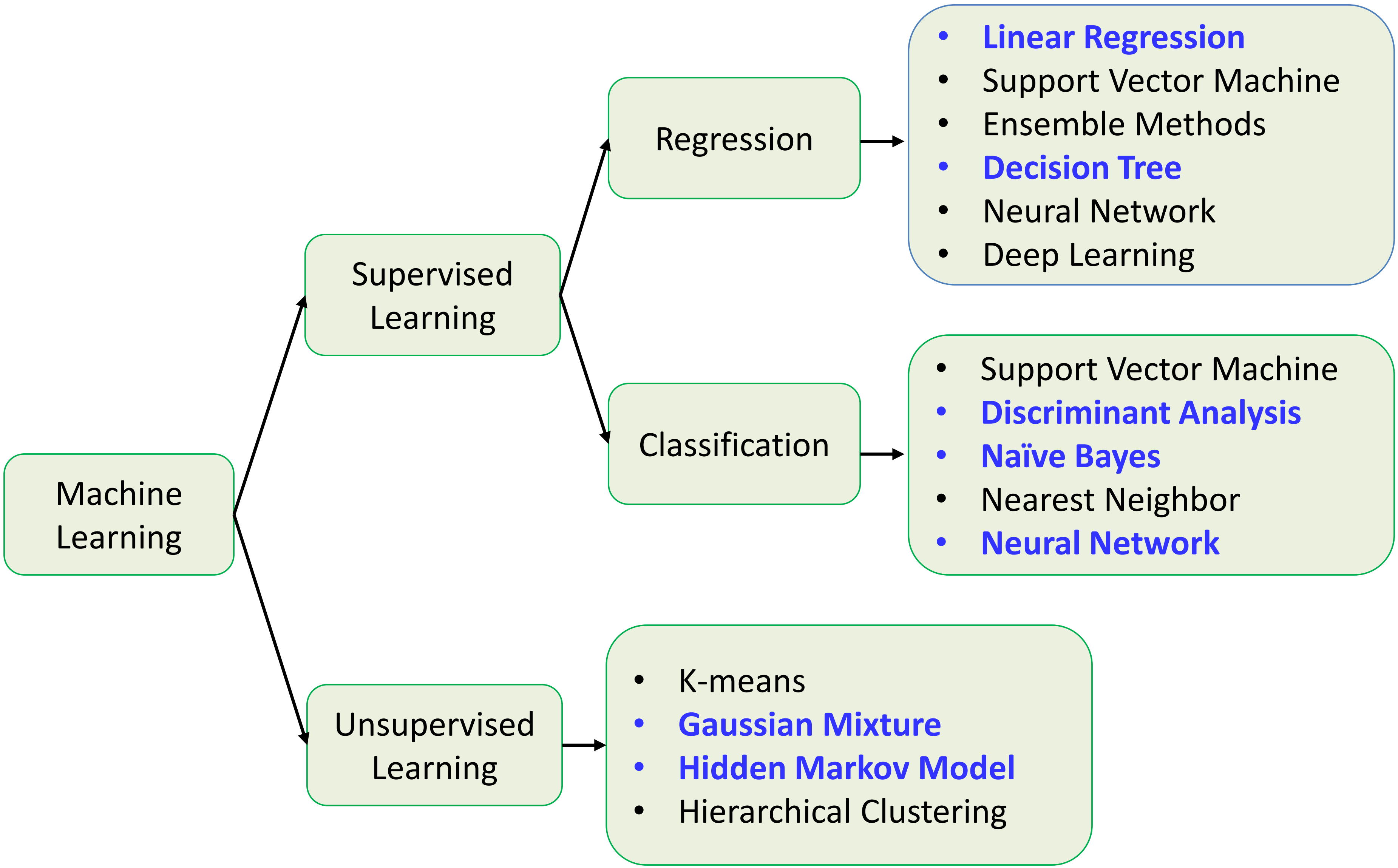
https://github.com/JWarmenhoven/ISLR-python

What is Machine Learning?

data + model = prediction

- Data : observations, experience,…
- Model: a form of prior knowledge, assumptions, belief
  - ✓ Functional model
  - ✓ Probabilistic model
- Prediction : the new knowledge obtained by combining the data and model
  - ✓ Regression
  - ✓ Classification
  - ✓ Clustering

What is Machine Learning?



Supervised learning

Data

Training data set D

$D = \{(x_i, y_i); i = 1, \ldots, m\}$

$\boldsymbol{x}_i = (x_{i1}, \ \cdots, x_{in})$

$x_i = (x_{i1}, \ \cdots, x_{in})$

m input Feature vectors

x_1	x_2	$\cdots$	x_n	y
x_{11}	x_{12}	$\cdots$	x_{1n}	y_1
x_{21}	x_{22}	$\cdots$	x_{2n}	y_2
$\vdots$	$\vdots$		$\vdots$	$\vdots$
x_{m1}	x_{m2}	$\cdots$	x_{mn}	y_m

m outputs

model

Functional form $f(x; \theta)$
Is usually given

Learning Algorithm

Using training data set, a learning algorithm finds the best hypothesis function $h(x)$ that **is believed to accurately predict** the output y for a given query input x

$\hat{\theta}$

prediction

Query input

x_*

Hypothesis
$f(x; \theta^*)$

Predicted output

y_*

Input feature vector

$x_* = (x_{*1}, x_{*2}, \ldots x_{*n})$

if $y_* \in \mathbb{R}$: **Regression**

if $y_* \in \{1, \ldots, N\}$: **Classification**

Two different learning approaches

- **Machine Learning as Optimization**
  - ✓ Relate variables through a basis function (parametric function)
  - ✓ Formulate learning problem as an optimization problem
  - ✓ Employ optimization algorithm to solve the formulated problem

- **Machine Learning as Probabilistic Modeling (not necessarily Bayesian)**
  - ✓ Relate variables through probability distributions
  - ✓ Formulate learning problem as inference
  - ✓ If Bayesian, treat parameters with probability distributions
  - ✓ Requires inference methods (integral or sampling) to solve the formulated problem

Let’s explore different views on Machine Learning by taking a linear regression as an example

nature International weekly journal of science

NATURE | INSIGHT | REVIEW

Probabilistic machine learning and artificial intelligence

Zoubin Ghahramani

Nature **521**, 452–459 (28 May 2015) | doi:10.1038/nature14541
Received 12 February 2015 | Accepted 21 April 2015 | Published online 27 May 2015

PDF | Citation | Reprints | Rights & permissions | Article metrics

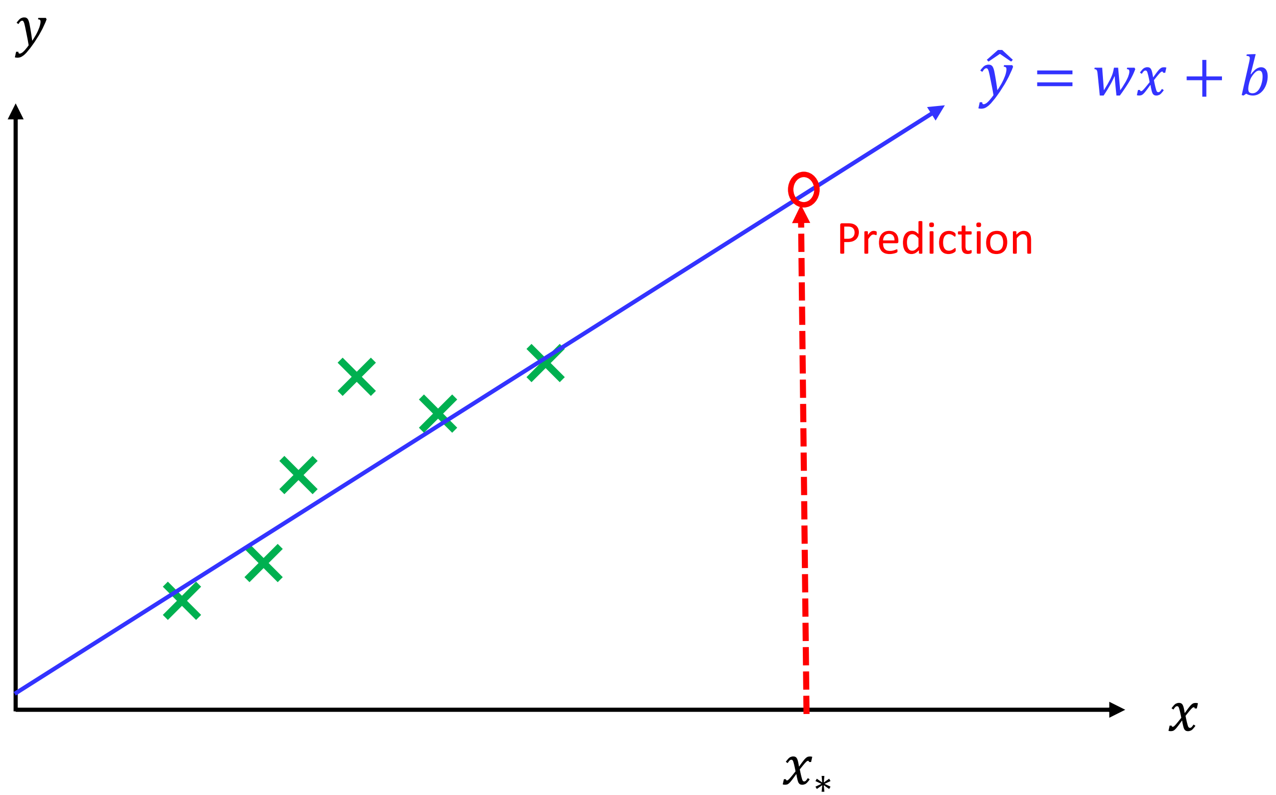
Abstract

How can a machine learn from experience? Probabilistic modelling provides a framework for understanding what learning is, and has therefore emerged as one of the principal theoretical and practical approaches for designing machines that learn from data acquired through experience. The probabilistic framework, which describes how to represent and manipulate uncertainty about models and predictions, has a central role in scientific data analysis, machine learning, robotics, cognitive science and artificial intelligence. This Review provides an introduction to this framework, and discusses some of the state-of-the-art advances in the field, namely, probabilistic programming, Bayesian optimization, data compression and automatic model discovery.

Load Map

1. Optimization Approach (Normal Equation)
2. Maximum Likelihood Estimation (MLE) Approach
3. Maximum A Posteriori Estimation (MAP) Approach
4. Full Bayesian Approach
   - ✓ Analytical approach
   - ✓ Sampling approach
5. Regularization regression (Ridge and Lasso)
   - ✓ Optimization view
   - ✓ Bayesian View

1D Linear Regression

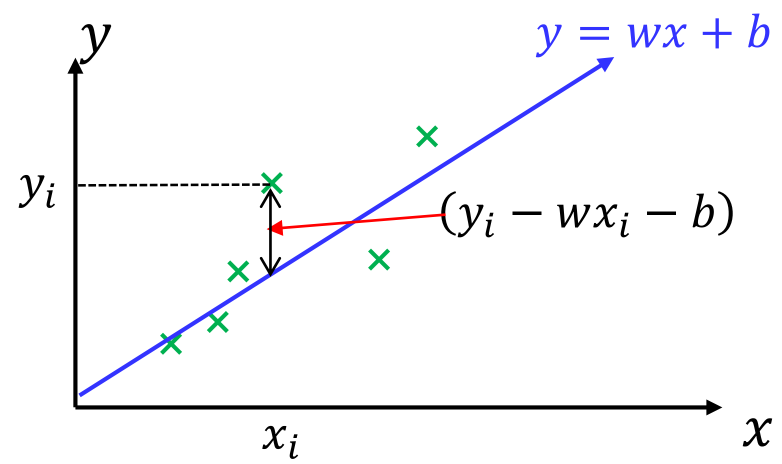


- **Data** : $(x_1, y_1), \dots, (x_m, y_m)$
- **Model**: Linera model $\hat{y} = wx + b$ ($y = w^T x + b$ for multidimensional)
- **Learning**: What are w and b?
- **Prediction** : What is $\hat{y}_* = wx_* + b$

Learning as optimization

- Define an objective (cost) function

$$J(w,b) = \sum_{i=1}^{m} (y_i - wx_i - b)^2$$



- Minimize the error function with respect to w and b

$$\frac{dJ(w,b)}{dw} = -2\sum_{i=1}^{m} x_i(y_i - wx_i - b) \; = 0 \rightarrow w^* = \frac{\sum_{i=1}^{m}(y_i - b)x_i}{\sum_{i=1}^{n} x_i^2}$$

$$\frac{dJ(w,b)}{db} = -2\sum_{i=1}^{m} (y_i - wx_i - b) \; = 0 \rightarrow b^* = \frac{\sum_{i=1}^{m}(y_i - wx_i)}{n}$$

Learning as optimization

Notation for general cases $x_i \in \mathbb{R}^n$

- A linear regression model

$$\hat{y}_i = w_0 + w_1 x_{i1} + \cdots + w_n x_{in}$$

with $w = \left(w_{0,} w_1, \dots, w_n\right)^T$ and $x_i = (x_{i1}, \dots, x_{in})^T$

- If we introduce $x_{i0} = 1$,

$$\hat{y}_i = w^T x_i$$

with $w = (w_0, w_1, \dots, w_n)^T$ and $x_i = (\underset{\substack{\uparrow \\ 1}}{x_{i0}}, x_{i1}, \dots, x_{in})^T$

- In a Matrix form

$$\begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_m \end{pmatrix} = \begin{pmatrix} -x_1^T - \\ \vdots \\ -x_m^T - \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} x_1^T w \\ \vdots \\ x_m^T w \end{pmatrix} \quad \rightarrow \hat{y} = Xw$$

m: # of data points

with $\hat{y} = (\hat{y}_1, \dots, \hat{y}_m)^T, \quad X = \begin{pmatrix} -x_1^T - \\ \vdots \\ -x_m^T - \end{pmatrix}$

Learning as optimization (Normal Equation)

- The cost function for the optimization can be defined as :

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}\left(x_i^T w - y_i\right)^2 = \frac{1}{2}\|y - wX\|_2^2 = \frac{1}{2}(Xw - y\,)^T(Xw - y\,)$$

$$\sqrt{\Sigma_{i=1}^{m} z_i^2} \quad = \quad \|z\|_2 \quad = \quad \sqrt{z^T z}$$

- The optimum parameters $\widehat{w}$ cam be computed as one minizing the cost function :

$$\widehat{w} = \arg\min_{w} J(w) = \arg\min_{w} \frac{1}{2}\|y - wX\|_2^2$$

- For reference, other norms are summarized here :

$$\|z\|_1 = \sqrt{\Sigma_{i=1}^{n}|z_i|}, \qquad \|z\|_p = \sqrt{\Sigma_{i=1}^{n}|z_i|^p}, \qquad \|z\|_\infty = \max_i |z_i|$$

Learning as optimization (Normal Equation)

Linear Algebra Approach for finding the optimum parameters:

$$\hat{w} = \arg\min_{w} J(w) = \arg\min_{w} \frac{1}{2} \|y - wX\|_2^2$$

Optimality condition : $\nabla_w J(w) = 0$ at $\hat{w}$

$$\begin{aligned}
\nabla_w J(w) &= \nabla_w \frac{1}{2}(Xw - y)^T (Xw - y) \\
&= \frac{1}{2} \nabla_w (w^T X^T X w - w^T X^T y - y^T X w + y^T y) \\
&= \frac{1}{2} \nabla_w \mathrm{tr}(w^T X^T X w - w^T X^T y - y^T X w + y^T y) \\
&= \frac{1}{2} \nabla_w \big(\mathrm{tr}(w^T X^T X w) - 2\mathrm{tr}(y^T X w)\big) \\
&= \frac{1}{2} (X^T X w + X^T X w - 2X^T y) \\
&= X^T X w - X^T y
\end{aligned}$$

$$\nabla_w J(w) = X^T X w - X^T y = 0$$

$$\rightarrow X^T X w = X^T y$$

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

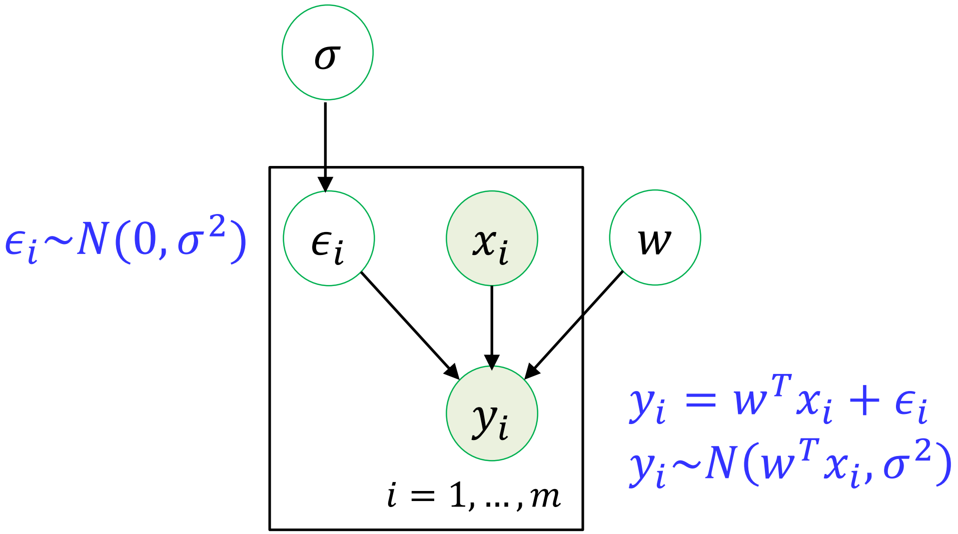
Probabilistic view on linear regression

- Assume there is uncertainty in the predicted value :

$$y_i = w^T x_i + \epsilon_i \text{ with } \epsilon_i \sim N(0, \sigma^2)$$

- Then the probabilistic model on output y_i can be represented as

$$y_i \sim N(w^T x_i, \sigma^2) \quad \text{or} \quad p(y_i | w^T x_i, \sigma) = N(y_i | w^T x_i, \sigma^2)$$



An error ϵ_i is independently identically distributed (i.i.d assumption)

- The likelihood of the data is defined as

$$p(y|X, w, \sigma) = \prod_{i=1}^{m} N(y_i | w^T x_i, \sigma^2) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right)$$

Learning as probabilistic model (MLE Approach)

- The log likelihood is

$$
\begin{aligned}
L(w,\sigma) &= \log p(y|X,w,\sigma) \\
&= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right) \\
&= \log \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m \exp\left(-\sum_{i=1}^{m} \frac{(y_i - w^T x_i)^2}{2\sigma^2}\right) \\
&= m \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{\sigma^2} \frac{1}{2} \sum_{i=1}^{m} (y_i - w^T x_i)^2 \\
&= m \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{\sigma^2} J(w)
\end{aligned}
$$

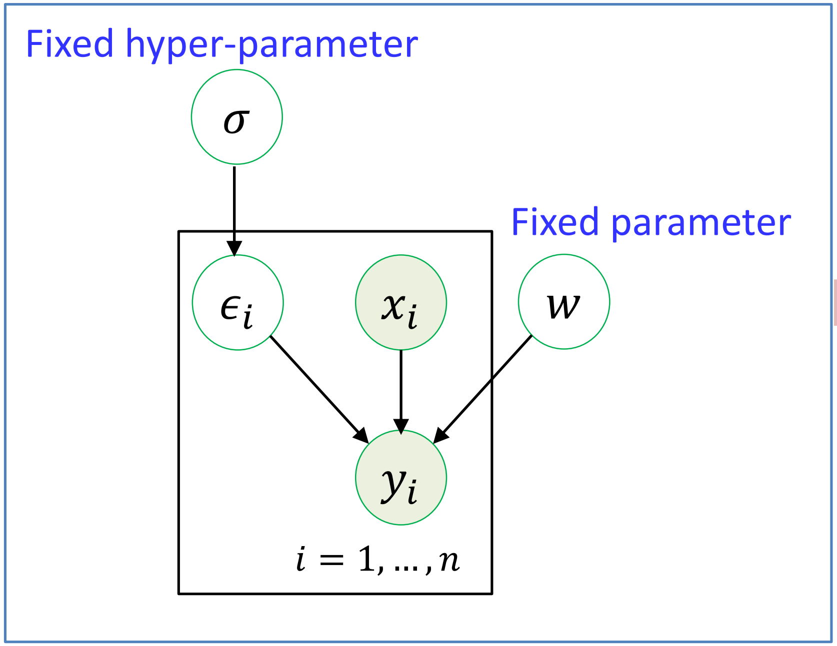
- The optimum parameters is determined by maximizing log likelihood

$$
(w^*, \sigma) = \max_{(w,\sigma)} L(w,\sigma) = \max_{(w,\sigma)} \log p(y|X,w,\sigma)
$$

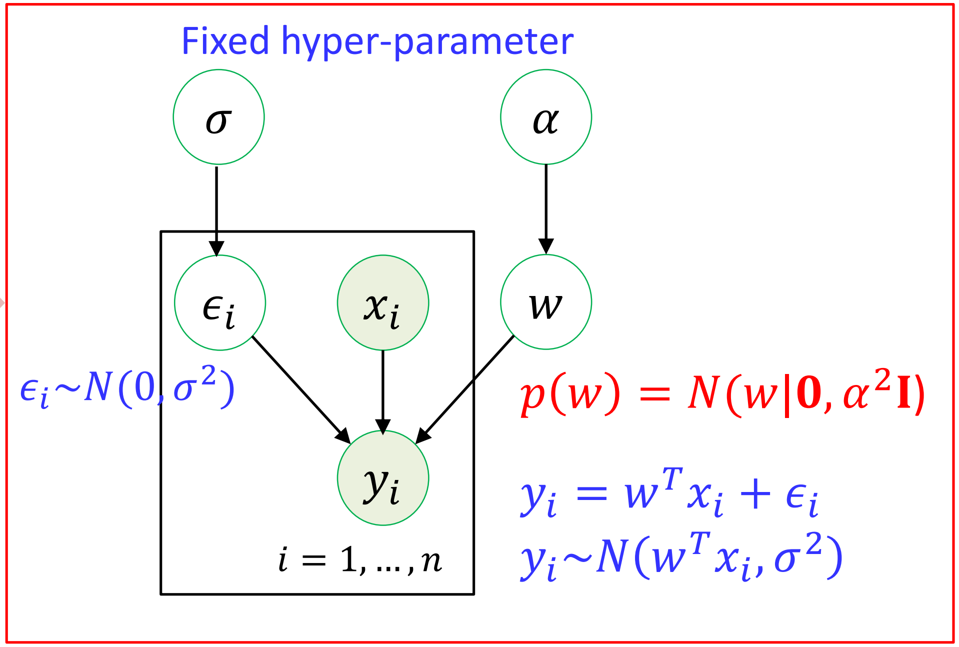
Minimizing the square error sum $J(w) =$

Maximizing the log likelihood $\log p(y|X,w,\sigma)$ with respect to w

MLE approach (point estimation)



Bayesian approach



- Consider the parameter w as stochastic variables (represented as a distribution)
- (Assume σ is known for simple derivation)
- Find the distribution on parameter w

$$p(w|y, X) = \frac{p(y|X, w)p(w|X)}{\int_w p(y|X, w)p(w|X)dw} \quad \rightarrow \quad p(w|y) = \frac{p(y|\, w)p(w)}{\int_w p(y|\, w)p(w)dw}$$

We will assume X is fixed for the data y

Learning as probabilistic model (Bayesian Approach)

- Multivariate regression likelihood is

$$
\begin{aligned}
p(y|w) &= \prod_{i=1}^{m} p(y_i|x_i, w) \\
&= \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - w^T x_i)^2\right)
\end{aligned}
$$

m = # of data points

- Multivariate Gaussian prior on parameter w

$$p(w) = N(w|\mathbf{0}, \alpha^2\mathbf{I})$$

$$p(w) = \frac{1}{(2\pi\alpha^2)^{n/2}} \exp\left(-\frac{1}{2\alpha^2} w^T w\right)$$

n = Dimension of w

Learning as probabilistic model (Bayesian Approach)

- **We want to find the posterior**

$$p(w|X,y) \propto p(y|X,w)p(w)$$
$$= \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - w^T x_i)^2\right)\frac{1}{(2\pi\alpha^2)^{k/2}} \exp\left(-\frac{1}{2\alpha^2} w^T w\right)$$

- **Take log:**

$$\log p(w|X,y) = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - w^T x_i)^2 - \frac{1}{2\alpha^2} w^T w + const$$
$$= -\frac{1}{2\sigma^2}\sum_{i=1}^{m} y_i^2 + \frac{1}{\sigma^2}\sum_{i=1}^{m} y_i x_i^T w - \frac{1}{2\sigma^2}\sum_{i=1}^{m} w^T x_i x_i^T w - \frac{1}{2\alpha^2} w^T w + const$$
$$= -\frac{1}{2\sigma^2} y^T y + \frac{1}{\sigma^2} y^T X w - \frac{1}{2\sigma^2} w^T X^T X w - \frac{1}{2\alpha^2} w^T w + const$$
$$= -\frac{1}{2\sigma^2} y^T y + \frac{1}{\sigma^2} y^T X w - \frac{1}{2} w^T \left[\frac{1}{\sigma^2} X^T X + \frac{1}{\alpha^2} I\right] w + const$$

- **Posterior distribution is**

$$p(w|X,y) = N(w|\mu_w, \Sigma_w)$$

$$\mu_w = \Sigma_w\left(\frac{1}{\sigma^2} X^T y\right) \qquad \Sigma_w = \left[\frac{1}{\sigma^2} X^T X + \frac{1}{\alpha^2} I\right]^{-1}$$

Learning as probabilistic model (Bayesian Approach)

- Predicative distribution

$$p(y_*|x_*, X, y) = \int_w p(y_*|x_*, w)p(w|X, y)dw$$

Jupyter Demo Simulation
Bayesian Regression Analytical

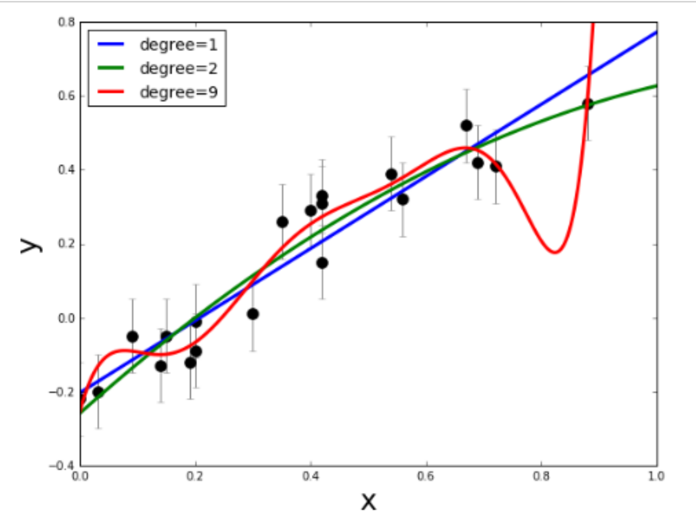
Learning as probabilistic model (Bayesian Approach)

- Predicative distribution

$$p(y_*|x_*, X, y) = \int_w p(y_*|x_*, w)p(w|X, y)dw$$

Jupyter Demo Simulation
Bayesian Regression (Sampling using PyMC)

Regularized linear regression



What is a good regression function?

- The goal of regression is to come up with some good prediction function:

$$\hat{f}(x) = \widehat{w}^T x^T$$

- So far, we have found $\widehat{w}$ by finding (Ordinary Least Square Estimation)

$$\widehat{w} = \arg\min_{w} J(w) = \arg\min_{w} \frac{1}{2} \|y - wX\|_2^2$$

- To see $\hat{f}(x)$ is good candidate, we need to check

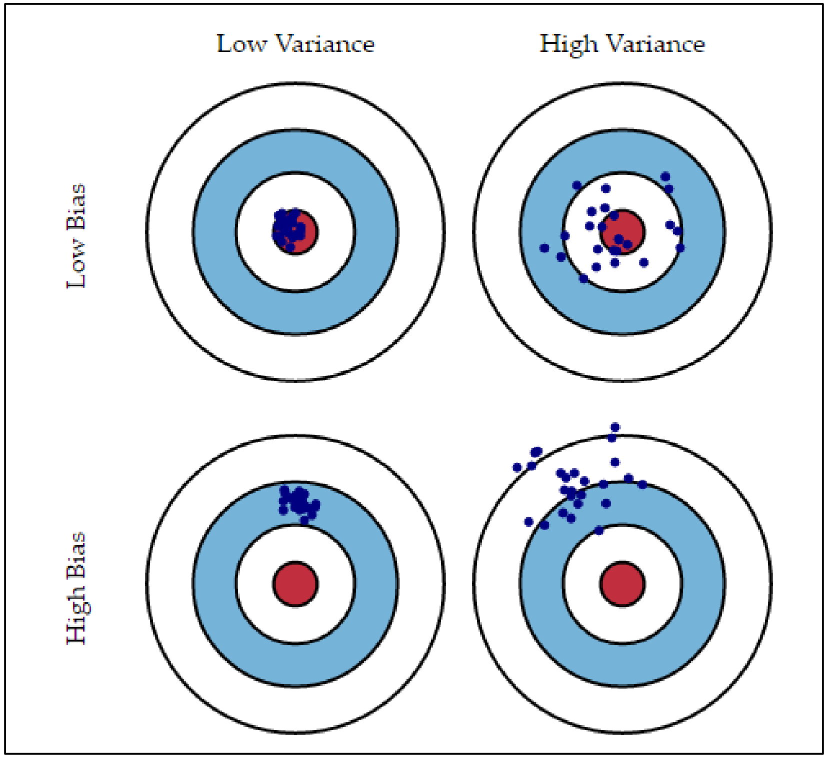
  ✓ Is $\widehat{w}$ close to the true w

  ✓ Will $\hat{f}(x)$ fit future observation well? (Generalization)

Regularized linear regression

Jupyter Demo Simulation
Regularization Motivation Example

The Bias and Variance Trade off



Each hit represents an individual realization of our model, given the chance variability in the training data we gather.

- Imagine you could repeat the whole model building process more than once: each time you gather new data and run a new analysis creating a new model.

- Due to randomness in the underlying data sets, the resulting models will have a range of predictions.
  - **Bias** measures how far off in general these models' predictions are from the correct value
  - **Variance** measures he variability of a model prediction for a given data point

Figure source : http://scott.fortmann-roe.com/docs/BiasVariance.html

The Bias and Variance Trade off

Estimation : $\hat{f}(x) = x^T \hat{w}$
True : $f(x)$
Observation : $y = f(x) + \epsilon, \epsilon \sim N(0, \sigma^2)$

- *The expected prediction error* of a regression fit $\hat{f}(x_0)$, using square-loss error :

$$\begin{aligned}
\mathrm{EPP}(x_0) &= \mathrm{E}\left[\left(y - \hat{f}(x_0)\right)^2 \middle| x_0\right] \\
&= \mathrm{E}\left[y^2 + \hat{f}(x_0)^2 - 2y\hat{f}(x_0) \middle| x_0\right] \\
&= \mathrm{E}[y^2|x_0] + \mathrm{E}\left[\hat{f}(x_0)^2\right] - \mathrm{E}\left[2y\hat{f}(x_0) \middle| x_0\right] \\
&= \mathrm{E}[y^2|x_0] + \mathrm{E}\left[\hat{f}(x_0)^2\right] - 2f(x_0)\mathrm{E}\left[\hat{f}(x_0)\right]
\end{aligned}$$

$$\begin{aligned}
\mathrm{E}\left[2y\hat{f}(x_0) \middle| x_0\right] &= \mathrm{E}\left[2(f(x) + \epsilon)\hat{f}(x_0) \middle| x_0\right] \\
&= 2\mathrm{E}\left[f(x)\hat{f}(x_0) \middle| x_0\right] + 2\mathrm{E}\left[\epsilon\hat{f}(x_0) \middle| x_0\right] \\
&= 2f(x_0)\mathrm{E}\left[\hat{f}(x_0)\right] + 2\mathrm{E}\left[\epsilon\hat{f}(x_0) \middle| x_0\right] \quad (\because f(x_0) \text{ is constant}) \\
&= 2f(x_0)\mathrm{E}\left[\hat{f}(x_0)\right] \quad (\because \epsilon \perp \hat{f}(x_0))
\end{aligned}$$

The Bias and Variance Trade off

Estimation : $\hat{f}(x) = x^T \hat{w}$
True : $f(x)$
Observation : $y = f(x) + \epsilon, \epsilon \sim N(0, \sigma^2)$

- *The expected prediction error* of a regression fit $\hat{f}(x_0)$, using square-loss error :

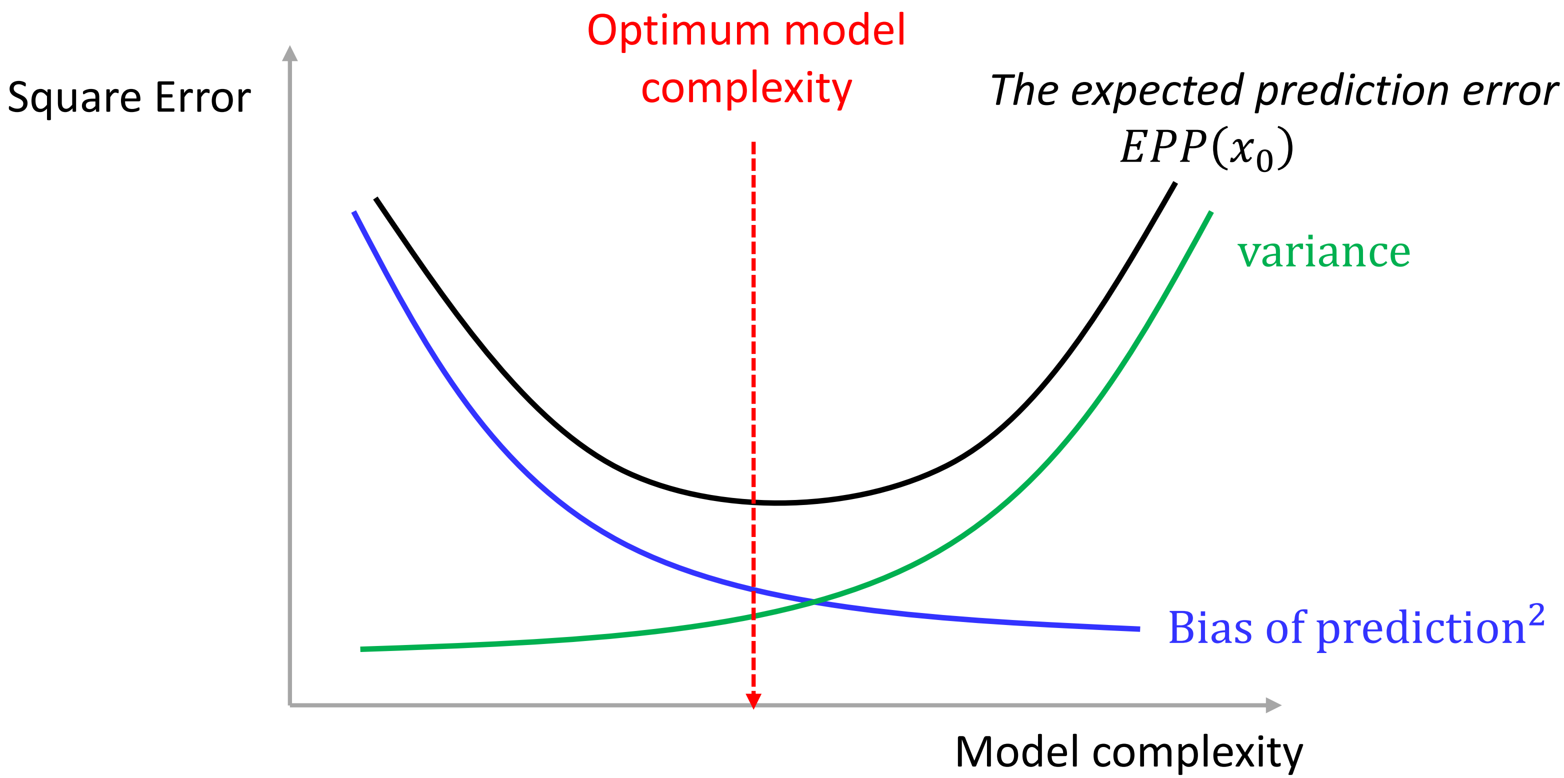
E is over the training data

$$
\begin{aligned}
\mathrm{EPP}(x_0) &= \mathrm{E}\left[\left(y - \hat{f}(x_0)\right)^2 \middle| x_0\right] \\
&= \mathrm{E}\left[y^2 + \hat{f}(x_0)^2 - 2y\hat{f}(x_0) \middle| x_0\right] \\
&= \mathrm{E}[y^2|x_0] + \mathrm{E}\left[\hat{f}(x_0)^2\right] - \mathrm{E}\left[2y\hat{f}(x_0)\middle|x_0\right] \\
&= \mathrm{E}[y^2|x_0] + \mathrm{E}\left[\hat{f}(x_0)^2\right] - 2f(x_0)\mathrm{E}\left[\hat{f}(x_0)\right] \\
&= \mathrm{var}[y|x_0] + \mathrm{E}[y|x_0]^2 + \mathrm{var}\left[\hat{f}(x_0)\right] + \mathrm{E}\left[\hat{f}(x_0)\right]^2 - 2f(x_0)\mathrm{E}\left[\hat{f}(x_0)\right] \\
&= \mathrm{var}[y|x_0] + \mathrm{var}\left[\hat{f}(x_0)\right] + \mathrm{E}[y|x_0]^2 + \mathrm{E}\left[\hat{f}(x_0)\right]^2 - 2f(x_0)\mathrm{E}\left[\hat{f}(x_0)\right] \\
&= \mathrm{var}[y|x_0] + \mathrm{var}\left[\hat{f}(x_0)\right] + f(x_0)^2 + \mathrm{E}\left[\hat{f}(x_0)\right]^2 - 2f(x_0)\mathrm{E}\left[\hat{f}(x_0)\right] \\
&= \mathrm{var}[y|x_0] + \mathrm{var}\left[\hat{f}(x_0)\right] + \left(f(x_0) - \mathrm{E}\left[\hat{f}(x_0)\right]\right)^2 \\
&= \mathrm{var}[y|x_0] + \mathrm{E}\left[\left(\hat{f}(x_0) - \mathrm{E}\left[\hat{f}(x_0)\right]\right)^2\right] + \left(f(x_0) - \mathrm{E}\left[\hat{f}(x_0)\right]\right)^2 \\
&= \quad \sigma^2 \quad + \text{variance of prediction} \quad + \text{Bias of prediction}^2
\end{aligned}
$$

Irreducible

Can balance

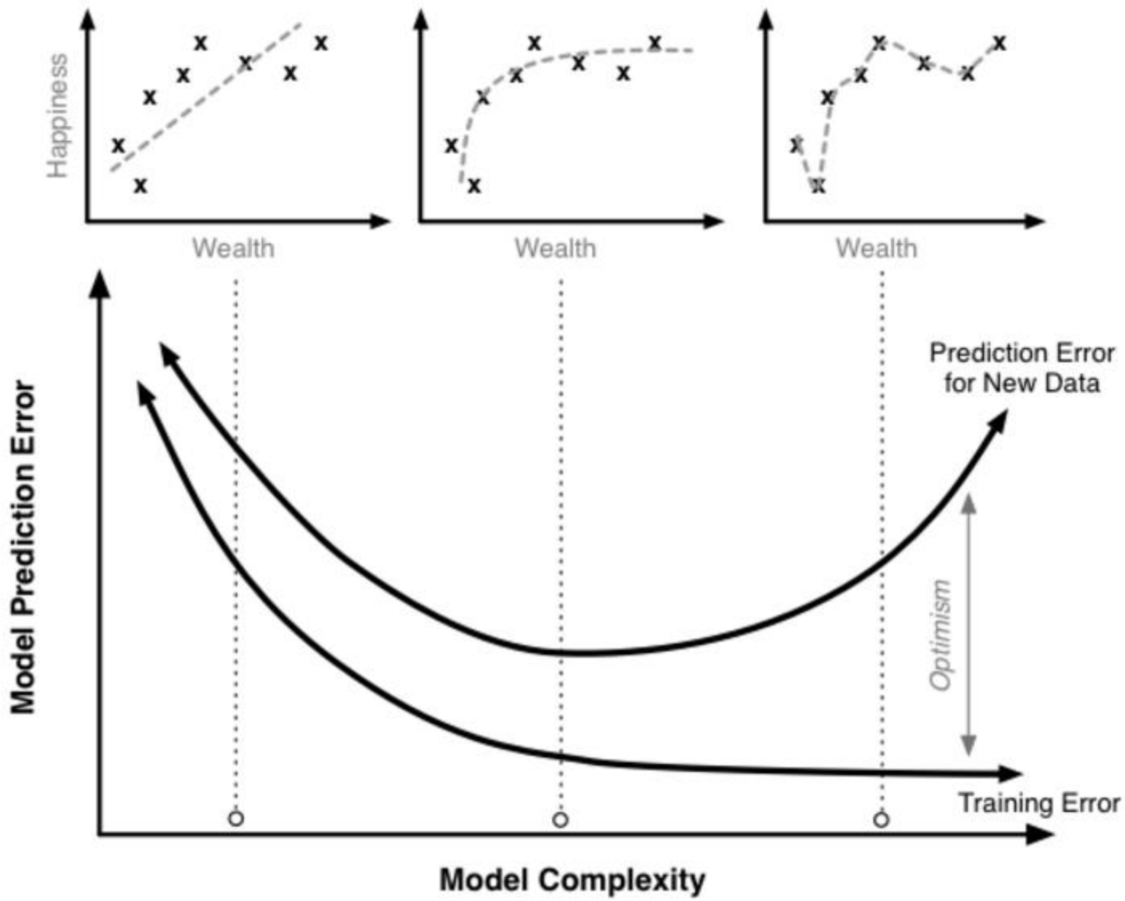
The Bias and Variance Trade off



Model complexity is related with

- The number of model parameters
- The size of model parameters
- …

How to measure prediction error



How to measure (estimate) model prediction error?

- Statistical measure (i.e., R^2 value)
- Information Theoretic Approaches (i.e., BIC, AIC measure)
- Holdout set or Cross Validation (Training data vs Test data set)

Ordinary Least Square Estimation

- OLS estimates find the parameter that minimize the bias between the predicted and true values :

$$\hat{w} = \underset{w}{\operatorname{argmin}} \|y - wX\|_2^2$$

- OLS estimates often have low bias but large variance

→ Poor generalization toward unseen test data set

- All features have a weight

→ Smaller subset with strong effects is more interpretable

- $w_i's$ are unconstrained

→ They can explore and hence are susceptible to very high variance

We nee some shrinkage (or regulation) to constraint $\hat{w}$

Regularized linear regression

Ridge regression

- Ridge regression introduces a regularization with the L-2 norm:

$$\hat{w} = \underset{w}{\operatorname{argmin}} \|y - wX\|_2^2 + \lambda_2 \|w\|_2^2 \qquad \|w\|_2 = \sqrt{\Sigma_{i=1}^{k} w_i^2},$$

- Sacrifice a little of bias to reduce the variance of predicted values

→ More stable and generalize better

- Keep all the repressors in the model

→ Not easily interpretable model

Lasso (Least Absolute Shrinkage and Selection Operator)

- Lasso regression introduces a regularization with the L-1 norm:

$$\hat{w} = \underset{w}{\operatorname{argmin}} \|y - wX\|_2^2 + \lambda_1 \|w\|_1 \qquad \|w\|_1 = \Sigma_{i=1}^{k} |w_i|$$

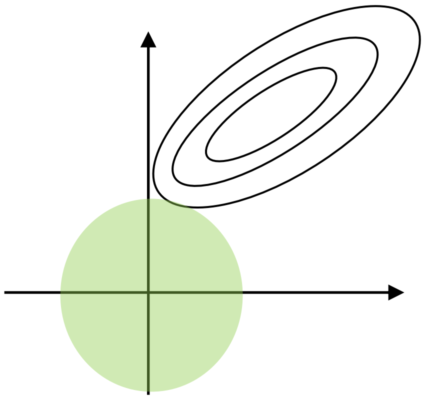
- Only a small subset of features with $\hat{w}_i \neq 0$ are selected

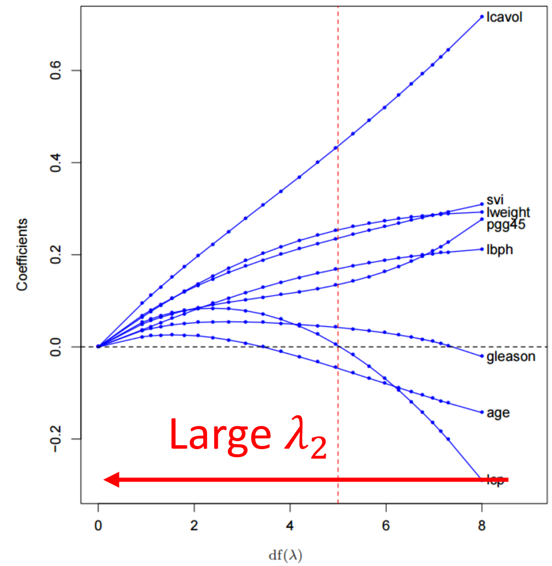
→ Increases the interpretability

- More difficult to implement than Ridge Regression

Ridge regression

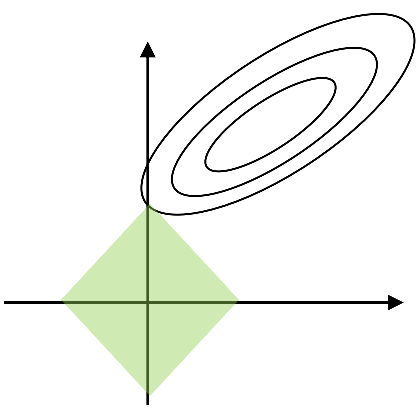
$$\hat{w} = \underset{w}{\operatorname{argmin}} \|y - wX\|_2^2 + \lambda_2 \|w\|_2^2$$

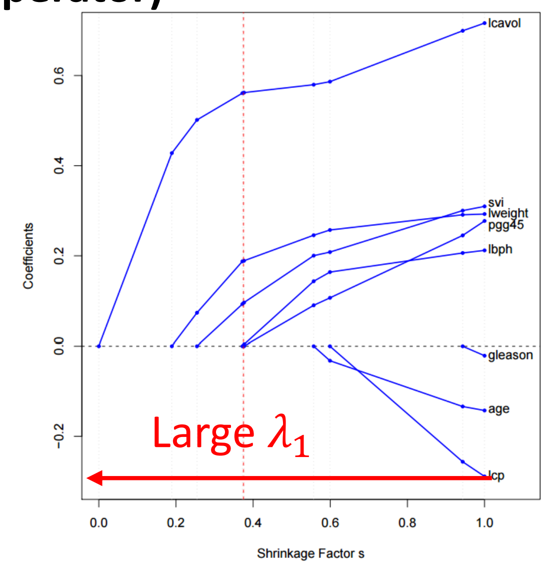




Lasso (Least Absolute Shrinkage and Selection Operator)

$$\hat{w} = \underset{w}{\operatorname{argmin}} \|y - wX\|_2^2 + \lambda_1 \|w\|_1$$

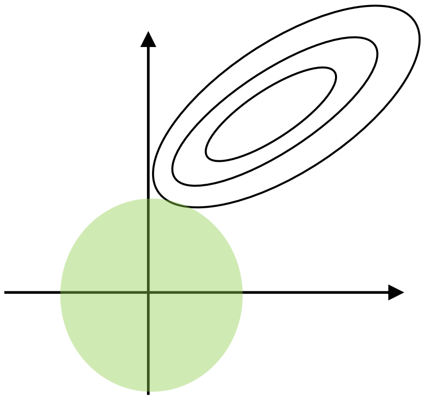




Bayesian view on Ridge regression

Ridge regression

$$\hat{w} = \underset{w}{\operatorname{argmin}} \|y - wX\|_2^2 + \lambda_2 \|w\|_2^2$$



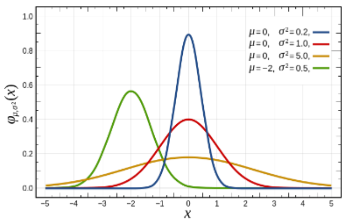
MAP estimation view

MAP estimation view

$$\hat{w} = \underset{w}{\operatorname{argmax}} \log p(w|X,y) = p(y|X,w)p(w)$$

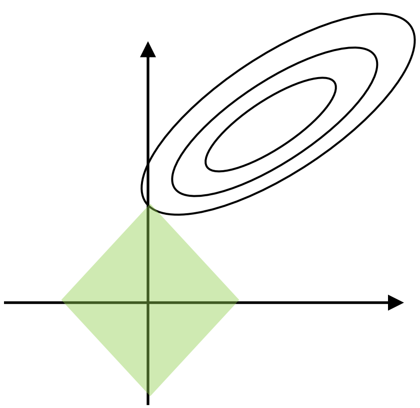
Gaussian prior

$$p(w) = \frac{1}{(2\pi\alpha^2)^{k/2}} \exp\left(-\frac{1}{2\alpha^2} w^T w\right)$$



Lasso (Least Absolute Shrinkage and Selection Operator)

$$\hat{w} = \underset{w}{\operatorname{argmin}} \|y - wX\|_2^2 + \lambda_1 \|w\|_1$$

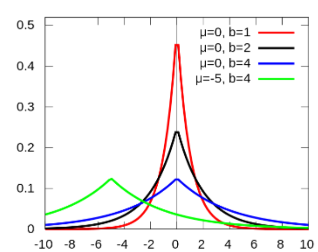


MAP estimation view

$$\hat{w} = \underset{w}{\operatorname{argmax}} \log p(w|X,y) = p(y|X,w)p(w)$$

Laplace prior

$$p(w) = \prod_{i=1}^{k} \frac{\lambda}{2\sqrt{\tau^2}} \exp\left(-\frac{\lambda|w_i|}{\sqrt{\tau^2}}\right)$$



Bayesian view on Ridge regression

- $p(y|X,w) = \prod_{i=1}^{m} \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right)$

m : number of data points
n : dimension of w

- $p(w) = N(w|\mathbf{0}, \tau^2\mathbf{I}) = \prod_{i=1}^{n} \frac{1}{(2\pi\tau^2)^{1/2}} \exp\left(-\frac{(w_i - 0)^2}{2\tau^2}\right) = \frac{1}{(2\pi\tau^2)^{n/2}} \exp\left(-\frac{1}{2\tau^2} w^T w\right)$

- $p(w|X,y) \propto p(y|X,w)p(w)$

$$= \prod_{i=1}^{m} \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right) \frac{1}{(2\pi\tau^2)^{k/2}} \exp\left(-\frac{1}{2\tau^2} w^T w\right)$$

$$= \left(\frac{1}{(2\pi\sigma^2)^{1/2}}\right)^m \frac{1}{(2\pi\tau^2)^{n/2}} \exp\left(-\sum_{i=1}^{m} \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \frac{1}{2\tau^2} w^T w\right)$$

- $\log p(w|X,y) = m \log \frac{1}{(2\pi\sigma^2)^{1/2}} + \log \frac{1}{(2\pi\tau^2)^{n/2}} - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{m} (y_i - w^T x_i)^2 + \frac{\sigma^2}{\tau^2} w^T w\right)$

- Maximum A Posteriori (MAP) estimation with Gaussian prior = **Ridge Regression**

$$(w^*) = \underset{w}{\operatorname{argmax}} \log p(w|X,y) = \underset{w}{\operatorname{argmin}} \|y - wX\|_2^2 + \lambda_2 \|w\|_2^2$$

Bayesian view on Lasso regression

- $p(y|X,w) = \prod_{i=1}^{m} \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right)$

m : number of data points
n : dimension of w

- $p(w) = \mathrm{Lap}(w|\lambda,\tau) = \prod_{i=1}^{n} \frac{\lambda}{2\sqrt{\tau^2}} \exp\left(-\frac{\lambda|w_i|}{\sqrt{\tau^2}}\right)$

- $p(w|X,y) \propto p(y|X,w)p(\mathrm{w})$

$$= \prod_{i=1}^{m} \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right) \left(\frac{\lambda}{2\sqrt{\tau^2}}\right)^n \exp\left(-\frac{\lambda}{\sqrt{\tau^2}} \Sigma_{i=1}^{n}|w_i|\right)$$

$$= \left(\frac{1}{(2\pi\sigma^2)^{1/2}}\right)^m \left(\frac{\lambda}{2\sqrt{\tau^2}}\right)^n \exp\left(-\sum_{i=1}^{m} \frac{(y_i - w^T x_i)^2}{2\sigma^2} - \frac{\lambda}{\sqrt{\tau^2}} \Sigma_{i=1}^{n}|w_i|\right)$$

- $\log p(w|X,y) = m \log \frac{1}{(2\pi\sigma^2)^{1/2}} + n\log \frac{\lambda}{2\sqrt{\tau^2}} - \frac{1}{2\sigma^2}\left(\sum_{i=1}^{m}(y_i - w^T x_i)^2 + \frac{2\sigma^2\lambda}{\sqrt{\tau^2}} \Sigma_{i=1}^{n}|w_i|\right)$

- Maximum A Posteriori estimation with Laplacian prior = **Lasso regression**

$$(w^*) = \underset{w}{\operatorname{argmax}} \log p(w|X,y) = \underset{w}{\operatorname{argmin}} \|y - wX\|_2^2 + \lambda_1 \|w\|_1$$

Bayesian view on Lasso regression

Jupyter Demo Simulation

Ridge and Lasso Simulation

Bayesian Model Selection

Bayesian Approach for Model Selection

- **Model fitting** proceeds by assuming a particular model is true, and tuning the model so it provides the best possible fit to the data

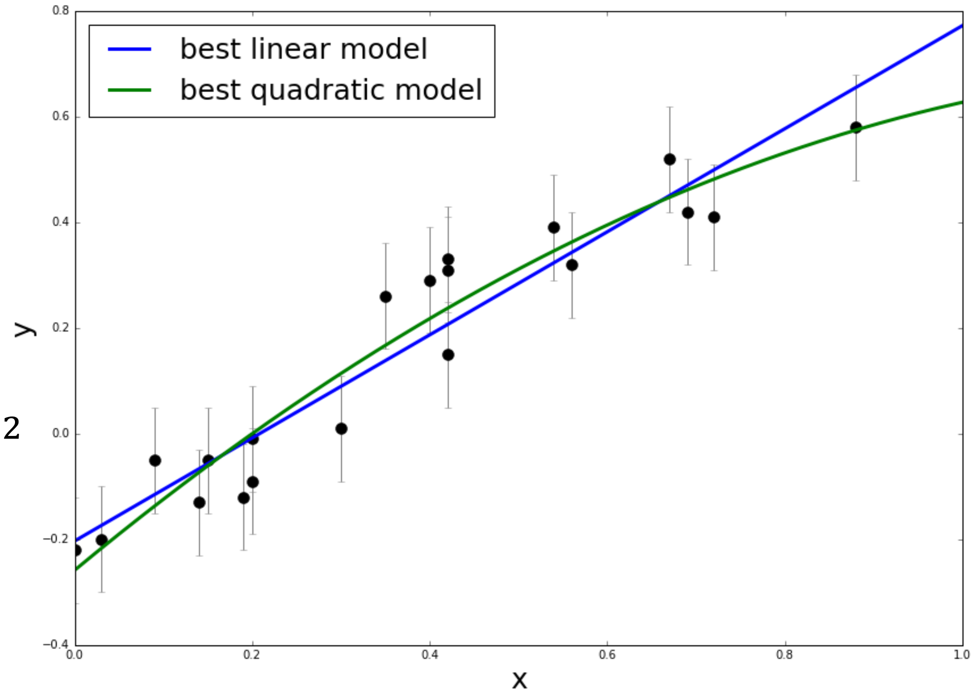
- **Model selection**, on the other hand, asks the larger question of whether the assumptions of the model are compatible with the data.

Linear model:

$$y_{M1} = f_{M1}(x; w) = w_0 + w_1 x$$
$$y \sim N(y_{M1}, \sigma_y^2)$$

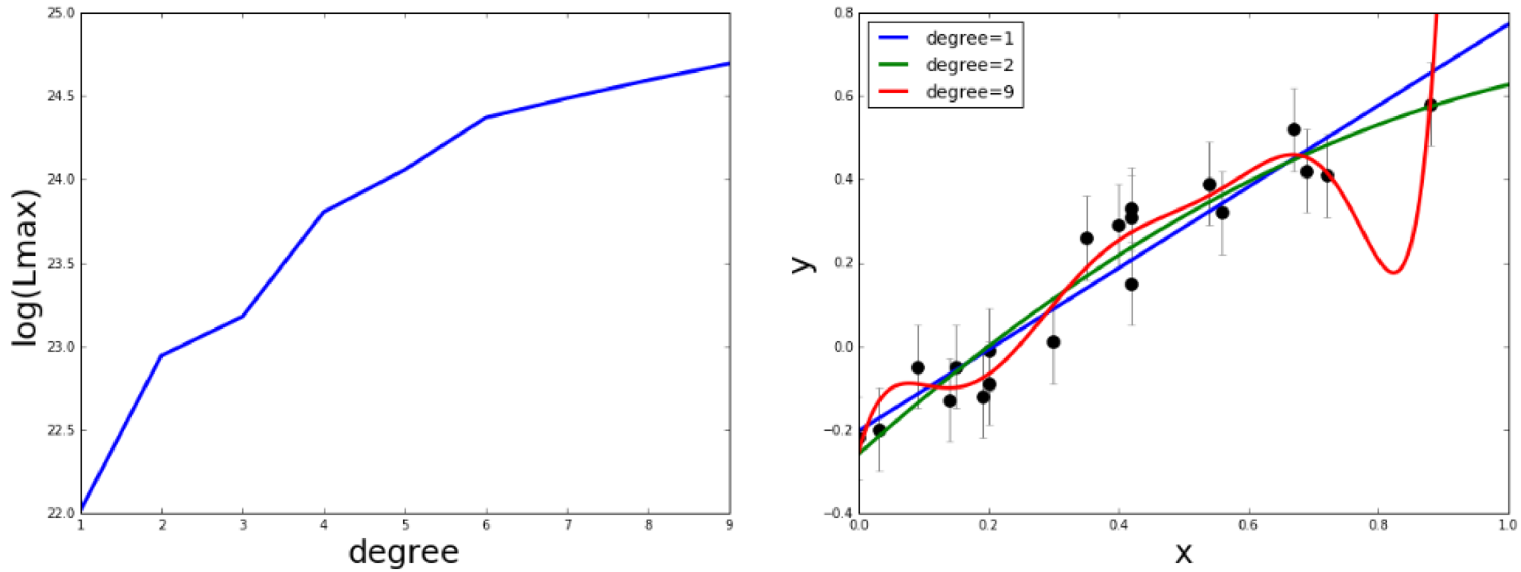
Quadratic model:

$$y_{M2} = f_{M2}(x; w) = w_0 + w_1 x + w_2 x^2$$
$$y \sim N(y_{M2}, \sigma_y^2)$$



Which model is better ?

Model Complexity and Generality



- Comparing maximum likelihood $p(D|w,M_1)$ and $p(D|w,M_2)$ is not a good idea

$$\lim_{w} p(D|w,M_1) \quad v.s. \quad \lim_{w} p(D|w,M_2)$$

- As more complex model is used, model better fits the data, however, this model cannot predict well on unseen test data
- Balancing between model fitting and generalization is a fundamental question in ML
- In frequentist approach, a complex model is penalized by additional regularization term

The Bayesian approach addresses this by integrating over the model parameter space, which in effect acts to automatically penalize overly-complex models.

Bayesian Approach for Model Selection

- **The parameter posterior** given the model M is expressed

$$p(w|D,M) = \frac{p(D|w,M)p(w|M)}{p(D|M)}$$

- **The model posterior** can be expressed

$$p(M|D) = \frac{p(D|M)p(M)}{p(D)}$$

where $$p(D|M) = \int_w p(D,w|M) = \int_\Omega p(D|w,M)p(w|M)d\theta$$

(Integration over the entire parameter space $w \in \Omega$)

- **The odd ratio between two models,** M_1 and M_2, can be expressed

$$O_{21} = \frac{p(M_2|D)}{p(M_1|D)} = \frac{p(D|M_2)}{p(D|M_1)}\frac{p(M_2)}{p(M_1)}$$

$0_{21} >$ thredhold
Choose M_2

$\frac{p(D|M_2)}{p(D|M_1)}$: Bayes factors $\frac{p(M_2)}{p(M_1)}$: Prior odd ratio