ECE595 / STAT598: Machine Learning I
Lecture 13 Connecting Bayesian with Linear Regression

Spring 2020
Stanley Chan

School of Electrical and Computer Engineering
Purdue University
In linear discriminant analysis (LDA), there are generally two types of approaches:

- **Generative approach**: Estimate model, then define the classifier
- **Discriminative approach**: Directly define the classifier
Outline

Generative Approaches
- Lecture 9 Bayesian Decision Rules
- Lecture 10 Evaluating Performance
- Lecture 11 Parameter Estimation
- Lecture 12 Bayesian Prior
- Lecture 13 Connecting Bayesian and Linear Regression

Today’s Lecture
- Linear Regression Review
  - Linear regression in the context of classification
  - Linking linear regression with MLE and MAP
- Connection between Linear Regression and Bayesian
  - Expected Loss
  - Main Result
  - Implications
Linear Regression Reviewed

- Linear regression is actually a **discriminative method**.
- Do not require a distributional model.
- Construct the hypothesis function directly:

\[
h(x) = \begin{cases} +1, & \text{if } g(x) > 0, \\ -1, & \text{if } g(x) < 0. \end{cases}
\]

- Consider a binary classification problem with discriminant function:

\[g(x) = w^T x + w_0\]

- The goal is to determine the parameters \( \theta = \{w, w_0\} \)

- Training data: \((x_n, y_n)_{n=1}^N\)
  - \(x_n \in \mathbb{R}^d\) is the input vector
  - \(y_n \in \{-1, +1\}\) is the corresponding label
Geometry of Linear Regression

- The discriminant function $g(x)$ is linear
- The hypothesis function $h(x) = \text{sign}(g(x))$ is a unit step
Loss Function

- All discriminant algorithms have a **Training Loss Function**

\[
J(\theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(g(x_n), y_n).
\]

- In linear regression,

\[
J(\theta) = \frac{1}{N} \sum_{n=1}^{N} (g(x_n) - y_n)^2 \\
= \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^T \mathbf{x}_n + w_0 - y_n)^2 \\
= \frac{1}{N} \left\| \begin{bmatrix} x_1^T & 1 \\ \vdots & \vdots \\ x_N^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ w_0 \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \right\|^2 = \frac{1}{N} \left\| \mathbf{A} \theta - \mathbf{y} \right\|^2.
\]
Solution of Linear Regression

Theorem (Linear Regression Solution)

The loss function of a linear regression model is given by

$$J(\theta) = \|A\theta - y\|^2,$$

of which the minimizer is

$$\theta^* = (A^TA)^{-1}A^Ty.$$

- Take derivative and setting to zero:

$$\nabla_\theta J(\theta) = \nabla_\theta \{\|A\theta - y\|^2\} = 2A^T(A\theta - y) = 0.$$

- So solution is $\theta^* = (A^TA)^{-1}A^Ty$, assuming $A^TA$ is invertible.
When $A^T A$ is large

- Computing $(A^T A)^{-1}$ directly is infeasible for large-scale datasets with a large number of variables.
- Consider using iterative algorithms such as gradient descent.
- The gradient descent is given by the iteration:

$$
\theta^{(k+1)} = \theta^{(k)} - \eta \nabla \theta J(\theta^{(k)}) \\
= \theta^{(k)} - \eta (2A^T A \theta^{(k)} - 2A^T y)
$$

- A pictorial illustration of the gradient descent step:
Treating Linear Regression as Maximum-Likelihood

- Minimizing $J(\theta)$ is the same as solving a maximum-likelihood:

$$\theta^* = \arg\min_{\theta} \| A\theta - y \|^2$$

$$= \arg\min_{\theta} \sum_{n=1}^{N} (a_n^T \theta - y_n)^2$$

$$= \arg\max_{\theta} \exp \left\{ - \sum_{n=1}^{N} (a_n^T \theta - y_n)^2 \right\}$$

$$= \arg\max_{\theta} \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi}\sigma^2} \exp \left\{ - \frac{(a_n^T \theta - y_n)^2}{2\sigma^2} \right\} \right\}$$

- Assume noise is i.i.d. Gaussian with variance $\sigma^2$. 
Treating Linear Regression as Maximum-a-Posteriori

- We can modify the MLE by adding a prior

\[ p_\Theta(\theta) = \exp \left\{ - \frac{\rho(\theta)}{\beta} \right\}. \]

- Then, we have a MAP problem:

\[
\theta^* = \arg\max_{\theta} \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi}\sigma^2} \exp \left\{ - \frac{(a_n^T\theta - y_n)^2}{2\sigma^2} \right\} \right\} \exp \left\{ - \frac{\rho(\theta)}{\beta} \right\}
\]

\[
= \arg\min_{\theta} \frac{1}{2\sigma^2} \sum_{n=1}^{N} (a_n^T\theta - y_n)^2 + \frac{1}{\beta} \rho(\theta)
\]

\[
= \arg\min_{\theta} \|A\theta - y\|^2 + \lambda \rho(\theta), \quad \text{where} \quad \lambda = \frac{2\sigma^2}{\beta}.
\]

- \( \rho(\cdot) \) is called **regularization function**.
- Useful when \( A^T A \) is not invertible.
Ridge Regression

- One option: Choose a Gaussian prior

\[
\exp \left\{ - \frac{\rho(\theta)}{\beta} \right\} = \exp \left\{ - \frac{\|\theta\|^2}{2\sigma_0^2} \right\}
\]

- Then, the MAP becomes

\[
\theta^* = \arg\max_{\theta} \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi}\sigma^2} \exp \left\{ - \frac{(a_n^T\theta - y_n)^2}{2\sigma^2} \right\} \right\} \exp \left\{ - \frac{\|\theta\|^2}{2\sigma_0^2} \right\}
\]

\[
= \arg\min_{\theta} \sum_{n=1}^{N} (a_n^T\theta - y_n)^2 + \frac{\sigma^2}{\sigma_0^2} \|\theta\|^2
\]

\[
= \arg\min_{\theta} \|A\theta - y\|^2 + \lambda\|\theta\|^2
\]

- This is called **Tikhonov regularization** or **Ridge regression**.
Outline

Generative Approaches
- Lecture 9 Bayesian Decision Rules
- Lecture 10 Evaluating Performance
- Lecture 11 Parameter Estimation
- Lecture 12 Bayesian Prior
- Lecture 13 Connecting Bayesian and Linear Regression

Today’s Lecture
- Linear Regression Review
  - Linear regression in the context of classification
  - Linking linear regression with MLE and MAP
- Connection between Linear Regression and Bayesian
  - Expected Loss
  - Main Result
  - Implications
Connection with Bayesian Decision Rule

- With infinite training samples, $J(\theta)$ converges almost surely to its expectation

$$\frac{1}{N} \sum_{n=1}^{N} (g(x_n) - y_n)^2 \xrightarrow{p} \mathbb{E}_{x,y} [g(x) - y]^2].$$

- Minimizing $J(\theta)$ is essentially minimizing the expectation

$$\theta^* = \arg\min_{w,w_0} \quad \frac{1}{N} \sum_{n=1}^{N} (g(x_n) - y_n)^2 \approx \arg\min_{w,w_0} \mathbb{E}_{x,y} \left[ (w^T x + w_0 - y)^2 \right].$$
Summary of the Result

Theorem (Conditions for Linear Regression = Bayes)

Suppose that all the following three conditions are satisfied:

(i) The likelihood $p(x|i)$ is Gaussian satisfying

$$p(x|i) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp \left\{ -\frac{1}{2} (x - \mu_i)^T \Sigma^{-1} (x - \mu_i) \right\}, \ i \in \{-1, +1\}$$

(ii) The prior is uniform: $p_y(+1) = p_y(-1) = \frac{1}{2}$.

(iii) The number of training samples goes to infinity.

Then, the linear regression model parameter $(\mathbf{w}, w_0)$ is given by

$$\mathbf{w} = \tilde{\Sigma}^{-1} (\mu_1 - \mu_{-1}), \quad w_0 = -\frac{1}{2} (\mu_1 + \mu_{-1}) \tilde{\Sigma}^{-1} (\mu_1 - \mu_{-1}),$$

where $\tilde{\Sigma} \overset{\text{def}}{=} \Sigma/2$, and $\Sigma$ is the covariance of the Gaussian.
Sketch of Proof

Let us make some assumptions:

- **Likelihood**: Gaussian with equal covariance:

\[
p(x_n|y = +1) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left\{ -\frac{1}{2} (x_n - \mu_+)^T \Sigma^{-1} (x_n - \mu_+) \right\}
\]

\[
p(x_n|y = -1) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left\{ -\frac{1}{2} (x_n - \mu_-)^T \Sigma^{-1} (x_n - \mu_-) \right\}
\]

- **Prior**: Equal prior:

\[
p_y(+1) = \frac{1}{2}
\]

\[
p_y(-1) = \frac{1}{2}.
\]
Sketch of Proof

- Taking derivative w.r.t. \((w, w_0)\) yields

\[
\frac{d}{dw} E_{x,y} \left[(w^T x + w_0 - y)^2\right] = 2 \left( E[xx^T] w + E[x] w_0 - E[xy] \right)
\]

\[
\frac{d}{dw_0} E_{x,y} \left[(w^T x + w_0 - y)^2\right] = 2 \left( E[x]^T w + w_0 - E[y] \right)
\]

- What is \(E[x]\)?

\[
E[x] = E[x|y = 1] p_y(1) + E[x|y = -1] p_y(-1)
\]

\[
= \mu_1 \left( \frac{1}{2} \right) + \mu_{-1} \left( \frac{1}{2} \right) = \frac{1}{2} (\mu_1 + \mu_{-1}).
\]

- What is \(E[xy]\)?

\[
E[xy] = E[xy|y = 1] p_y(1) + E[xy|y = -1] p_y(-1)
\]

\[
= (+\mu_1) \left( \frac{1}{2} \right) + (-\mu_{-1}) \left( \frac{1}{2} \right) = \frac{1}{2} (\mu_1 - \mu_{-1}).
\]
Sketch of Proof

- What is $\mathbb{E} \left[ (x - \mathbb{E}[x])(x - \mathbb{E}[x])^T \right]$?

\[
\mathbb{E} \left[ (x - \mathbb{E}[x])(x - \mathbb{E}[x])^T \right] = \mathbb{E} \left[ (x - \mathbb{E}[x])(x - \mathbb{E}[x])^T | y = +1 \right] p_y(+1) \\
+ \mathbb{E} \left[ (x - \mathbb{E}[x])(x - \mathbb{E}[x])^T | y = -1 \right] p_y(-1) \\
= \frac{1}{2} \Sigma + \frac{1}{2} \Sigma = \Sigma.
\]

- This will allow us to compute $\mathbb{E}[xx^T]$:

\[
\mathbb{E} \left[ (x - \mathbb{E}[x])(x - \mathbb{E}[x])^T \right] = \mathbb{E}[xx^T] - \mathbb{E}[x]\mathbb{E}[x]^T.
\]

- The remaining is just linear algebra. See Appendix.
Implication

- Linear regression assumes equal covariance for both classes.

- Bayesian allows different variance $\Sigma_i$.
- They are equal only when the number of training samples is large.
**Example 1**: When the classes are intrinsically unbalanced.

Bayesian gives nonlinear decision boundary
When will Linear Regression Go Wrong? (1)

- When the classes are intrinsically unbalanced.
- One class has a significantly larger variance than the other.
- Nothing to do with the number of training samples.
- Regression goes wrong because the big variance class dominates the sum square error.
- So you spend more effort to make that class “happy”.

Bayesian decision rule takes care of this by allowing different $\Sigma_i$. 
**Example 2:** When training samples are unbalanced.

- Bayesian performs equally bad.
When will Linear Regression Go Wrong? (2)

- When training samples are unbalanced.
- One class has more training samples than the other class.
- Nothing to do with the intrinsic distribution. You just did not sample the training samples uniformly from the true distribution.
- Regression goes wrong because the more sample class dominate the sum square error.
- So you spend more effort to make the majority “happy”.

Bayesian suffers too because it has a bad estimate of the mean.
Does Regularization Help?

- We can put regularization to linear regression
  \[ J(\theta) = \|A\theta - y\|^2 + \lambda\|\theta\|^2 \]
- Can help some bizarre cases when \( A \) is rank deficient.
- But what regularization to use? How to control \( \lambda \)?
- Prior in Bayesian is a lot more intuitive.

\[
\hat{\mu} = \frac{\sigma^2}{N\sigma^2_0 + \sigma^2}\mu_0 + \frac{N\sigma^2_0}{N\sigma^2 + \sigma^2}\mu_{ML}.
\]
- When \( N \) is small, we have the prior to control the estimate.
- Linear regression does not have this capability, unless you know what the decision weights should look like.
Example 3: “Outliers”

One sample point appears “abnormally”

Bayesian suffers from the same issue

But Bayesian can use the prior term to mitigate outliers

Of course, you can also do data pre-processing in linear regression to remove outliers
Reading List

Linear Regression and Bayesian Decision

- Chris Bishop’s *Pattern Recognition*, Chapter 3.1, 4.1
- Hastie-Tibshirani-Friedman’s *Elements of Statistical Learning*, Chapter 3.2, 3.4
- Stanford CS 229 Discriminant Algorithms
  
Appendix
Proof of Main Result

By following the steps in the proof sketch, we have shown that

\[
\frac{d}{dw} \mathbb{E}_{x,y} \left[ (w^T x + w_0 - y)^2 \right] = 2 \left( \mathbb{E}[x x^T]w + \mathbb{E}[x]w_0 - \mathbb{E}[x y] \right) = 0
\]

\[
\frac{d}{dw_0} \mathbb{E}_{x,y} \left[ (w^T x + w_0 - y)^2 \right] = 2 \left( \mathbb{E}[x]^T w + w_0 - \mathbb{E}[y] \right) = 0
\]

- Look at the second equation

\[
-\mathbb{E}[x] \mathbb{E}[x]^T w - \mathbb{E}[x]w_0 + \mathbb{E}[x] \mathbb{E}[y] = 0
\]
\[
+ \mathbb{E}[x x^T] w + \mathbb{E}[x]w_0 - \mathbb{E}[x y] = 0
\]

- This gives us

\[
(\mathbb{E}[x x^T] - \mathbb{E}[x] \mathbb{E}[x]^T)w + 0 - (\mathbb{E}[x y] - \mathbb{E}[x] \mathbb{E}[y]) = 0.
\]
Proof of Main Result

- Therefore, we have
  \[
  \left(\mathbb{E}[xx^T] - \mathbb{E}[x]\mathbb{E}[x]^T\right)\Sigma^{-1} + 0 - \left(\mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]\right) = 0.
  \]

- This means that
  \[
  \Sigma w = \frac{1}{2}(\mu_+ - \mu_-),
  \]

- which gives us
  \[
  w = \frac{1}{2} \Sigma^{-1}(\mu_+ - \mu_-).
  \]

- Compare to the Bayesian decision rule for equal covariance:
  \[
  w = \Sigma^{-1}(\mu_+ - \mu_-).
  \]

- The only difference is the factor $1/2$. 
Proof of Main Result

- Now let us determine $w_0$.
- Look at the second equation again:
  \[
  \mathbb{E}[x]^T w + w_0 - \mathbb{E}[y] = 0
  \]
- This means
  \[
  w_0 = \mathbb{E}[y] - \mathbb{E}[x]^T w
  = 0 - \left( \frac{1}{2}(\mu_{+1} + \mu_{-1}) \right)^T w
  = 0 - \left( \frac{1}{2}(\mu_{+1} + \mu_{-1}) \right)^T \left( \frac{1}{2} \Sigma^{-1}(\mu_{+1} - \mu_{-1}) \right)
  = -\frac{1}{4}(\mu_{+1} + \mu_{-1})\Sigma^{-1}(\mu_{+1} - \mu_{-1}).
  \]
Proof of Main Result

- If we want to write the decision boundary as $\mathbf{w}^T (\mathbf{x} - \mathbf{x}_0) = 0$,
- then we can show that

$$\mathbf{w}^T (\mathbf{x} - \mathbf{x}_0) = \left( \frac{1}{2} \Sigma^{-1} (\mu_+ - \mu_-) \right) (\mathbf{x} - \mathbf{x}_0).$$

- Since

$$w_0 = -\frac{1}{4} (\mu_+ - \mu_-) \Sigma^{-1} (\mu_+ + \mu_-),$$

- in order to make $w_0 = \mathbf{w}^T \mathbf{x}_0$, we should choose

$$\mathbf{x}_0 = \frac{1}{2} (\mu_+ + \mu_-).$$

- This is the same as the Bayesian decision rule with equal covariance.