6.867 Machine learning and neural networks

Tommi Jaakkola
MIT AI Lab
tommi@ai.mit.edu

Lecture 2: linear/additive regression

Topics

- Linear regression, additive models
  - Loss functions, fitting, generalization
  - Statistical view, bias and variance

Regression

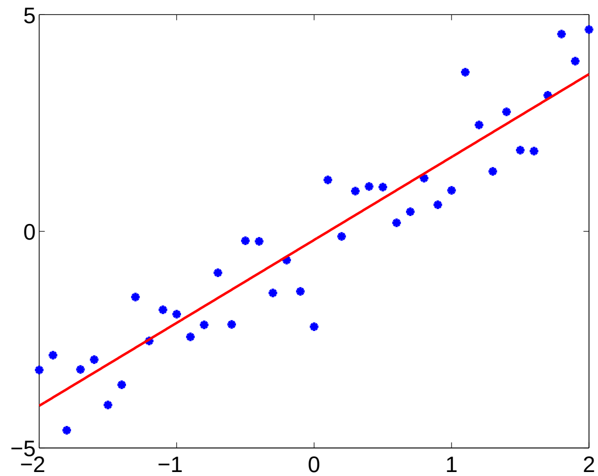
- We need to define a **function class** and **fitting criterion (loss)**
- Example: linear functions of one variable (two parameters)

$$f(x; \mathbf{w}) = w_0 + w_1 x$$

with a squared loss: $\text{Loss}(y, f(x; \mathbf{w})) = (y - f(x; \mathbf{w}))^2/2$.

Estimation based on minimizing the *empirical* loss

$$J_n(\mathbf{w}) = \sum_{i=1}^{n} \text{Loss}(y_i, f(x_i; \mathbf{w}))$$



Linear regression: estimation

- We minimize the *empirical* squared loss

$$J_n(\mathbf{w}) = \sum_{i=1}^{n} \mathsf{Loss}(y_i, f(x_i;\mathbf{w})) = \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2/2$$

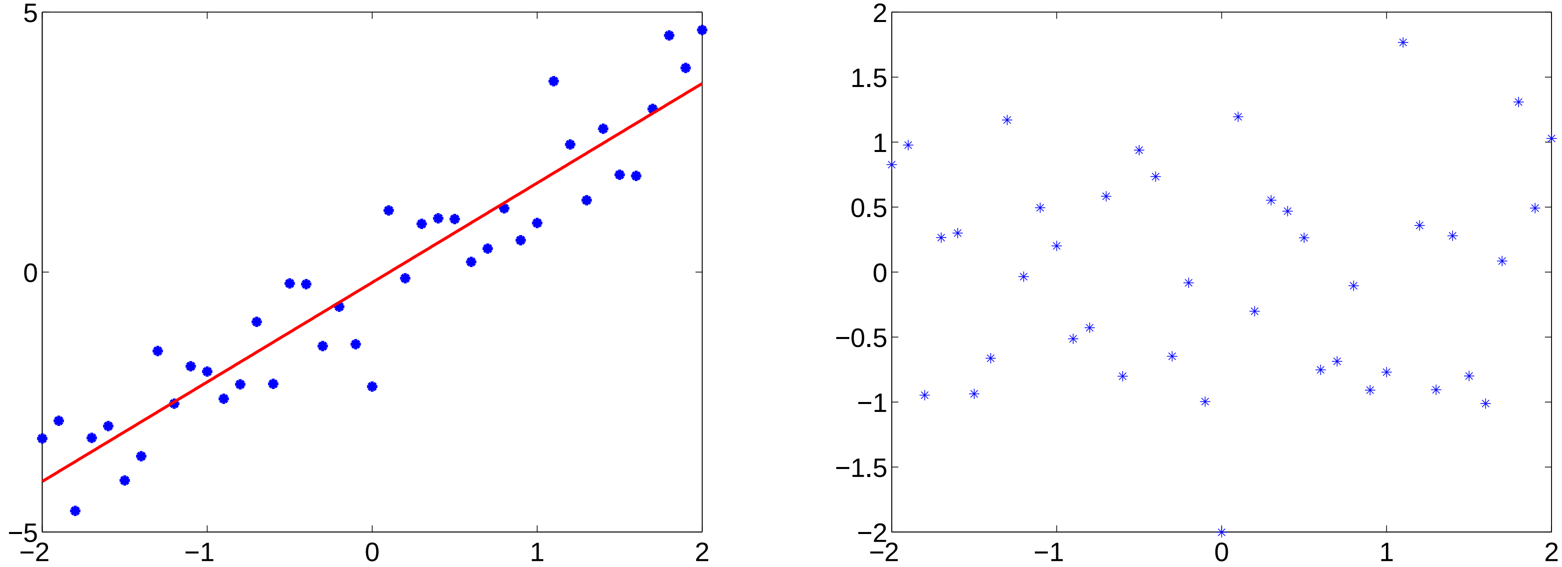
Setting the derivatives with respect to w_0 and w_1 to zero we get necessary conditions for the "optimal" parameter values

$$\begin{aligned} \frac{\partial}{\partial w_0} J_n(\mathbf{w}) &= -\sum_{i=1}^{n} (y_i - w_0 - w_1 x_i) = 0 \\ \frac{\partial}{\partial w_1} J_n(\mathbf{w}) &= -\sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)\, x_i = 0 \end{aligned}$$

Note: These conditions mean that the prediction error $(y_i - w_0 - w_1 x_i)$ has zero mean and is decorrelated with the inputs x_i

Linear regression: estimation

- The prediction error $(y_i - w_0 - w_1 x_i)$ is decorrelated with the inputs x_i



Linear regression: estimation

$$\frac{\partial}{\partial w_0} J_n(\mathbf{w}) = -\sum_{i=1}^{n} (y_i - w_0 - w_1 x_i) = 0$$

$$\frac{\partial}{\partial w_1} J_n(\mathbf{w}) = -\sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)\, x_i = 0$$

- Solution via matrix inversion

$$\begin{aligned} w_0 \left(\textstyle\sum_{i=1}^n 1\right) &+ w_1 \left(\textstyle\sum_{i=1}^n x_i\right) = \textstyle\sum_{i=1}^n y_i \\ w_0 \left(\textstyle\sum_{i=1}^n x_i\right) &+ w_1 \left(\textstyle\sum_{i=1}^n x_i^2\right) = \textstyle\sum_{i=1}^n y_i x_i \end{aligned}$$

or $\Phi \mathbf{w} = b$, where

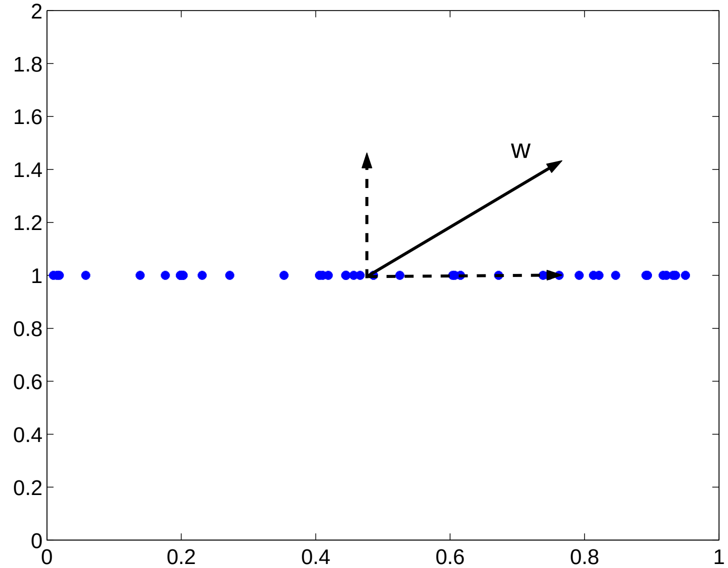
$$\Phi = \begin{bmatrix} \sum_{i=1}^n 1 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix}, \quad b = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n y_i x_i \end{bmatrix}$$

- If Φ is invertible, we get our parameter estimates via $\hat{\mathbf{w}} = \Phi^{-1} b$

Linear regression: pseudo-inverse

- 2-D example:

$$y_i \approx f(\mathbf{x}_i; \mathbf{w}) = w_0 + w_1 x_{i1} + w_2 x_{i2}$$



- We find the solution in the subspace spanned by the examples (weight vector set to zero in the "unused" dimensions)

Linear regression

- In a matrix notation, we minimize:

$$\frac{1}{2}\left\| \begin{bmatrix} y_1 \\ \cdots \\ y_n \end{bmatrix} - \begin{bmatrix} 1 & x_1 \\ \cdots & \cdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \right\|^2$$

or $\frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$

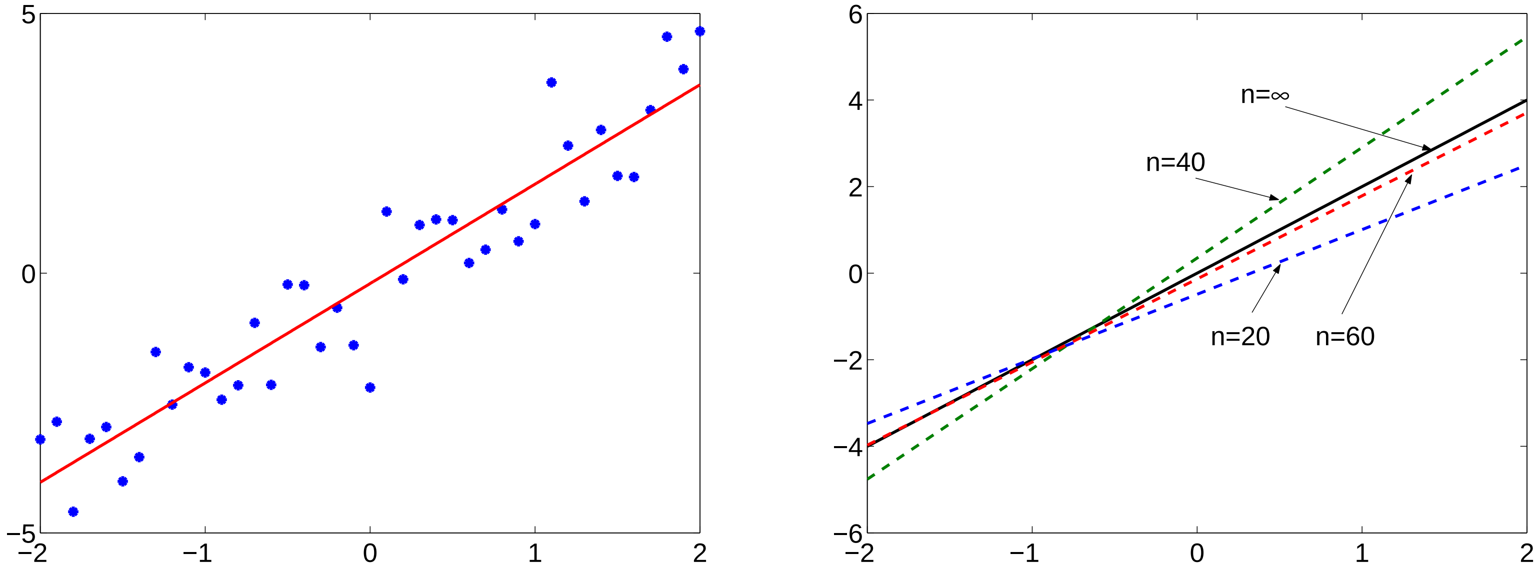
By setting the derivatives to zero, we get

$$\mathbf{X}^T\mathbf{y} - \mathbf{X}^T\mathbf{X}\mathbf{w} = 0 \;\Rightarrow\; \hat{\mathbf{w}} = \underbrace{(\mathbf{X}^T\mathbf{X})}_{\Phi}{}^{-1}\underbrace{\mathbf{X}^T\mathbf{y}}_{b}$$

Note: the solution is a linear function of the outputs y

Linear regression: generalization

- Generalization performance as a function of the number of training examples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$



- This makes no sense unless we assume that there is a systematic relation between x and y: each training example (x, y) is an *independent* sample from a fixed but unknown distribution $P(x, y)$.

Linear regression: generalization

Training examples	$\{(x_1, y_1), \ldots, (x_n, y_n)\}$
Test examples	$\{(x_{n+1}, y_{n+1}), \ldots, (x_{n+N}, y_{n+N})\}$

Let $\hat{\mathbf{w}}_n$ be the least squares parameter estimates on the basis of the training examples.

$$\begin{aligned}
\text{Mean training error} &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mathbf{w}}_0 - \hat{\mathbf{w}}_1 x_i)^2 \\
\text{Mean test error} &= \frac{1}{N}\sum_{i=n+1}^{n+N}(y_i - \hat{\mathbf{w}}_0 - \hat{\mathbf{w}}_1 x_i)^2 \\
\text{``Generalization'' error} &= E_{(x,y)\sim P}\left\{(y - \hat{\mathbf{w}}_0 - \hat{\mathbf{w}}_1 x)^2\right\}
\end{aligned}$$

(note: $\hat{\mathbf{w}}_0$ and $\hat{\mathbf{w}}_1$ are themselves random variables as they are computed on the basis of the randomly sampled training examples)

Linear regression: generalization

- We can decompose the "generalization" error

$$E_{(x,y)\sim P}\left\{(y - \hat{\mathbf{w}}_0 - \hat{\mathbf{w}}_1 x)^2\right\}$$

into two terms:

1. error of the best predictor in the class

$$E_{(x,y)\sim P}\left\{(y - \mathbf{w}_0^* - \mathbf{w}_1^* x)^2\right\}$$

2. how well we approximate the best predictor

$$E_{(x,y)\sim P}\left\{(\mathbf{w}_0^* + \mathbf{w}_1^* x - \hat{\mathbf{w}}_0 - \hat{\mathbf{w}}_1 x)^2\right\}$$

Linear regression and extensions

- Linear in the parameters $\mathbf{w}$, not necessarily in the inputs $\mathbf{x}$

1. Simple linear prediction $f : \mathcal{R} \rightarrow \mathcal{R}$

$$f(x;\mathbf{w}) = w_0 + w_1 x$$

2. m^{th} order polynomial prediction $f : \mathcal{R} \rightarrow \mathcal{R}$

$$f(x;\mathbf{w}) = w_0 + w_1 x + \ldots + w_{m-1} x^{m-1} + w_m x^m$$

3. Multi-dimensional linear prediction $f : \mathcal{R}^d \rightarrow R$

$$f(\mathbf{x};\mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_{d-1} x_{d-1} + w_d x_d$$

where $\mathbf{x} = [x_1 \ldots x_{d-1}\; x_d]^T$, $d = dim(\mathbf{x})$

Additive models

4. Prediction via linear combination of basis functions (features) $\{\phi_1(\mathbf{x}), \ldots, \phi_m(\mathbf{x})\}$, where each $\phi_i(\mathbf{x}) : \mathcal{R}^d \to \mathcal{R}$, and

$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 \phi_1(\mathbf{x}) + \ldots + w_{m-1} \phi_{m-1}(\mathbf{x}) + w_m \phi_m(\mathbf{x})$$

- For example:

  If $\phi_i(x) = x^i$, $i = 1, \ldots, m$, then

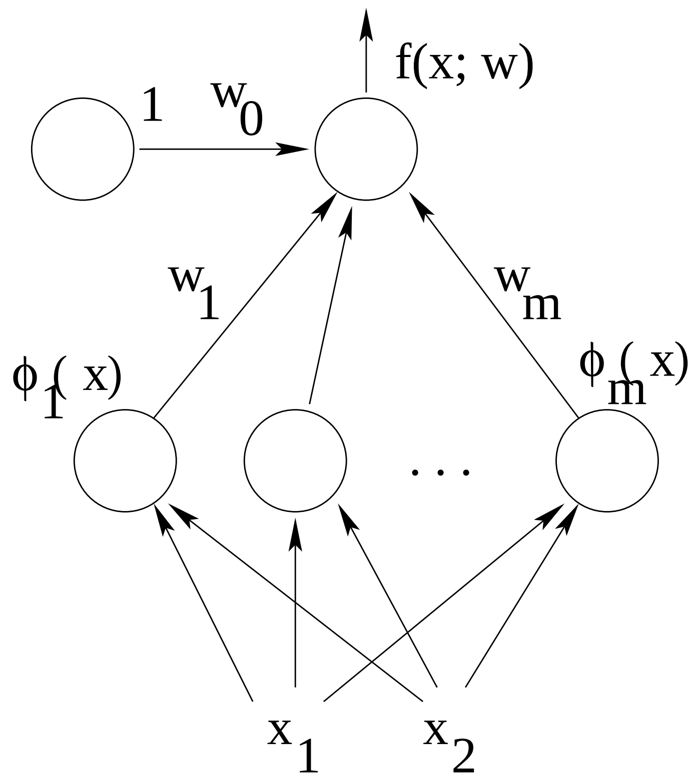
$$f(x; \mathbf{w}) = w_0 + w_1 x + \ldots + w_{m-1} x^{m-1} + w_m x^m$$

  If $m = d$, $\phi_i(\mathbf{x}) = x_i$, $i = 1, \ldots, d$, then

$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_{d-1} x_{d-1} + w_d x_d$$

Additive models

- Graphical representation of additive models



- What if the basis functions themselves can be adjusted?

Additive models

- Example: we have m prototypes of examples $\mu_1, \ldots, \mu_m$

  The basis functions can be used to (softly) select the closest prototype to each example $\mathbf{x}$

$$\phi_k(\mathbf{x}) \;=\; \exp\{\, -\frac{1}{2}\|\mathbf{x} - \mu_k\|^2 \,\}$$

- Example: the "basis functions" may also be constructed by computing various relevant features from the examples

$$\phi_k(\mathbf{x}) \;=\; \begin{cases} 1, \text{ if interest rate is up} \\ 0, \text{ otherwise} \end{cases}$$

Statistical view of linear regression

- A statistical regression model

$$\begin{aligned}\textbf{Observed output} &= \textbf{function} + \textbf{noise} \\ y &= f(\mathbf{x}; \mathbf{w}) + \epsilon\end{aligned}$$

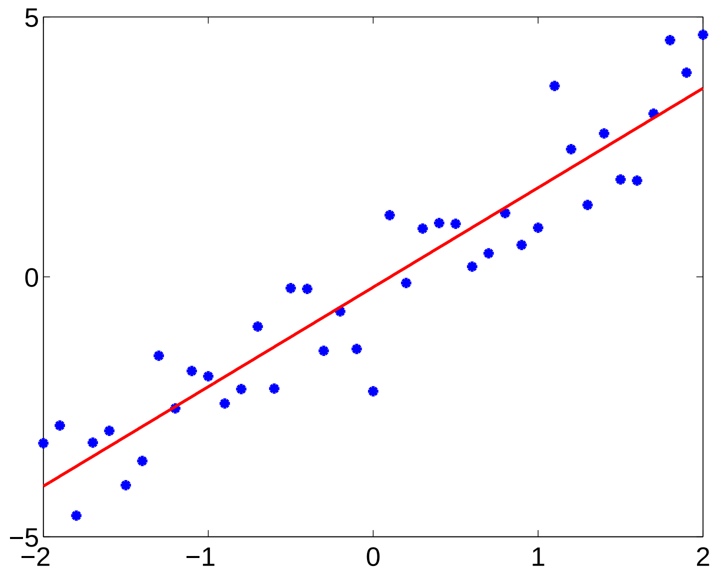
  where, e.g., $\epsilon \sim N(0, \sigma^2)$.

- Whatever we cannot capture with our chosen family of functions will be *interpreted* as noise

Statistical view of linear regression

- Our function $f(\mathbf{x}; \mathbf{w})$ here is trying to capture the mean of the observations y given a specific input $\mathbf{x}$:

$$E\{\, y \,|\, \mathbf{x} \,\} = f(\mathbf{x}; \mathbf{w})$$



Statistical view of linear regression

- According to our statistical model

$$y = f(\mathbf{x}; \mathbf{w}) + \epsilon, \ \ \epsilon \sim N(0, \sigma^2)$$

  the outputs y given $\mathbf{x}$ are normally distributed with mean $f(\mathbf{x}; \mathbf{w})$ and variance σ^2:

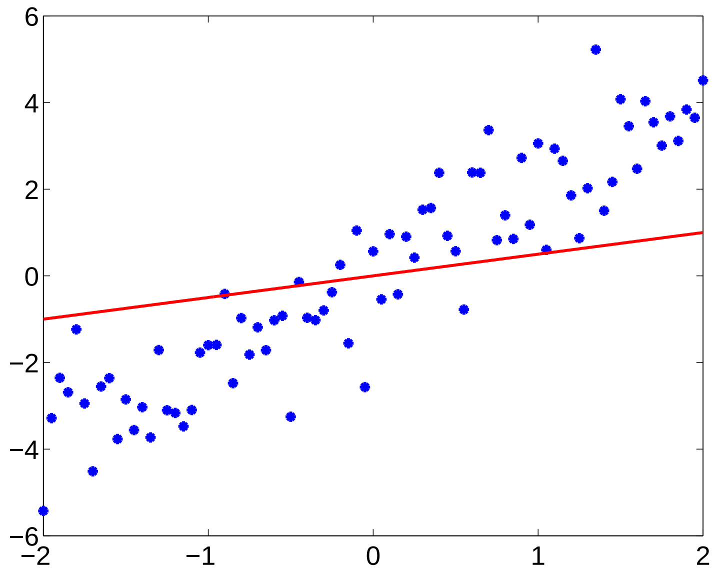
$$P(y|\mathbf{x}, \mathbf{w}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{ -\frac{1}{2\sigma^2}(y - f(\mathbf{x}; \mathbf{w}))^2 \}$$

- As a result we can also measure the uncertainty in the predictions, not just the mean
- Loss function? Estimation?

Maximum likelihood estimation

- Given observations $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ we find the parameters $\mathbf{w}$ that maximize the likelihood of the observed outputs

$$L(D; \mathbf{w}, \sigma^2) = \prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \mathbf{w}, \sigma^2)$$



Why is this a bad fit according to the likelihood criterion?

Maximum likelihood estimation

Likelihood of the observed outputs:

$$L(D; \mathbf{w}, \sigma^2) = \prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \mathbf{w}, \sigma^2)$$

- It is often easier (but equivalent) to try to maximize the log-likelihood:

$$\begin{aligned}
l(D; \mathbf{w}, \sigma^2) &= \log L(D; \mathbf{w}, \sigma^2) = \sum_{i=1}^{n} \log P(y_i | \mathbf{x}_i, \mathbf{w}, \sigma^2) \\
&= \sum_{i=1}^{n} \left(-\frac{1}{2\sigma^2} (y_i - f(\mathbf{x}_i; \mathbf{w}))^2 - \log \sqrt{2\pi\sigma^2} \right) \\
&= \left(-\frac{1}{2\sigma^2} \right) \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i; \mathbf{w}))^2 + \ldots
\end{aligned}$$

- This should look familiar...

Maximum likelihood estimation cont'd

- The noise distribution and the loss-function are intricately related

$$\text{Loss}(y, f(\mathbf{x}; \mathbf{w})) = -\log P(y|\mathbf{x}, \mathbf{w}, \sigma^2) + \text{ const.}$$

Maximum likelihood estimation cont'd

- General fitting criterion: likelihood of the observed outputs

$$L(D; \mathbf{w}, \sigma^2) = \prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \mathbf{w}, \sigma^2)$$

- We can just as easily fit the noise variance σ^2 by maximizing the log-likelihood $l(D; \mathbf{w}, \sigma^2)$ with respect to σ^2

  What might the answer be?

Maximum likelihood estimation cont'd

- General fitting criterion: likelihood of the observed outputs

$$L(D; \mathbf{w}, \sigma^2) = \prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \mathbf{w}, \sigma^2)$$

- We can just as easily fit the noise variance σ^2 by maximizing the log-likelihood $l(D; \mathbf{w}, \sigma^2)$ with respect to σ^2

  Let $\hat{\mathbf{w}}$ be the maximum likelihood parameters for the linear model $f(\mathbf{x}; \mathbf{w})$, we can compute σ^2 as

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i; \hat{\mathbf{w}}))^2$$

  i.e., it is the mean squared prediction error of the best linear predictor.

Bias and variance

- Assume that the outputs were actually generated from a linear model with parameters $\mathbf{w}^*$, i.e.,

$$y \;=\; \overbrace{w_0^* + w_1^* x}^{y^*} + \epsilon$$

where $\epsilon \sim N(0, \sigma^2)$.

- Based on n training examples, we find a weight vector

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y}^* + \epsilon) = \mathbf{w}^* + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\epsilon$$

- We can (in principle) characterize how the estimate depends on the noise by computing its bias and variance

Bias: $\mathbf{w}^* - E\{\,\hat{\mathbf{w}}\,\} = 0$

where the expectation is over the noise terms ϵ. The linear model is *unbiased*

Variance: $E\left\{\,(\hat{\mathbf{w}} - E\{\,\hat{\mathbf{w}}\,\})\,(\hat{\mathbf{w}} - E\{\,\hat{\mathbf{w}}\,\})^T\,\right\} = \ldots = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$

The covariance depends on both the location of the input examples and the noise variance σ^2.