

# CS195-5 : Introduction to Machine Learning

## Lecture 3

Greg Shakhnarovich

September 11, 2006

Revised October 24th, 2006

---

# Announcements

- Next lecture (Wed 9/13): Lubrano
- Matlab / CS accounts
- Books
- CS241

---

# Review

- Learning by estimating parameters  $\mathbf{w}^*$  that minimize empirical loss  $L_N(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N L(f(\mathbf{x}_i; \mathbf{w}), y_i)$
- Expected loss (risk)  $R(\mathbf{w}) = E_{(\mathbf{x}_0, y_0) \sim p(\mathbf{x}, y)} [L(f(\mathbf{x}_0; \mathbf{w}), y_0)]$
- Least squares:  $f(\mathbf{x}; \mathbf{w}) = w_0 + \sum_{j=1}^d w_j x_j$ ,

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2 = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- Prediction errors  $y_i - f(\mathbf{x}_i; \mathbf{w})$  have zero mean and are uncorrelated with any linear function of the inputs  $\mathbf{x}$ .
- As  $N$  increases,  $L_N$  goes up but  $R$  goes down.

---

# Today

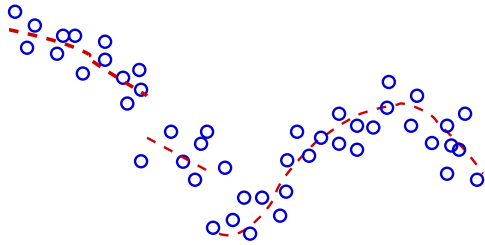
- The optimal regression function
- Error decomposition for parametric regression
- Statistical view of regression

---

# Best unrestricted predictor

- What is the *best possible* predictor of  $y$ , in terms of expected squared loss, if we do not restrict  $\mathcal{F}$  at all?

$$f^* = \operatorname{argmin}_{f: \mathcal{X} \rightarrow \mathbb{R}} E_{(\mathbf{x}_0, y_0) \sim p(\mathbf{x}, y)} \left[ (f(\mathbf{x}_0) - y_0)^2 \right]$$

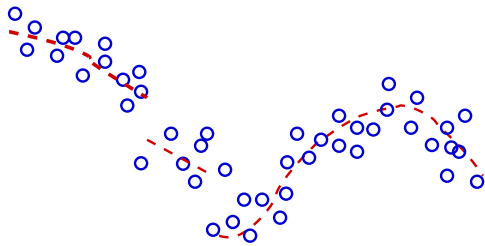


*Any*  $f : \mathcal{X} \rightarrow \mathbb{R}$  is allowed.

# Best unrestricted predictor

- What is the *best possible* predictor of  $y$ , in terms of expected squared loss, if we do not restrict  $\mathcal{F}$  at all?

$$f^* = \operatorname{argmin}_{f: \mathcal{X} \rightarrow \mathbb{R}} E_{(\mathbf{x}_0, y_0) \sim p(\mathbf{x}, y)} \left[ (f(\mathbf{x}_0) - y_0)^2 \right]$$



Any  $f : \mathcal{X} \rightarrow \mathbb{R}$  is allowed.

The *product rule* of probability:  $p(\mathbf{x}, y) = p(y|\mathbf{x})p(\mathbf{x})$

By definition:  $E_{p(y, \mathbf{x})} [g(y, \mathbf{x})] = \int_{\mathbf{x}} \int_y g(y, \mathbf{x}) p(y|\mathbf{x}) p(\mathbf{x}) dy d\mathbf{x}$

$$E_{(\mathbf{x}_0, y_0) \sim p(\mathbf{x}, y)} \left[ (f(\mathbf{x}_0) - y_0)^2 \right] = E_{\mathbf{x}_0 \sim p(\mathbf{x})} \left[ E_{y_0 \sim p(y|\mathbf{x})} \left[ (f(\mathbf{x}_0) - y_0)^2 \mid \mathbf{x}_0 \right] \right]$$

---

## Best unrestricted predictor

$$E_{(\mathbf{x}_0, y_0) \sim p(\mathbf{x}, y)} \left[ (f(\mathbf{x}_0) - y_0)^2 \right] = \int_{\mathbf{x}_0} \left\{ E_{y_0 \sim p(y|\mathbf{x})} \left[ (f(\mathbf{x}_0) - y_0)^2 \mid \mathbf{x}_0 \right] \right\} p(\mathbf{x}_0) d\mathbf{x}_0$$

- If we minimize the inner conditional expectation for each  $\mathbf{x}_0$  separately, we will necessarily minimize the whole integral (outer expectation).
  - Must predict optimally  $y$  for *any*  $\mathbf{x}$ .

$$\begin{aligned} \frac{\delta}{\delta f(\mathbf{x})} E_{p(y|\mathbf{x})} \left[ (f(\mathbf{x}_0) - y_0)^2 \mid \mathbf{x}_0 \right] &= 2 E_{p(y|\mathbf{x})} [f(\mathbf{x}_0) - y_0 \mid \mathbf{x}_0] \\ &= 2 (f(\mathbf{x}_0) - E_{p(y|\mathbf{x})} [y_0 \mid \mathbf{x}_0]) = 0 \end{aligned}$$

- We minimize the expected loss by setting  $f$  to the conditional expectation of  $y$ :

$$f^*(\mathbf{x}_0) = E_{p(y|\mathbf{x})} [y_0 \mid \mathbf{x}_0]$$

---

# Generative versus discriminative learning

- A generative approach:
  - Infer the joint probability density  $p(\mathbf{x}, y)$
  - *Normalize* to find the conditional density  $p(y|\mathbf{x})$
  - Given a specific  $\mathbf{x}_0$ , *marginalize* to find the conditional expectation  $\hat{y} = E_{p(y|\mathbf{x})} [y_0|\mathbf{x}_0]$ .

---

# Generative versus discriminative learning

- A generative approach:
  - Infer the joint probability density  $p(\mathbf{x}, y)$
  - *Normalize* to find the conditional density  $p(y|\mathbf{x})$
  - Given a specific  $\mathbf{x}_0$ , *marginalize* to find the conditional expectation  $\hat{y} = E_{p(y|\mathbf{x})} [y_0|\mathbf{x}_0]$ .
- A discriminative approach:
  - Estimate/infer the conditional density  $p(y|\mathbf{x})$  *directly* from the data; don't bother with  $p(\mathbf{x}, y)$ .
  - Marginalize and obtain  $\hat{y}$ .

---

# Generative versus discriminative learning

- A generative approach:
  - Infer the joint probability density  $p(\mathbf{x}, y)$
  - *Normalize* to find the conditional density  $p(y|\mathbf{x})$
  - Given a specific  $\mathbf{x}_0$ , *marginalize* to find the conditional expectation  $\hat{y} = E_{p(y|\mathbf{x})} [y_0|\mathbf{x}_0]$ .
- A discriminative approach:
  - Estimate/infer the conditional density  $p(y|\mathbf{x})$  *directly* from the data; don't bother with  $p(\mathbf{x}, y)$ .
  - Marginalize and obtain  $\hat{y}$ .
- Non-probabilistic approach: ignore probabilities, estimate  $f(\mathbf{x})$  directly from the data.

---

## Decomposition of error

Let's take a closer look at the expected loss:

- $\hat{\mathbf{w}} = [\hat{w}_0, \hat{w}_1]^T$  are LSQ estimates from training data (assuming 1D case).
- $\mathbf{w}^* = [w_0^*, w_1^*]^T$  are *optimal* linear regression parameters (generally unknown!)
- $y - \hat{w}_0 - \hat{w}_1 x = (y - w_0^* - w_1^* x) + (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)$

---

## Decomposition of error

Let's take a closer look at the expected loss:

- $\hat{\mathbf{w}} = [\hat{w}_0, \hat{w}_1]^T$  are LSQ estimates from training data (assuming 1D case).
- $\mathbf{w}^* = [w_0^*, w_1^*]^T$  are *optimal* linear regression parameters (generally unknown!)
- $y - \hat{w}_0 - \hat{w}_1 x = (y - w_0^* - w_1^* x) + (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)$

$$\begin{aligned} E_{p(x,y)} \left[ (y - \hat{w}_0 - \hat{w}_1 x)^2 \right] &= E_{p(x,y)} \left[ (y - w_0^* - w_1^* x)^2 \right] \\ &+ 2E_{p(x,y)} \left[ (y - w_0^* - w_1^* x) (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x) \right] \\ &+ E_{p(x,y)} \left[ (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)^2 \right]. \end{aligned}$$

---

## Decomposition of error

Let's take a closer look at the expected loss:

- $\hat{\mathbf{w}} = [\hat{w}_0, \hat{w}_1]^T$  are LSQ estimates from training data (assuming 1D case).
- $\mathbf{w}^* = [w_0^*, w_1^*]^T$  are *optimal* linear regression parameters (generally unknown!)
- $y - \hat{w}_0 - \hat{w}_1 x = (y - w_0^* - w_1^* x) + (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)$

$$\begin{aligned} E_{p(x,y)} \left[ (y - \hat{w}_0 - \hat{w}_1 x)^2 \right] &= E_{p(x,y)} \left[ (y - w_0^* - w_1^* x)^2 \right] \\ &\quad + 2E_{p(x,y)} \left[ (y - w_0^* - w_1^* x) (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x) \right] \\ &\quad + E_{p(x,y)} \left[ (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)^2 \right]. \end{aligned}$$

- The second term vanishes since prediction errors  $y_0 - w_0^* - w_1^* x$  are uncorrelated with *any* linear function of  $x$  including  $w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x$ .

---

## Decomposition of error

$$E_{p(x,y)} \left[ (y - \hat{w}_0 - \hat{w}_1 x)^2 \right] = E_{p(x,y)} \left[ (y - w_0^* - w_1^* x)^2 \right] \\ + E_{p(x,y)} \left[ (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)^2 \right].$$

- *Structural error*  $E_{p(x,y)} \left[ (y - w_0^* - w_1^* x)^2 \right]$  measures inherent limitations of the chosen hypothesis class (linear function). This error will remain even with infinite training data.
- *Approximation error*  $E_{p(x,y)} \left[ (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)^2 \right]$  measures how close to the optimal  $\mathbf{w}^*$  is  $\hat{\mathbf{w}}$  estimated with finite training data.
- Note: since training data  $X, Y$  are random variables drawn from  $p(\mathbf{x}, y)$ , the estimated  $\hat{\mathbf{w}}$  is a random variable as well.

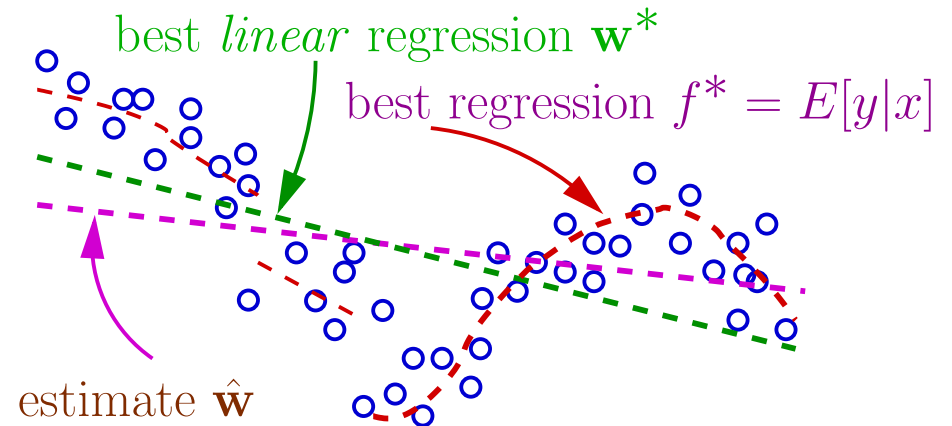
# Decomposition of error

- Structural error

$$E_{p(x,y)} \left[ (y - w_0^* - w_1^*x)^2 \right]$$

- Approximation error

$$E_{p(x,y)} \left[ (w_0^* + w_1^*x - \hat{w}_0 - \hat{w}_1x)^2 \right]$$



- For a *consistent* estimation procedure,  $\lim_{N \rightarrow \infty} \hat{\mathbf{w}} = \mathbf{w}^*$ , and so the approximation error decreases.
- The structural error can not be removed without changing the hypothesis class (e.g., moving from linear to quadratic regression).

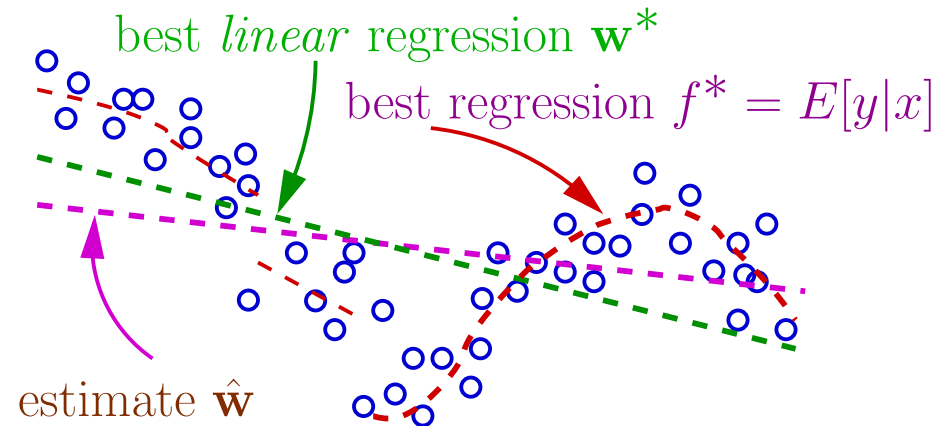
# Decomposition of error

- Structural error

$$E_{p(x,y)} \left[ (y - w_0^* - w_1^*x)^2 \right]$$

- Approximation error

$$E_{p(x,y)} \left[ (w_0^* + w_1^*x - \hat{w}_0 - \hat{w}_1x)^2 \right]$$



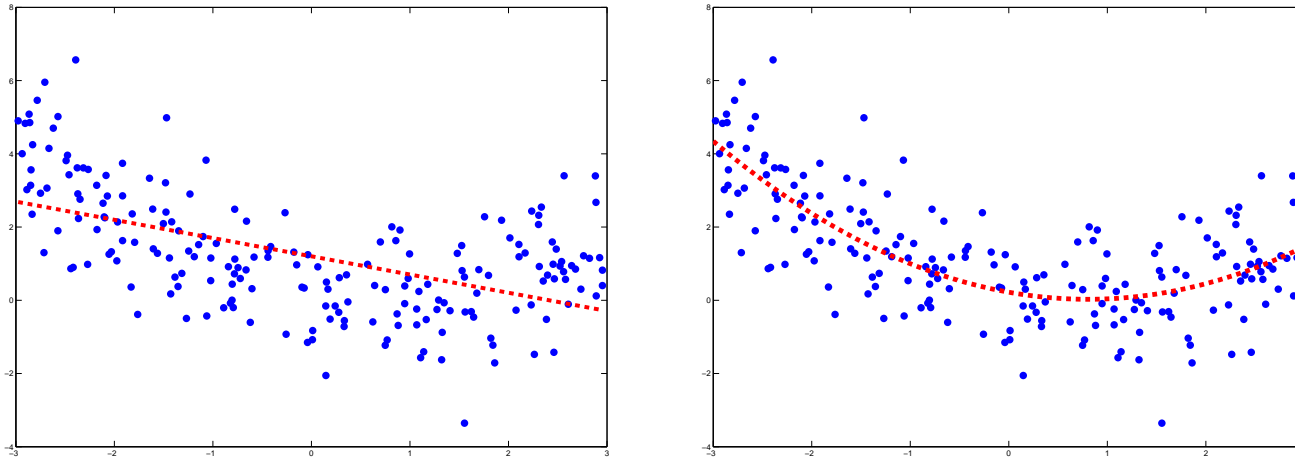
- For a *consistent* estimation procedure,  $\lim_{N \rightarrow \infty} \hat{\mathbf{w}} = \mathbf{w}^*$ , and so the approximation error decreases.
- The structural error can not be removed without changing the hypothesis class (e.g., moving from linear to quadratic regression).
- Structural error is minimized if  $f^* \in \mathcal{F}$ .

# Statistical view of regression

- We will now explicitly model the randomness in the data:

$$y = f(\mathbf{x}; \mathbf{w}) + \nu$$

where the *noise*  $\nu$  accounts for everything not captured by  $f$ .



- This definition of “noise” may include meaningful components, which are no longer part of noise once we move to a more complex  $f$ .

---

# Statistical view of regression

$$y = f(\mathbf{x}; \mathbf{w}) + \nu$$

- Under this model, the best predictor is

$$E_{p(y|\mathbf{x})} [f(\mathbf{x}; \mathbf{w}) + \nu | \mathbf{x}] = f(\mathbf{x}; \mathbf{w}) + E_{p(\nu)} [\nu]$$

- Typically,  $E_{p(\nu)} [\nu] = 0$  (*white noise*).
- Under such a model,  $f(\mathbf{x}; \mathbf{w})$  captures the expected value of  $y|\mathbf{x}$  if we believe the distribution in the model.
  - If (and only if) the model is “correct”,  $f$  is the optimal predictor!

# Gaussian noise model

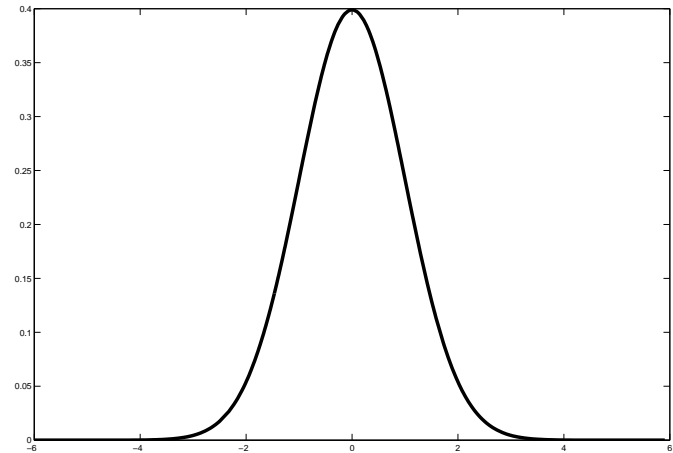
- Typical choice:  $p(\nu) \equiv \mathcal{N}(\nu; 0, \sigma^2)$

$$\text{1D Gaussian distribution: } \mathcal{N}(z; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right)$$

$$E_{\mathcal{N}(z; \mu, \sigma^2)}[z] = \mu,$$

$$E_{\mathcal{N}(z; \mu, \sigma^2)}[(z - E[z])^2] = \sigma^2.$$

- Gaussian is:
  - Symmetric;
  - Light-tail: probability of value far from mean is low;
  - Unimodal: single peak in the density function at  $\mu$ .



---

# Gaussian noise model

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

- Given the input  $\mathbf{x}$ , the label  $y$  is a random variable

$$p(y|\mathbf{x}; \mathbf{w}, \sigma) = \mathcal{N}(y; f(\mathbf{x}; \mathbf{w}), \sigma^2)$$

that is,

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

- This is an explicit *predictive* model of  $y$  that allows us, for instance, to *sample*  $y$  for a given  $\mathbf{x}$ .

---

# Likelihood

- The *likelihood* of the parameters  $\mathbf{w}$  given the observed data  $X = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ ,  $Y = [y_1, \dots, y_N]^T$  is defined as

$$\mathcal{P}(Y; \mathbf{w}, \sigma) \triangleq p(Y|X; \mathbf{w}, \sigma)$$

i.e., the probability of observing these  $y$ s for the given  $\mathbf{x}$ s, under the model parametrized by  $\mathbf{w}$  and  $\sigma$ .

- Under the assumption that data are i.i.d. (independently, identically distributed) according to  $p(\mathbf{x})$ ,

$$\mathcal{P}(Y; \mathbf{w}, \sigma) = \prod_{i=1}^N p(y_i | \mathbf{x}_i, \mathbf{w}, \sigma)$$

---

# Maximum likelihood estimation

- *Maximum likelihood* (ML) estimation principle:

$$\hat{\mathbf{w}}_{ML} = \underset{\mathbf{w}}{\operatorname{argmax}} \mathcal{P}(Y; \mathbf{w}, \sigma)$$

- Note: here we focus on  $\mathcal{P}$  as a function of  $\mathbf{w}$ .
- In case of Gaussian noise model:

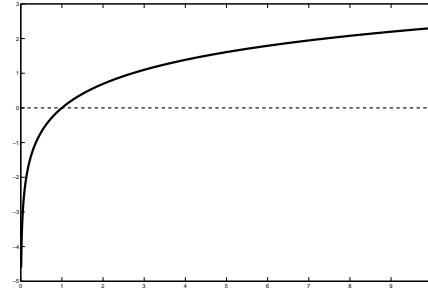
$$\hat{\mathbf{w}}_{ML} = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^N \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - f(\mathbf{x}_i; \mathbf{w}))^2}{2\sigma^2}\right)$$

- This may become numerically unwieldy. . .

---

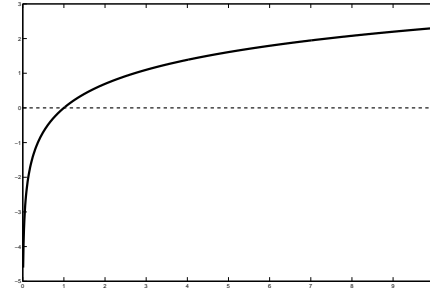
# Log-likelihood

- Properties of  $\log$ :
  - Defined for any  $x > 0$ .
  - Monotonically increasing.
  - $\log(AB) = \log A + \log B$ ,  $\log A^B = B \log A$ .



# Log-likelihood

- Properties of log:
  - Defined for any  $x > 0$ .
  - Monotonically increasing.
  - $\log(AB) = \log A + \log B$ ,  $\log A^B = B \log A$ .



- Maximizing  $\mathcal{P}(Y; \mathbf{w}, \sigma)$  is equivalent to maximizing *log-likelihood*  $\ell$ :

$$\begin{aligned}\ell(Y; \mathbf{w}, \sigma) &\triangleq \log \mathcal{P}(Y; \mathbf{w}, \sigma) = \log \prod_{i=1}^N p(y_i | \mathbf{x}_i, \mathbf{w}, \sigma) \\ &= \sum_{i=1}^N \log p(y_i | \mathbf{x}_i, \mathbf{w}, \sigma)\end{aligned}$$

---

# Log-likelihood, Gaussian noise

$$\begin{aligned}\ell(Y; \mathbf{w}, \sigma) &= \sum_{i=1}^N \log p(y_i | \mathbf{x}_i, \mathbf{w}, \sigma) \\ &= \sum_{i=1}^N \left[ -\frac{(y_i - f(\mathbf{x}_i; \mathbf{w}))^2}{2\sigma^2} - \log \sigma \sqrt{2\pi} \right]\end{aligned}$$

---

## Log-likelihood, Gaussian noise

$$\begin{aligned}\ell(Y; \mathbf{w}, \sigma) &= \sum_{i=1}^N \log p(y_i | \mathbf{x}_i, \mathbf{w}, \sigma) \\ &= \sum_{i=1}^N \left[ -\frac{(y_i - f(\mathbf{x}_i; \mathbf{w}))^2}{2\sigma^2} - \log \sigma \sqrt{2\pi} \right] \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - f(\mathbf{x}_i; \mathbf{w}))^2 - N \log \sigma \sqrt{2\pi}.\end{aligned}$$

- The second term is independent of  $\mathbf{w}$ .

---

# Maximum likelihood

- We can define a new loss function: *log-loss* (negative log-probability)

$$L(f(\mathbf{x}; \mathbf{w}), y) = -\log p(y|\mathbf{x}; \mathbf{w}, \sigma)$$

- Maximizing log-likelihood is equivalent to *minimizing* empirical log-loss.
- When the noise is Gaussian, this in turn is equivalent to minimizing average squared loss:

$$\begin{aligned} \operatorname{argmax}_{\mathbf{w}} \ell(Y; \mathbf{w}, \sigma) &= \operatorname{argmin}_{\mathbf{w}} -\sum_{i=1}^N \log p(y_i|\mathbf{x}_i; \mathbf{w}, \sigma) \\ &= \operatorname{argmin}_{\mathbf{w}} \sum_{i=1}^N (y_i - f(\mathbf{x}_i; \mathbf{w}))^2. \end{aligned}$$

---

# Maximum likelihood and least squares

- So, the ML estimate under the Gaussian noise model

$$\hat{\mathbf{w}}_{ML} = \operatorname{argmax}_{\mathbf{w}} \ell(Y; \mathbf{w}, \sigma)$$

is equivalent to the least squares principle (minimizing empirical squared loss):

$$\hat{\mathbf{w}}_{LSQ} = \operatorname{argmin}_{\mathbf{w}} \sum_{i=1}^N (y_i - f(\mathbf{x}_i; \mathbf{w}))^2$$

- Is it the case for *any* noise model?

---

## Next time

We will investigate the behavior of  $\hat{\mathbf{w}}$ , and discuss extensions of the simple linear regression model.

Time permitting, we will then move on to *classification* problems.