Announcements

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

- Learning by estimating parameters $w^*$ that minimize empirical loss $L_N(w) = \frac{1}{N} \sum_{i=1}^{N} L(f(x_i; w), y_i)$

- Expected loss (risk) $R(w) = E_{(x_0, y_0) \sim p(x, y)} [L(f(x_0; w), y_0)]$

- Least squares: $f(x; w) = w_0 + \sum_{j=1}^{d} w_j x_j$

$$w^* = \arg\min_{w} \frac{1}{N} \sum_{i=1}^{N} (f(x_i, w) - y_i)^2 = (X^T X)^{-1} X^T y$$

- Prediction errors $y_i - f(x_i; w)$ have zero mean and are uncorrelated with any linear function of the inputs $x$.
- 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?

$$f^* = \arg\min_{f: \mathcal{X} \rightarrow \mathbb{R}} E_{(x_0, y_0) \sim p(x, y)} \left[ (f(x_0) - y_0)^2 \right]$$

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?

$$f^* = \arg\min_{f : \mathcal{X} \to \mathbb{R}} E_{(x_0, y_0) \sim p(x, y)} [(f(x_0) - y_0)^2]$$

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

The product rule of probability: $p(x, y) = p(y|x)p(x)$

By definition: $E_{p(y, x)} [g(y, x)] = \int_x \int_y g(y, x)p(y|x)p(x)dydx$

$$E_{(x_0, y_0) \sim p(x, y)} [(f(x_0) - y_0)^2] = E_{x_0 \sim p(x)} \left[E_{y_0 \sim p(y|x)} [(f(x_0) - y_0)^2 | x_0] \right]$$
Best unrestricted predictor

\[ E_{(x_0, y_0) \sim p(x, y)} \left[ (f(x_0) - y_0)^2 \right] = \int_{x_0} \left\{ E_{y_0 \sim p(y|x)} \left[ (f(x_0) - y_0)^2 \mid x_0 \right] \right\} p(x_0) \, dx_0 \]

- If we minimize the inner conditional expectation for each \( x_0 \) separately, we will necessarily minimize the whole integral (outer expectation).
  - Must predict optimally \( y \) for any \( x \).

\[
\frac{\delta}{\delta f(x)} E_{p(y|x)} \left[ (f(x_0) - y_0)^2 \mid x_0 \right] = 2E_{p(y|x)} \left[ f(x_0) - y_0 \mid x_0 \right] \\
= 2 \left( f(x_0) - E_{p(y|x)} [y_0 \mid x_0] \right) = 0
\]

- We minimize the expected loss by setting \( f \) to the conditional expectation of \( y \):

\[
f^*(x_0) = E_{p(y|x)} [y_0 \mid x_0]
\]
Generative versus discriminative learning

- A generative approach:
  - Infer the joint probability density $p(x, y)$
  - Normalize to find the conditional density $p(y|x)$
  - Given a specific $x_0$, marginalize to find the conditional expectation $\hat{y} = E_{p(y|x)}[y_0|x_0]$. 
Generative versus discriminative learning

- A generative approach:
  - Infer the joint probability density $p(x, y)$
  - Normalize to find the conditional density $p(y|x)$
  - Given a specific $x_0$, marginalize to find the conditional expectation $\hat{y} = E_{p(y|x)}[y_0|x_0]$.

- A discriminative approach:
  - Estimate/infer the conditional density $p(y|x)$ directly from the data; don't bother with $p(x, y)$.
  - Marginalize and obtain $\hat{y}$. 
Generative versus discriminative learning

• A generative approach:
  – Infer the joint probability density \( p(x, y) \)
  – *Normalize* to find the conditional density \( p(y|x) \)
  – Given a specific \( x_0 \), *marginalize* to find the conditional expectation \( \hat{y} = E_{p(y|x)}[y_0|x_0] \).

• A discriminative approach:
  – Estimate/infer the conditional density \( p(y|x) \) *directly* from the data; don't bother with \( p(x, y) \).
  – Marginalize and obtain \( \hat{y} \).

• Non-probabilistic approach: ignore probabilities, estimate \( f(x) \) 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!)
- \( y - \hat{w}_0 - \hat{w}_1 x = (y - w_0^* - w_1^* x) + (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x) \)

\[
E_p(x,y) \left[ (y - \hat{w}_0 - \hat{w}_1 x)^2 \right] = E_p(x,y) \left[ (y - w_0^* - w_1^* x)^2 \right] \\
+ 2E_p(x,y) \left[ (y - w_0^* - w_1^* x) (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x) \right] \\
+ E_p(x,y) \left[ (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)^2 \right].
\]
Decomposition of error

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

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

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

- 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 \( w^* \) is \( \hat{w} \) estimated with finite training data.

- **Note:** since training data \( X, Y \) are random variables drawn from \( p(x, y) \), the estimated \( \hat{w} \) 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**
  \[
  E_p(x,y) \left[ (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)^2 \right]
  \]

- For a *consistent* estimation procedure, \( \lim_{N \to \infty} \hat{w} = w^* \), and so the approximation error decreases.

- The structural error cannot 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**
  \[ E_p(x,y) \left[ (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)^2 \right] \]

- For a *consistent* estimation procedure, \( \lim_{N \to \infty} \hat{w} = w^* \), 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 \( f^* \in \mathcal{F} \). 

\[\text{best linear regression } w^*\]

\[\text{best regression } f^* = E[y|x]\]

\[\text{estimate } \hat{w}\]
We will now explicitly model the randomness in the data:

\[ y = f(x; w) + \nu \]

where the \textit{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

\[ y = f(x; w) + \nu \]

- Under this model, the best predictor is

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

- Typically, \( E_{p(\nu)} [\nu] = 0 \) (white noise).

- Under such a model, \( f(x; w) \) captures the expected value of \( y|x \) 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: \( p(\nu) \equiv \mathcal{N}(\nu; 0, \sigma^2) \)

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

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

- Given the input \( x \), the label \( y \) is a random variable

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

that is,

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

- This is an explicit \textit{predictive} model of \( y \) that allows us, for instance, to \textit{sample} \( y \) for a given \( x \).
Likelihood

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

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

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

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

$$P(Y; \mathbf{w}, \sigma) = \prod_{i=1}^{N} p(y_i|x_i, \mathbf{w}, \sigma)$$
Maximum likelihood estimation

- **Maximum likelihood (ML) estimation principle:**

\[
\hat{w}_{ML} = \arg\max_w P(Y; w, \sigma)
\]

- **Note:** here we focus on \( P \) as a function of \( w \).

- **In case of Gaussian noise model:**

\[
\hat{w}_{ML} = \arg\max_w \prod_{i=1}^{N} \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{(y_i - f(x_i; w))^2}{2\sigma^2} \right)
\]

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

  $$
  \ell(Y; w, \sigma) \triangleq \log P(Y; w, \sigma) = \log \prod_{i=1}^{N} p(y_i|x_i, w, \sigma)
  = \sum_{i=1}^{N} \log p(y_i|x_i, w, \sigma)
  $$
Log-likelihood, Gaussian noise

\[
\ell(Y; w, \sigma) = \sum_{i=1}^{N} \log p(y_i|x_i, w, \sigma)
\]

\[
= \sum_{i=1}^{N} \left[ -\frac{(y_i - f(x_i; w))^2}{2\sigma^2} - \log \sigma \sqrt{2\pi} \right]
\]
Log-likelihood, Gaussian noise

\[
\ell(Y; w, \sigma) = \sum_{i=1}^{N} \log p(y_i | x_i, w, \sigma)
\]

\[
= \sum_{i=1}^{N} \left[ -\frac{(y_i - f(x_i; w))^2}{2\sigma^2} - \log \sigma \sqrt{2\pi} \right]
\]

\[
= -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - f(x_i; w))^2 - N \log \sigma \sqrt{2\pi}.
\]

- The second term is independent of \(w\).
Maximum likelihood

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

\[
L(f(x; w), y) = -\log p(y|x; w, \sigma)
\]

- 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:

\[
\text{argmax}_{w} \ell(Y; w, \sigma) = \text{argmin}_{w} - \sum_{i=1}^{N} \log p(y_{i}|x; w, \sigma)
\]

\[
= \text{argmin}_{w} \sum_{i=1}^{N} (y_{i} - f(x_{i}; w))^{2}.
\]
Maximum likelihood and least squares

• So, the ML estimate under the Gaussian noise model

\[ \hat{w}_{ML} = \arg\max_w \ell(Y; w, \sigma) \]

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

\[ \hat{w}_{LSQ} = \arg\min_{w} \sum_{i=1}^{N} (y_i - f(x_i; w))^2 \]

• Is it the case for any noise model?
Next time

We will investigate the behavior of $\hat{w}$, and discuss extensions of the simple linear regression model.
Time permitting, we will then move on to classification problems.