Lecture 6: Machine learning
• My office hours cancelled this week
  - Can do them by appointment instead
• PS3 out tonight
Today

• Temporal filtering
• Introduction to machine learning
Temporal filtering
Temporal filtering

Source: Freeman, Torralba, Isola
Videos

Source: Freeman, Torralba, Isola
Cube size = 128x128x90

Source: Freeman, Torralba, Isola
Videos

Cube size = 128x128x90

Source: Freeman, Torralba, Isola
Color amplification

Original

Phase-based motion attenuation + color amplification

[Wadhwa et al., Phase-based Video Motion Processing. SIGGRAPH 2013, Wu et al. SIGGRAPH 2012]
Motion magnification

Original

- Just spatial-temporal filtering!
- You’ll do a simplified version in PS3

[Wu et al., SIGGRAPH 2012]
Machine learning
Last week: hole filling

Source: A. Efros
[Hays and Efros. Scene Completion Using Millions of Photographs. SIGGRAPH 2007.]
2 Million Flickr Images

Source: A. Efros
Why does it work?
Nearest neighbors from a collection of 20 thousand images

Source: A. Efros
Nearest neighbors from a collection of 2 million images

Source: A. Efros
“Unreasonable Effectiveness of Data”  
[Halevy, Norvig, Pereira 2009]

Parts of our world can be explained by elegant mathematics  
- physics, chemistry, astronomy, etc.

But much cannot  
- psychology, economics, genetics, etc.

Enter The Data!

“For many tasks, once we have a billion or so examples, we essentially have a closed set that represents (or at least approximates) what we need…”

Source: A. Efros
ML intro slides from: Isola, Torralba, Freeman
What does ☆ do?

2 ☆ 3 = 36
7 ☆ 1 = 49
5 ☆ 2 = 100
2 ☆ 2 = 16

Source: Isola, Torralba, Freeman
2 \star 3 = 36
7 \star 1 = 49
5 \star 2 = 100
2 \star 2 = 16

Your brain

\[ x \star y \rightarrow (xy)^2 \]

3 \star 5 \rightarrow \[ x \star y \rightarrow (xy)^2 \] \rightarrow 225

Source: Isola, Torralba, Freeman
Learning from examples
(aka *supervised learning*)

Training data

\[
\begin{align*}
\{ & \text{input: } [2, 3], \text{output: } 36 \\
& \text{input: } [7, 1], \text{output: } 49 \\
& \text{input: } [5, 2], \text{output: } 100 \\
& \text{input: } [2, 2], \text{output: } 16 \\
\end{align*}
\]

\[\rightarrow\]

\[\text{Learner} \rightarrow f\]

Source: Isola, Torralba, Freeman
Learning from examples
(aka \textit{supervised learning})

Training data

\begin{align*}
\{x_1, y_1\} \\
\{x_2, y_2\} & \quad \rightarrow \quad \text{Learner} \\
\{x_3, y_3\} \\
\ldots
\end{align*}

\rightarrow f : X \rightarrow Y

Source: Isola, Torralba, Freeman
Learning from examples

(aka supervised learning)

Training data

\[
\begin{align*}
X & \quad Y \\
\{ & \quad \}, \quad \}, \quad \}, \quad \} \rightarrow \quad \text{Learner} \rightarrow \quad f : X \rightarrow Y
\end{align*}
\]

Source: Isola, Torralba, Freeman
Case study #1: Linear least squares
Training data

Test query

Source: Isola, Torralba, Freeman
The relationship between X and Y is roughly linear:

\[ y \approx \theta_1 x + \theta_0 \]
Training data

Search for the parameters, $\theta = \{\theta_0, \theta_1\}$, that best fit the data.

$$f_\theta(x) = \theta_1 x + \theta_0$$

Best fit in what sense?

Source: Isola, Torralba, Freeman
Search for the parameters, $\theta = \{\theta_0, \theta_1\}$, that best fit the data.

$$f_\theta(x) = \theta_1 x + \theta_0$$

Best fit in what sense?

The least-squares objective (aka loss) says the best fit is the function that minimizes the squared error between predictions and target values:

$$\mathcal{L}(\hat{y}, y) = (\hat{y} - y)^2 \quad \hat{y} \equiv f_\theta(x)$$

Source: Isola, Torralba, Freeman
Search for the parameters, \( \theta = \{\theta_0, \theta_1\} \), that best fit the data.

\[
f_\theta(x) = \theta_1 x + \theta_0
\]

Best fit in what sense?

The least-squares objective (aka loss) says the best fit is the function that minimizes the squared error between predictions and target values:

\[
\mathcal{L}(\hat{y}, y) = (\hat{y} - y)^2 \quad \hat{y} \equiv f_\theta(x)
\]

Source: Isola, Torralba, Freeman
Training data

Complete learning problem:

\[ \theta^* = \arg\min_{\theta} \sum_{i=1}^{N} (f_{\theta}(x_i) - y_i)^2 \]

\[ = \arg\min_{\theta} \sum_{i=1}^{N} (\theta_1 x_i + \theta_0 - y_i)^2 \]

Source: Isola, Torralba, Freeman
Training data

\{x_i, y_i\}_{i=1}^{N}

Test query

\(x', y'\)

Source: Isola, Torralba, Freeman
Training data

Test query

Source: Isola, Torralba, Freeman
Why is squared error a good objective?

Follows from two modeling assumptions

1. Observed Y’s are corrupted by Gaussian noise:

\[ Y \approx f_\theta(X) \]
\[ \epsilon = Y - f_\theta(X) \quad \epsilon \sim \mathcal{N}(0, \sigma) \]
\[ Y = f_\theta(X) + \epsilon \]
\[ P_\theta(Y = y | X = x) \propto \exp \left( -\frac{(y - f_\theta(x))^2}{2\sigma^2} \right) \]

2. Training datapoints are independently and identically distributed (iid):

\[ P_\theta(\{y_i\}_{i=1}^N | \{x_i\}_{i=1}^N) = \prod_{i=1}^N P_\theta(y_i | x_i) \]
Why is squared error a good objective?

\[ \theta^* = \arg \max_\theta \prod_{i=1}^N P_\theta(y_i|x_i) \]

\[ = \arg \max_\theta \prod_{i=1}^N \exp -\frac{(y - f_\theta(x))^2}{2\sigma^2} \]

\[ = \arg \max_\theta \log \prod_{i=1}^N \exp -\frac{(y - f_\theta(x))^2}{2\sigma^2} \]

\[ = \arg \max_\theta \sum_{i=1}^N -\frac{(y - f_\theta(x))^2}{2\sigma^2} \]

\[ = \arg \min_\theta \sum_{i=1}^N \frac{(y - f_\theta(x))^2}{2\sigma^2} \]

\[ = \arg \min_\theta \sum_{i=1}^N (y - f_\theta(x))^2 \]

Source: Isola, Torralba, Freeman
Why is squared error a good objective?

Under the assumption that the data is distributed as:

\[
Y = f_\theta(X) + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma)
\]

\[
P_\theta(\{y_i\}_{i=1}^N|\{x_i\}_{i=1}^N) = \prod_{i=1}^N P_\theta(y_i|x_i)
\]

The most probable estimate of \( \theta \) is:

\[
\theta^* = \arg \min_{\theta} \sum_{i=1}^N (f_\theta(x_i) - y_i)^2
\]

This is called the **maximum likelihood** estimator of \( \theta \).

Source: Isola, Torralba, Freeman
How to minimize the objective w.r.t. $\theta$?

$$
\theta^* = \arg \min_{\theta} \sum_{i=1}^{N} (f_{\theta}(x_i) - y_i)^2
$$

We’ll use methods from optimization.
How to minimize the objective w.r.t. $\theta$?

In the linear case:

$$
\theta^* = \arg \min_{\theta} \sum_{i=1}^{N} (\theta_1 x_i + \theta_0 - y_i)^2
$$

$$
J(\theta) = \sum_{I=1}^{N} (\theta_1 x_i + \theta_0 - y_i)^2
= (y - X\theta)^T (y - X\theta)
$$

$$
\theta^* = \arg \min_{\theta} J(\theta)
$$

$$
\frac{\partial J(\theta)}{\partial \theta} = 0
$$

$$
\frac{\partial J(\theta)}{\partial \theta} = 2(X^T X\theta - X^T y)
$$

Learning problem

$$
X = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{pmatrix} \quad \theta = (\theta_1 \quad \theta_0) \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}
$$

Solution

$$
2(X^T X\theta^* - X^T y) = 0
$$

$$
X^T X\theta^* = X^T y
$$

$$
\theta^* = (X^T X)^{-1} X^T y
$$

Source: Isola, Torralba, Freeman
Empirical Risk Minimization
(formalization of supervised learning)

\[
\theta^* = \arg \min_{\theta} \sum_{i=1}^{N} (\theta_1 x_i + \theta_0 - y_i)^2
\]

Linear least squares learning problem

Source: Isola, Torralba, Freeman
Empirical Risk Minimization
(formalization of supervised learning)

\[ f^* = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(x_i), y_i) \]

Objective function
(loss)

Hypothesis space

Training data

Source: Isola, Torralba, Freeman
Example 1: Linear least squares

Data

Learner

Source: Isola, Torralba, Freeman
Example 2: Program induction

Data

Training → Learner →

Learner

\[
\text{def predict}(x):
\]

\[
y = 0.8x + 2
\]

\[
\text{return } y
\]

Input →

\[
\text{def predict}(x):
\]

\[
y = 0.8x + 2
\]

\[
\text{return } y
\]

Output

Source: Isola, Torralba, Freeman
Example 3: Deep learning

Source: Isola, Torralba, Freeman
Learning for vision

Questions:

1. How do you represent the input and output?

2. What is the objective?

3. What is the hypothesis space? (e.g., linear, polynomial, neural net?)

4. How do you optimize? (e.g., gradient descent, Newton’s method?)

5. What data do you train on?

Source: Isola, Torralba, Freeman
Learning for vision

Questions:

1. How do you represent the input and output?

2. What is the objective?

3. What is the hypothesis space? (e.g., linear, polynomial, neural net?)

4. How do you optimize? (e.g., gradient descent, Newton’s method?)

5. What data do you train on?

Source: Isola, Torralba, Freeman
Image classification

image $x$ → Classifier $\rightarrow$ “Fish”

label $y$

Source: Isola, Torralba, Freeman
Image classification

Source: Isola, Torralba, Freeman
Image classification

Source: Isola, Torralba, Freeman
Image classification

Source: Isola, Torralba, Freeman
Training data

\[
\begin{align*}
\mathbf{x} & , \quad \text{"Fish"} \\
\mathbf{y} & ,  \\
\mathbf{x} & , \quad \text{"Grizzly"} \\
\mathbf{x} & , \quad \text{"Chameleon"} \\
\vdots
\end{align*}
\]

\[
\mathbf{x}
\]

\[
\arg \min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}_i), y_i)
\]

Source: Isola, Torralba, Freeman
How to represent class labels?

Training data

\[
\begin{align*}
\{ & \text{“Fish”} \\
\{ & \text{“Grizzly”} \\
\{ & \text{“Chameleon”} \\
\vdots & 
\end{align*}
\]

One-hot vector

\[
\begin{align*}
\{ & [0,0,1] \\
\{ & [0,1,0] \\
\{ & [1,0,0] \\
\vdots & 
\end{align*}
\]

Source: Isola, Torralba, Freeman
What should the loss be?

0-1 loss (number of misclassifications)

\[ L(\hat{y}, y) = 1(\hat{y} = y) \quad \text{← discrete; hard to optimize!} \]

Cross entropy

\[ L(\hat{y}, y) = H(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k \quad \text{← continuous, differentiable, convex} \]
Ground truth label $y$

$x$

$[0,0,0,0,0,1,0,0,...]$
The diagram illustrates the process of predicting animal species from an image. The function $f_\theta : X \rightarrow \mathbb{R}^K$ takes an input image $X$ and predicts a vector $\hat{y}$ containing probabilities for each of the $K$ possible animal species. The ground truth label $y$ is also shown, and the loss function $H(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k$ is used to measure the difference between the predicted and actual labels. The source of the diagram is Isola, Torralba, Freeman.
Loss
\[ H(y, \hat{y}) = - \sum_{k=1}^{K} y_k \log \hat{y}_k \]

Source: Isola, Torralba, Freeman
Softmax regression (a.k.a. multinomial logistic regression)

\[ f_\theta : X \rightarrow \mathbb{R}^K \]

\[ z = f_\theta(x) \quad \text{← logits: vector of K scores, one for each class} \]

\[ \hat{y} = \text{softmax}(z) \quad \text{← squash into a non-negative vector that sums to 1} \]

\[ \hat{y}_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_j}} \]

Source: Isola, Torralba, Freeman
Softmax regression (a.k.a. multinomial logistic regression)

Probabilistic interpretation:

\[ \hat{y} = [P_\theta(Y = 1|X = x), \ldots, P_\theta(Y = K|X = x)] \]  \hspace{1cm} \text{predicted probability of each class given input } x

\[ H(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k \] \hspace{1cm} \text{picks out the -log likelihood of the ground truth class } y \text{ under the model prediction } \hat{y}

\[ f^* = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^{N} H(y_i, \hat{y}_i) \] \hspace{1cm} \text{maximum likelihood}

Source: Isola, Torralba, Freeman
Softmax regression (a.k.a. multinomial logistic regression)

\[ f_\theta : X \to \mathbb{R}^K \]

\[ z = f_\theta(x) \]

\[ \hat{y} = \text{softmax}(z) \]

Source: Isola, Torralba, Freeman
The Problem of Generalization
Linear regression

\[ f_\theta(x) = \theta_0 + \theta_1 x \]

Source: Isola, Torralba, Freeman
Linear regression

\[ f_\theta(x) = \theta_0 + \theta_1 x \]

Source: Isola, Torralba, Freeman
Polynomial regression

$$f_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

K-th degree polynomial regression

Source: Isola, Torralba, Freeman
Training data

\[ \{x_i, y_i\}_{i=1}^N \]

Source: Isola, Torralba, Freeman
True data-generating process

$\rho_{\text{data}}$

Source: Isola, Torralba, Freeman
True data-generating process

\[ p_{\text{data}} \]

\[ \{x_i^{(\text{train})}, y_i^{(\text{train})}\}_{i=1}^{N} \sim p_{\text{data}} \]

\[ \{x_i^{(\text{test})}, y_i^{(\text{test})}\}_{i=1}^{M} \sim p_{\text{data}} \]

Source: Isola, Torralba, Freeman
Training objective:

\[
\sum_{i=1}^{N} (f_\theta(x_{i \text{train}}^{\text{train}}) - y_{i \text{train}})^2
\]

Test time evaluation:

\[
\sum_{i=1}^{M} (f_\theta(x_{i \text{test}}^{\text{test}}) - y_{i \text{test}})^2
\]

Source: Isola, Torralba, Freeman
**Training objective:**

\[
\sum_{i=1}^{N} (f_{\theta}(x_{i}^{\text{train}}) - y_{i}^{\text{train}})^2
\]

**True objective:**

\[
\mathbb{E}_{\{x,y\} \sim p_{\text{data}}} [(f_{\theta}(x) - y)^2]
\]
Generalization

“The central challenge in machine learning is that our algorithm must perform well on new, previously unseen inputs—not just those on which our model was trained. The ability to perform well on previously unobserved inputs is called generalization.

… [this is what] separates machine learning from optimization.”

— Deep Learning textbook (Goodfellow et al.)
What does ☆ do?

$2 \times 3 = 36$

$7 \times 1 = 49$

$5 \times 2 = 100$

$2 \times 2 = 16$

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

\[ f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k \]

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

$$f_{\theta}(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

$$f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k$$

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

\[ f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k \]

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

\[ f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k \]

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

\[ f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k \]

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

\[ f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k \]

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

\[ f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k \]

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

\[ f_{\theta}(x) = \sum_{k=0}^{K} \theta_k x^k \]

Source: Isola, Torralba, Freeman
What happens as we add more basis functions?

This phenomenon is called **overfitting**.

It occurs when we have too high capacity a model, e.g., too many free parameters, too few data points to pin these parameters down.
When the model does not have the capacity to capture the true function, we call this **underfitting**.

An underfit model will have high error on the training points. This error is known as **approximation error**.

Source: Isola, Torralba, Freeman
The true function is a quadratic, so a quadratic model \((K=2)\) fits quite well.

\[ \{x_i, y_i\}_{i=1}^{N} \]
Now we have zero approximation error — the curve passes exactly through each training point.

But we have high **generalization error**, reflected in the gap between the true function and the fit line. We want to do well on novel queries, which will be sampled from the green curve (plus noise).

Source: Isola, Torralba, Freeman
Underfitting

**K = 1**

- **High error on train set**
- **High error on test set**

Appropriate model

**K = 2**

- **Low error on train set**
- **Low error on test set**

Overfitting

**K = 10**

- **Lowest error on train set**
- **High error on test set**

Source: Isola, Torralba, Freeman
Underfitting

\[ K = 1 \]

\[ \{x_i, y_i\}_{i=1}^N \]

\[ X \]

\[ Y \]

Appropriate model

\[ K = 2 \]

\[ \{x_i, y_i\}_{i=1}^N \]

\[ X \]

\[ Y \]

Overfitting

\[ K = 10 \]

\[ \{x_i, y_i\}_{i=1}^N \]

\[ X \]

\[ Y \]

Source: Isola, Torralba, Freeman
We need to control the **capacity** of the model (e.g., use the appropriate number of free parameters).

The capacity may be defined as the number of hypotheses under consideration in the hypothesis space.

Complex models with many free parameters have high capacity.

Simple models have low capacity.

Source: Isola, Torralba, Freeman
Training error versus generalization error

Source: Isola, Torralba, Freeman

[“Deep Learning”, Goodfellow et al.]
How do we know if we are underfitting or overfitting?

Cross validation: measure prediction error on validation data

Source: Isola, Torralba, Freeman
Fitting just right

Underfitting?
1. add more parameters (more features, more layers, etc.)

Overfitting?
1. remove parameters
2. add regularizers

Selecting a hypothesis space of functions with just the right capacity is known as model selection

Source: Isola, Torralba, Freeman
Regularization

Empirical risk minimization:

\[ \theta^* = \arg \min_{\theta} \sum_{i=1}^{N} \mathcal{L}(f_{\theta}(x_i), y_i) + R(\theta) \]

Source: Isola, Torralba, Freeman
Regularized least squares

\[ f_\theta(x) = \sum_{k=0}^{K} \theta_k x^k \]

\[ R(\theta) = \lambda \|\theta\|_2^2 \]

Only use polynomial terms if you really need them! Most terms should be zero.

_ridge regression_, a.k.a., _Tikhonov regularization_

(Probabilistic interpretation: \( R \) is a Gaussian _prior_ over values of the parameters.)

Source: Isola, Torralba, Freeman
$$\theta^* = \arg \min_\theta \sum_{i=1}^{N} \mathcal{L}(f_\theta(x_i), y_i) + \lambda \|\theta\|_2^2$$

A

B

C

Low $\lambda$ — ?
Medium $\lambda$ — ?
High $\lambda$ — ?

Source: Isola, Torralba, Freeman

[Adapted from “Deep Learning”, Goodfellow et al.]
\[ \theta^* = \arg \min_{\theta} \sum_{i=1}^{N} \mathcal{L}(f_{\theta}(x_i), y_i) + \lambda \| \theta \|^2_2 \]

Source: Isola, Torralba, Freeman

High $\lambda$ — A
Medium $\lambda$ — B
Low $\lambda$ — C

[Adapted from “Deep Learning”, Goodfellow et al.]
Regularized polynomial least squares regression

Data: \( \{x_i, y_i\}_{i=1}^N \) →

Objective:
\[
\sum_{i=1}^N (f_\theta(x_i) - y_i)^2 + \lambda \|\theta\|_2^2
\]

Hypothesis space:
\[
f_\theta(x) = \sum_{k=0}^K \theta_k x^k
\]

Optimizer:
\[
\theta^* = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T y
\]

Source: Isola, Torralba, Freeman
Underfitting

\[ K = 1 \]

Simple model
Doesn’t fit the training data

Appropriate model

\[ K = 2 \]

Simple model
Fits the training data

Overfitting

\[ K = 10 \]

Complex model
Fits the training data

Source: Isola, Torralba, Freeman
Next lecture: neural networks