TODAY’S LECTURE

Data collection → Data processing → Exploratory analysis & Data viz → Analysis, hypothesis testing, & ML → Insight & Policy Decision

BIG THANKS: Zico Kolter (CMU) & Amol Deshpande (UMD)
Score $\psi$ of an instance $x$ and class $y$ is the sum of the weights for the features in that class:

$$\psi_{xy} = \sum \theta_n f_n(x, y) = \theta^T f(x, y)$$

Let's compute $\psi_{x1, y=hates\_cats}$:

- $\psi_{x1, y=hates\_cats} = \theta^T f(x_1, y = hates\_cats = 0)$
- $= 0*1 + -1*1 + 1*0 + -0.1*1 + 0*0 + 1*0 + -1*0 + 0.5*0 + 1*1$
- $= -1 - 0.1 + 1 = -0.1$

$$\theta^T = \begin{bmatrix} 0 & -1 & 1 & -0.1 & 0 & 1 & -1 & 0.5 & 1 \end{bmatrix}$$

$$f(x_1, y = 0) = \begin{bmatrix} 1 & l & \text{hates\_cats} & \text{likes\_cats} \end{bmatrix}$$
RECALL: EXPLICIT EXAMPLE OF STUFF FROM NLP CLASS

Saving the boring stuff:

• \( \psi_{x1,y=hates\_cats} = -0.1; \psi_{x1,y=likes\_cats} = +2.5 \)
  
• \( \psi_{x2,y=hates\_cats} = +1.9; \psi_{x2,y=likes\_cats} = +0.5 \)

We want to predict the class of each document:

\[
\hat{y} = \arg \max \theta^\top f(x, y)
\]

Document 1: \( \arg \max \{ \psi_{x1,y=hates\_cats}, \psi_{x1,y=likes\_cats} \} \)  

Document 2: \( \arg \max \{ \psi_{x2,y=hates\_cats}, \psi_{x2,y=likes\_cats} \} \)
We used a linear model to classify input documents. The model parameters $\theta$ were given to us a priori:
- (created by hand.)
- Typically, we cannot specify a model by hand.

Supervised machine learning provides a way to automatically infer the predictive model from labeled data.

Training Data

- $(x^{(1)}, y^{(1)})$
- $(x^{(2)}, y^{(2)})$
- $(x^{(3)}, y^{(3)})$
- ...

ML Algorithm

Hypothesis function

$y^{(i)} = h(x^{(i)})$

Predictions

New example $x$

$y = h(x)$
TERMINOLOGY

Input features: \( x^{(i)} \in \mathbb{R}^n, i = 1, \ldots, m \)

\[
\begin{array}{ccc}
  & \text{like} & \text{hate} & \text{cats} \\
\hline
x^{(1)T} &=& 1 & 1 & 0 & 1 \\
x^{(2)T} &=& 1 & 0 & 1 & 1 \\
\end{array}
\]

Outputs: \( y^{(i)} \in Y, i = 1, \ldots, m \)

\( y^{(i)} \in \{0, 1\} = \{ \text{hates\_cats, likes\_cats} \} \)

Model parameters:
\[
\theta \in \mathbb{R}^n
\]

\[
\theta^T = 0 \quad -1 \quad 1 \quad -0.1 \quad 0 \quad 1 \quad -1 \quad 0.5 \quad 1
\]
TERMINOLOGY

Hypothesis function: \( h_\theta : \mathbb{R}^n \rightarrow y \)
E.g., linear classifiers predict outputs using:

\[
h_\theta(x) = \theta^T x = \sum_{j=1}^{n} \theta_j \cdot x_j
\]

Loss function:
- Measures difference between prediction and the true output
- E.g., squared loss:
  \[
  \ell(\hat{y}, y) = (\hat{y} - y)^2
  \]
- E.g., hinge loss:
  \[
  \ell(y) = \max(0, 1 - t \cdot y)
  \]

Output \( t = \{-1,+1\} \) based on -1 or +1 class label
Classifier score \( y \)
THE CANONICAL MACHINE LEARNING PROBLEM

At the end of the day, we want to learn a hypothesis function that predicts the actual outputs well.

Choose the parameterization that minimizes loss!

\[
\text{minimize}_{\theta} \sum_{i=1}^{m} \ell(h_{\theta}(x^{(i)}), y^{(i)})
\]

*Not actually what we want – want it over the world of inputs – will discuss later …
HOW DO I MACHINE LEARN?

1. **What is the hypothesis function?**
   • Domain knowledge and EDA can help here.

2. **What is the loss function?**
   • We’ve discussed two already: squared and absolute.

3. **How do we solve the optimization problem?**
   • (We’ll cover gradient descent and stochastic gradient descent in class, but if you are interested, take CMSC422!)

First GIS result for “optimization”
ASIDE: LOSS FUNCTIONS
QUICK ASIDE ABOUT LOSS FUNCTIONS

Say we’re back to classifying documents into:
• hates_cats, translated to label y = -1
• likes_cats, translated to label y = +1

We want some parameter vector θ such that:
• $\psi_{xy} > 0$ if the feature vector x is of class likes_cat; (y = +1)
• $\psi_{xy} < 0$ if x’s label is y = -1

We want a hyperplane that separates positive examples from negative examples.

Why not use 0/1 loss; that is, the number of wrong answers?

$$\arg \min_\theta \sum_{i=1}^{n} 1 \left[ y^{(i)} \cdot \langle \theta, x^{(i)} \rangle \leq 0 \right]$$
MINIMIZING 0/1 LOSS IN A SINGLE DIMENSION

\[
\sum_{i=1}^{n} 1 \left[ y^{(i)} \cdot \langle \theta, x^{(i)} \rangle \leq 0 \right]
\]

Each time we change \( \theta \) such that the example is right (wrong) the loss will increase (decrease)
**MINIMIZING 0/1 LOSS OVER ALL $\Theta$**

$$\arg \min_\theta \sum_{i=1}^{n} 1 \left[ y^{(i)} \cdot \langle \theta, x^{(i)} \rangle \leq 0 \right]$$

This is NP-hard.

- Small changes in any $\theta$ can have large changes in the loss (the change isn’t continuous)
- There can be many local minima
- At any give point, we don’t have much information to direct us towards any minima

Maybe we should consider other loss functions.
What are some desirable properties of a loss function?

- **Continuous** so we get a local indication of the direction of minimization
- Only one (i.e., **global**) minimum
**CONVEX FUNCTIONS**

“A function is convex if the line segment between any two points on its graph lies above it.”

Formally, given function $f$ and two points $x, y$:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \ \forall \lambda \in [0, 1]$$
SURROGATE LOSS FUNCTIONS

For many applications, we really would like to minimize the 0/1 loss.

A surrogate loss function is a loss function that provides an upper bound on the actual loss function (in this case, 0/1).

We’d like to identify convex surrogate loss functions to make them easier to minimize.

Key to a loss function is how it scores the difference between the actual label y and the predicted label y’.
0/1 loss: \( \ell(\hat{y}, y) = 1 [y\hat{y} \leq 0] \)

Any ideas for surrogate loss functions ?????????
Want: a function that is continuous and convex and upper bounds the 0/1 loss.

• Hinge: \( \ell(\hat{y}, y) = \max(0, 1 - y\hat{y}) \)

• Exponential: \( \ell(\hat{y}, y) = e^{-y\hat{y}} \)

• Squared: \( \ell(\hat{y}, y) = (y - \hat{y})^2 \)

What do each of these penalize??????????
**SURROGATE LOSS FUNCTIONS**

0/1 loss:
\[ \ell(\hat{y}, y) = 1 \left[ y\hat{y} \leq 0 \right] \]

Hinge:
\[ \ell(\hat{y}, y) = \max(0, 1 - y\hat{y}) \]

Exponential:
\[ \ell(\hat{y}, y) = e^{-y\hat{y}} \]

Squared loss:
\[ \ell(\hat{y}, y) = (y - \hat{y})^2 \]

(Recall: \( y \) in \{-1, +1\})
## SOME ML ALGORITHMS

<table>
<thead>
<tr>
<th>Name</th>
<th>Hypothesis Function</th>
<th>Loss Function</th>
<th>Optimization Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Least squares</td>
<td>Linear</td>
<td>Squared</td>
<td>Analytical or GD</td>
</tr>
<tr>
<td>Linear regression</td>
<td>Linear</td>
<td>Squared</td>
<td>Analytical or GD</td>
</tr>
<tr>
<td>Support Vector Machine (SVM)</td>
<td>Linear, Kernel</td>
<td>Hinge</td>
<td>Analytical or GD</td>
</tr>
<tr>
<td>Perceptron</td>
<td>Linear</td>
<td>Perceptron criterion (~Hinge)</td>
<td>Perceptron algorithm, others</td>
</tr>
<tr>
<td>Neural Networks</td>
<td>Composed nonlinear</td>
<td>Squared, Hinge</td>
<td>SGD</td>
</tr>
<tr>
<td>Decision Trees</td>
<td>Hierarchical halfplanes</td>
<td>Many</td>
<td>Greedy</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>Linear</td>
<td>Joint probability</td>
<td>#SAT</td>
</tr>
</tbody>
</table>

RECALL: LINEAR REGRESSION

Scatterplot of Listing vs IncomePC

- Listing against IncomePC
- Points scattered around a linear trend line
- Axes range from 10000 to 900000 for Listing, and 15000 to 32500 for IncomePC
LINEAR REGRESSION AS MACHINE LEARNING

Let’s consider linear regression that minimizes the sum of squared error, i.e., least squares …

1. Hypothesis function:  
   • Linear hypothesis function
   \[ h_\theta(x) = \theta^T x \]

2. Loss function:  
   • Squared error loss
   \[ \ell(\hat{y}, y) = \frac{1}{2} (\hat{y} - y)^2 \]

4. Optimization problem:  
   \[ \text{minimize}_\theta \sum_{i=1}^{m} (\theta^T x^{(i)} - y^{(i)})^2 \]