I. Decision Theory: From Model to Answers

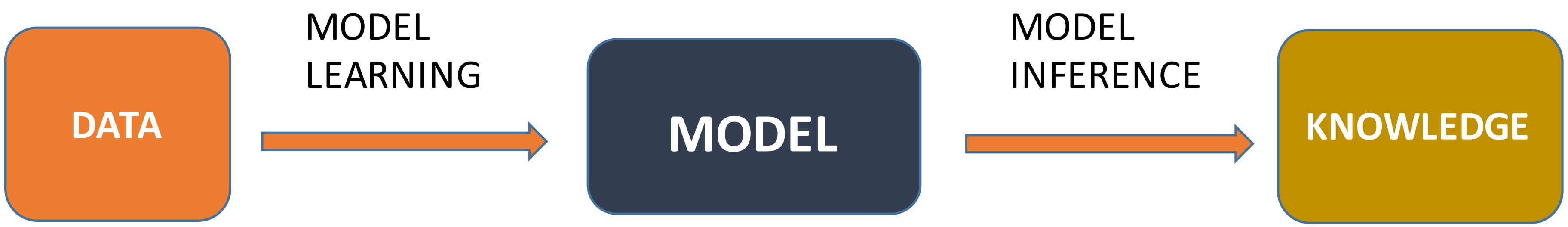
II. Empirical Risk Minimization

Pradeep Ravikumar
Co-instructor: Aarti Singh

Machine Learning
Feb 8, 2017

MACHINE LEARNING DEPARTMENT
Carnegie Mellon
School of Computer Science

Recall: Model-based ML



- Learning: From data to model
  - A model thus is a summary of the data
  - But also a story of how the data was generated
  - Could thus be used to describe how future data can be generated
  - **E.g. given (symptoms, diseases) data, a model explains how symptoms and diseases are related**

- Inference: From model to knowledge
  - Given the model, how can we answer questions relevant to us
  - **E.g. given (symptom, disease) model, given some symptoms, what is the disease?**

Model to Knowledge

- You know how to learn a model from data, with guarantees
- How do we go from model to knowledge?

- i.e. How do we get the answers we seek from the model?
- E.g. Recall “coin flip” example: the Billionaire might be really after answers to questions such as:
  - Which side is more likely in the next flip?
  - If a bookie gives 3 to 5 odds on tails, should he take the bet?

Model to Knowledge: Plugin Estimates

- In most cases, the knowledge we seek is a fixed function **f(P)** of the distribution of the data
  - E.g. is the coin fair? Does the coin have better odds than 3/5, etc.
- Once we learn a model, we have an estimate of the distribution of the data: $P_{\widehat{\theta}}$
- So we can simply "plugin" the model for the distribution to get our answers: $f(P_{\widehat{\theta}})$
- Is the coin fair: $\mathbb{I}(\theta == 1/2)$
  - Plugin Estimate: $\mathbb{I}(\widehat{\theta} == 1/2)$

Specification of Knowledge

- In the previous, the specification of what knowledge we were seeking was through an explicit function of the distribution
  - E.g. is the coin fair? Does the coin have better odds than 3/5, etc.
- But such an explicit specification is not always possible
- Think of the knowledge we seek as some “decision” given the underlying data
- QUESTIONS:
  - How do we characterize such decisions?
  - What is the optimal decision we can make?
  - How do we characterize optimality?
  - Falls under **decision theory** in economics

Specification of Knowledge

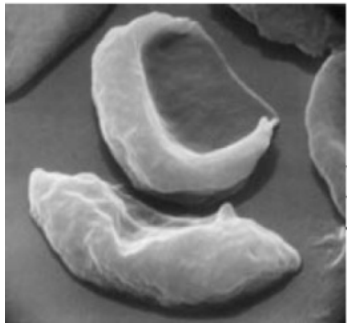
- In the previous, the specification of what knowledge we were seeking was through an explicit function of the distribution
  - E.g. is the coin fair? Does the coin have better odds than 3/5, etc.
- But such an explicit specification is not always possible
- An important ingredient in machine learning is to use **decision theory** to characterize the knowledge we seek
  - Through **performance measures** (also known as **loss/utility functions**, borrowing language from decision theory)
  - Whenever you encounter a task, you should automatically think about the appropriate **performance measure/loss function**

Example: Supervised Learning Prediction Task

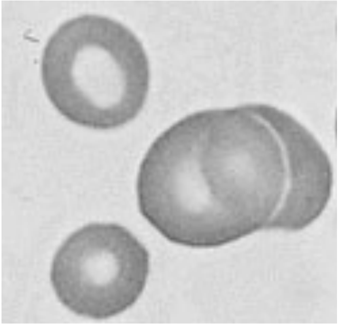
Task:

Given $X \in \mathcal{X}$, predict $Y \in \mathcal{Y}$.

$\equiv$ Construct **prediction rule** $f : \mathcal{X} \to \mathcal{Y}$



"Lupus (0)"

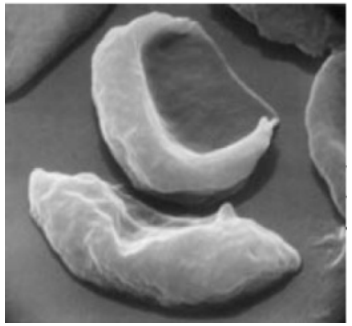


"Healthy (1)"

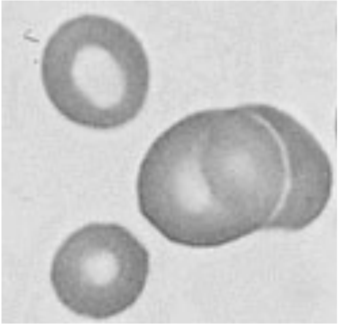
Example: Supervised Learning Prediction Task

Task: Given $X \in \mathcal{X}$, predict $Y \in \mathcal{Y}$.

$\equiv$ Construct **prediction rule** $f : \mathcal{X} \to \mathcal{Y}$



→ "Lupus cell (0)"



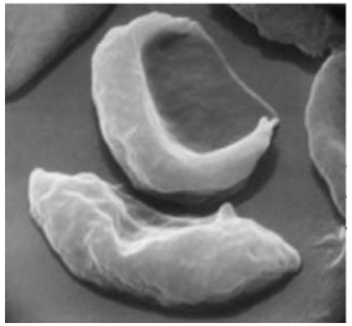
→ "Healthy cell (1)"

But I can always come up with a prediction rule: always say it's not LUPUS!

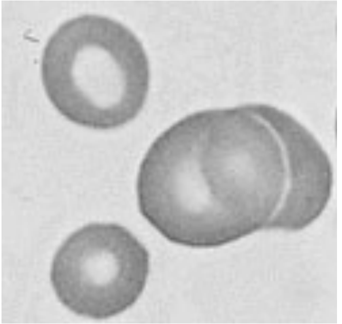
Example: Supervised Learning Prediction Task

Task: Given $X \in \mathcal{X}$, predict $Y \in \mathcal{Y}$.

$\equiv$ Construct **prediction rule** $f : \mathcal{X} \rightarrow \mathcal{Y}$



"Lupus (0)"



"Healthy (1)"

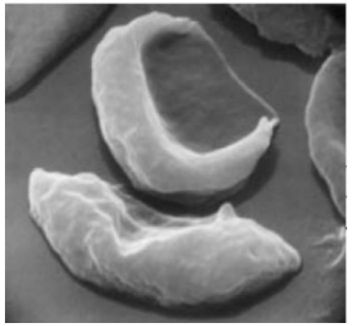
To complete the specification of the task, we need something more!!!

Characterize Task using Performance Measures

What is the “loss” I suffer when I take decision **f**?

Performance Measure:

$\text{loss}(Y, f(X))$ - Measure of closeness between true label *Y* and prediction *f*(*X*)



X	*Y*	*f*(*X*)	$\text{loss}(Y, f(X))$
	“Lupus”	“Lupus”	0
		“Healthy”	1

$$\text{loss}(Y, f(X)) = 1_{\{f(X) \neq Y\}}$$ **0/1 loss**

Performance Measures

Performance:
Measure:

$\text{loss}(Y, f(X))$ - Measure of closeness between true label *Y* and prediction *f*(*X*)

Don't just want label of one test data (cell image), but any cell image $X \in \mathcal{X}$

$$(X, Y) \sim P_{XY}$$

Given a cell image drawn randomly from the collection of all cell images, how well does the predictor perform on average?

$$\boxed{\text{Risk } R(f) \equiv \mathbb{E}_{XY}\left[\text{loss}(Y, f(X))\right]}$$

Performance Measures

Performance:
Measure:

$\text{loss}(Y, f(X))$ - Measure of closeness between true label *Y* and prediction *f*(*X*)

Don't just want label of one test data (cell image), but any cell image $X \in \mathcal{X}$

$$(X, Y) \sim P_{XY}$$

Given a cell image drawn randomly from the collection of all cell images, how well does the predictor perform on average?

What is the "risk" of taking decision **f**?

$$\boxed{\text{Risk } R(f) \equiv \mathbb{E}_{XY}\left[\text{loss}(Y, f(X))\right]}$$

Bayes Optimal Rule

Knowledge That we seek: Construct **prediction rule** $f^* : \mathcal{X} \to \mathcal{Y}$

$$f^*(P) = \arg\min_f \mathbb{E}_{(X,Y)\sim P}[\text{loss}(Y, f(X))]$$

Bayes optimal rule

Best possible performance:

Bayes Risk $R(f^*) \leq R(f)$ for all f

Bayes Optimal Rule

<u>Knowledge</u>
<u>That we seek</u>:

Construct **prediction rule** $f^* : \mathcal{X} \to \mathcal{Y}$

$$f^*(P) = \arg\min_f \mathbb{E}_{(X,Y)\sim P}[\text{loss}(Y, f(X))]$$ Bayes optimal rule

loss$(Y, f(X))$	Risk $R(f)$	Bayes Optimal Rule $f^*(P)$
$1_{\{f(X)\neq Y\}}$ **0/1 loss**	$P(f(X) \neq Y)$ **Probability of Error**	$f^*(P) = \mathbb{I}(P(Y=1\|X) > 1/2)$
$(f(X) - Y)^2$ **square loss**	$\mathbb{E}[(f(X) - Y)^2]$ **Mean Square Error**	$f^*(P) = \mathbb{E}(Y\|X)$

Bayes Optimal Rule

Knowledge That we seek: Construct **prediction rule** $f^* : \mathcal{X} \to \mathcal{Y}$

$$f^*(P) = \arg\min_f \mathbb{E}_{(X,Y)\sim P}[\text{loss}(Y, f(X))]$$

Bayes optimal rule

Best possible performance:

Bayes Risk $\quad R(f^*) \leq R(f)$ for all f

Model-free Methods

Knowledge That we seek: Construct **prediction rule** $f^* : \mathcal{X} \rightarrow \mathcal{Y}$

$$f^*(P) = \arg\min_f \mathbb{E}_{(X,Y)\sim P}[\text{loss}(Y, f(X))]$$

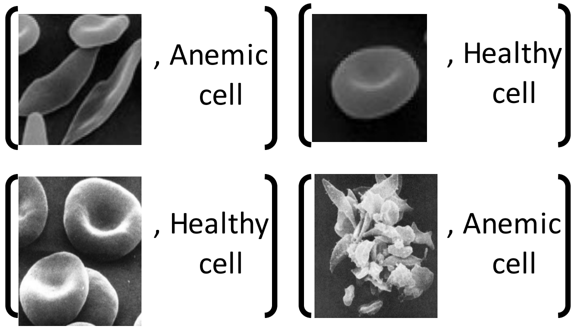
Bayes optimal rule

Optimal rule is not computable
- depends on unknown distribution P over (X,Y) !

MODEL BASED METHODS: Use a model for P_{XY} !

MODEL-FREE METHODS: Estimate the knowledge through some learning algorithm that does not go through a model for P_{XY}

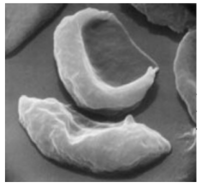
Model-free Methods



Learning algorithm $\Rightarrow \widehat{f}_n$

Data $\{(X_i, Y_i)\}_{i=1}^n$

$\widehat{f}_n$ is a mapping from $\mathcal{X} \rightarrow \mathcal{Y}$

$\widehat{f}_n$  = "Anemic cell"

Test data X

Popular Approach for model-free ML: **Empirical Risk Minimization**

Knowledge That we seek: Construct **prediction rule** $f^* : \mathcal{X} \to \mathcal{Y}$

$$f^*(P) = \arg\min_f \mathbb{E}_{(X,Y)\sim P}[\text{loss}(Y, f(X))]$$ Bayes optimal rule

Given $\{X_i, Y_i\}_{i=1}^n$, **learn** prediction rule $\widehat{f}_n : \mathcal{X} \to \mathcal{Y}$

Empirical Risk Minimizer:

$$\widehat{f}_n = \arg\min_f \frac{1}{n} \sum_{i=1}^n [\text{loss}(Y_i, f(X_i))]$$

$$\boxed{\frac{1}{n} \sum_{i=1}^n [\text{loss}(Y_i, f(X_i))] \xrightarrow[\text{Numbers}]{\text{Law of Large}} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]}$$

Empirical Risk Minimization

- Very Popular Approach in ML
- Given a loss function, and data, estimate decision function by minimizing “empirical risk”
- Typically restrict decision to lie within some restricted set
  - This restricted set is NOT a statistical model
  - Could capture our prior information
  - Or just be for computational convenience

$$\widehat{f} = \arg\inf_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathrm{loss}(Y_i, f(X_i)) \right\}$$

Empirical Risk Minimization: Considerations

$$\widehat{f} = \arg \inf_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \text{loss}(Y_i, f(X_i)) \right\}$$

- **Computational Considerations**: How do we solve the above optimization problem in a computationally tractable manner?

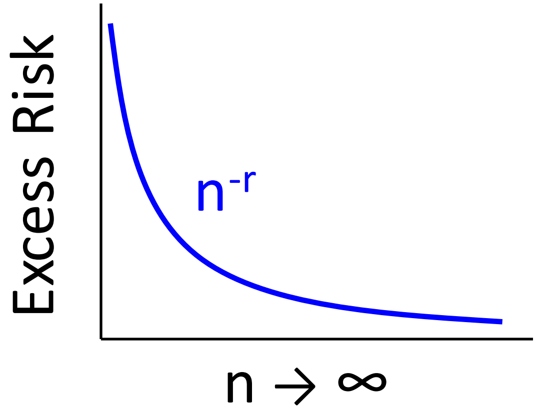
- **Statistical Considerations**: What guarantees do I have for the empirical risk minimizer (ERM) estimator?

Statistical Considerations: Consistency and Rate of Convergence

- How does the performance of the algorithm compare with ideal performance?

Excess Risk $\quad \mathbb{E}_{D_n}\left[R(\widehat{f}_n)\right] - R(f^*)$

- **Consistent** algorithm if Excess Risk → 0 as n → ∞
- **Rate of Convergence**



More later …

Computational Considerations

$$\widehat{f} = \arg \inf_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathrm{loss}(Y_i, f(X_i)) \right\}$$

- Even when class of functions is simple (e.g. class of linear functions), the above optimization need not be **convex**
- This non-convexity, and consequently, computational intractability holds for 0-1 loss classification

0-1 Loss for Classification

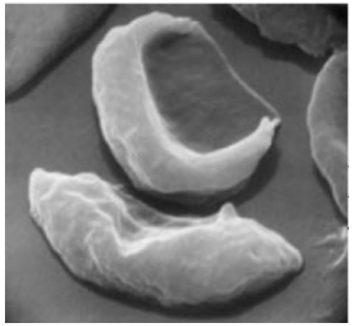
$$\ell_{0/1}(Y, f(X)) = \mathbb{I}(Y \neq f(X))$$

The loss is either zero or one (hence its name)

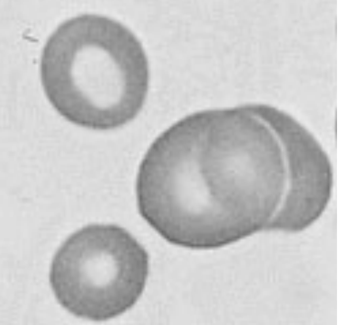
Loss is zero if the classifier outputs the label exactly

Loss is one if not

Binary Classification



“Lupus (-1)”



“Healthy (1)”

Binary Classification: Setup

Output Label: $Y \in \{-1, 1\}$

Input Features: $X \in \mathcal{X}$

Classifier: $f : \mathcal{X} \mapsto \{-1, 1\}$

Discriminant: $f : \mathcal{X} \mapsto \mathbb{R}$.
Given a discriminant, we use $\mathrm{sign}(f(x))$ as the corresponding *classifier*

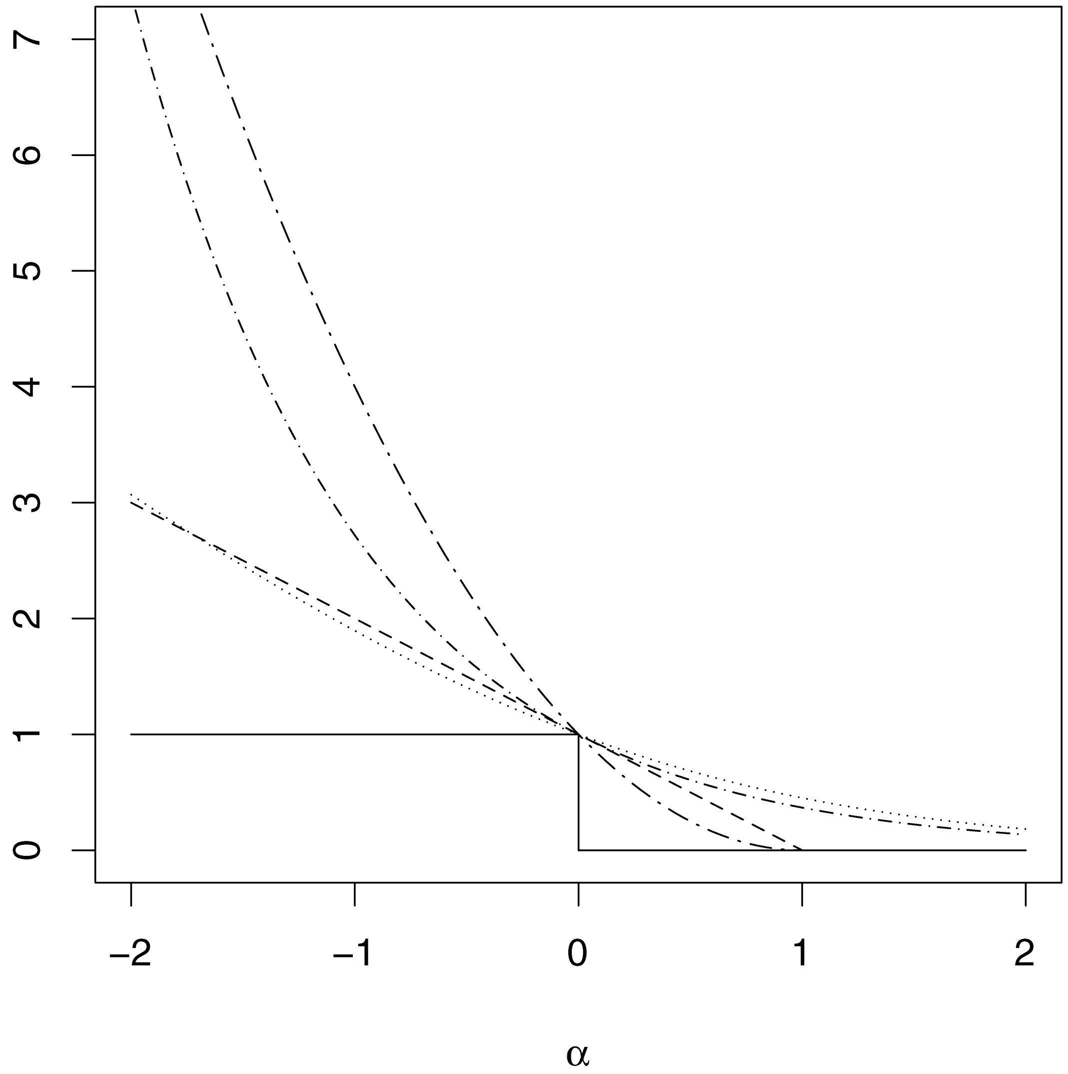
Binary Classification and 0-1 Loss

$$\ell_{0/1}(Y, f(X)) := \mathbb{I}(Y \neq \text{sign}(f(X)))$$
$$= \mathbb{I}(Y\, f(X) < 0)$$

Can write it as $\ell_{0/1}(Y, f(X)) = \ell(Y\, f(X))$
where $\ell(\alpha) = \mathbb{I}(\alpha < 0)$

Empirical Risk Minimizer with respect to 0-1 loss is computationally intractable in large part because 0-1 loss \ell(\alpha) above is **non-convex**

Binary Classification: Convex Surrogates



Different loss functions $\ell(\alpha)$
where use in classification would be as:
$\ell(Y, f(X)) = \ell(Y\, f(X))$

Binary Classification

$$\widehat{f}_{0/1} = \arg \inf_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell_{0/1}(Y_i \, f(X_i)) \right\}$$

ERM with respect to 0-1 loss: computationally intractable

$$\widehat{f}_{\phi} = \arg \inf_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \phi(Y_i \, f(X_i)) \right\}$$

ERM with respect to convex surrogate loss: computationally tractable!
Basis of all modern classifiers: boosting, support vector machines, logistic regression, etc.