I. Decision Theory: From Model to Answers

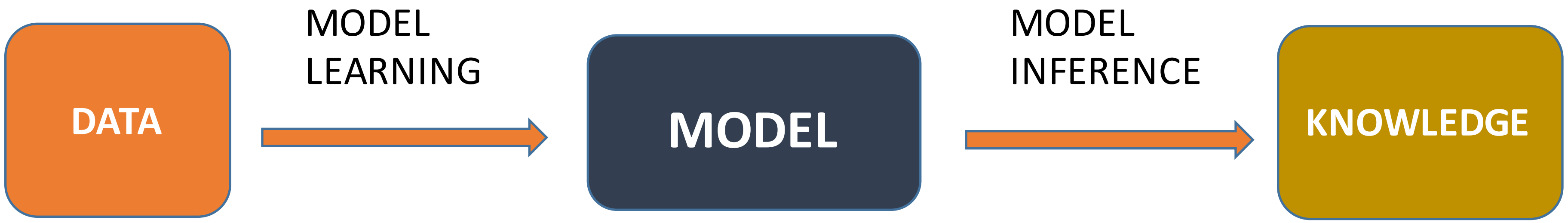
II. Empirical Risk Minimization

Pradeep Ravikumar
Co-instructor: Manuela Veloso

Machine Learning
Jan 29, 2018

ML MACHINE LEARNING DEPARTMENT
Carnegie Mellon.
School of Computer Science

Recall: Model-based ML



- Learning: From data to model
  - A model explains how the data was 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 model is the Bernoulli distribution
  - The Billionaire might not care about Bernoulli distribution per se, as much as 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 model **P** of the data
  - is the coin fair: I(p == ½)?
  - does the coin have better odds than 3/5: I(p >= 3/5)

Model to Knowledge: Plugin Estimates

- In most cases, the knowledge we seek is a fixed function **f(P)** of the distribution **P** of the data
  - is the coin fair: I(p == ½)?
  - does the coin have better odds than 3/5: I(p >= 3/5)

- 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**

Specification of Knowledge

- **Decision theory** can be used to characterize the decision to take
  - Through **performance measures** (also known as **loss/utility functions**)
  - Whenever you encounter a task, you should automatically think about the appropriate **performance measure/loss function**

From model to knowledge: the general case

Given model parameter θ, we then pick our action a from a set A by solving a decision-theoretic optimization problem:

$$\min_{a \in A} \ell(\theta, a).$$

Here $\ell(\theta, a)$ is the loss of taking action a when model parameter is θ.

Example: Finance

- Suppose you have a model that specifies the value of apple stock, or how much the stock price will go up in next 1 hour (instead of bias of Billionaire's coins)
- Your set of actions are how much apple stock to buy or sell
- Given the model, you have to minimize some loss function that balances risk and reward to decide the action **a in A** (how much stock to buy or sell).

Example: Electricity Generation

- Suppose you have a model that predicts consumer demand of electricity
- Your set of actions are based on how to schedule the generators of electrical plant
- Given the model, have to minimize some loss function to decide which and how and when of the generation of electricity

Unsupervised vs Supervised

- Learning a Bernoulli distribution as the model for a sequence of coin flips is an example of an "unsupervised learning" problem
- In a "supervised learning" problem, you have an input and an output, and the goal is to predict an output **given an input**
- Coin Flips: Predict coin flips given "features" about coin
- Finance: predict how much apple stock will move up given economic indicators
- Electricity Generation: predict consumer demand given past demand, weather

Supervised Learning, and Going from Model to Knowledge

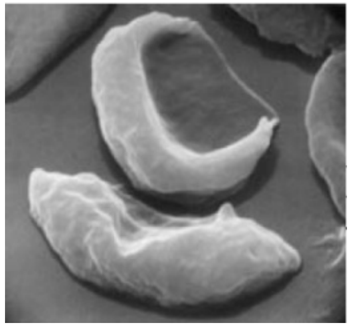
- For supervised learning problems, it is natural to talk about the knowledge first, and model second

- In particular, a decision-theoretic **loss function** is part of the problem specification in supervised learning

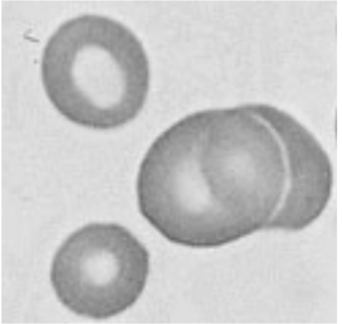
Supervised Learning Prediction Task

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

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



“Lupus cell (0)”

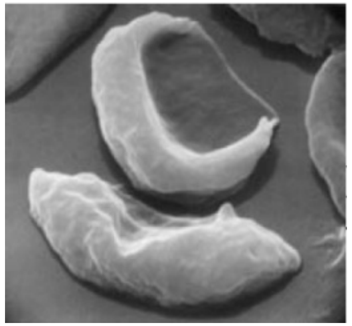


“Healthy cell (1)”

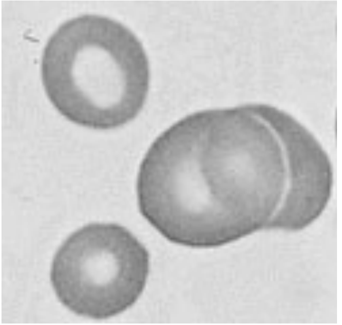
Example: Supervised Learning Prediction Task

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

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



"Lupus cell (0)"



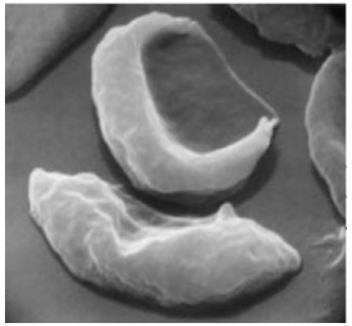
"Healthy cell (1)"

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

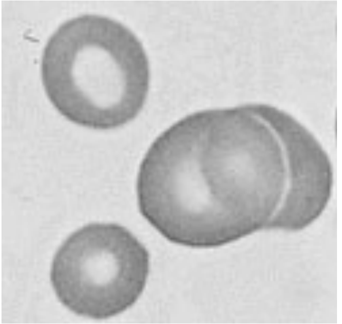
Example: Supervised Learning Prediction Task

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

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



"Lupus (0)"



"Healthy (1)"

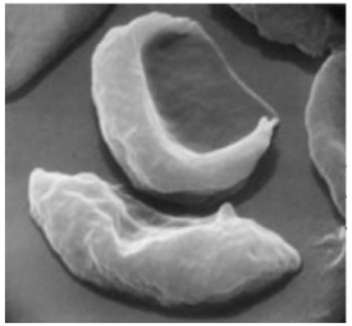
To complete the specification of the task, we need something more!!!

Characterize Task using Performance Measures

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

Performance Measure:

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

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

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

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	Share price, Y	$f(X)$	$\text{loss}(Y, f(X))$
Past performance, trade volume etc. as of Sept 8, 2010	"\$24.50"	"\$24.50"	0
		"\$26.00"	1?
		"\$26.10"	2?

$$\boxed{\text{loss}(Y, f(X)) = (f(X) - Y)^2 \quad \textbf{square loss}}$$

Performance Measures

Performance:

Measure:

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

We don't just want to correctly label one test sample (in this case, cell image), but most cell images $X \in \mathcal{X}$

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*)

We don't just want to correctly label one test sample (in this case, cell image), but most cell images $X \in \mathcal{X}$

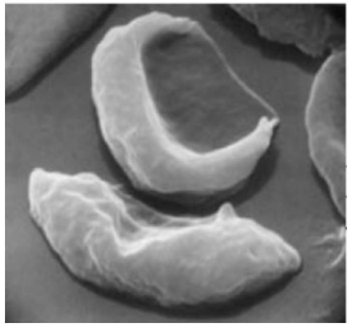
Given a cell image drawn randomly from the collection of all cell images, how well does the predictor perform on average?

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

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

Performance Measures

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



→ “Anemic cell”

→ Share Price “$ 24.50”

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

Bayes Optimal Rule

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

Best possible performance:

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

Bayes Optimal Rule

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

$\text{loss}(Y, f(X))$	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 $R(f^*) \leq R(f)$ for all f

Model-free Methods

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

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

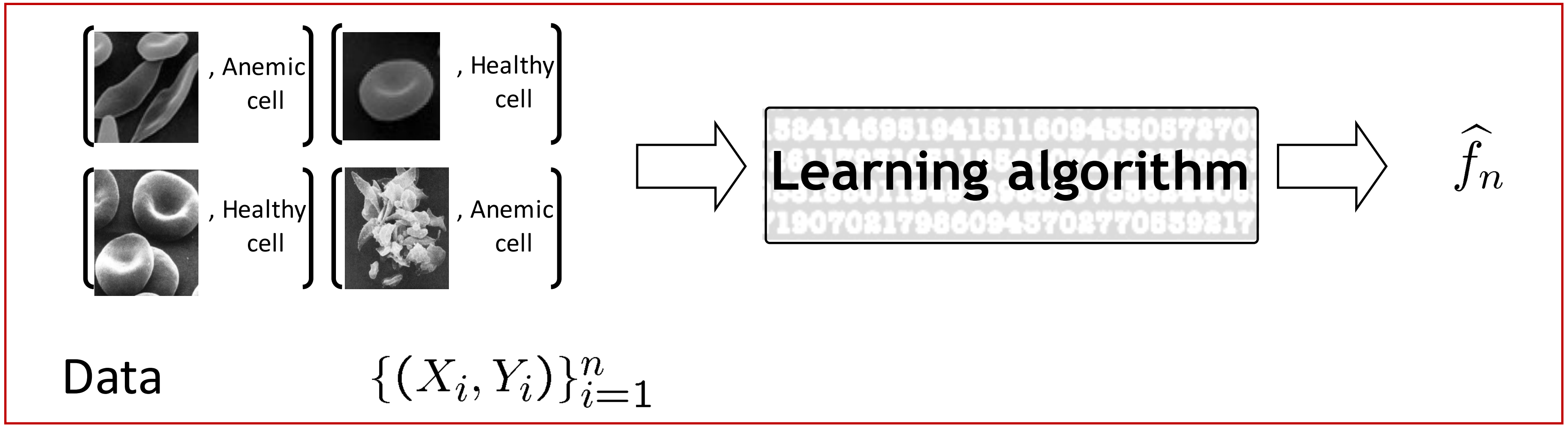
Bayes optimal rule

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

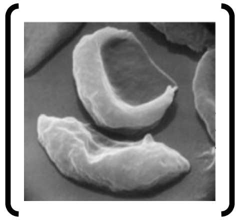
MODEL BASED METHODS: Use a model for P_{XY} !

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

Model-free Methods



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

$\widehat{f}_n$  = “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
    - 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

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

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

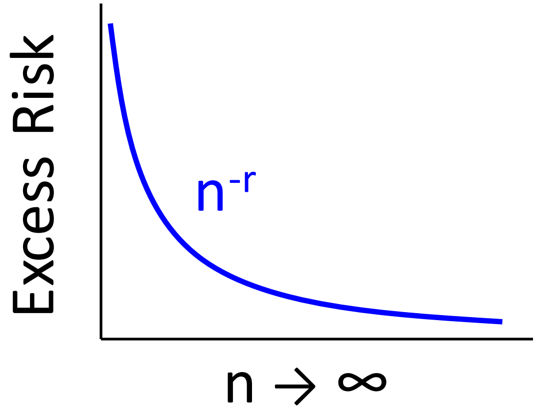
- **Statistical Considerations**: What guarantees do I have for the empirical risk minimizer (ERM) estimator?

Statistical Considerations: Consistency and Rate of Convergence

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

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

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



More later …

Computational Considerations

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

- 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