I. Parametric Models: Prior Information
II. From Models to Answers

Pradeep Ravikumar

Machine Learning
Jan 25, 2017

MACHINE LEARNING DEPARTMENT
Carnegie Mellon.
School of Computer Science

Recall: Your first consulting job

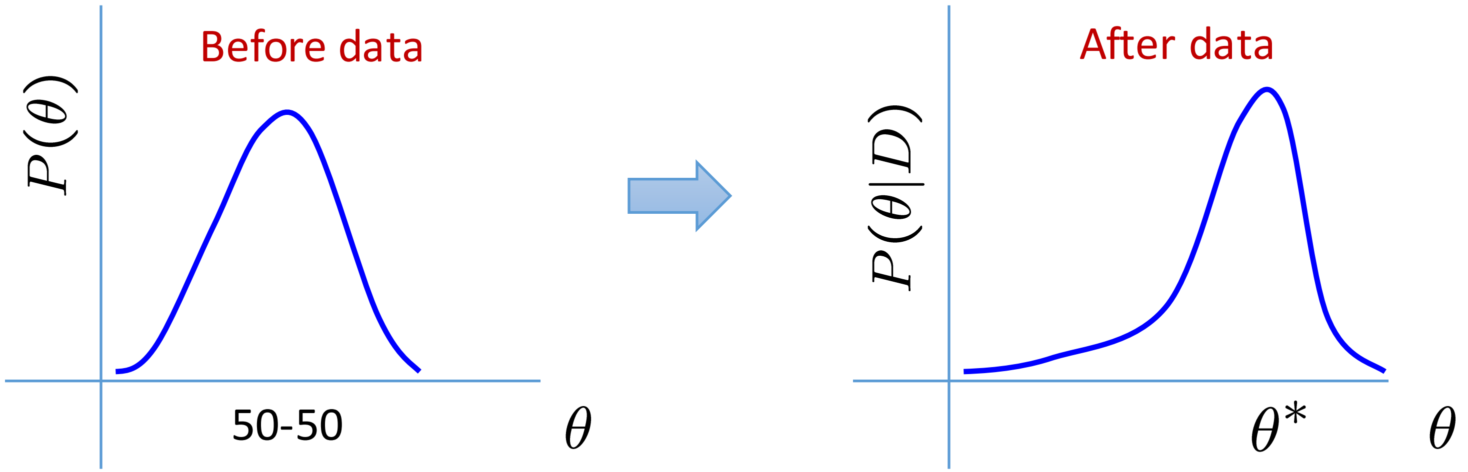
- A billionaire from the suburbs of Seattle asks you a question:
  - He says: I have a coin, if I flip it, what's the probability it will fall with the head up?
  - You say: Please flip it a few times:

  - You say: The probability is: **3/5** because… frequency of heads in all flips
  - **He says: But can I put money on this estimate?**
  - You say: ummm…. Maybe not.
    - Not enough flips (less than sample complexity)

What about prior knowledge?

- Billionaire says: Wait, I know that the coin is “close” to 50-50. What can you do for me now?
- **You say: I can learn it the Bayesian way…**

- Rather than estimating a single θ, we obtain a distribution over possible values of θ



Bayesian Learning

- Use Bayes rule:

$$\underbrace{P(\theta}_{\text{Parameters}} \mid \underbrace{\mathcal{D})}_{\text{Data}} = \frac{\overbrace{P(\mathcal{D} \mid \theta)}^{\text{likelihood}}\overbrace{P(\theta)}^{\text{prior}}}{P(\mathcal{D})}$$

Bayes, Thomas (1763) An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, **53:370-418**

Bayesian Learning

- Use Bayes rule:

$$P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta) P(\theta)}{P(\mathcal{D})}$$

- Or equivalently:

$$\underbrace{P(\theta \mid \mathcal{D})}_{\text{posterior}} \propto \underbrace{P(\mathcal{D} \mid \theta)}_{\text{likelihood}} \underbrace{P(\theta)}_{\text{prior}}$$

Bayes, Thomas (1763) An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, **53:370-418**

AIDS test (Bayes rule)

Data

- **Approximately 0.1% are infected**
- **Test detects all infections**
- **Test reports positive for 1% healthy people**

[Slide from Prof. Barnabas]

AIDS test (Bayes rule)

Data

- **Approximately 0.1% are infected**
- **Test detects all infections**
- **Test reports positive for 1% healthy people**

Probability of having AIDS if test is positive:

$$P(a=1|t=1) = \frac{P(t=1|a=1)P(a=1)}{P(t=1)}$$

$$= \frac{P(t=1|a=1)P(a=1)}{P(t=1|a=1)P(a=1) + P(t=1|a=0)P(a=0)}$$

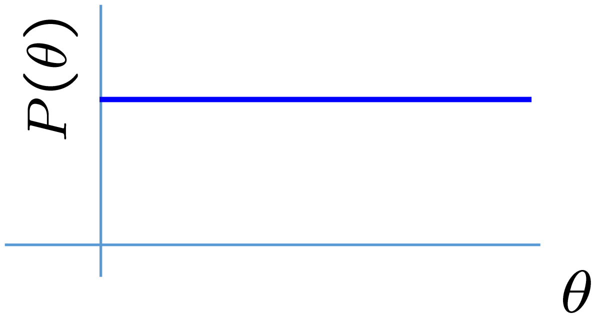
$$= \frac{1 \cdot 0.001}{1 \cdot 0.001 + 0.01 \cdot 0.999} = 0.091$$

Only 9%!...

[Slide from Prof. Barnabas]

Prior distribution

- From where do we get the prior?
  - Represents expert knowledge (philosophical approach)
  - Simple posterior form (engineer's approach)
- Uninformative priors:
  - Uniform distribution



- Conjugate priors:
  - Closed-form representation of posterior
  - $P(\theta)$ and $P(\theta|D)$ have the same algebraic form as a function of \theta

Conjugate Prior

- P(θ) and P(θ|D) have the same form as a function of theta

Eg. 1 Coin flip problem

Likelihood given Bernoulli model:

$$P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

If prior is Beta distribution,

$$P(\theta) = \frac{\theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T)$$

Then posterior is Beta distribution

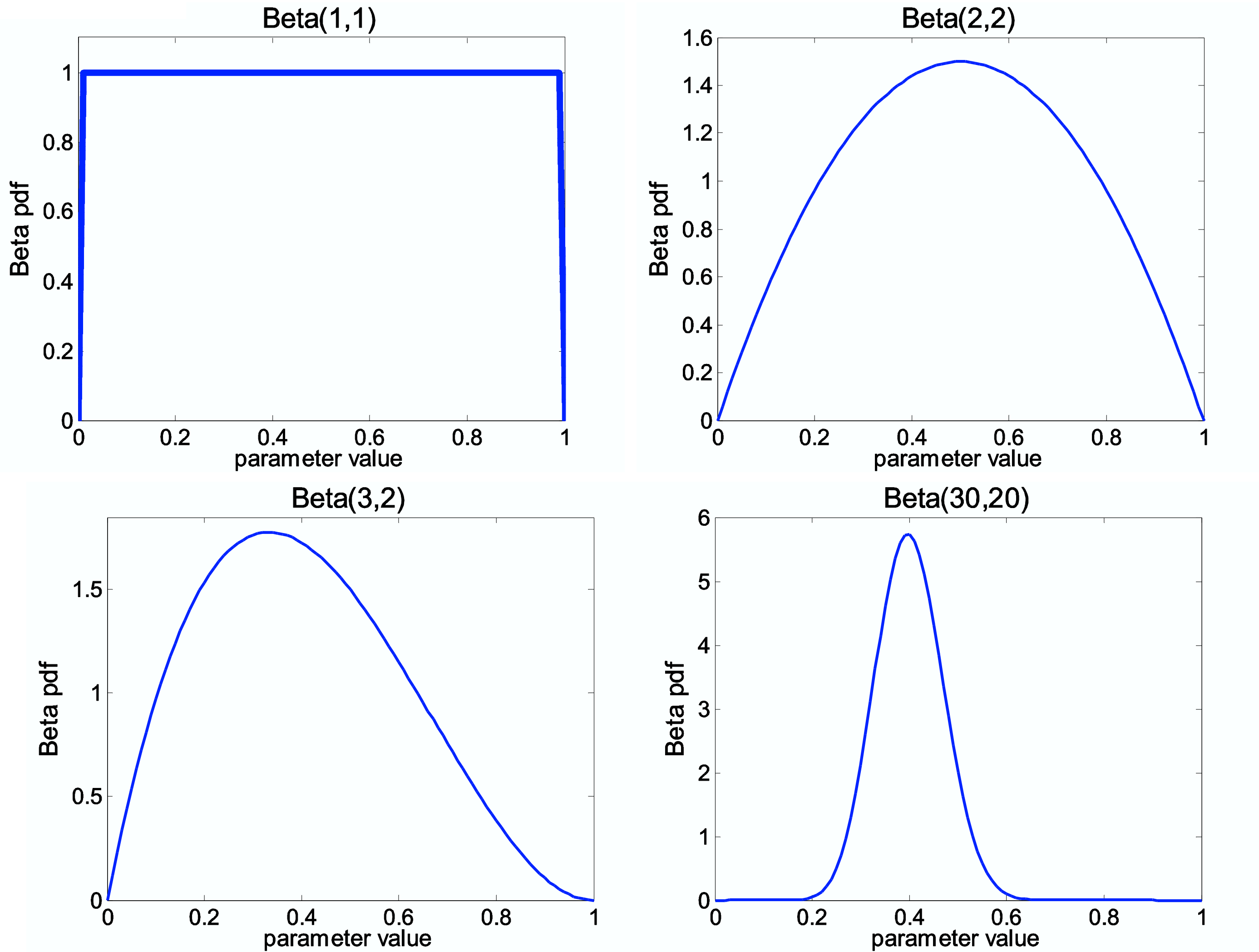
$$P(\theta|D) \sim Beta(\beta_H + \alpha_H, \beta_T + \alpha_T)$$

For Binomial, conjugate prior is Beta distribution.

Beta distribution

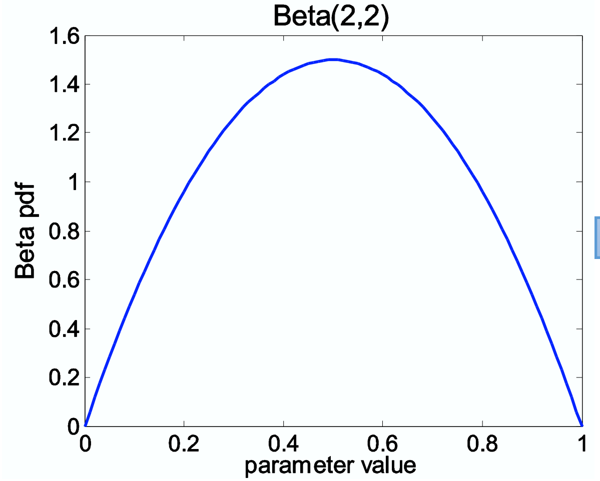
$Beta(\beta_H, \beta_T)$

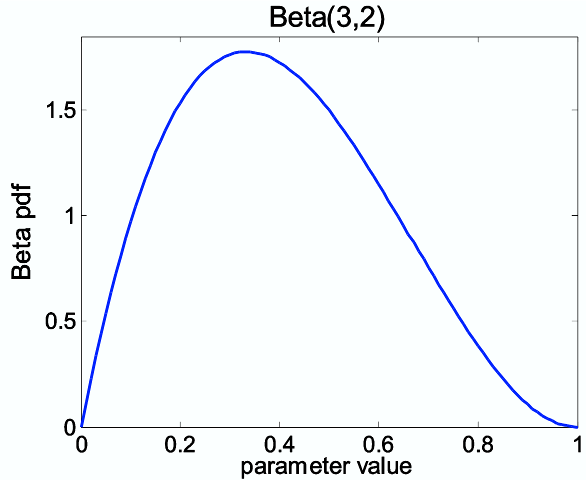
More concentrated as values of β_H, β_T increase

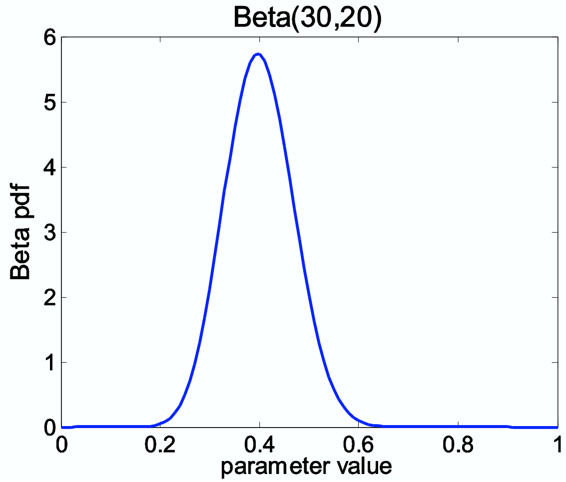


Beta conjugate prior

$$P(\theta) \sim Beta(\beta_H, \beta_T) \qquad P(\theta|D) \sim Beta(\beta_H + \alpha_H, \beta_T + \alpha_T)$$







As $n = \alpha_H + \alpha_T$ increases

As we get more samples, effect of prior is "washed out"

Conjugate Prior

- $P(\theta)$ and $P(\theta|D)$ have the same form

Eg. 2 Dice roll problem (6 outcomes instead of 2)

Likelihood is ~ Multinomial($\theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$)

$$P(\mathcal{D} \mid \theta) = \theta_1^{\alpha_1}\theta_2^{\alpha_2}\ldots\theta_k^{\alpha_k}$$

If prior is Dirichlet distribution,

$$P(\theta) = \frac{\prod_{i=1}^{k}\theta_i^{\beta_i - 1}}{B(\beta_1, \ldots, \beta_k)} \sim \text{Dirichlet}(\beta_1, \ldots, \beta_k)$$

Then posterior is Dirichlet distribution

$$P(\theta|D) \sim \text{Dirichlet}(\beta_1 + \alpha_1, \ldots, \beta_k + \alpha_k)$$

For Multinomial, conjugate prior is Dirichlet distribution.

Posterior Distribution

- The approach seen so far is what is known as a **Bayesian** approach
- Prior information encoded as a **distribution** over possible values of parameter
- Using the Bayes rule, you get an updated **posterior** distribution over parameters, which you provide with flourish to the Billionaire
- But the billionaire is not impressed
  - Distribution? I just asked for one number: is it 3/5, 1/2, what is it?
  - How do we go from a distribution over parameters, to a single estimate of the true parameters?

Maximum A Posteriori Estimation

Choose θ that maximizes a posterior probability

$$\begin{aligned}\widehat{\theta}_{MAP} &= \arg\max_{\theta} \quad P(\theta \mid D) \\ &= \arg\max_{\theta} \quad P(D \mid \theta)P(\theta)\end{aligned}$$

MAP estimate of probability of head:

$$P(\theta|D) \sim Beta(\beta_H + \alpha_H, \beta_T + \alpha_T)$$

$$\widehat{\theta}_{MAP} = \frac{\alpha_H + \beta_H - 1}{\alpha_H + \beta_H + \alpha_T + \beta_T - 2}$$

Mode of Beta distribution

MLE vs. MAP

- Maximum Likelihood estimation (MLE)

  Choose value that maximizes the probability of observed data

$$\widehat{\theta}_{MLE} = \arg\max_{\theta} P(D|\theta)$$

- Maximum *a posteriori* (MAP) estimation

  Choose value that is most probable given observed data and prior belief

$$\begin{aligned}\widehat{\theta}_{MAP} &= \arg\max_{\theta} P(\theta|D) \\ &= \arg\max_{\theta} P(D|\theta)P(\theta)\end{aligned}$$

When is MAP same as MLE?

MLE vs. MAP

$$\widehat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

What if we toss the coin too few times?

- You say: Probability next toss is a head = 0
- Billionaire says: You're fired! ...with prob 1 ☺

$$\widehat{\theta}_{MAP} = \frac{\alpha_H + \beta_H - 1}{\alpha_H + \beta_H + \alpha_T + \beta_T - 2}$$

- Beta prior equivalent to extra coin flips
- As $n \longrightarrow$ infty, prior is "forgotten"
- **But, for small sample size, prior is important!**

MLE vs MAP

You are no good when sample is small

You give a different answer for different priors

MAP for Gaussian mean and variance

- Conjugate priors
  - Mean: Gaussian prior
  - Variance: Wishart Distribution

- Prior for mean:

$$P(\mu \mid \eta, \lambda) = \frac{1}{\lambda\sqrt{2\pi}} e^{\frac{-(\mu-\eta)^2}{2\lambda^2}} \quad = N(\eta, \lambda^2)$$

MAP for Gaussian Mean

$$\widehat{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\widehat{\mu}_{MAP} = \frac{\frac{1}{\sigma^2}\sum_{i=1}^{n} x_i + \frac{\eta}{\lambda^2}}{\frac{n}{\sigma^2} + \frac{1}{\lambda^2}}$$

MAP of Gaussian variance - Later

Prior Information

- In the Bayesian approach, the prior information is encoded through a prior distribution over the parameters
- Seems onerous: the distribution typically seems to be obtained from convenience (conjugate distribution)
- What other ways can we encode our prior knowledge about the parameters?
- A non-Bayesian approach is via constraints

Encoding prior information via constraints

MLE:

$$\max_{\theta} \log \mathbb{P}(D; \theta).$$

Encoding prior information via constraints

MLE:

$$\max_{\theta} \log \mathbb{P}(D; \theta).$$

Constrained MLE:

$$\max_{\theta} \log \mathbb{P}(D; \theta)$$
$$\text{s.t.}\, \mathcal{R}(\theta) <= C.$$

Encoding prior information via constraints

MLE:

$$\max_{\theta} \log \mathbb{P}(D;\theta).$$

Constrained MLE:

$$\max_{\theta} \log \mathbb{P}(D;\theta)$$
$$\text{s.t.}\, \mathcal{R}(\theta) <= C.$$

When $\mathcal{R}(\theta)$ is convex, constrained MLE is equivalent to regularized MLE:

$$\max_{\theta} \left\{\log \mathbb{P}(D;\theta) + \lambda\, \mathcal{R}(\theta)\right\}.$$

Regularized MLE

Regularized MLE:

$$\max_{\theta} \left\{ \log \mathbb{P}(D; \theta) + \lambda \mathcal{R}(\theta) \right\}.$$

Trades off maximizing the log-likelihood (i.e. fit to data), against the "prior" constraints encoded by regularization (which do not involve the data at all).

Regularized MLE

Regularized MLE:

$$\max_{\theta} \left\{ \log \mathbb{P}(D; \theta) + \lambda \mathcal{R}(\theta) \right\}.$$

Trades off maximizing the log-likelihood (i.e. fit to data), against the "prior" constraints encoded by regularization (which do not involve the data at all).

The MAP estimator can be seen to be a special case by simply setting

$$\lambda \mathcal{R}(\theta) = \log\, P(\theta).$$

Regularized MLE

Regularized MLE:

$$\max_{\theta} \left\{ \log \mathbb{P}(D; \theta) + \lambda \mathcal{R}(\theta) \right\}.$$

Trades off maximizing the log-likelihood (i.e. fit to data), against the "prior" constraints encoded by regularization (which do not involve the data at all).

The MAP estimator can be seen to be a special case by simply setting

$$\lambda \mathcal{R}(\theta) = \log P(\theta).$$

Here, the tradeoff between likelihood and prior is naturally captured by setting the regularization function equal to the log of the prior distribution.

Popular Regularization functions

- ℓ_2 **regularization**:

$$\mathcal{R}(\theta) = \|\theta\|_2^2 = \sum_{j=1}^{p} \theta_j^2 .$$

This regularization encodes the prior information that the parameter values are not too large (where how large is determined by the regularization tradeoff parameter λ).

This regularization is thus a “general purpose” regularization function (who wants their parameters to be very large?)

Popular Regularization functions

- ℓ_1 **regularization**:

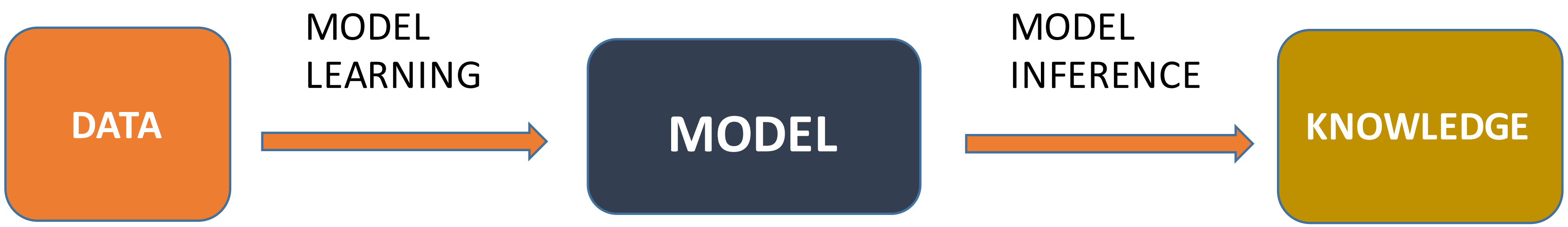
$$\mathcal{R}(\theta) = \|\theta\|_1 = \sum_{j=1}^{p} |\theta_j|.$$

This regularization encodes the prior information that the parameter values be **sparse**: i.e. with many zero values.

This is a very important prior constraint in big data settings:
with very large number of parameters, we expect the true model to depend on only a few non-zero parameters.

Widely used in high-dimensional model learning: called LASSO when used with linear regression models.

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

- We now 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. 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: $\mathbb{I}(\widehat{\theta} == 1/2)$

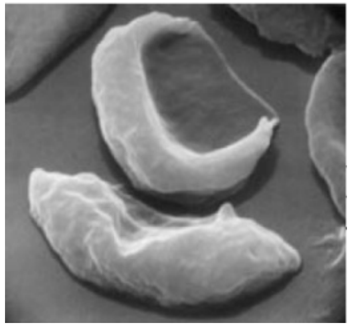
Specification of Knowledge

- 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
- An important construct in machine learning is a language for an implicit specification of task/what knowledge we seek
  - Through "performance measures"
  - Whenever you encounter a task, you should automatically think about the appropriate performance measure

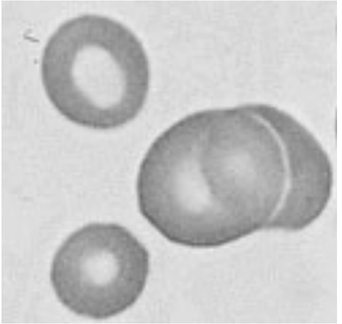
Supervised Learning Prediction Task

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

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



"Anemic cell (0)"

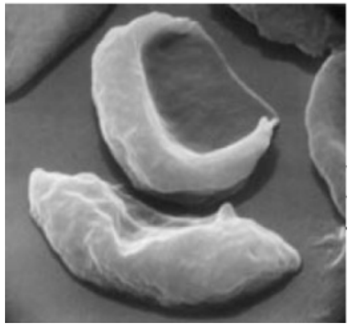


"Healthy cell (1)"

Performance Measures

Performance Measure:

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

X	*Y*	*f*(*X*)	$\text{loss}(Y, f(X))$
	"Anemic cell"	"Anemic cell"	0
		"Healthy cell"	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*)

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*)

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

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

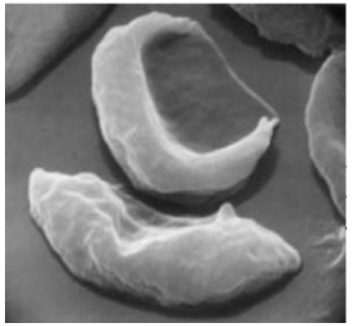
Given a cell image drawn randomly from the collection of all cell images, how well does the predictor perform on average?

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

Performance Measures

Performance:
Measure:

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



“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** $f^* : \mathcal{X} \to \mathcal{Y}$

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

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

Bayes optimal rule

Best possible performance:

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

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

Use a model for P_{XY} !

Model-free Methods

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

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

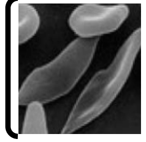
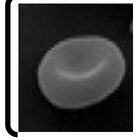
Bayes optimal rule

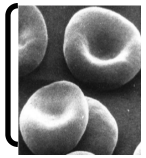
We could use a model for P_{XY} !

But can we estimate the knowledge through some learning algorithm that does not go through a model?

A model-free approach for ML

Model-free Methods

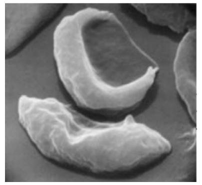
(, Anemic cell) (, Healthy cell)

(, Healthy cell) (, Anemic cell)

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

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

$\widehat{f}_n$ 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:

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

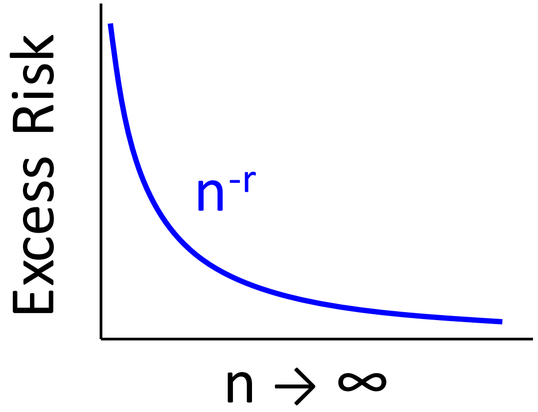
$$\boxed{\frac{1}{n} \sum_{i=1}^n [\text{loss}(Y_i, f(X_i))] \xrightarrow[\text{Numbers}]{\text{Law of Large}} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]}$$

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 …