# CS 1571 Introduction to AI
# Lecture 23

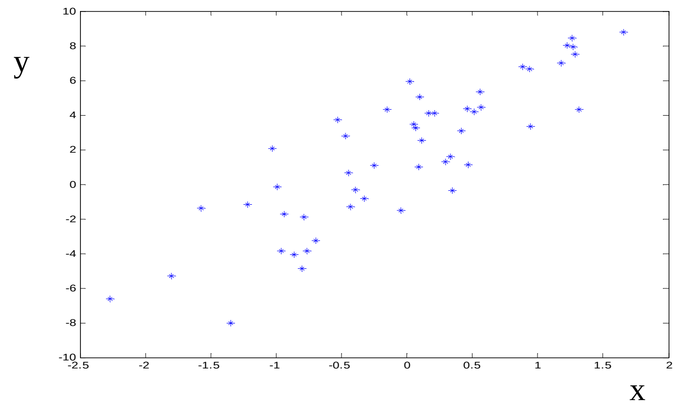
# Machine learning

**Milos Hauskrecht**
milos@cs.pitt.edu
5329 Sennott Square



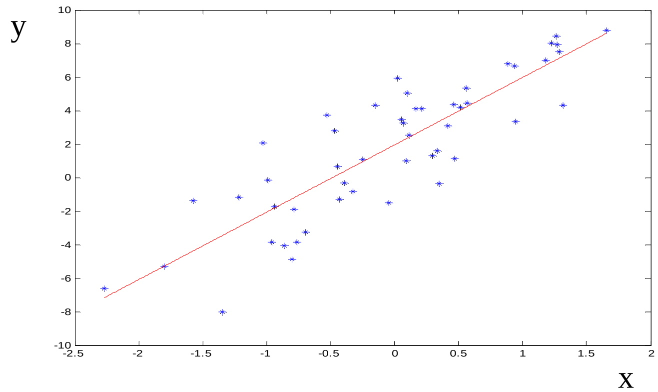
# Learning

- Assume we see examples of pairs (**x** , *y*) and we want to learn the mapping $f : X \to Y$ to predict future ys for values of **x**
- We get the data what should we do?



## Learning bias

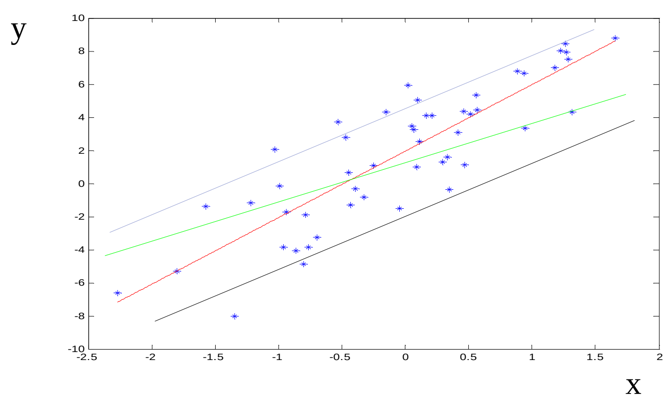
- Problem is easier when we make an assumption about the model, say, $f(x) = ax + b$
- Restriction to a linear model narrows down the possibilities





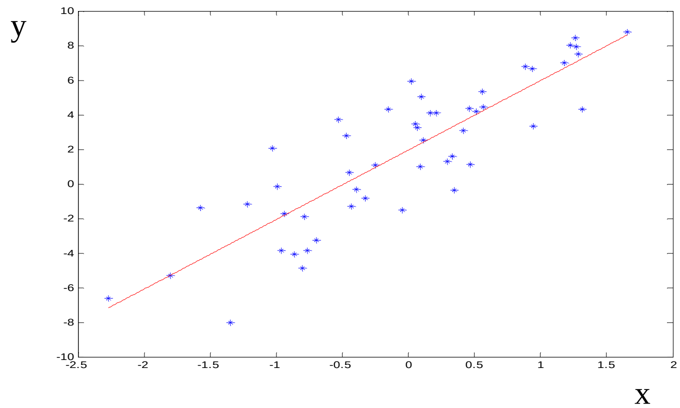

## Learning bias

- Choosing a parametric model $f(x) = ax + b$
- Many possible functions: One for every pair of parameters *a*, *b*



## Fitting the data to the model

- **Error function:**
  - Least squares fit with the linear model
  - minimizes $\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$






## Typical learning

**Three basic steps:**

- **Select a model** or a set of models (with parameters)

  E.g. $y = ax + b + \varepsilon \quad \varepsilon = N(0, \sigma)$
- **Select the error function** to be optimized

  E.g. $\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$
- **Find the set of parameters optimizing the error function**
  - The model and parameters with the smallest error represent the best fit of the model to the data

But there are problems one must be careful about …

# Learning

**Problem**

- We fit the model based on past experience (past examples seen)
- But ultimately we are interested in learning the mapping that performs well on the whole population of examples

**Training data:** **Data used to fit the parameters of the model**

**Training error:** $$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

**True (generalization) error** (over the whole unknown population):

$$E_{(x,y)}(y - f(x))^2$$ Expected squared error

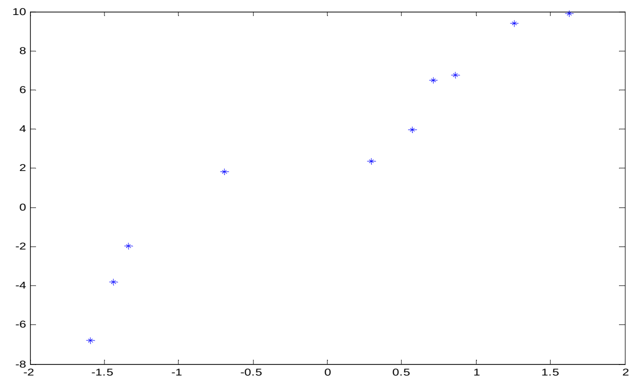
**Training error tries to approximate the true error !!!!**

Does a good training error imply a good generalization error ?



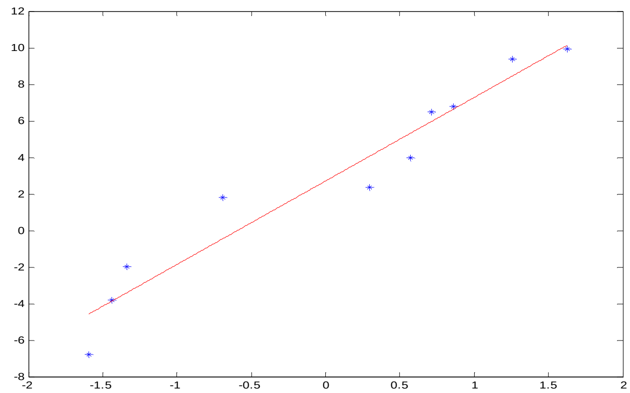
# Overfitting

- Assume we have a set of 10 points and we consider polynomial functions as our possible models



# Overfitting

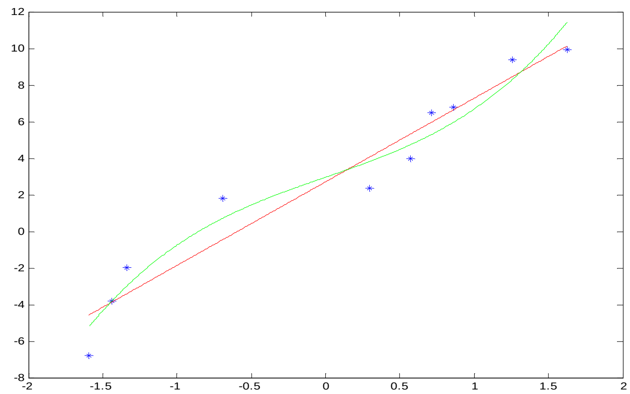
- Fitting a linear function with mean-squares error
- Error is nonzero



CS 1571 Intro to AI

# Overfitting

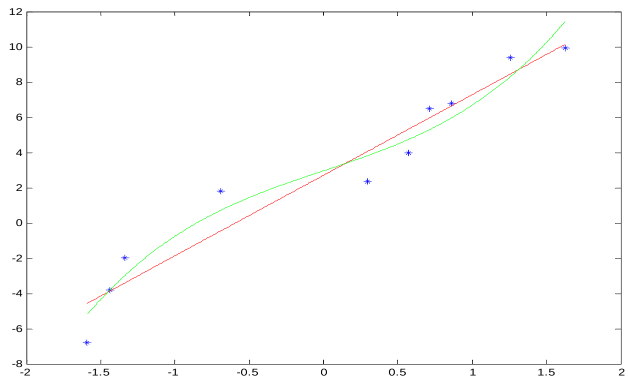
- Linear vs. cubic polynomial
- Higher order polynomial leads to a better fit, smaller error



CS 1571 Intro to AI

# Overfitting

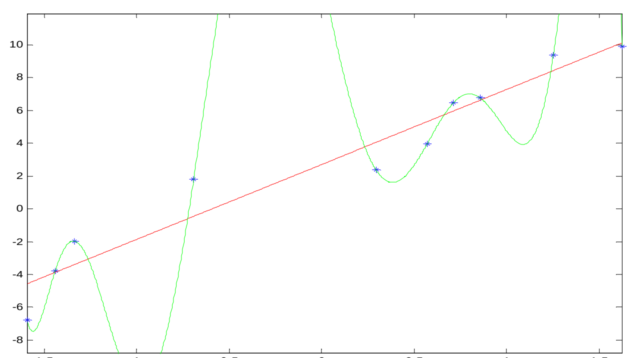
- Is it always good to minimize the error of the observed data?





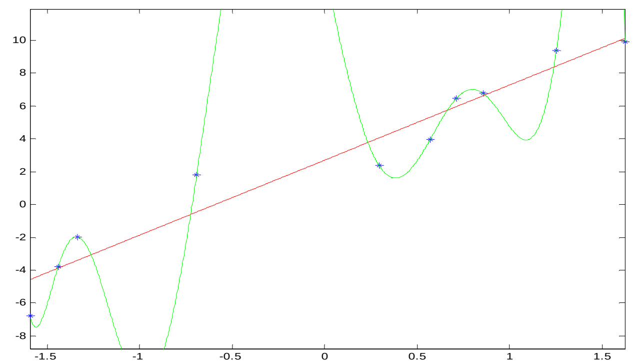

# Overfitting

- For 10 data points, degree 9 polynomial gives a perfect fit (Lagrange interpolation). Error is zero.
- Is it always good to minimize the training error?



## Overfitting

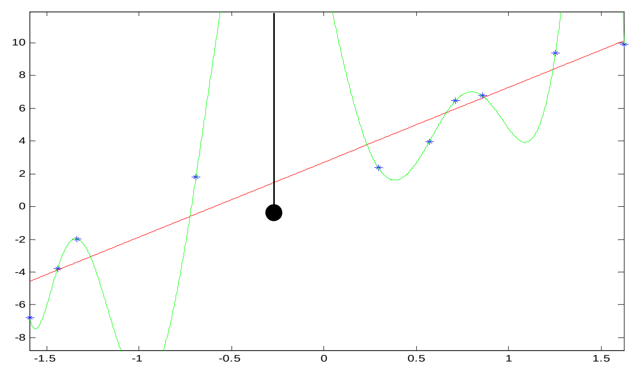
- For 10 data points, degree 9 polynomial gives a perfect fit (Lagrange interpolation). Error is zero.
- Is it always good to minimize the training error? NO !!
- More important: How do we perform on the unseen data?






## Overfitting

- Situation when the training error is low and the generalization error is high. Causes of the phenomenon:
  - Model with more degrees of freedom (more parameters)
  - Small data size (as compared to the complexity of model)



## How to evaluate the learner's performance?

- **Generalization error** is the true error for the population of examples we would like to optimize

$$E_{(x,y)}(y - f(x))^2$$

- **But it cannot be computed exactly**
- **Optimizing (mean) training error can lead to overfit, i.e.** training error may not reflect properly the generalization error

$$\frac{1}{n}\sum_{i=1,..n}(y_i - f(x_i))^2$$

- So how to test the generalization error?



## How to assess the learner's performance?

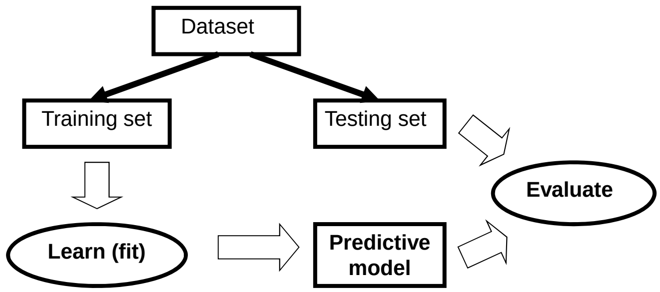
- **Generalization error** is the true error for the population of examples we would like to optimize

$$E_{(x,y)}[(y - f(x))^2]$$

- **Sample mean only approximates it**
- How to measure the generalization error?
- **Two ways:**
  - **Theoretical: Law of Large numbers**
    - statistical bounds on the difference between the true and sample mean errors
  - **Practical:** Use a separate data set with *m* data samples to test
    - **(Mean) test error** $\frac{1}{m}\sum_{j=1,..m}(y_j - f(x_j))^2$

## Testing of learning models

- **Simple holdout method**
  - Divide the data to the training and test data




  - Typically 2/3 training and 1/3 testing

## Basic experimental setup to test the learner's performance

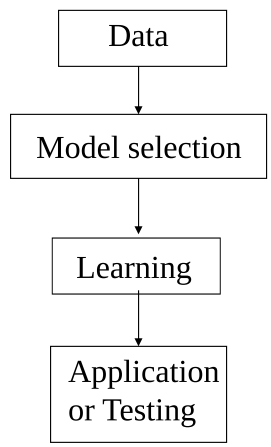
**1. Take a dataset D and divide it into:**
- **Training data set**
- **Testing data set**

**2. Use the training set and your favorite ML algorithm to train the learner**

**3. Test (evaluate) the learner on the testing data set**

- The results on the testing set can be used to compare different learners powered with different models and learning algorithms

## Design of a learning system (first view)






## Design of a learning system.

**1. Data:** $D = \{d_1, d_2, .., d_n\}$

**2. Model selection:**

- **Select a model** or a set of models (with parameters)

  E.g. $y = ax + b$

- **Select the error function** to be optimized

  E.g. $\frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$

**3. Learning:**

- **Find the set of parameters optimizing the error function**
  - The model and parameters with the smallest error

**4. Application:**

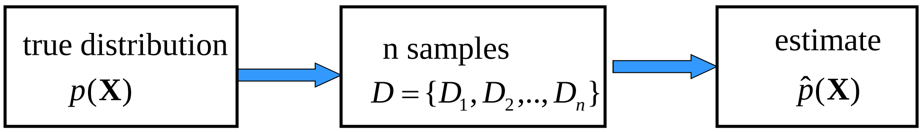
- **Apply the learned model**
  - E.g. predict ys for new inputs **x** using learned $f(\mathbf{x})$

# Density estimation

**Data:** $D = \{D_1, D_2, .., D_n\}$

$D_i = \mathbf{x}_i$ a vector of attribute values

**Objective:** try to estimate the underlying true probability distribution over variables $\mathbf{X}$ , $p(\mathbf{X})$ , using examples in $D$




**Standard (iid) assumptions: Samples**

- **are independent of each other**
- **come from the same (identical) distribution (fixed $p(\mathbf{X})$)**



# Learning via parameter estimation

In this lecture we consider **parametric density estimation**

**Basic settings:**

- A set of random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_d\}$
- **A model of the distribution** over variables in ***X*** with parameters $\Theta$
- **Data** $D = \{D_1, D_2, .., D_n\}$

**Objective:** find parameters $\hat{\Theta}$ that fit the data the best

- What is the best set of parameters?
  – There are various criteria one can apply here.

## Parameter estimation. Basic criteria.

- **Maximum likelihood (ML)**

  maximize $p(D \mid \Theta, \xi)$

  $\xi$ - represents prior (background) knowledge

- **Maximum a posteriori probability (MAP)**

  maximize $p(\Theta \mid D, \xi)$

  **Selects the mode of the posterior**

$$p(\Theta \mid D, \xi) = \frac{p(D \mid \Theta, \xi)\, p(\Theta \mid \xi)}{p(D \mid \xi)}$$



## Parameter estimation. Coin example.

**Coin example:** we have a coin that can be biased

**Outcomes:** two possible values -- head or tail

**Data:** $D$ a sequence of outcomes $x_i$ such that

- **head** $x_i = 1$
- **tail** $x_i = 0$

**Model:** probability of a head $\theta$

probability of a tail $(1-\theta)$

**Objective:**

We would like to estimate the probability of a **head** $\hat{\theta}$ from data

## Parameter estimation. Example.

- **Assume** the unknown and possibly biased coin
- Probability of the head is $\theta$
- **Data:**

  H H T T H H T H T H T T T H T H H H H T H H H H T
  - **Heads:** 15
  - **Tails:** 10

What would be your estimate of the probability of a head ?

$$\tilde{\theta} = ?$$



## Parameter estimation. Example

- **Assume** the unknown and possibly biased coin
- Probability of the head is $\theta$
- **Data:**

  H H T T H H T H T H T T T H T H H H H T H H H H T
  - **Heads:** 15
  - **Tails:** 10

What would be your choice of the probability of a head ?

**Solution:** use frequencies of occurrences to do the estimate

$$\tilde{\theta} = \frac{15}{25} = 0.6$$

This is **the maximum likelihood estimate** of the parameter $\theta$

## Probability of an outcome

**Data:** $D$ a sequence of outcomes $x_i$ such that

- **head** $x_i = 1$
- **tail** $x_i = 0$

**Model:** probability of a head $\theta$
probability of a tail $(1-\theta)$

**Assume:** **we know the probability** $\theta$
**Probability of an outcome of a coin flip** $x_i$

$$P(x_i \mid \theta) = \theta^{x_i}(1-\theta)^{(1-x_i)}$$ ← **Bernoulli distribution**

- Combines the probability of a head and a tail
- So that $x_i$ is going to pick its correct probability
- Gives $\theta$ for $x_i = 1$
- Gives $(1-\theta)$ for $x_i = 0$



## Probability of a sequence of outcomes.

**Data:** $D$ a sequence of outcomes $x_i$ such that

- **head** $x_i = 1$
- **tail** $x_i = 0$

**Model:** probability of a head $\theta$
probability of a tail $(1-\theta)$

**Assume:** **a sequence of independent coin flips**

**D = H H T H T H** **(encoded as D= 110101)**

What is the probability of observing the data sequence **D:**

$$P(D \mid \theta) = ?$$

## Probability of a sequence of outcomes.

**Data:** $D$ a sequence of outcomes $x_i$ such that

- **head** $x_i = 1$
- **tail** $x_i = 0$

**Model:** probability of a head $\theta$
probability of a tail $(1-\theta)$

**Assume: a sequence of coin flips D = H H T H T H**
**encoded as D= 110101**

What is the probability of observing a data sequence **D:**

$$P(D \mid \theta) = \theta\theta(1-\theta)\theta(1-\theta)\theta$$



## Probability of a sequence of outcomes.

**Data:** $D$ a sequence of outcomes $x_i$ such that

- **head** $x_i = 1$
- **tail** $x_i = 0$

**Model:** probability of a head $\theta$
probability of a tail $(1-\theta)$

**Assume: a sequence of coin flips D = H H T H T H**
**encoded as D= 110101**

What is the probability of observing a data sequence **D:**

$$P(D \mid \theta) = \theta\theta(1-\theta)\theta(1-\theta)\theta$$

**likelihood of the data**

## Probability of a sequence of outcomes.

**Data:** $D$ a sequence of outcomes $x_i$ such that

- **head** $x_i = 1$
- **tail** $x_i = 0$

**Model:** probability of a head $\theta$
probability of a tail $(1-\theta)$

**Assume: a sequence of coin flips D = H H T H T H**
**encoded as D= 110101**

What is the probability of observing a data sequence **D:**

$$P(D \mid \theta) = \theta\theta(1-\theta)\theta(1-\theta)\theta$$

$$P(D \mid \theta) = \prod_{i=1}^{6} \theta^{x_i}(1-\theta)^{(1-x_i)}$$

Can be rewritten using the Bernoulli distribution:



## The goodness of fit to the data.

**Learning: we do not know the value of the parameter** $\theta$

**Our learning goal**:

- Find the parameter $\theta$ that fits the data D the best?

**One solution to the "best":** Maximize the likelihood

$$P(D \mid \theta) = \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{(1-x_i)}$$

**Intuition:**

- more likely are the data given the model, the better is the fit

**Note:** Instead of an error function that measures how bad the data fit the model we have a measure that tells us how well the data fit :

$$Error(D,\theta) = -P(D \mid \theta)$$

## Maximum likelihood (ML) estimate.

**Likelihood of data:**

$$P(D \mid \theta, \xi) = \prod_{i=1}^{n} \theta^{x_i} (1-\theta)^{(1-x_i)}$$

**Maximum likelihood** estimate

$$\theta_{ML} = \arg\max_{\theta} P(D \mid \theta, \xi)$$

**Optimize log-likelihood (the same as maximizing likelihood)**

$$l(D,\theta) = \log P(D \mid \theta, \xi) = \log \prod_{i=1}^{n} \theta^{x_i} (1-\theta)^{(1-x_i)} =$$

$$\sum_{i=1}^{n} x_i \log\theta + (1-x_i)\log(1-\theta) = \log\theta \sum_{i=1}^{n} x_i + \log(1-\theta)\sum_{i=1}^{n}(1-x_i)$$

$N_1$ - number of heads seen   $N_2$ - number of tails seen



## Maximum likelihood (ML) estimate.

**Optimize log-likelihood**

$$l(D,\theta) = N_1 \log\theta + N_2 \log(1-\theta)$$

**Set derivative to zero**

$$\frac{\partial l(D,\theta)}{\partial \theta} = \frac{N_1}{\theta} - \frac{N_2}{(1-\theta)} = 0$$

**Solving** $$\theta = \frac{N_1}{N_1 + N_2}$$

**ML Solution:** $$\theta_{ML} = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2}$$

## Maximum likelihood estimate. Example

- **Assume** the unknown and possibly biased coin
- Probability of the head is $\theta$
- **Data:**

  H H T T H H T H T H T T T H T H H H H T H H H H T
  - **Heads:** 15
  - **Tails:** 10

What is the ML estimate of the probability of a head and a tail?



## Maximum likelihood estimate. Example

- Assume the unknown and possibly biased coin
- Probability of the head is $\theta$
- **Data:**

  H H T T H H T H T H T T T H T H H H H T H H H H T
  - **Heads:** 15
  - **Tails:** 10

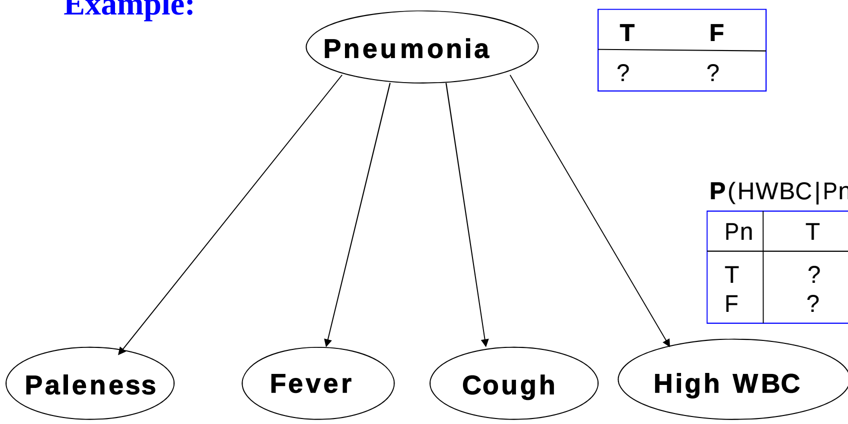
What is the ML estimate of the probability of head and tail ?

**Head:** $$\theta_{ML} = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2} = \frac{15}{25} = 0.6$$

**Tail:** $$(1 - \theta_{ML}) = \frac{N_2}{N} = \frac{N_2}{N_1 + N_2} = \frac{10}{25} = 0.4$$

# Learning of BBN parameters. Example.

**Example:**




**P**(Pneumonia)

| T | F |
|---|---|
| ? | ? |

**P**(HWBC|Pneum)

| Pn | T | F |
|---|---|---|
| T | ? | ? |
| F | ? | ? |

**P**(Palen|Pneum)

?

**P**(Fever|Pneum)

?

**P**(Cough|Pneum)

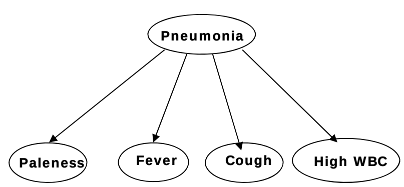
?



# Learning of BBN parameters. Example.

**Data D (different patient cases):**

| Pal | Fev | Cou | HWB | Pneu |
|---|---|---|---|---|
| T | T | T | T | F |
| T | F | F | F | F |
| F | F | T | T | T |
| F | F | T | F | T |
| F | T | T | T | T |
| T | F | T | F | F |
| F | F | F | F | F |
| T | T | F | F | F |
| T | T | T | T | T |
| F | T | F | T | T |
| T | F | F | T | F |
| F | T | F | F | F |



## Estimates of parameters of BBN

- Much like multiple **coin tosses**
- A "smaller" learning problem corresponds to the learning of exactly one conditional distribution
- **Example:**

$$\mathbf{P}(Fever \mid Pneumonia = T)$$

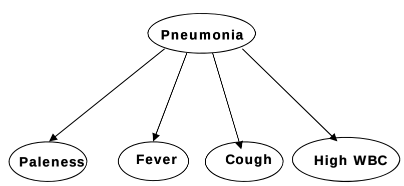
- **Problem:** How to pick the data to learn?



## Learning of BBN parameters. Example.

**Data D (different patient cases):**

| Pal | Fev | Cou | HWB | Pneu |
|---|---|---|---|---|
| T | T | T | T | F |
| T | F | F | F | F |
| F | F | T | T | T |
| F | F | T | F | T |
| F | T | T | T | T |
| T | F | T | F | F |
| F | F | F | F | F |
| T | T | F | F | F |
| T | T | T | T | T |
| F | T | F | T | T |
| T | F | F | T | F |
| F | T | F | F | F |




**How to estimate:**

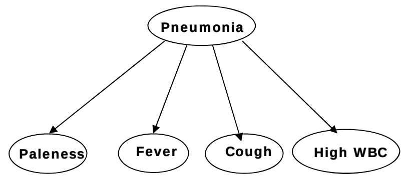
$$\mathbf{P}(Fever \mid Pneumonia = T) = ?$$

## Learning of BBN parameters. Example.

**Learn:** $\mathbf{P}(Fever \mid Pneumonia = T)$

**Step 1:** Select data points with Pneumonia=T

| Pal | Fev | Cou | HWB | Pneu |
|---|---|---|---|---|
| T | T | T | T | F |
| T | F | F | F | F |
| F | F | T | T | T |
| F | F | T | F | T |
| F | T | T | T | T |
| T | F | T | F | F |
| F | F | F | F | F |
| T | T | F | F | F |
| T | T | T | T | T |
| F | T | F | T | T |
| T | F | F | T | F |
| F | T | F | F | F |



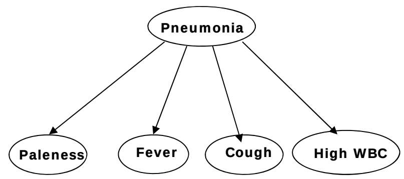



## Learning of BBN parameters. Example.

**Learn:** $\mathbf{P}(Fever \mid Pneumonia = T)$

**Step 1:** Ignore the rest

| Pal | Fev | Cou | HWB | Pneu |
|---|---|---|---|---|
| F | F | T | T | T |
| F | F | T | F | T |
| F | T | T | T | T |
| T | T | T | T | T |
| F | T | F | T | T |

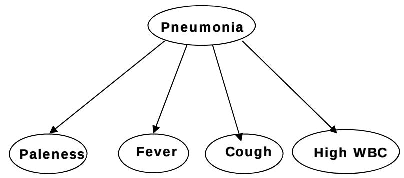


## Learning of BBN parameters. Example.

**Learn:** $\mathbf{P}(Fever \mid Pneumonia = T)$

**Step 2:** Select values of the random variable defining the distribution of Fever

| Pal | Fev | Cou | HWB | Pneu |
|---|---|---|---|---|
| F | F | T | T | T |
| F | F | T | F | T |
| F | T | T | T | T |
| T | T | T | T | T |
| F | T | F | T | T |



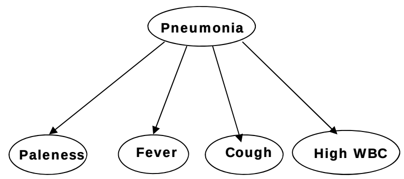



## Learning of BBN parameters. Example.

**Learn:** $\mathbf{P}(Fever \mid Pneumonia = T)$

**Step 2:** Ignore the rest

| Fev |
|---|
| F |
| F |
| T |
| T |
| T |

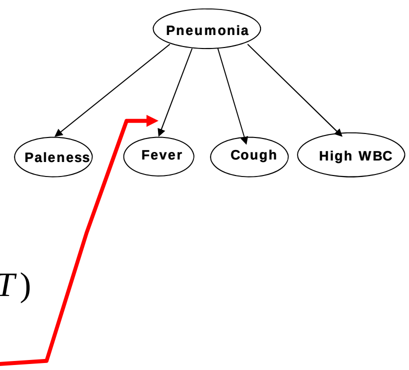


# Learning of BBN parameters. Example.

**Learn:** $\mathbf{P}(Fever \mid Pneumonia = T)$

**Step 3: Learning the ML estimate**

| Fev |
|---|
| F |
| F |
| T |
| T |
| T |




$\mathbf{P}(Fever \mid Pneumonia = T)$

| T | F |
|---|---|
| 0.6 | 0.4 |

CS 1571 Intro to AI