Introduction to Machine Learning

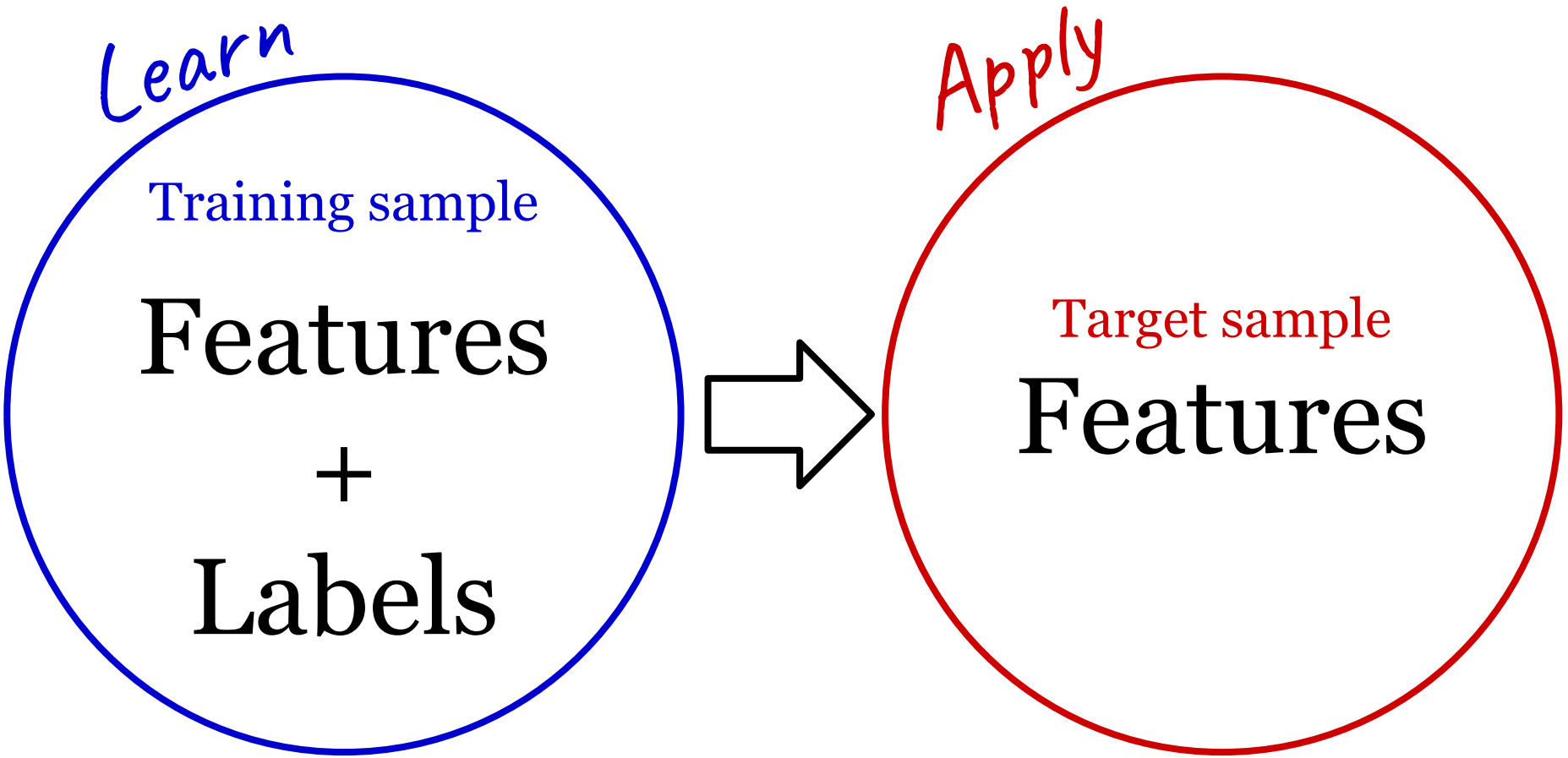
Theory II

ML@Cezeaux, 18 February 2020

Emille E. O. Ishida
Laboratoire de Physique de Clermont - Université Clermont-Auvergne
Clermont Ferrand, France

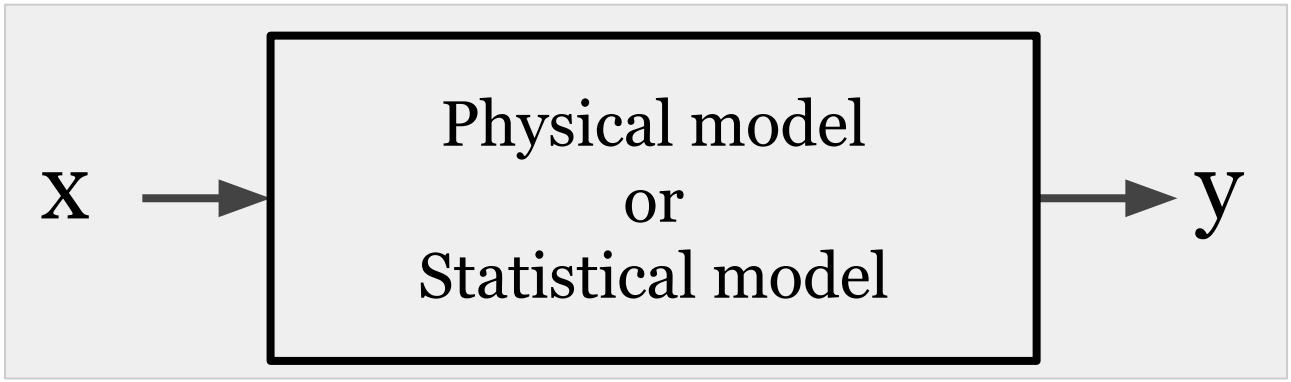
Supervised Learning

Learn by example

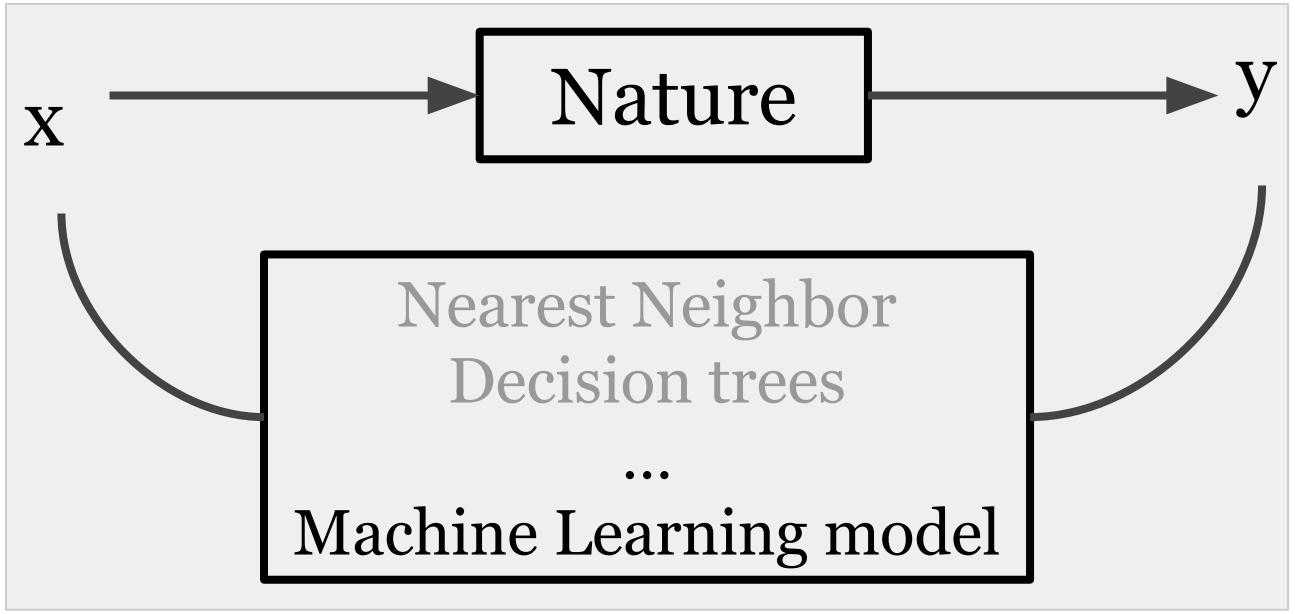


Hypothesis:

Physical modeling:



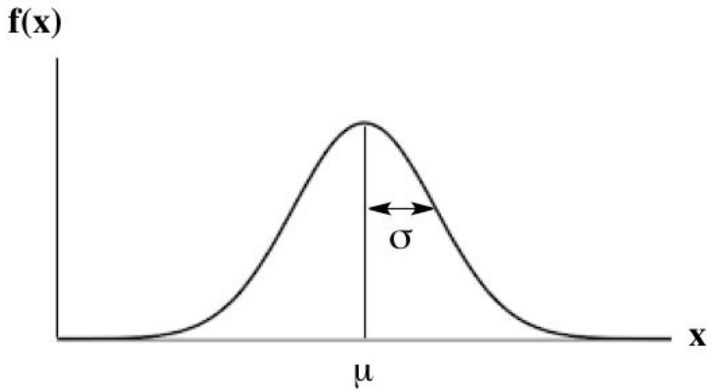
Algorithmic modeling:



Breiman, L., *Statistical Modeling: The Two Cultures*, Stat. Sci, Volume 16 (2001)

Representativeness

Probability distribution, *P*



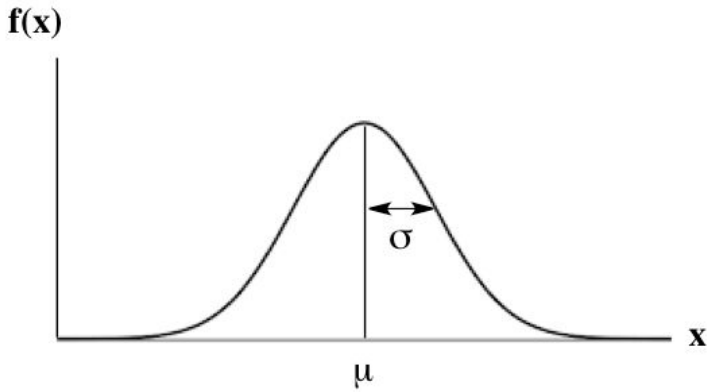
$$(\mu_P, \sigma_P)$$

Sample, S1

$$(\mu_{S_1}, \sigma_{S_1})$$

Representativeness

Probability distribution, *P*



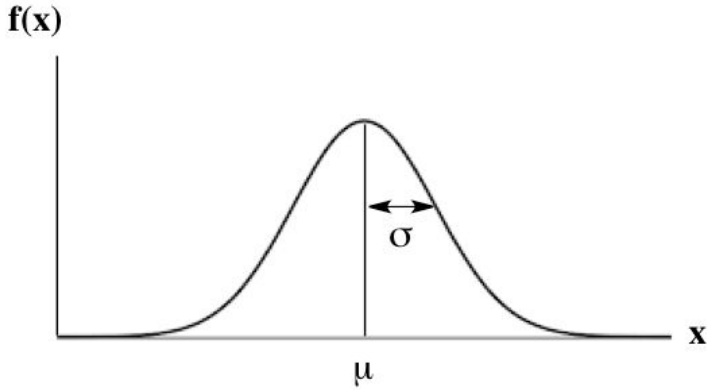
$$(\mu_P, \sigma_P)$$

Sample, S1

$$(\mu_{S_1}, \sigma_{S_1})$$

Representativeness

Probability distribution, *P*



(μ_P, σ_P)

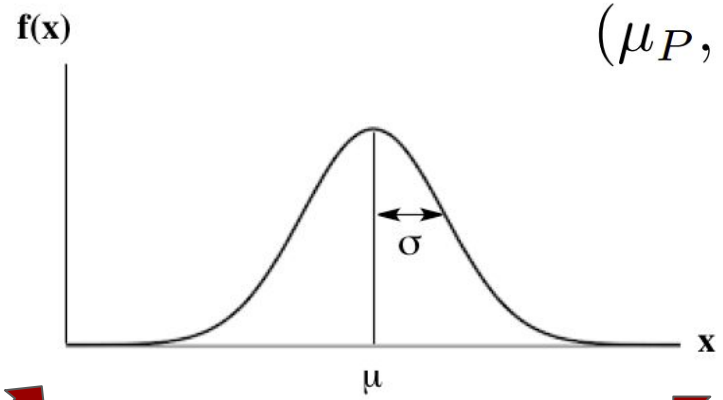
Sample, S1

$(\mu_{S_1}, \sigma_{S_1})$

S_1 is *representative* of *P*

Representativeness

Probability distribution, *P*



(μ_P, σ_P)

Sample, S_1

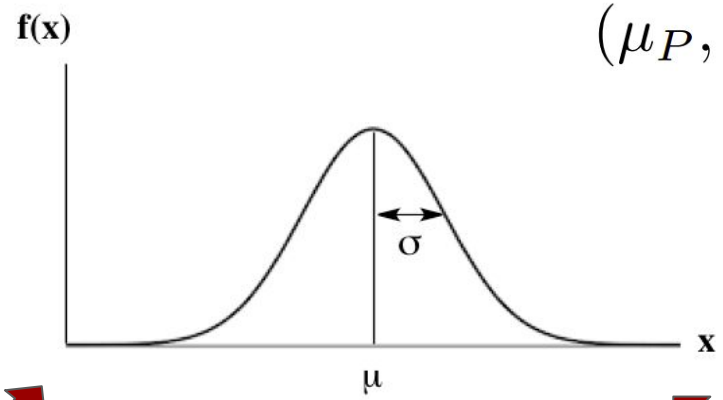
$(\mu_{S_1}, \sigma_{S_1})$

Sample, S_2

$(\mu_{S_2}, \sigma_{S_2})$

Representativeness

Probability distribution, *P*



(μ_P, σ_P)

Sample, S_1

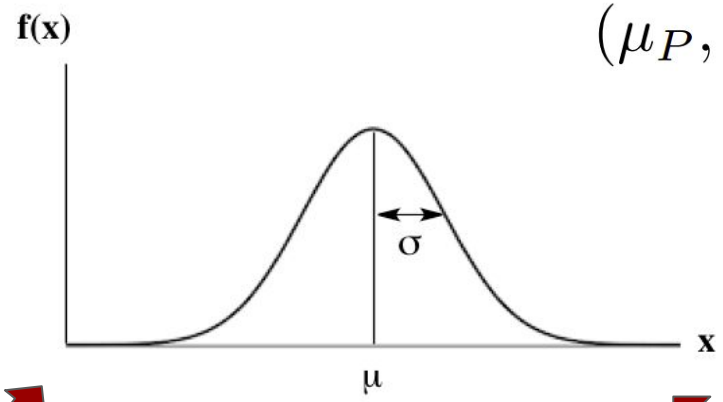
Sample, S_2

$(\mu_{S_1}, \sigma_{S_1})$

$(\mu_{S_2}, \sigma_{S_2})$

Representativeness

Probability distribution, *P*



(μ_P, σ_P)

Training

Features

+

Labels

Target

Features

+

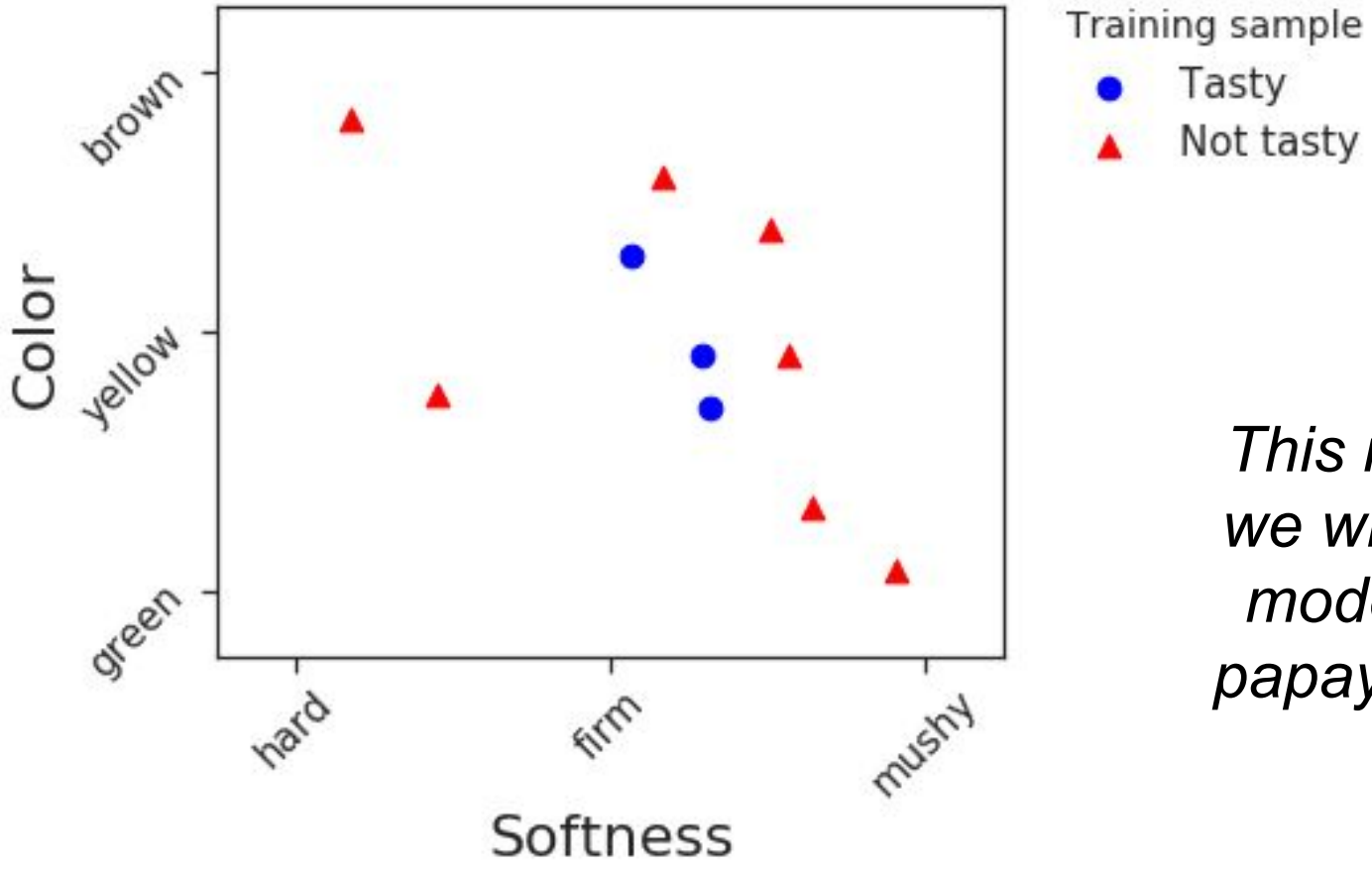
(?)

$(\mu_{S_1}, \sigma_{S_1})$

$(\mu_{S_2}, \sigma_{S_2})$

Papaya tasting

Binary classification

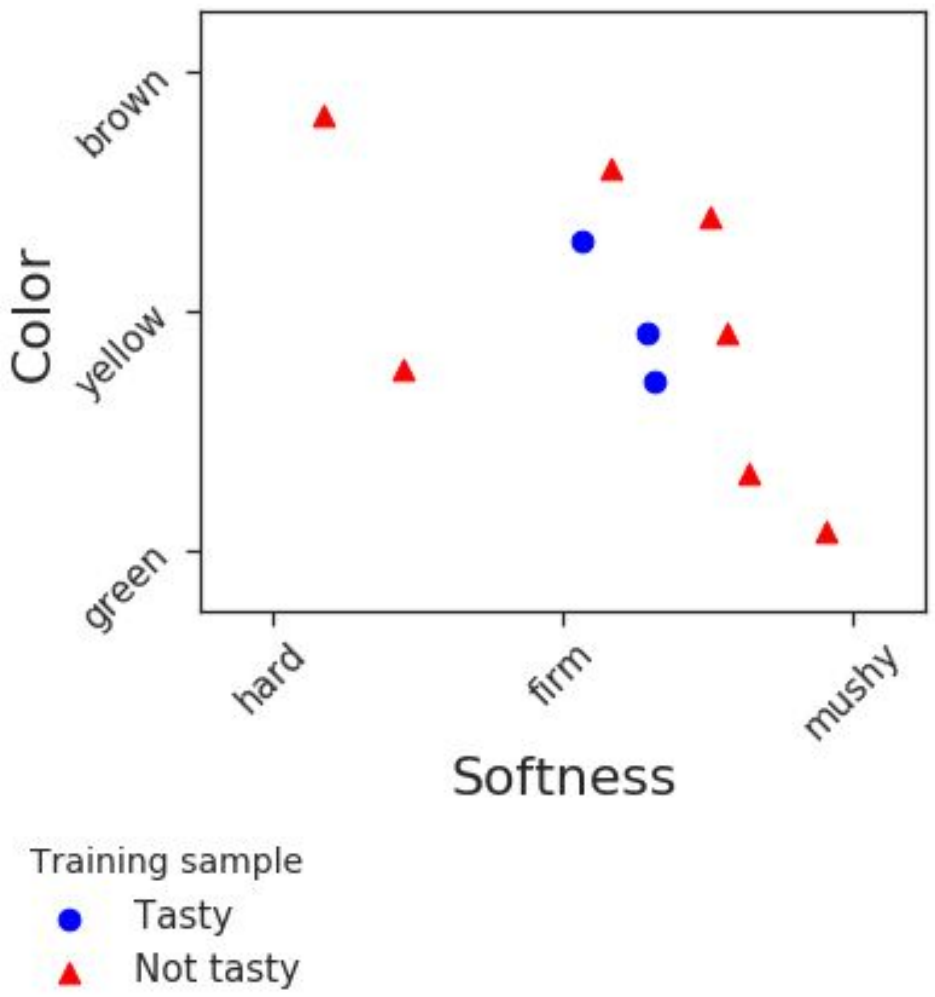


This is all the data we will input to the model about the papayas in the real world!

YouTube class on the papaya testing example:
https://www.youtube.com/watch?v=b5NlRg8SjZg&list=PLPW2keNyw-usgvmR7FTQ3ZRjfLs5jT4BO&index=2&t=0s

A controlled example:

Papaya tasting



X: set of all features,

$x = [softness, color]$

Y: set of possible labels,

$y = [tasty, not\ tasty]$

D: data generation model,

$D \Longrightarrow P(X)$

True Labelling function: $y = f(x)$

S: training sample: $[x_i, y_i]$, $i \in training$

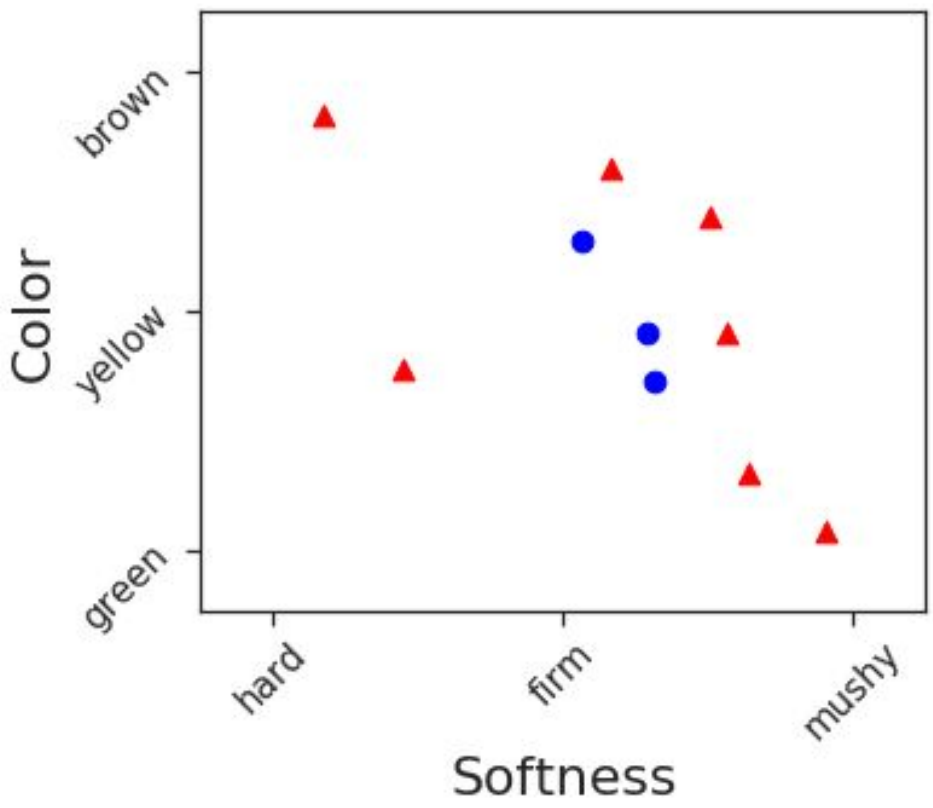
m: number of objects for training

h_S learner: $y_{est;i} = h_S(x_i)$

L metric: $L\,(y_{true;i} - y_{est;i})$, $i \in training$

A controlled example:

Papaya tasting



X: set of all features,

$x = [softness, color]$

Y: set of possible labels,

$y = [tasty, not\ tasty]$

D: data generation model,

$D \Longrightarrow P(X)$

True Labelling function: $y = f(x)$

S: training sample: $[x_i, y_i]$, $i \in training$

m: number of objects for training

h_S learner: $y_{est;i} = h_S(x_i)$

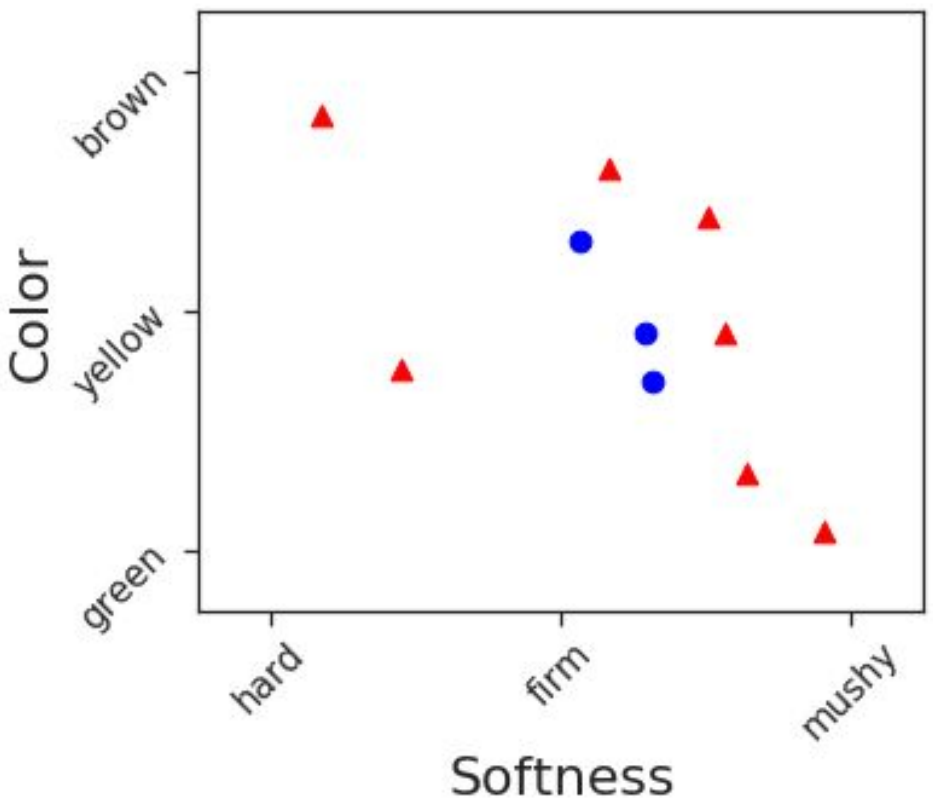
L metric: $L\,(y_{true;i} - y_{est;i})$, $i \in training$

Empirical Risk Minimization (ERM)

$L \rightarrow$ *fraction of incorrect predictions*

A controlled example:

Papaya tasting



X: set of all features,
x = [softness, color]
Y: set of possible labels,
y = [tasty, not tasty]
D: data generation model,
$D \Longrightarrow P(X)$
True Labelling function: $y = f(x)$

S: training sample: $[x_i, y_i]$, $i \in$ *training*
m: number of objects for training
h_S learner: $y_{est;i} = h_S(x_i)$
L metric: $L\,(y_{true;i} - y_{est;i})$, $i \in$ *training*

Empirical Risk Minimization (ERM)

$$L_{\mathcal{D}}(h_S) = \frac{|\{x \in \mathcal{D} : h_S(x) \neq f(x)\}|}{m}$$

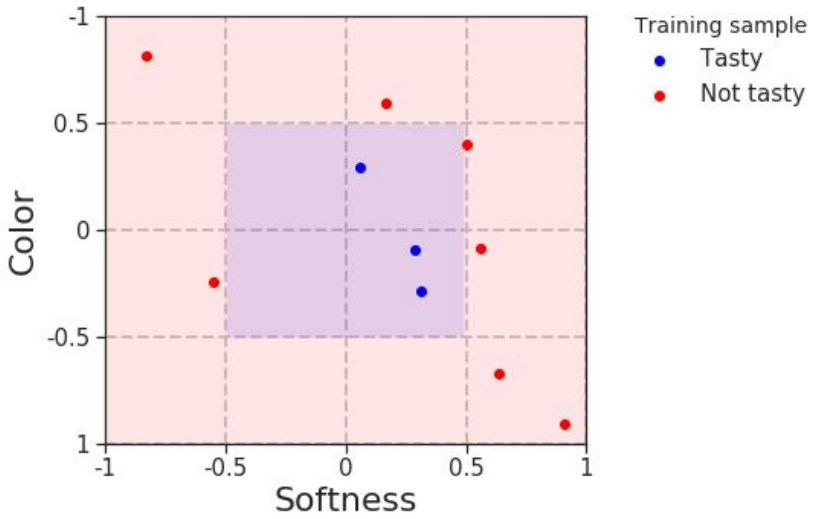
A controlled example:

Papaya tasting

Proposed learner:

$$h_S(x) = \begin{cases} y_i & \text{if } x = x_i \mid \{x_i \in S\} \\ 0 & \text{otherwise} \end{cases}$$

Toy model …



X: set of all features,
x = [softness, color]

Y: set of possible labels,
y = [tasty, not tasty]

D: data generation model,
$D \Longrightarrow P(X)$

True Labelling function: $y = f(x)$

S: training sample: $[x_i, y_i]$, $i \in$ *training*

m: number of objects for training

h_S learner: $y_{est;i} = h_S(x_i)$

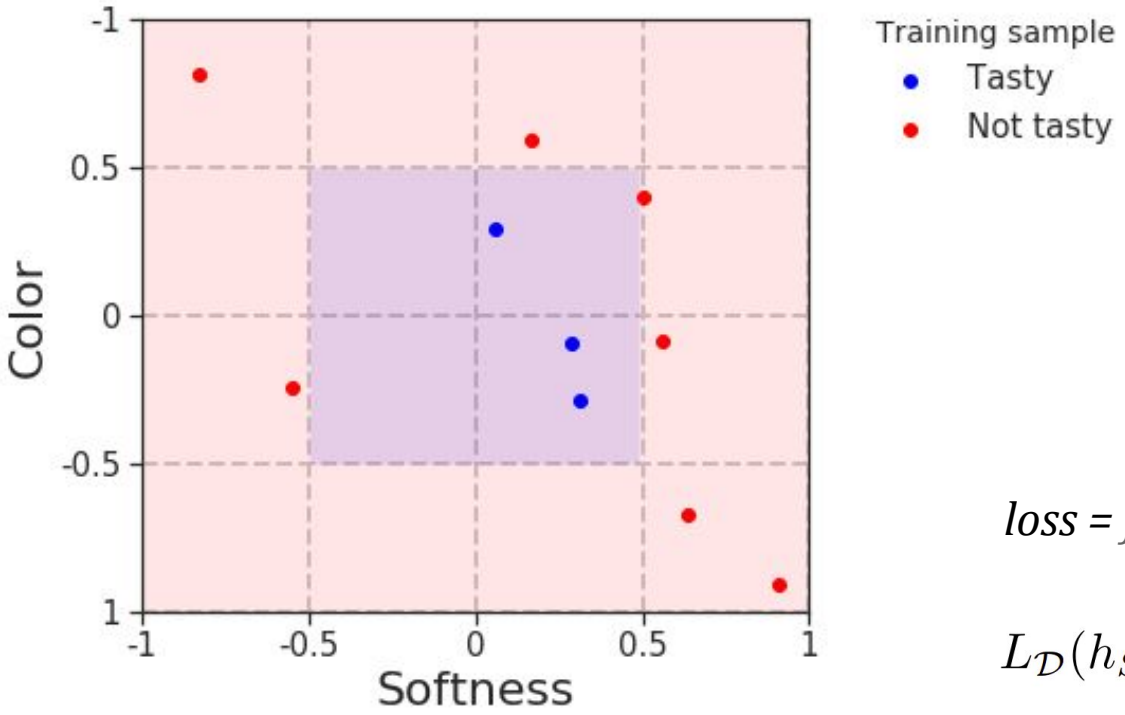
L metric: $L\,(y_{true;i} - y_{est;i})$, $i \in$ *training*

$$L_{\mathcal{D}}(h_S) = \frac{|\{x \in \mathcal{D} : h_S(x) \neq f(x)\}|}{m}$$

Question:

Proposed learner:

$$h_S(x) = \begin{cases} y_i & \text{if } x = x_i \mid \{x_i \in S\} \\ 0 & \text{otherwise} \end{cases}$$



[tasty, not tasty] = [1, 0]

What is the expected loss when this model is applied to an arbitrary test sample?

loss = fraction of incorrect predictions

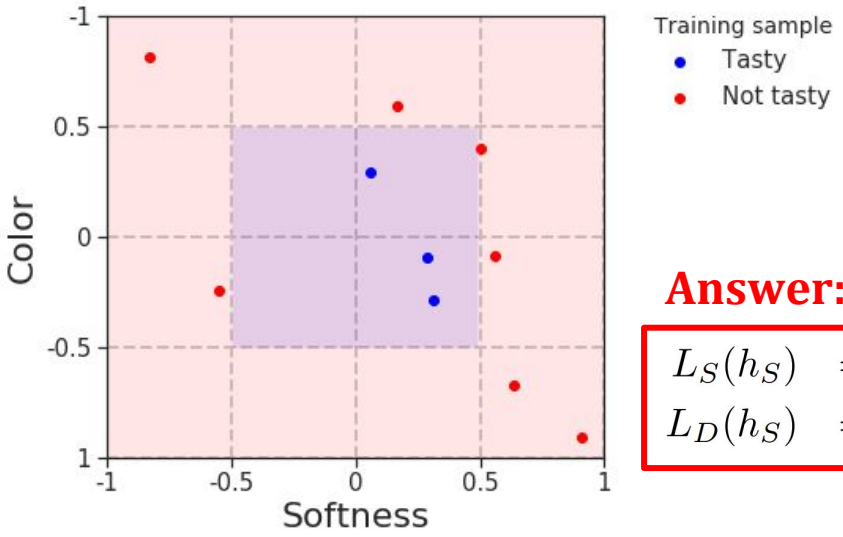
$$L_{\mathcal{D}}(h_S) = \frac{|\{x \in \mathcal{D} : h_S(x) \neq f(x)\}|}{m}$$

Papaya tasting

Proposed learner:

$$h_S(x) = \begin{cases} y_i & \text{if } x = x_i \mid \{x_i \in S\} \\ 0 & \text{otherwise} \end{cases}$$

Answer:



Answer:

$$L_S(h_S) = 0.0$$
$$L_D(h_S) = 0.25$$

X: set of all features,
x = [softness, color]
Y: set of possible labels,
y = [tasty, not tasty] = [1, 0]
D: data generation model,
$D \Longrightarrow P(X)$
True Labelling function: $y = f(x)$

S: training sample: $[x_i, y_i]$, $i \in$ *training*
m: number of objects for training
h_S learner: $y_{est;i} = h_S x_i$
L metric: $L\,(y_{true,i} - y_{est;i})$, $i \in$ *training*

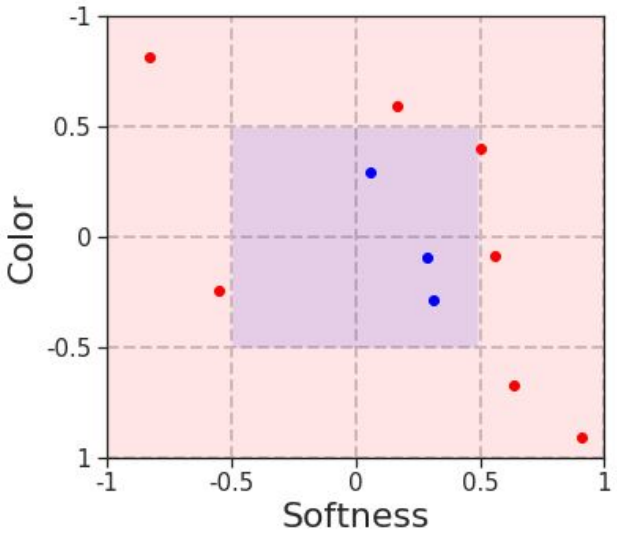
$$L_{\mathcal{D}}(h_S) = \frac{|\{x \in \mathcal{D} : h_S(x) \neq f(x)\}|}{m}$$

Papaya tasting

Proposed learner:

$$h_S(x) = \begin{cases} y_i & \text{if } x = x_i \mid \{x_i \in S\} \\ 0 & \text{otherwise} \end{cases}$$

Answer:



$$L_S(h_S) = 0.0$$
$$L_D(h_S) = 0.25$$

X: set of all features,
x = [softness, color]
Y: set of possible labels,
y = [tasty, not tasty]
D: data generation model,
$D \Longrightarrow P(X)$
True Labelling function: $y = f(x)$

S: training sample: $[x_i, y_i]$, $i \in$ *training*
m: number of objects for training
h_S learner: $y_{est;i} = h_S x_i$
L metric: $L\,(y_{true,i} - y_{est;i})$, $i \in$ *training*

$$L_{\mathcal{D}}(h_S) = \frac{|\{x \in \mathcal{D} : h_S(x) \neq f(x)\}|}{m}$$

Overfitting!

Question:

- How can we avoid overfitting?

Question:

- How can we avoid overfitting?

by adding prior knowledge ...

Choosing the learner

X: set of all features,
x = [softness, color]
Y: set of possible labels,
y = [tasty, not tasty]
D: data generation model,
$D \Longrightarrow P(X)$
True Labelling function: $y = f(x)$

S: training sample: $[x_i, y_i]$, $i \in$ *training*
m: number of objects for training
h_S learner: $y_{est;i} = h_S(x_i)$

$$h_S(x) = \begin{cases} y_i & \text{if } x = x_i \mid \{x_i \in S\} \\ 0 & \text{otherwise} \end{cases}$$

L: loss: $L\,(y_{true;i} - y_{est;i})$, $i \in$ *training*

$$L_{\mathcal{D}}(h_S) = \frac{|\{x \in \mathcal{D} : h_S(x) \neq f(x)\}|}{m}$$

Hypothesis class ($\mathcal{H}$):

$$h : \mathcal{X} \longrightarrow \mathcal{Y}; \qquad h \in \mathcal{H}$$

$$\mathrm{ERM}_{\mathcal{H}}(S) \in \operatorname*{argmin}_{h \in \mathcal{H}} L_S(h),$$

Choosing the learner

X: set of all features,
x = [softness, color]
Y: set of possible labels,
y = [tasty, not tasty]
D: data generation model,
$D \Longrightarrow P(X)$
True Labelling function: $y = f(x)$

S: training sample: $[x_i, y_i]$, $i \in$ *training*
h_S learner: $y_{est;i} = h_S(x_i)$

$$h_S(x) = \begin{cases} y_i & \text{if } x = x_i \mid \{x_i \in S\} \\ 0 & \text{otherwise} \end{cases}$$

L: loss: $L\,(y_{true;i} - y_{est;i})$, $i \in$ *training*

$$L_{\mathcal{D}}(h_S) = \frac{|\{x \in \mathcal{D} : h_S(x) \neq f(x)\}|}{m}$$

Hypothesis class ($\mathcal{H}$):

$$h : \mathcal{X} \longrightarrow \mathcal{Y}; \qquad h \in \mathcal{H}$$

$$\mathrm{ERM}_{\mathcal{H}}(S) \in \operatorname*{argmin}_{h \in \mathcal{H}} L_S(h),$$

- $\mathcal{H}$ is finite, $N_{\mathcal{H}}$ = number of hypothesis
- The true labelling function is part of $\mathcal{H}$:

$$f \in \mathcal{H}$$

ERM with inductive bias

χ: set of all features,
x = [softness, color]
Y: set of possible labels,
y = [tasty, not tasty]
D: data generation model,
$D \Longrightarrow P(\chi)$
True Labelling function: $y = f(x)$

S: training sample: $[x_i, y_i]$, $i \in$ *training*
h_S learner: $y_{est;i} = h_S(x_i)$

$$h_S(x) = \begin{cases} y_i & \text{if } x = x_i \mid \{x_i \in S\} \\ 0 & \text{otherwise} \end{cases}$$

L: loss: $L\,(y_{true;i} - y_{est;i})$, $i \in$ *training*

$$L_{\mathcal{D}}(h_S) = \frac{|\{x \in \mathcal{D} : h_S(x) \neq f(x)\}|}{m}$$

Hypothesis class ($\mathcal{H}$):

$$h : \mathcal{X} \longrightarrow \mathcal{Y}; \qquad h \in \mathcal{H}$$

$$\mathrm{ERM}_{\mathcal{H}}(S) \in \operatorname*{argmin}_{h \in \mathcal{H}} L_S(h),$$

- $\mathcal{H}$ is finite, $N_{\mathcal{H}}$ = number of hypothesis
- The true labelling function is part of $\mathcal{H}$:

$$f \in \mathcal{H}$$

- *S* is identically independently distributed (*i.i.d.*) from *D*

Representativeness

- A sample S1 is said to be representative of a probability distribution P if one can draw accurate conclusions about P from S1

- If two samples S1 and S2 are representative of P, S1 and S2 are representative in relation to each other

Representativeness

- A sample S1 is said to be representative of a probability distribution P if one can draw accurate conclusions about P from S1

- If two samples S1 and S2 are representative of P, S1 and S2 are representative in relation to each other

Question:

If a sample S_1 identically independently distributed (i.i.d.) from a distribution P, is this enough to guarantee that S_1 is representative of P?

Model assumptions

χ: set of all features,
x = [softness, color]

Y: set of possible labels,
y = [tasty, not tasty]

D: data generation model,
$D \Longrightarrow P(\chi)$

True Labelling function: $y = f(x)$

S: training sample: $[x_i, y_i]$, $i \in$ *training*

h_S learner: $y_{est;i} = h_S(x_i)$

$$h_S(x) = \begin{cases} y_i & \text{if } \exists i \in [m] \text{ s.t. } x_i = x \\ 0 & \text{otherwise.} \end{cases}$$

L: loss: $L\,(y_{true;i} - y_{est;i})$, $i \in$ *training*

$$L_{\mathcal{D}}(h_S) = \frac{|\{x \in \mathcal{D} : h_S(x) \neq f(x)\}|}{m}$$

Hypothesis class ($\mathcal{H}$):

$$h : \mathcal{X} \longrightarrow \mathcal{Y}; \qquad h \in \mathcal{H}$$

$$\mathrm{ERM}_{\mathcal{H}}(S) \in \operatorname*{argmin}_{h \in \mathcal{H}} L_S(h),$$

- $\mathcal{H}$ is finite, $N_{\mathcal{H}}$ = number of hypothesis
- The true labelling function is part of $\mathcal{H}$:

$$f \in \mathcal{H}$$

- *S* is identically independently distributed (*i.i.d.*) from *D*
- *S is statistically representative of D*

Bad hypothesis and samples

$\delta \longrightarrow$ probability of non-representative (bad) samples

$1 - \delta \longrightarrow$ confidence parameter

Bad hypothesis and samples

$\delta \longrightarrow$ probability of non-representative (bad) samples

$1 - \delta \longrightarrow$ confidence parameter

$\varepsilon \longrightarrow$ contamination. A failure will occur when $L_D(h_S) \geq \epsilon$

Good hypothesis: $\mathcal{H}_G := [h \in \mathcal{H} : L_S(h_S) = 0 \quad \& \quad L_D(h_S) < \epsilon]$

Bad hypothesis: $\mathcal{H}_B := [h \in \mathcal{H} : L_S(h_S) = 0 \quad \& \quad L_D(h_S) \geq \epsilon]$

Bad hypothesis and samples

$\delta \longrightarrow$ probability of non-representative (bad) samples

$1 - \delta \longrightarrow$ confidence parameter

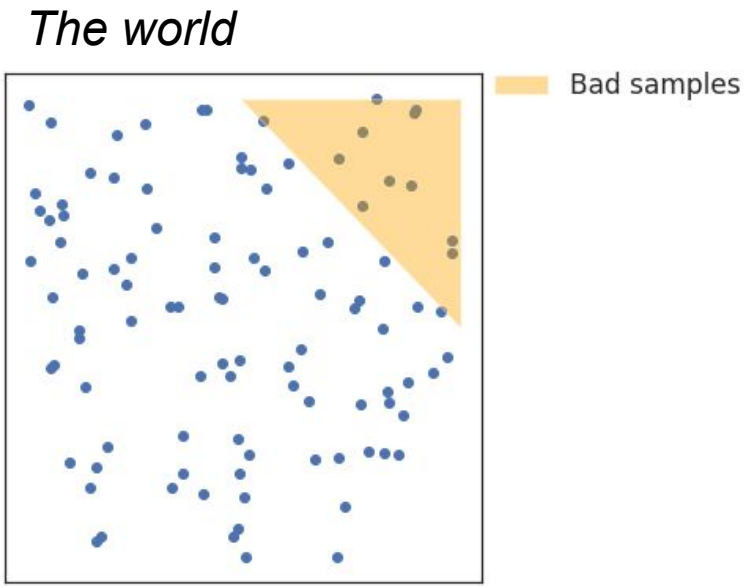
$\varepsilon \longrightarrow$ contamination. A failure will occur when $L_D(h_S) \geq \epsilon$

Good hypothesis: $\mathcal{H}_G := [h \in \mathcal{H} : L_S(h_S) = 0 \;\&\; L_D(h_S) < \epsilon]$

Bad hypothesis: $\mathcal{H}_B := [h \in \mathcal{H} : L_S(h_S) = 0 \;\&\; L_D(h_S) \geq \epsilon]$

Realizability assumption, $f \in \mathcal{H}$

Constructing misleading samples

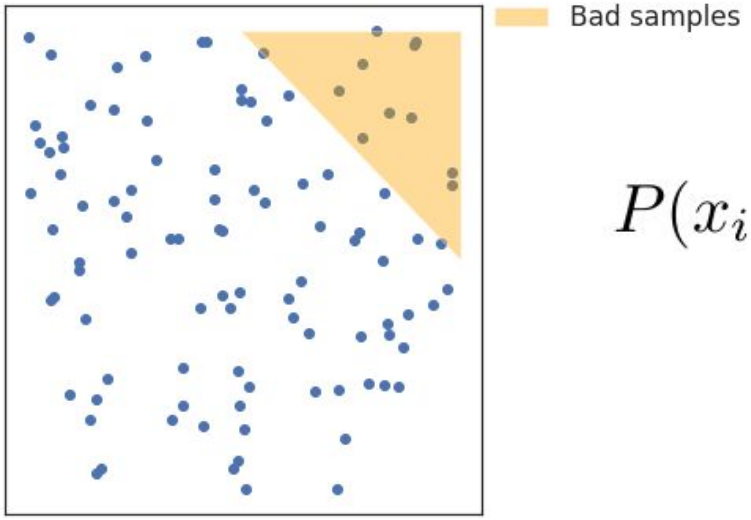


For 1 element in the training sample

$$x_i \quad | \quad h(x_i) \quad = \quad y_i$$

Constructing misleading samples

The world



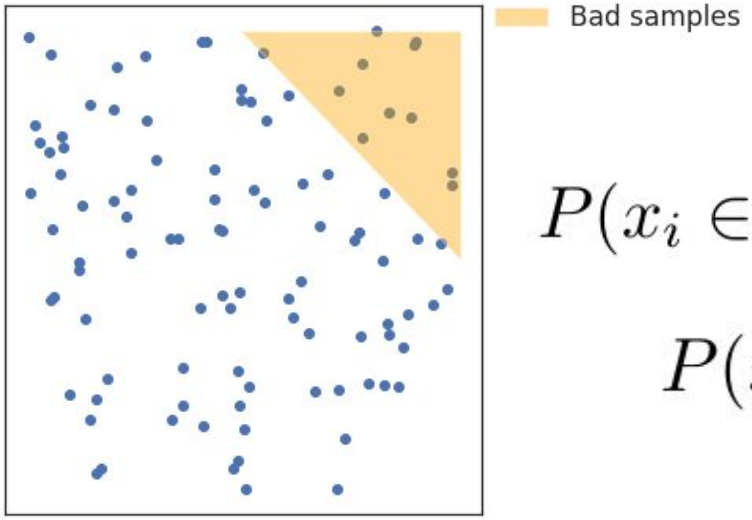
For 1 element in the training sample

$$x_i \quad | \quad h(x_i) \quad = \quad y_i$$

$$P(x_i \in \mathcal{D} : h(x_i) \quad = \quad y_i) = 1 - L_{\mathcal{D},f}(h)$$

Constructing misleading samples

The world



For 1 element in the training sample

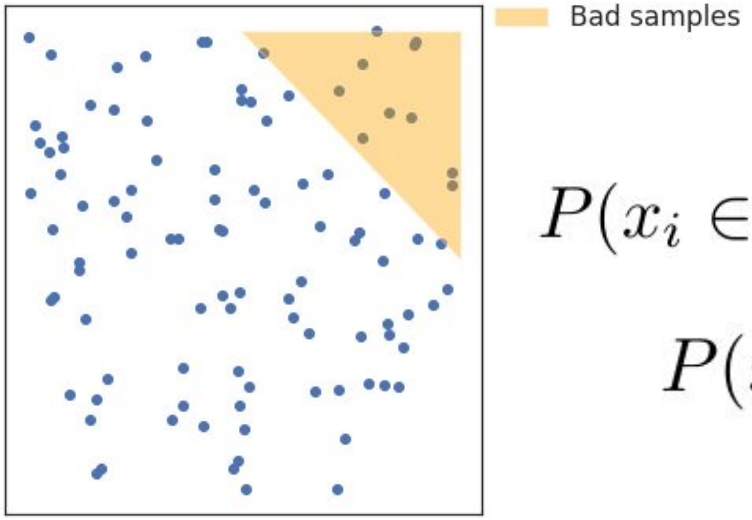
$$x_i \quad | \quad h(x_i) \quad = \quad y_i$$

$$P(x_i \in \mathcal{D} : h(x_i) \quad = \quad y_i) = 1 - L_{\mathcal{D},f}(h)$$

$$\dots$$

$$P(x_i \in \mathcal{D} : h(x_i) \quad = \quad y_i) \leq 1 - \epsilon$$

Constructing misleading samples



For 1 element in the training sample

$$x_i \quad | \quad h(x_i) \quad = \quad y_i$$

$$P(x_i \in \mathcal{D} : h(x_i) \quad = \quad y_i) = 1 - L_{\mathcal{D},f}(h)$$

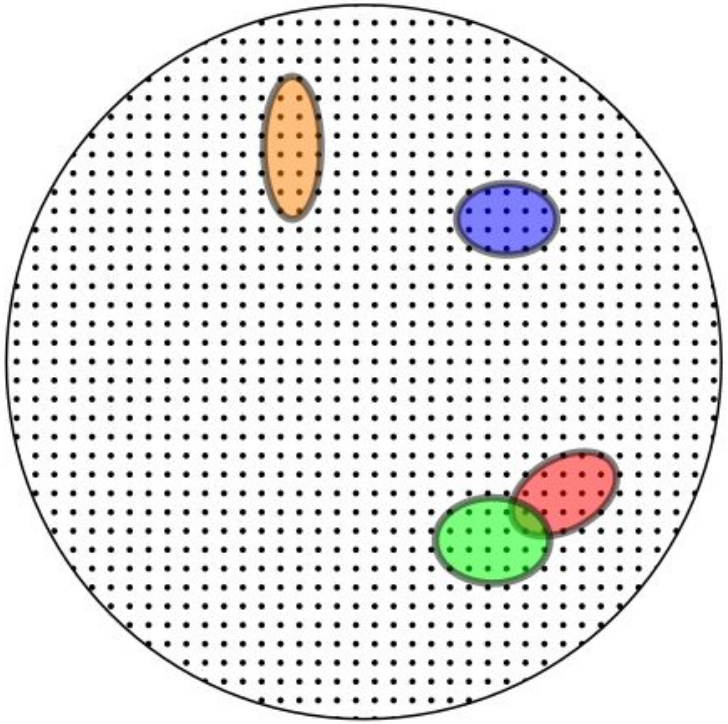
$$P(x_i \in \mathcal{D} : h(x_i) \quad = \quad y_i) \leq 1 - \epsilon$$

*For **m** elements in the training sample*

Since all elements in training are i.i.d.,

$$P(S_m : L_S(h) = 0) \quad \leq \quad \prod_{i=1}^{m} (1 - \epsilon) = (1 - \epsilon)^m$$

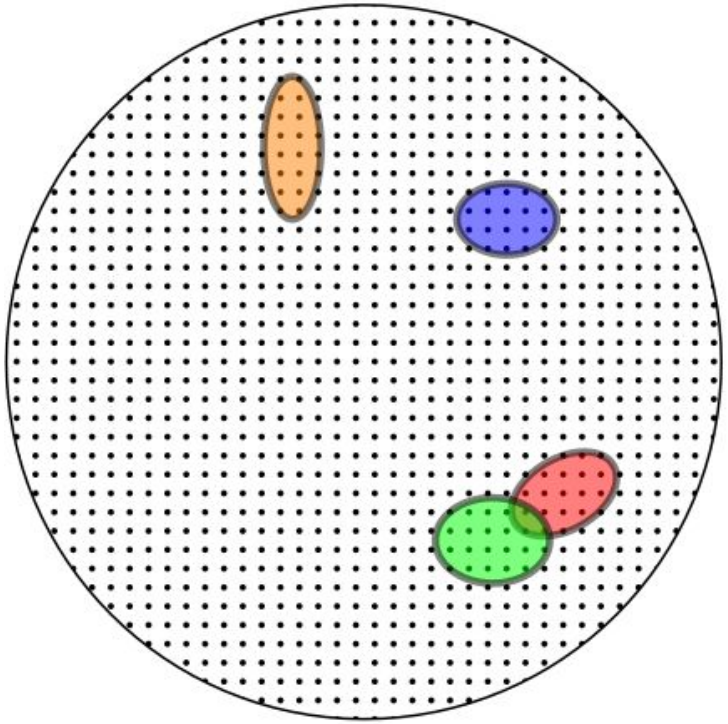
Considering bad hypothesis



For 1 hypothesis

$$P(S_m : L_S(h) = 0) \quad \leq \quad (1 - \epsilon)^m$$

Considering bad hypothesis



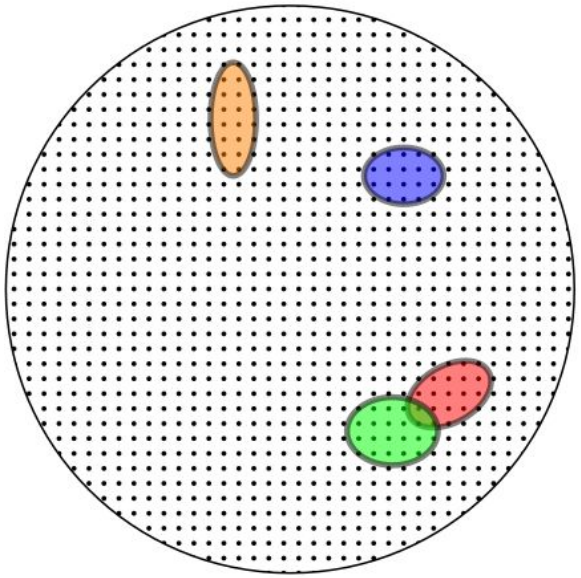
*For **1** hypothesis*

$$P(S_m : L_S(h) = 0) \quad \leq \quad (1-\epsilon)^m$$

For all bad hypothesis

$$P(A \cup B) \quad \leq \quad P(A) + P(B)$$

Considering bad hypothesis



For 1 hypothesis

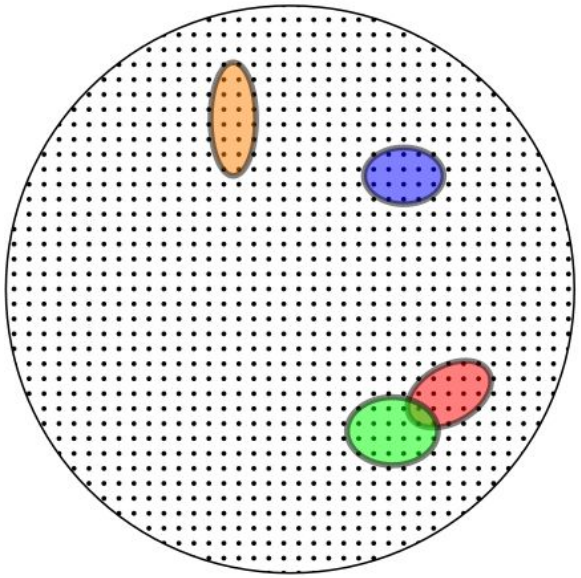
$$P(S_m : L_S(h) = 0) \quad \leq \quad (1-\epsilon)^m$$

For all bad hypothesis

$$P(A \cup B) \quad \leq \quad P(A) + P(B)$$

$$\delta = P(L_S(h) = 0, \forall h \in \mathcal{H}_B) \quad \leq \quad \sum_{h \in \mathcal{H}_B} (1-\epsilon)^m$$

Considering bad hypothesis



For 1 hypothesis

$$P(S_m : L_S(h) = 0) \quad \leq \quad (1-\epsilon)^m$$

For all bad hypothesis

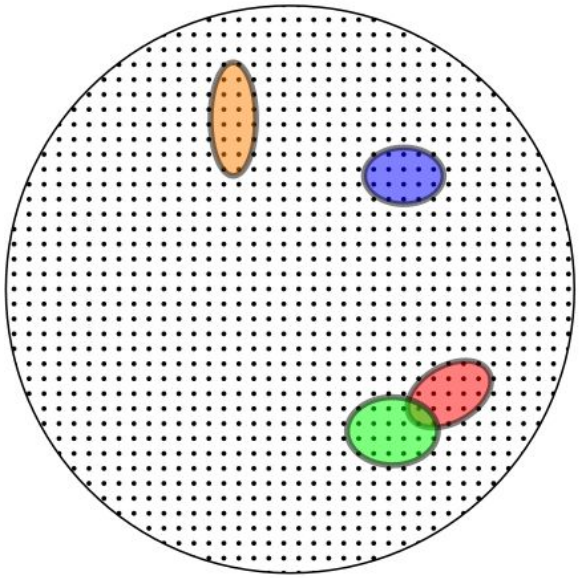
$$P(A \cup B) \quad \leq \quad P(A) + P(B)$$

$$\delta = P(L_S(h) = 0, \forall h \in \mathcal{H}_B) \quad \leq \quad \sum_{h \in \mathcal{H}_B} (1-\epsilon)^m$$

using...

$$(1-x)^y \quad \leq \quad \exp(-xy)$$

Considering bad hypothesis



For 1 hypothesis

$$P(S_m : L_S(h) = 0) \quad \leq \quad (1-\epsilon)^m$$

For all bad hypothesis

$$P(A \cup B) \quad \leq \quad P(A) + P(B)$$

$$\delta = P(L_S(h) = 0, \forall h \in \mathcal{H}_B) \quad \leq \quad \sum_{h \in \mathcal{H}_B} (1-\epsilon)^m$$

using...

$$(1-x)^y \quad \leq \quad \exp(-xy)$$

$$\delta \leq N_{\mathcal{H}} \exp(-\epsilon m)$$

PAC learning model

$$\delta \leq N_{\mathcal{H}} \exp(-\epsilon m)$$

Probably → with confidence 1 - δ over *m* samples
Approximately → within a contamination level ≤ ε
Correct

If,

$$m_{\mathcal{H}}(\epsilon, \delta) \geq \frac{\ln(N_{\mathcal{H}}/\delta)}{\epsilon}$$

every *h* from ERM,

$$L_{(\mathcal{D},f)}(h_S) \leq \epsilon.$$

L. Valiant. A theory of the learnable. Communications of the ACM, 27, 1984.

Papaya tasting

χ: set of $x \in$ *[softness, color]*

Y: set y = *[tasty, not tasty]*

D: data generation model: $D \Longrightarrow P(\chi)$

Papaya tasting

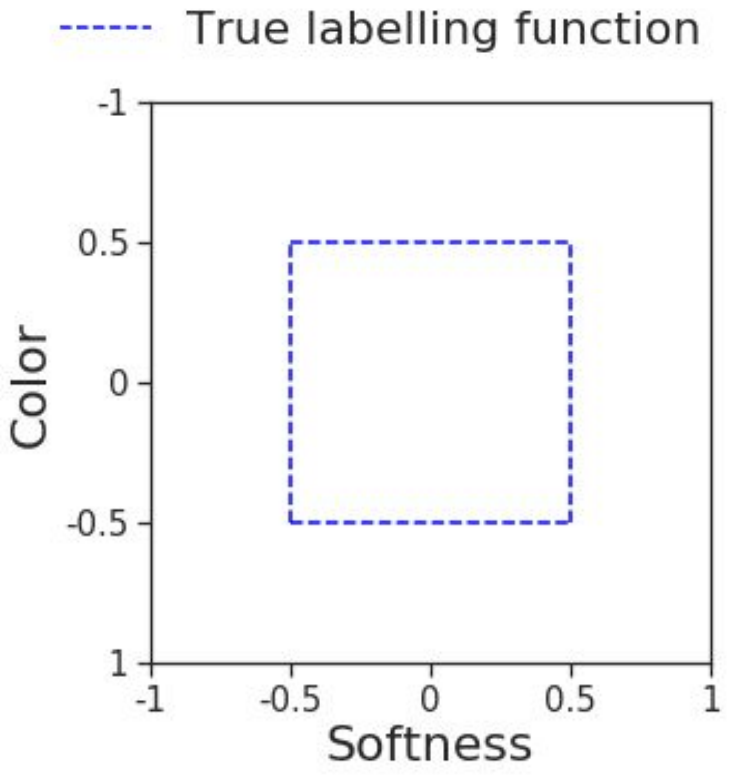
χ: set of $x \in$ *[softness, color]*

Y: set y = *[tasty, not tasty]*

D: data generation model: $D \Longrightarrow P(\chi)$

True Labelling function:

y = tasty if softness $\in$ *[-0.5, 0.5] and*
color $\in$ *[-0.5, 0.5]*



Papaya tasting

χ: set of $x \in$ *[softness, color]*

Y: set y = *[tasty, not tasty]*

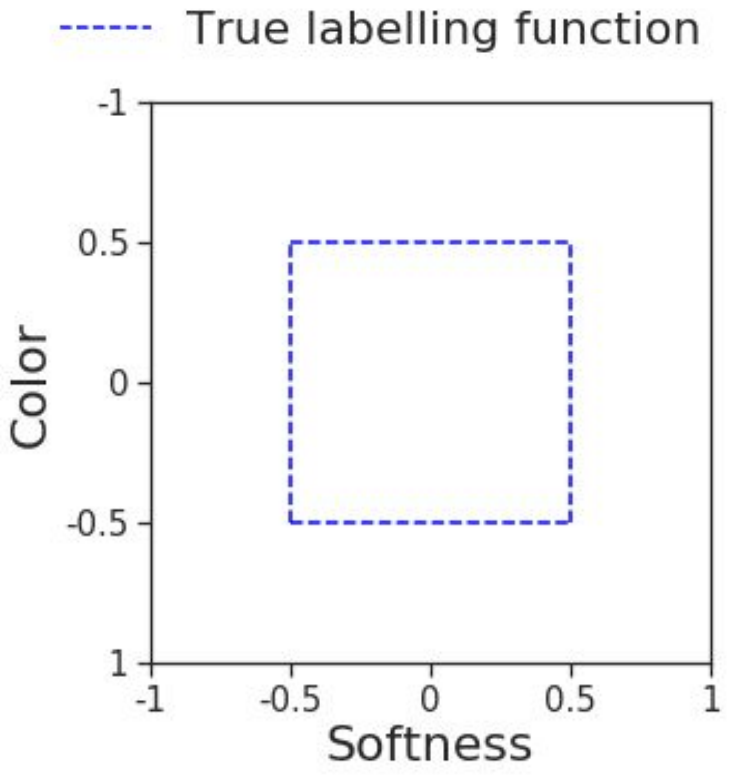
D: data generation model: $D \Longrightarrow P(\chi)$

True Labelling function:

y = tasty if softness $\in$ *[-0.5, 0.5] and*
color $\in$ *[-0.5, 0.5]*

S: training sample: $[x_i, y_i]$, $i \in$ *training*

m: number of objects for training



Papaya tasting

χ: set of $x \in$ *[softness, color]*

Y: set y = *[tasty, not tasty]*

D: data generation model: $D \Longrightarrow P(\chi)$

True Labelling function:

y = tasty if softness $\in$ *[-0.5, 0.5] and*
color $\in$ *[-0.5, 0.5]*

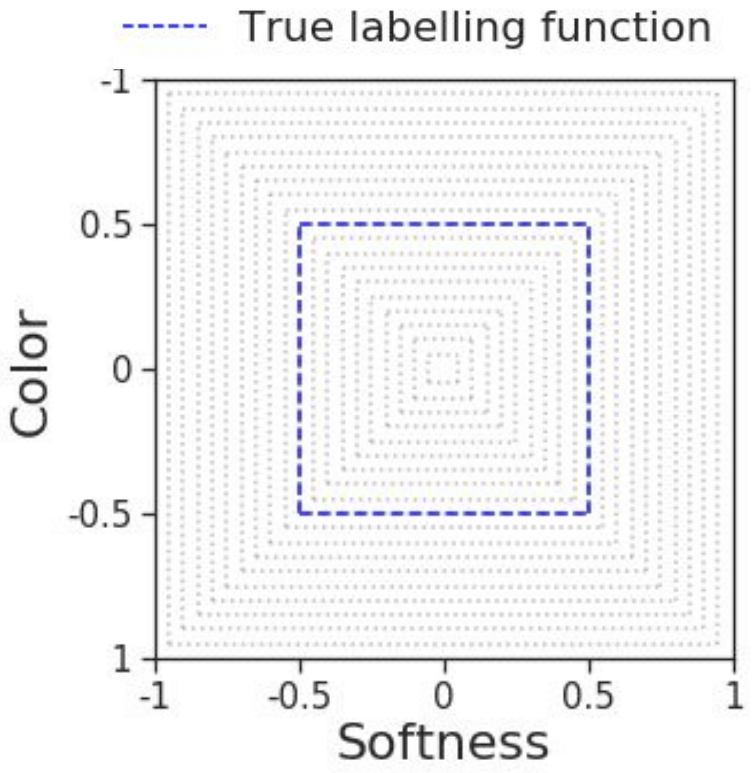
S: training sample: $[x_i, y_i]$, $i \in$ *training*

m: number of objects for training

$\mathcal{H}$: *hypothesis class:*

axis aligned squares in steps of 0.05

$N_H = 20$



Papaya tasting

χ: set of $x \in$ *[softness, color]*

Y: set y = *[tasty, not tasty]*

D: data generation model: $D \Longrightarrow P(\chi)$

True Labelling function:

y = tasty if softness $\in$ *[-0.5, 0.5] and color* $\in$ *[-0.5, 0.5]*

S: training sample: $[x_i, y_i]$, $i \in$ *training*

m: number of objects for training

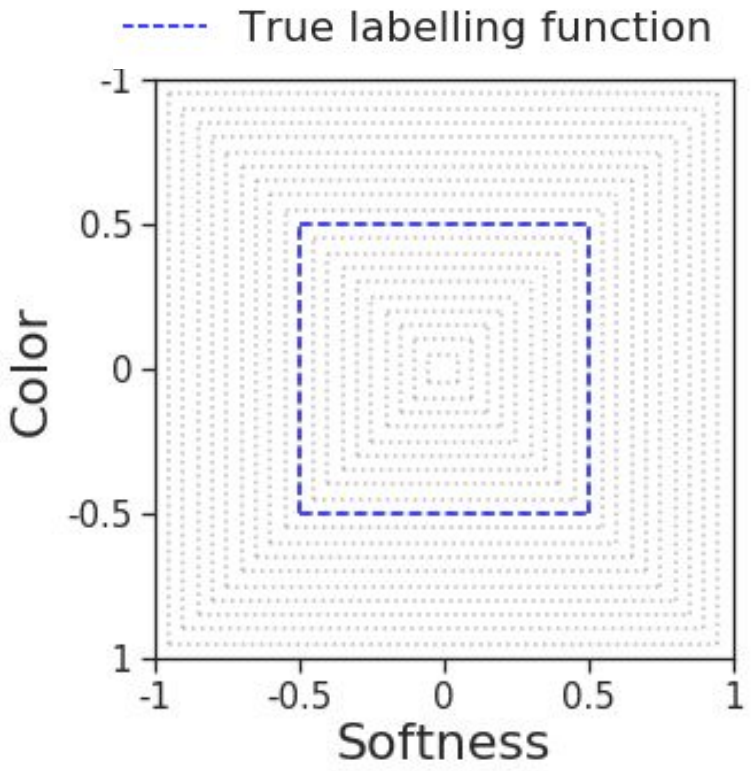
$\mathcal{H}$: *hypothesis class:*

axis aligned squares in steps of 0.05

$N_H = 20$

L: loss: $L\,(y_{true;i} - y_{est;i})$, $i \in$ *training*

$$L_{\mathcal{D}}(h_S) = \frac{|\{x \in \mathcal{D} : h_S(x) \neq f(x)\}|}{m}$$



Question:

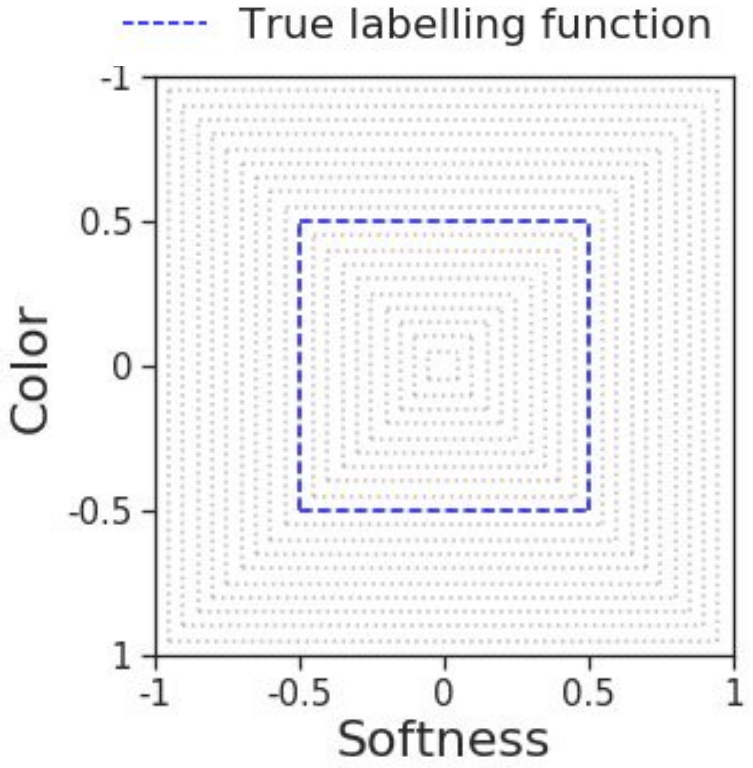
Data model: uniform distribution [-1,1] in both axis

$1 - \delta = 0.95 \leftarrow$ confidence

$\varepsilon = 0.05 \leftarrow$ contamination

$N_H = 20 \leftarrow$ number of possible squares

m = ??



What would you guess is the number of examples necessary for training?

Minimum number of samples

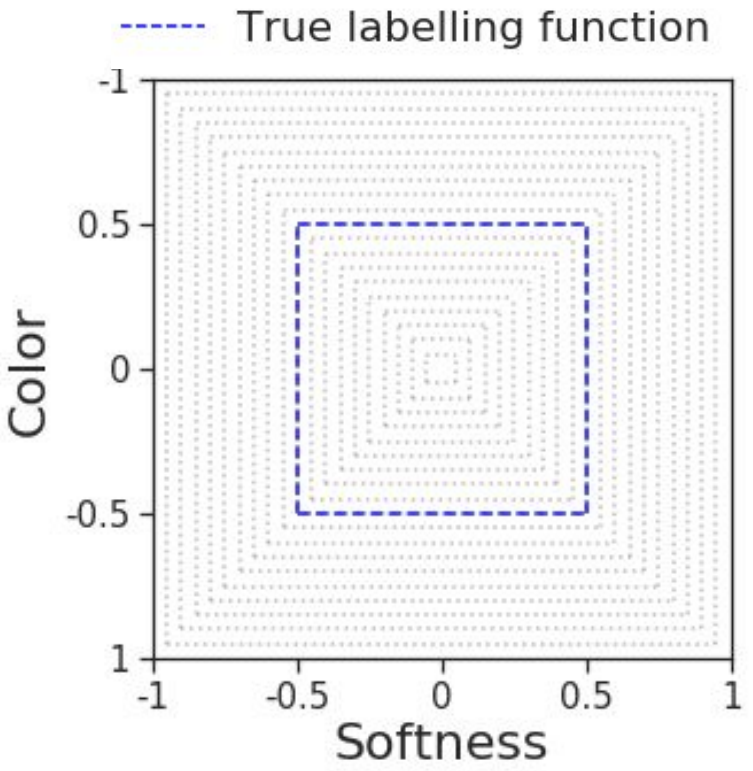
Data model: uniform distribution [-1,1] in both axis

$1 - \delta = 0.95 \leftarrow$ confidence

$\varepsilon = 0.05 \leftarrow$ contamination

$N_H = 20 \leftarrow$ number of possible squares

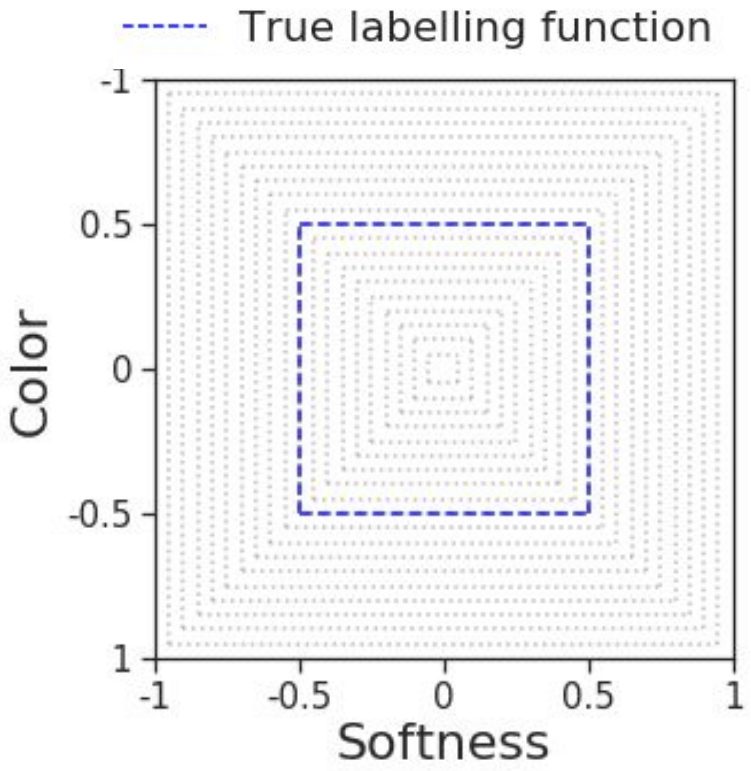
m ~ 120

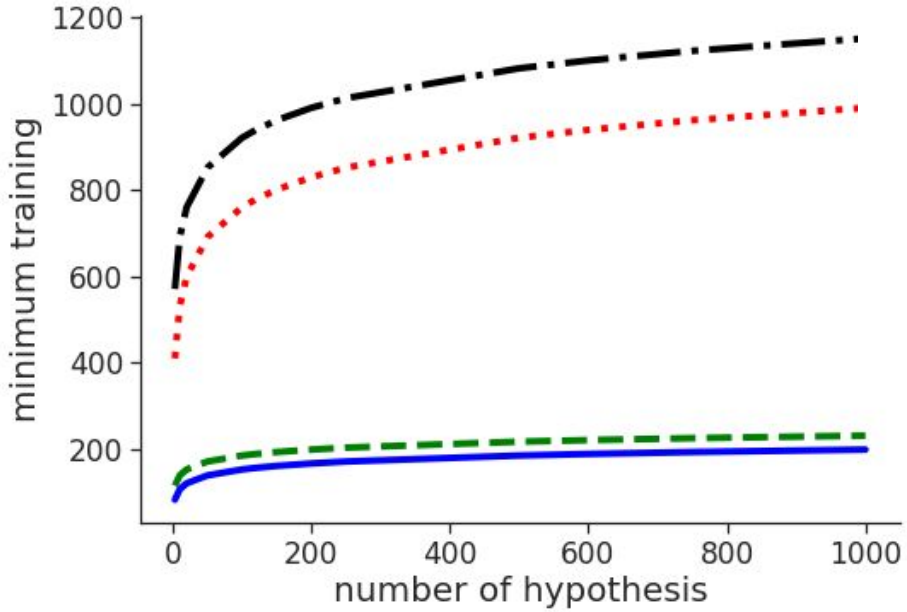


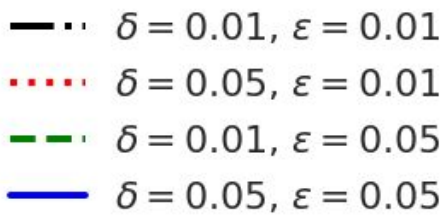
Minimum number of samples

Data model: uniform distribution
[-1,1] in both axis

$1 - \delta = 0.95$ ← confidence
$\varepsilon = 0.05$ ← contamination
$N_H = 20$ ← number of possible squares







If you still have stomach…

Next theory session:

- Agnostic PAC learning
- Uniform convergence
- Infinity number of hypothesis ...

Thank you!