Loss functions

Relevant chapters: Prince Ch 5

Miles Cranmer

Recipe for loss functions

1. Choose a suitable probability distribution $Pr(\mathbf{y}|\boldsymbol{\theta})$ that is defined over the domain of the predictions $\mathbf{y}$ and has distribution parameters $\boldsymbol{\theta}$.

Recipe for loss functions

1. Choose a suitable probability distribution $Pr(\mathbf{y}|\boldsymbol{\theta})$ that is defined over the domain of the predictions $\mathbf{y}$ and has distribution parameters $\boldsymbol{\theta}$.
2. Set the machine learning model $\mathbf{f}[\mathbf{x}, \phi]$ to predict one or more of these parameters so $\boldsymbol{\theta} = \mathbf{f}[\mathbf{x}, \phi]$ and $Pr(\mathbf{y}|\boldsymbol{\theta}) = Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \phi])$.

Recipe for loss functions

1. Choose a suitable probability distribution $Pr(\mathbf{y}|\boldsymbol{\theta})$ that is defined over the domain of the predictions $\mathbf{y}$ and has distribution parameters $\boldsymbol{\theta}$.
2. Set the machine learning model $\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}]$ to predict one or more of these parameters so $\boldsymbol{\theta} = \mathbf{f}[\mathbf{x}, \boldsymbol{\phi}]$ and $Pr(\mathbf{y}|\boldsymbol{\theta}) = Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}])$.
3. To train the model, find the network parameters $\hat{\boldsymbol{\phi}}$ that minimize the negative log-likelihood loss function over the training dataset pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$:

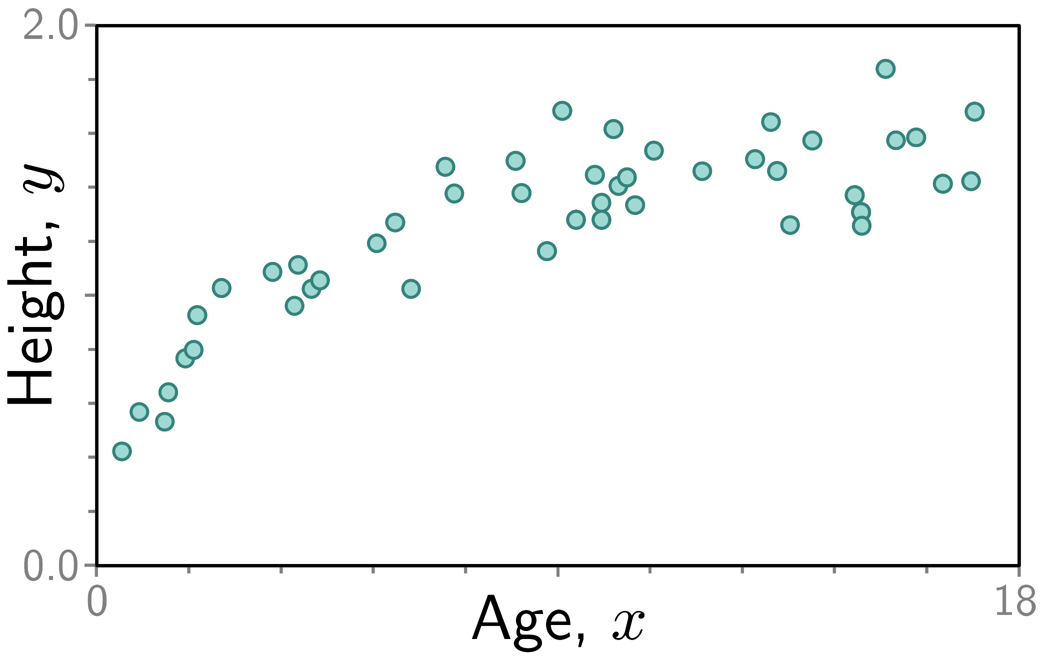
$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[L[\boldsymbol{\phi}]\right] = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[-\sum_{i=1}^{I}\log\Big[Pr(\mathbf{y}_i|\mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}])\Big]\right]. \tag{5.7}$$

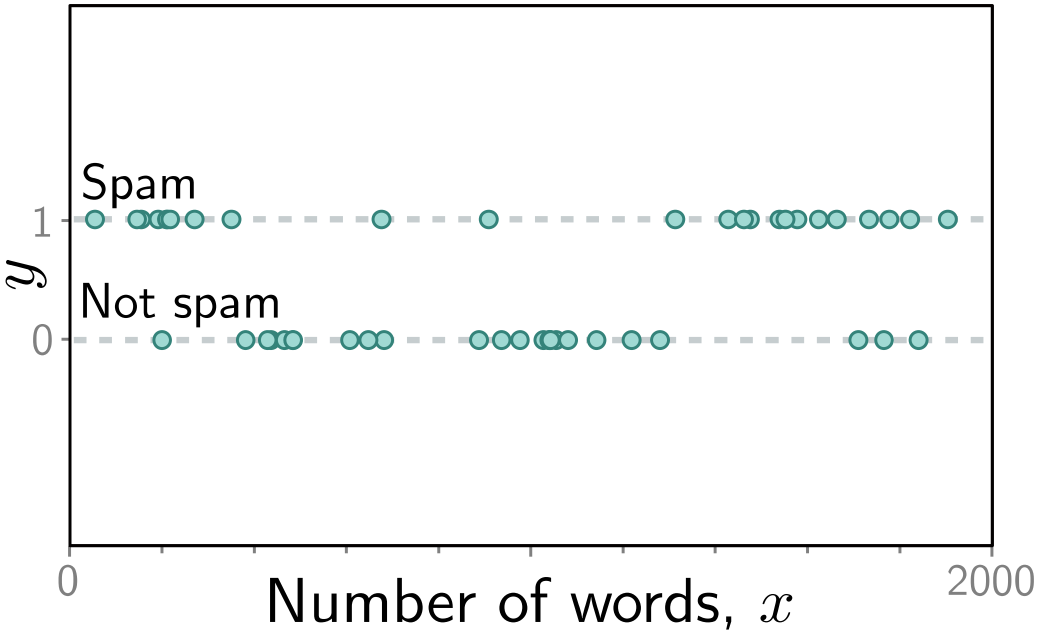
Recipe for loss functions

1. Choose a suitable probability distribution $Pr(\mathbf{y}|\boldsymbol{\theta})$ that is defined over the domain of the predictions $\mathbf{y}$ and has distribution parameters $\boldsymbol{\theta}$.
2. Set the machine learning model $\mathbf{f}[\mathbf{x}, \phi]$ to predict one or more of these parameters so $\boldsymbol{\theta} = \mathbf{f}[\mathbf{x}, \phi]$ and $Pr(\mathbf{y}|\boldsymbol{\theta}) = Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \phi])$.
3. To train the model, find the network parameters $\hat{\phi}$ that minimize the negative log-likelihood loss function over the training dataset pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$:

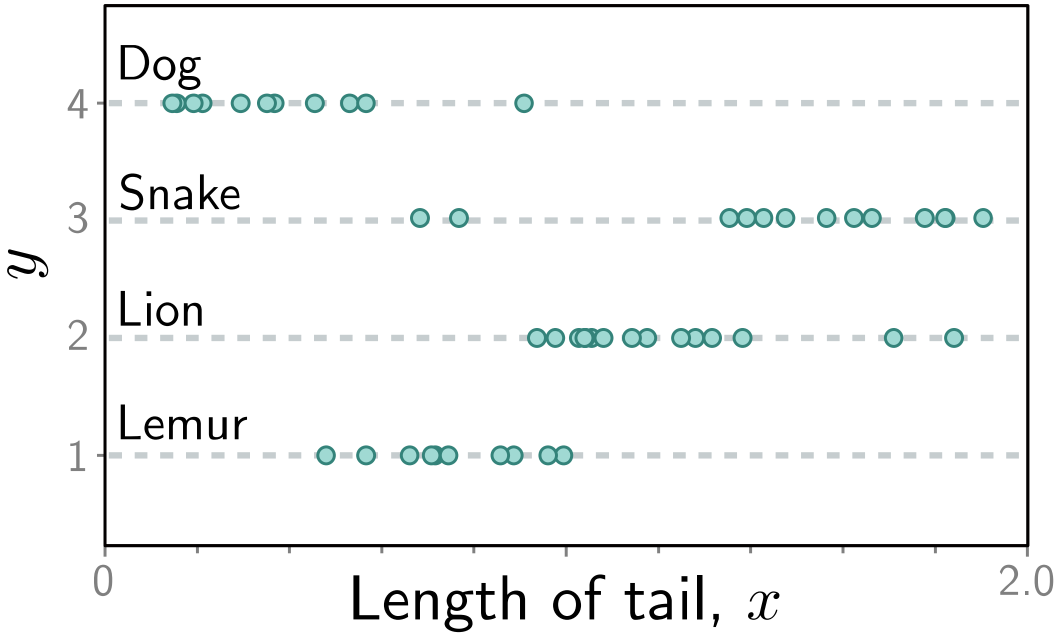
$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}} \left[L[\phi] \right] = \underset{\phi}{\operatorname{argmin}} \left[-\sum_{i=1}^{I} \log \left[Pr(\mathbf{y}_i|\mathbf{f}[\mathbf{x}_i, \phi]) \right] \right]. \tag{5.7}$$

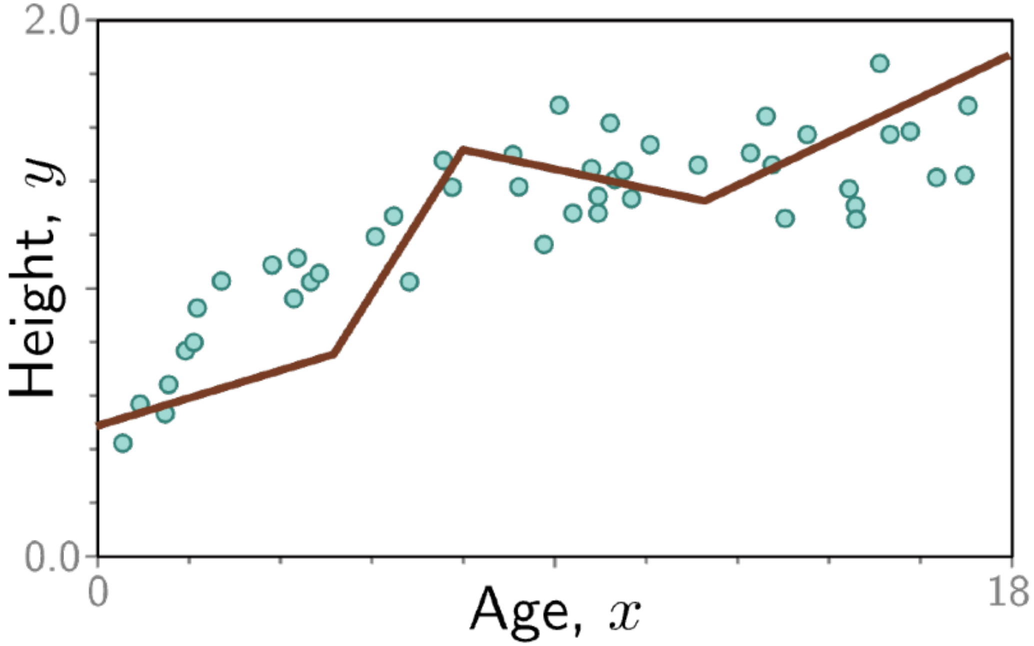
4. To perform inference for a new test example $\mathbf{x}$, return either the full distribution $Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \hat{\phi}])$ or the maximum of this distribution.

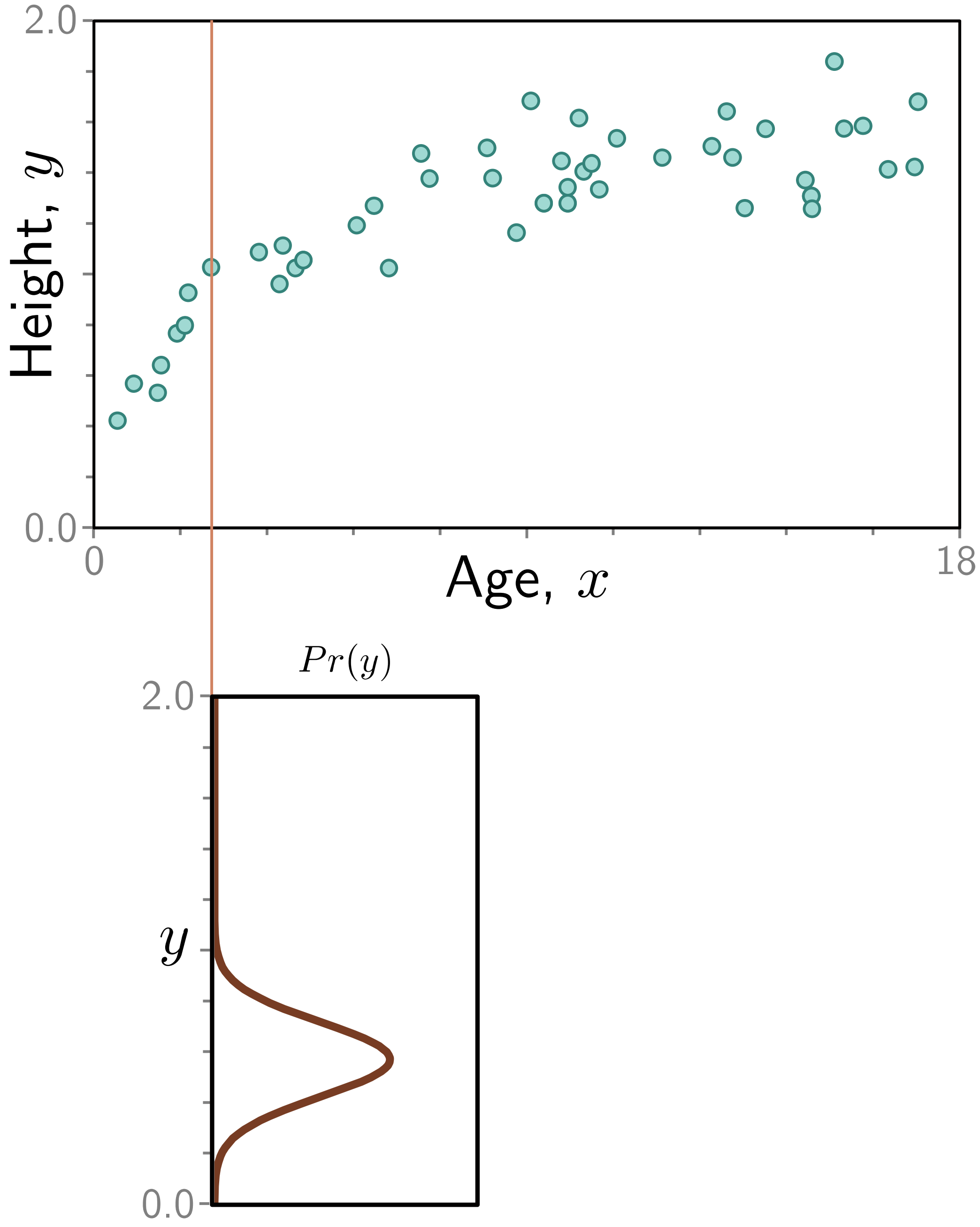

2.0
0.0
Height, y
0
18
Age, x

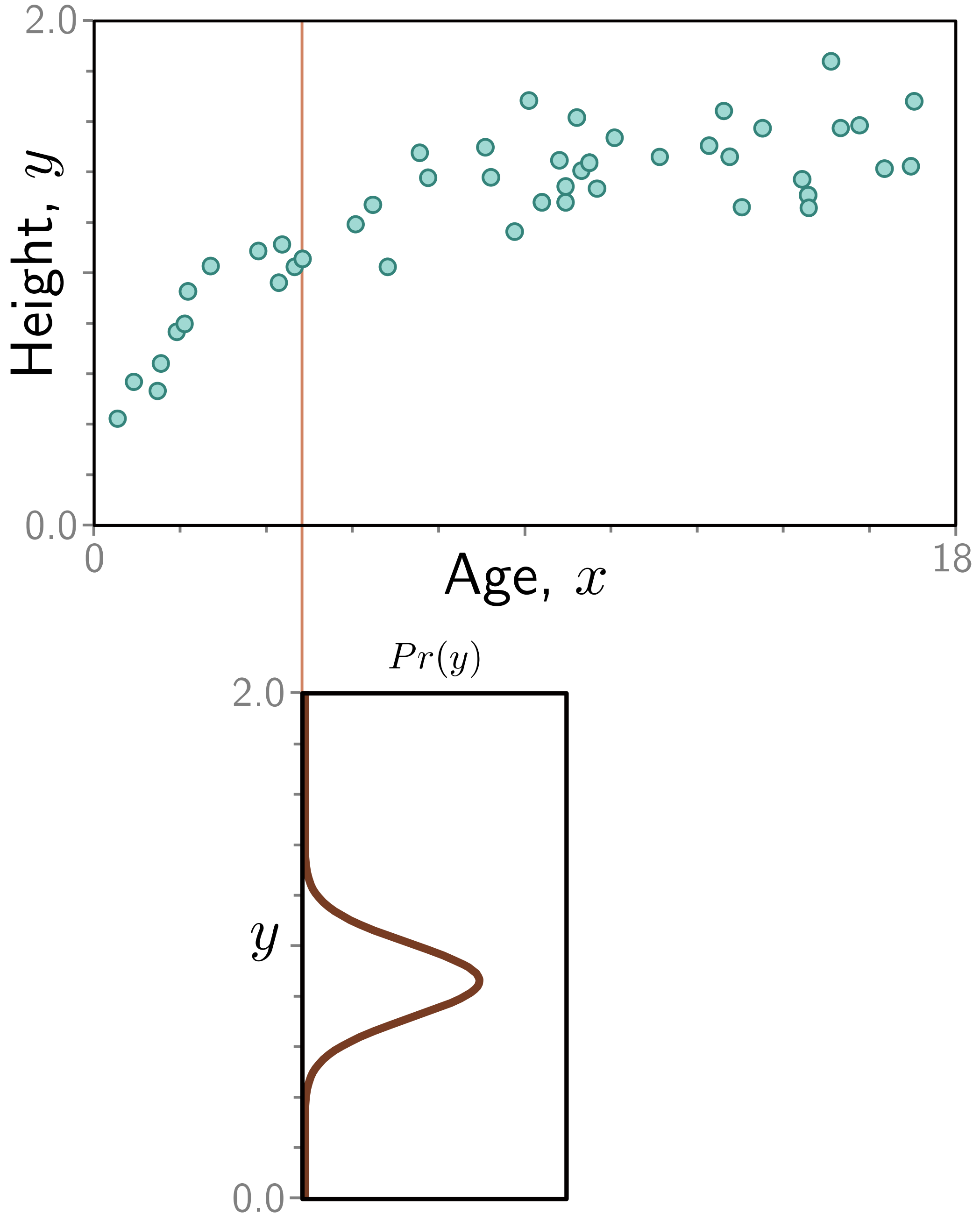


Spam
Not spam
1
0
y
0
2000
Number of words, x

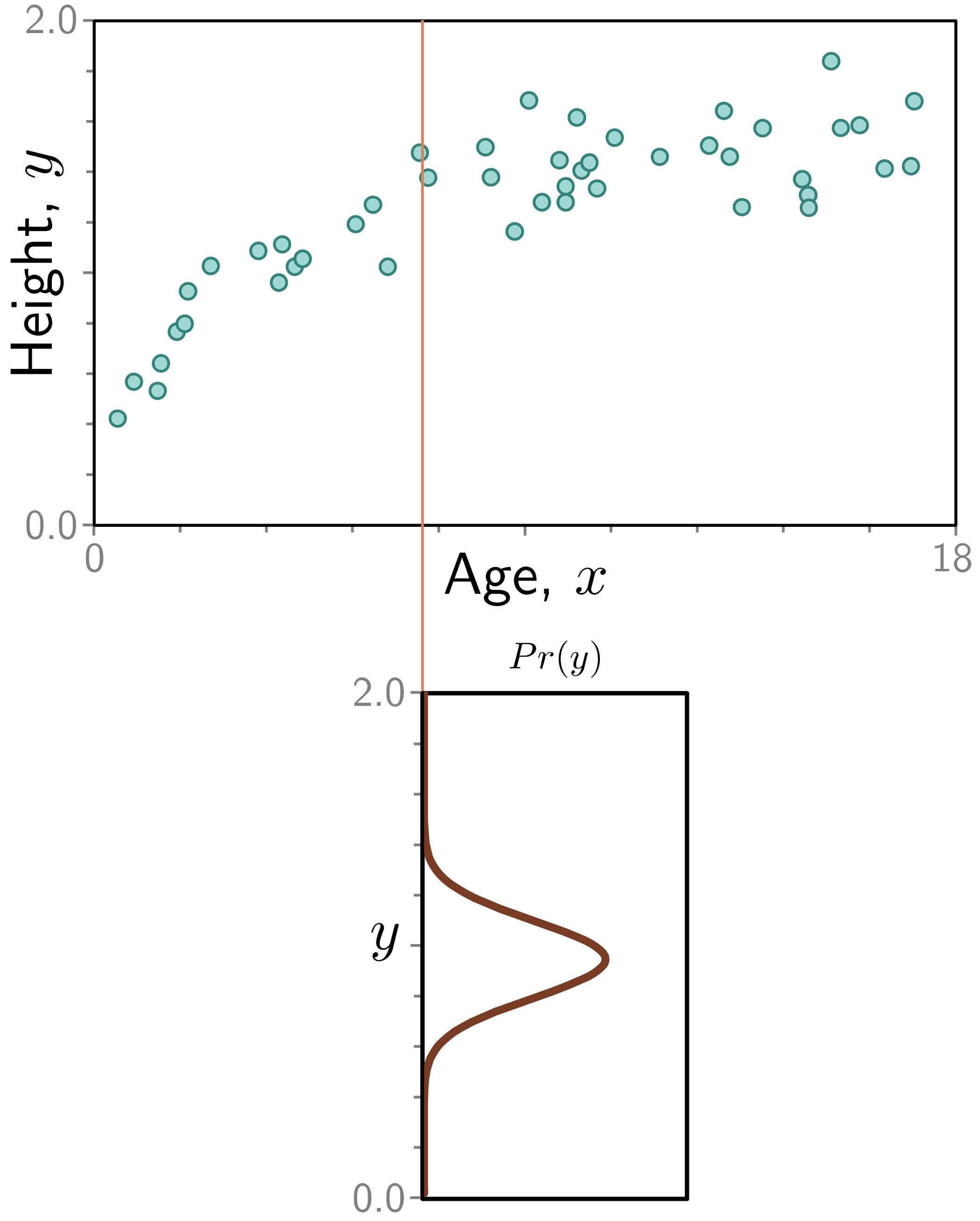

Dog
Snake
Lion
Lemur
4
3
2
1
y
0
2.0
Length of tail, x


2.0
0.0
Height, y
0
18
Age, x

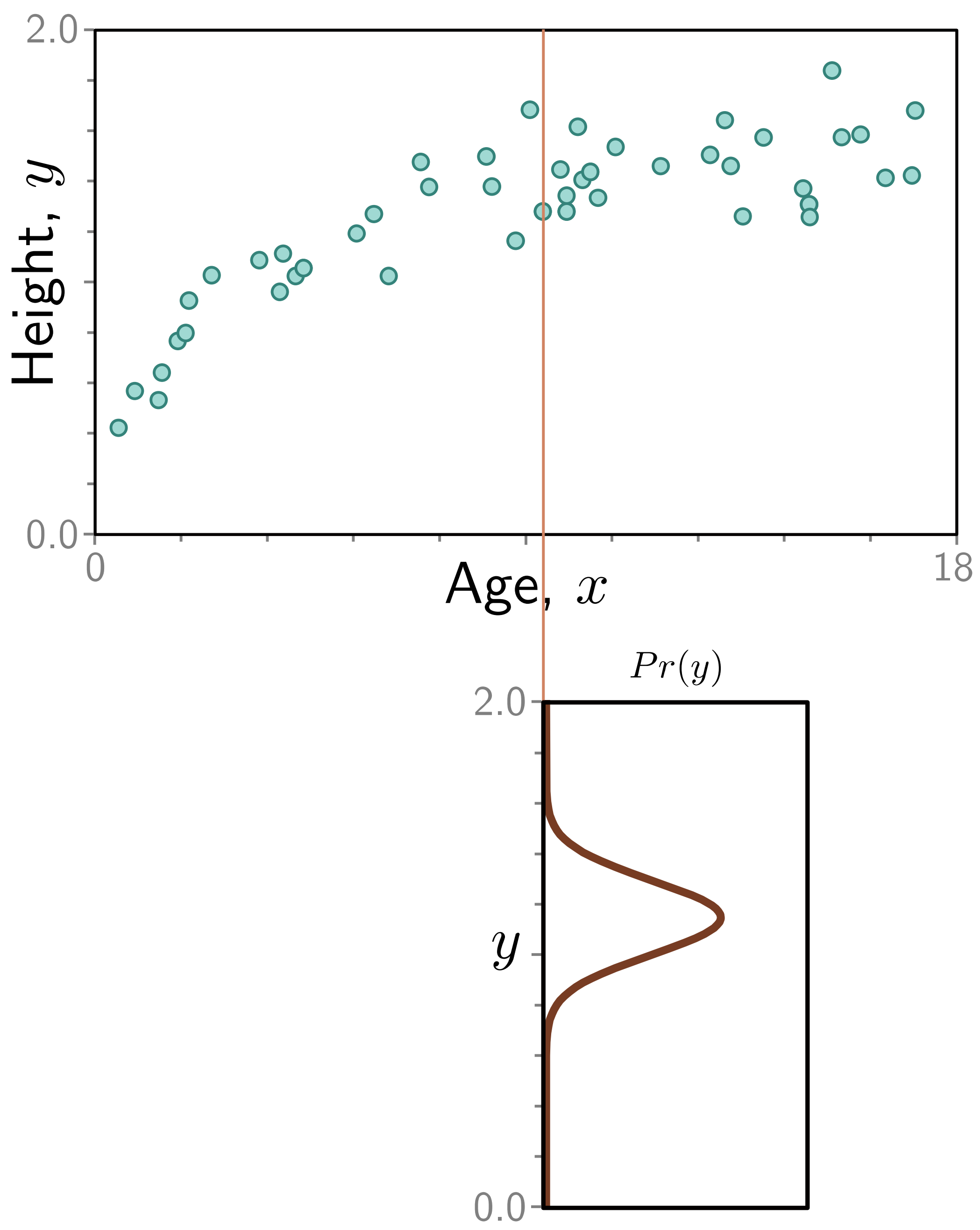

2.0
0.0
Height, y
0
18
Age, x
$Pr(y)$
2.0
y
0.0



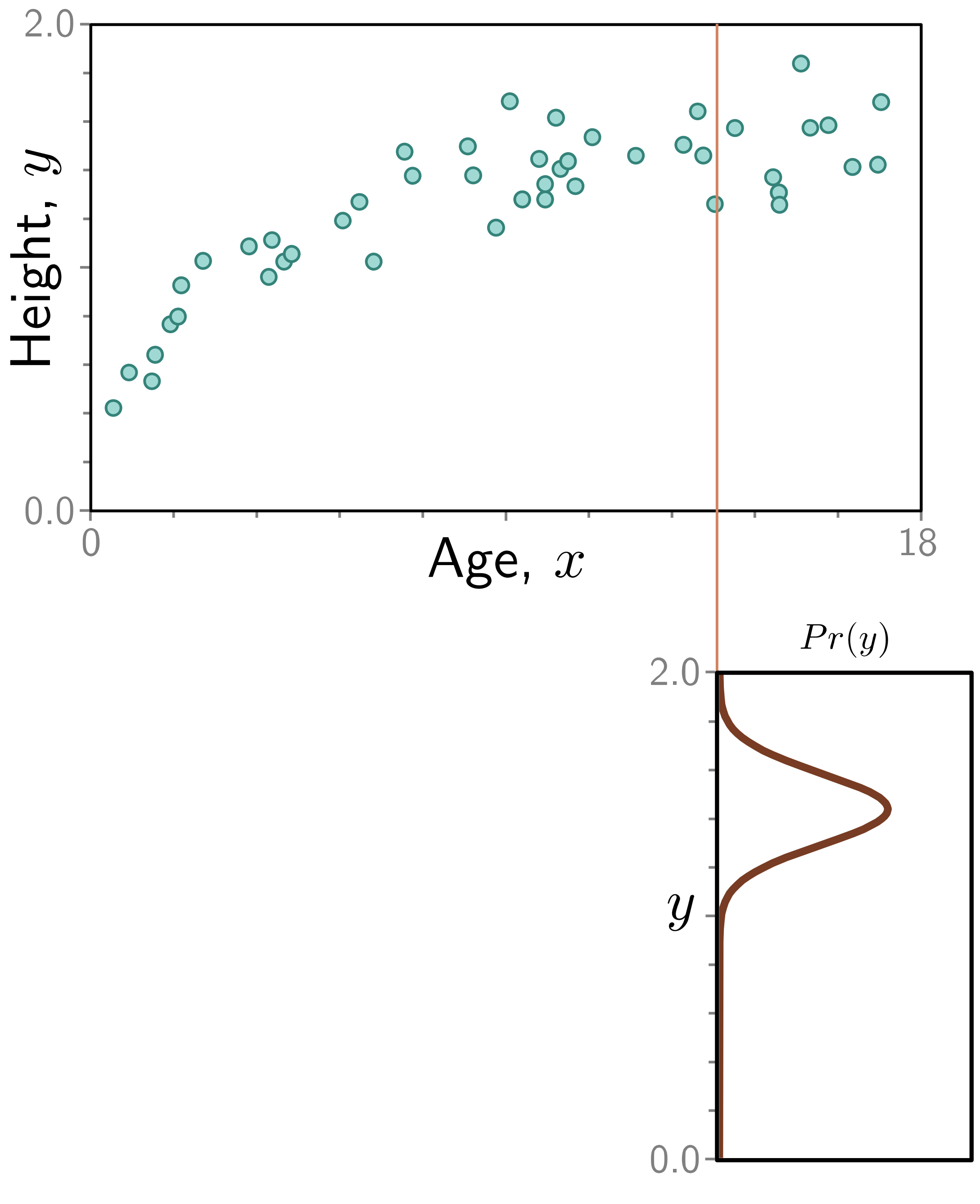
2.0
0.0
Height, y
0
18
Age, x
$Pr(y)$
2.0
y
0.0

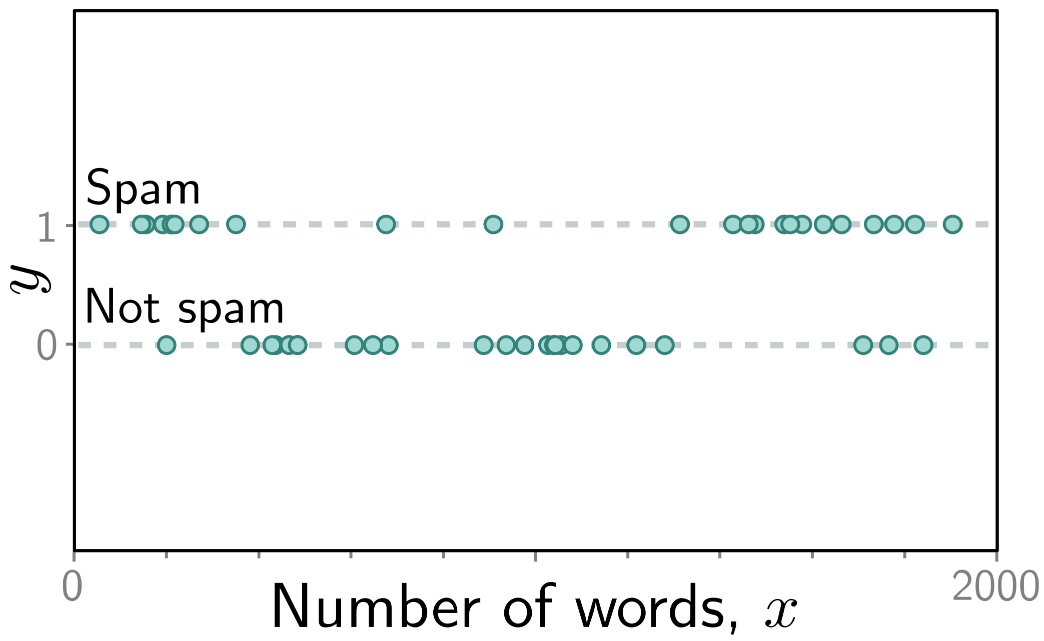


2.0
0.0
Height, y
0
18
Age, x
$Pr(y)$
2.0
y
0.0

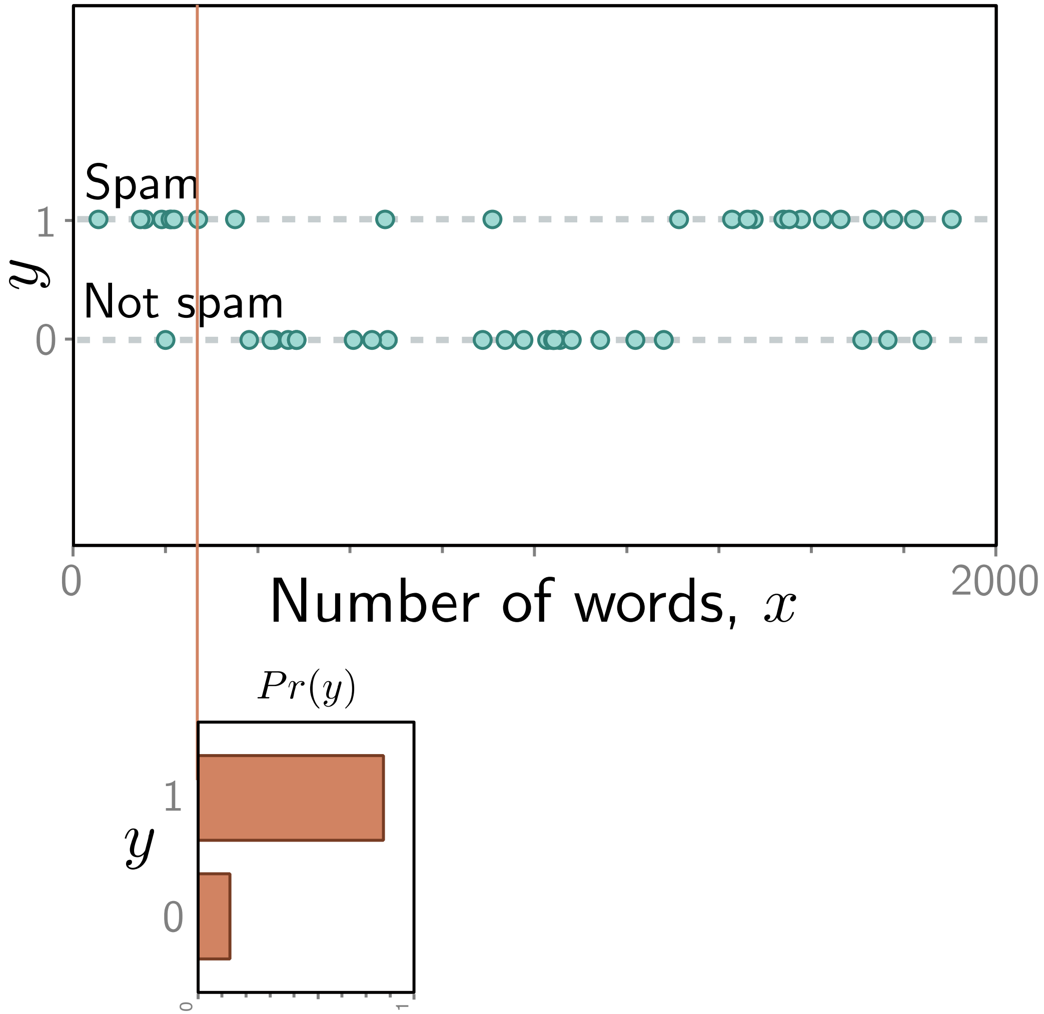


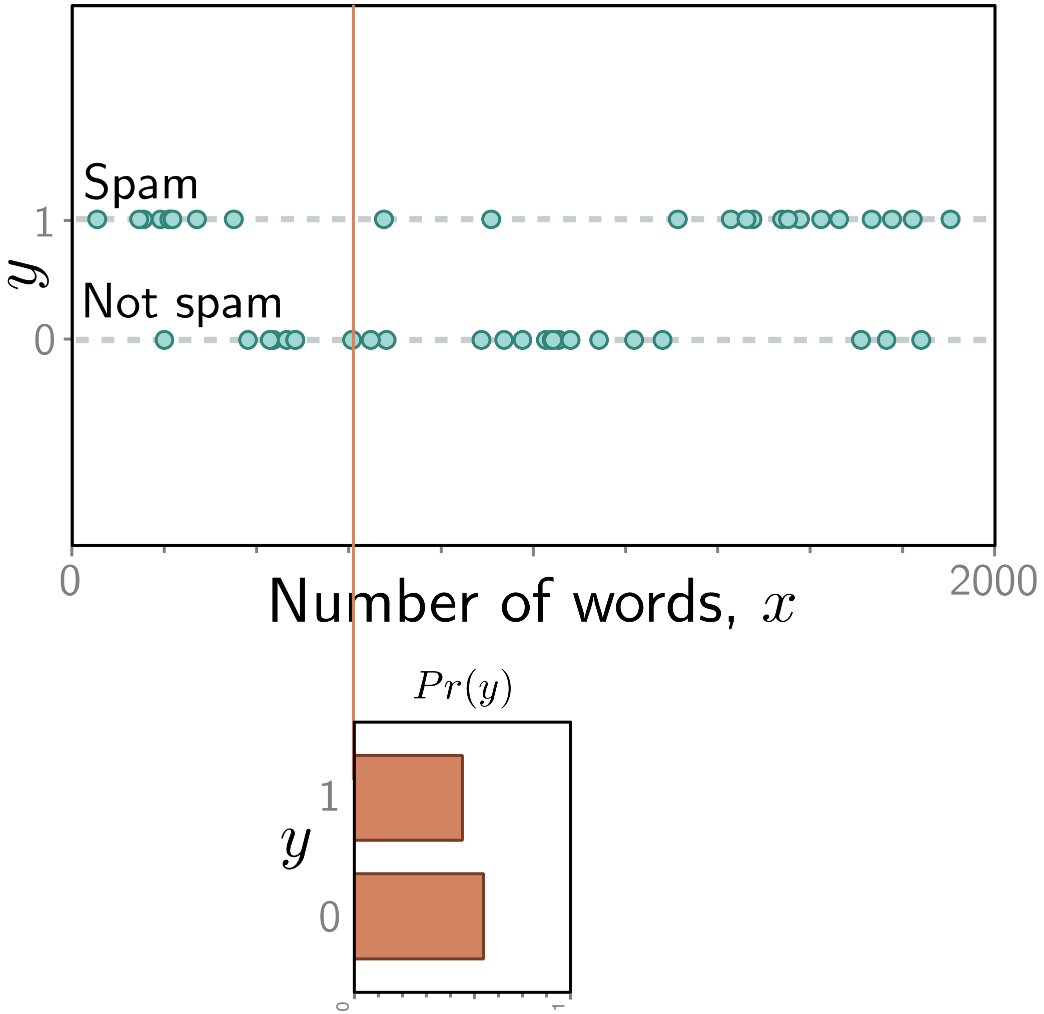
2.0
0.0
Height, y
0
18
Age, x
$Pr(y)$
2.0
y
0.0

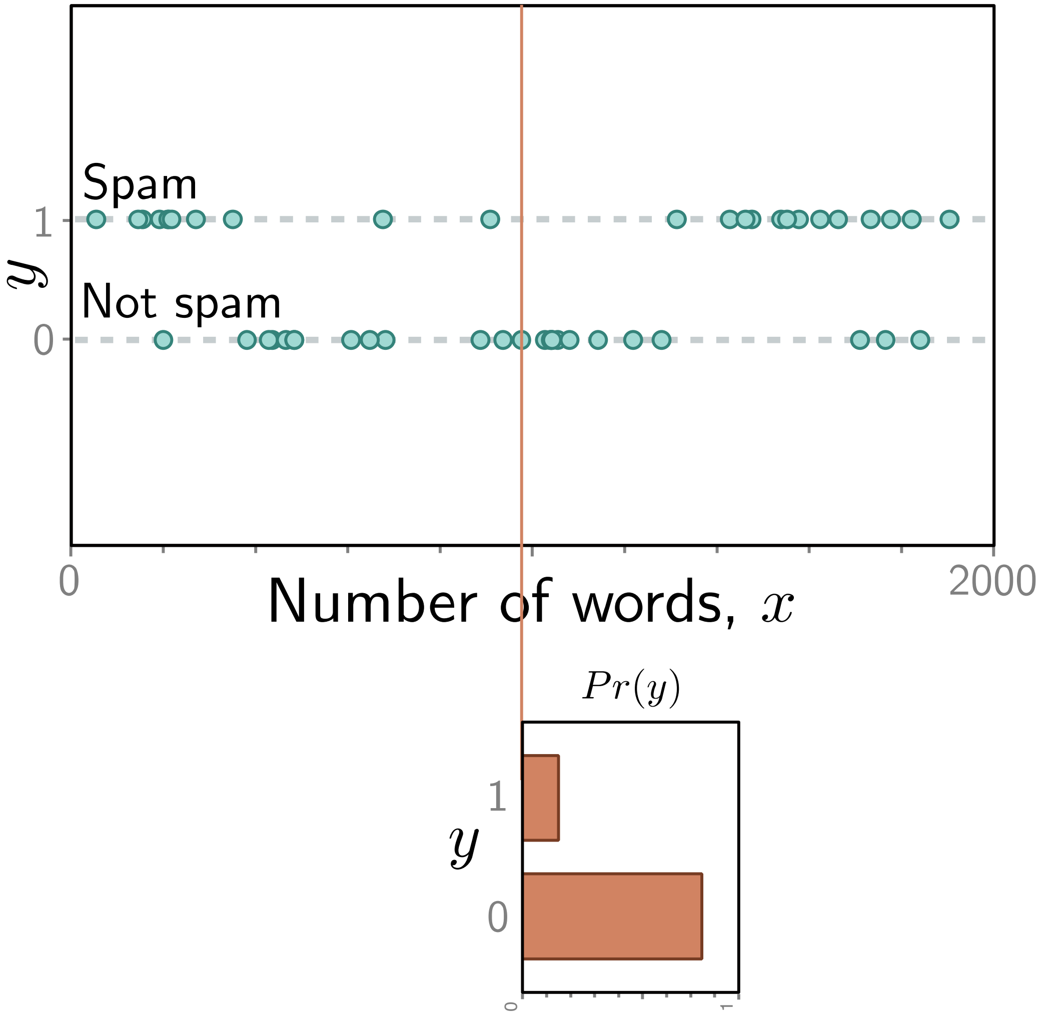

2.0
0.0
Height, y
0
18
Age, x
$Pr(y)$
2.0
y
0.0



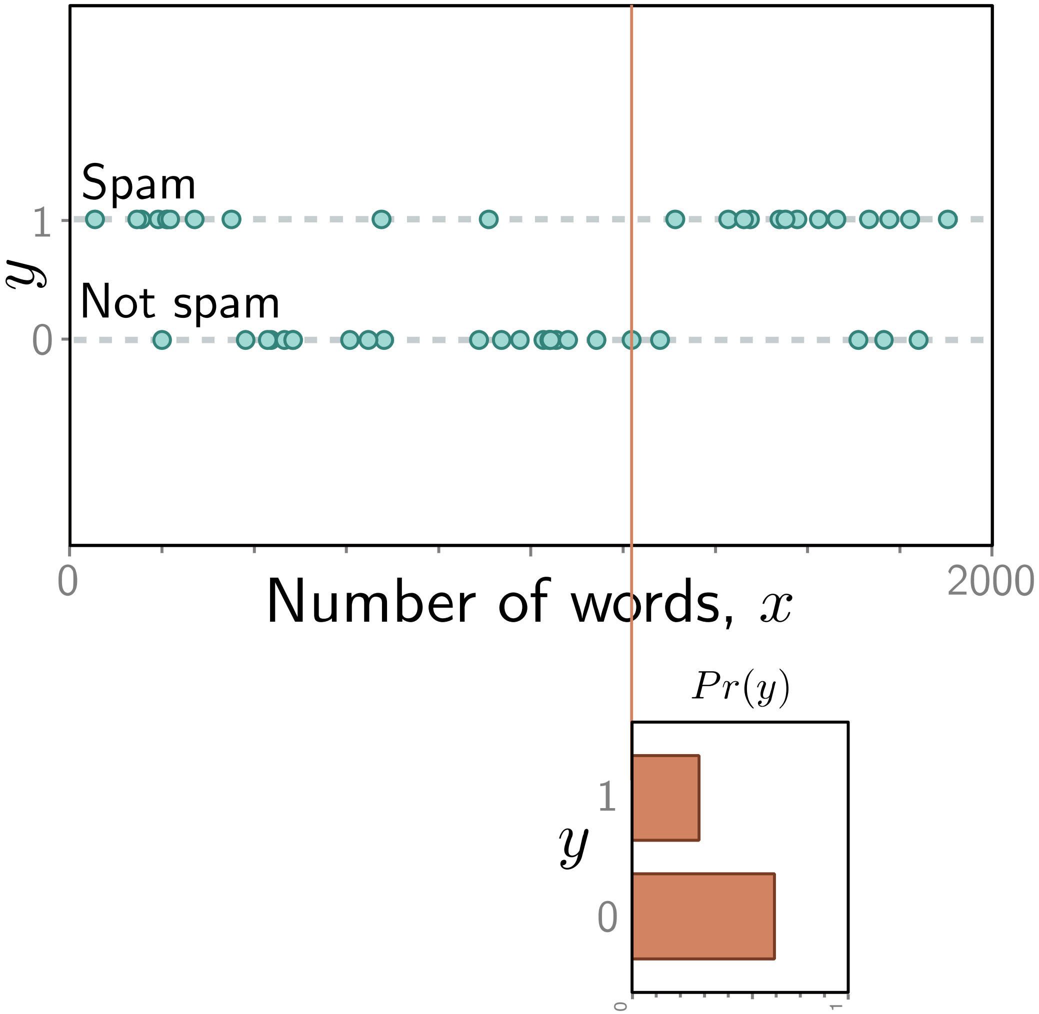
Spam
Not spam
1
0
y
0
2000
Number of words, x


Spam
Not spam
1
0
y
0
2000
Number of words, x
$Pr(y)$
1
0
y
0
1

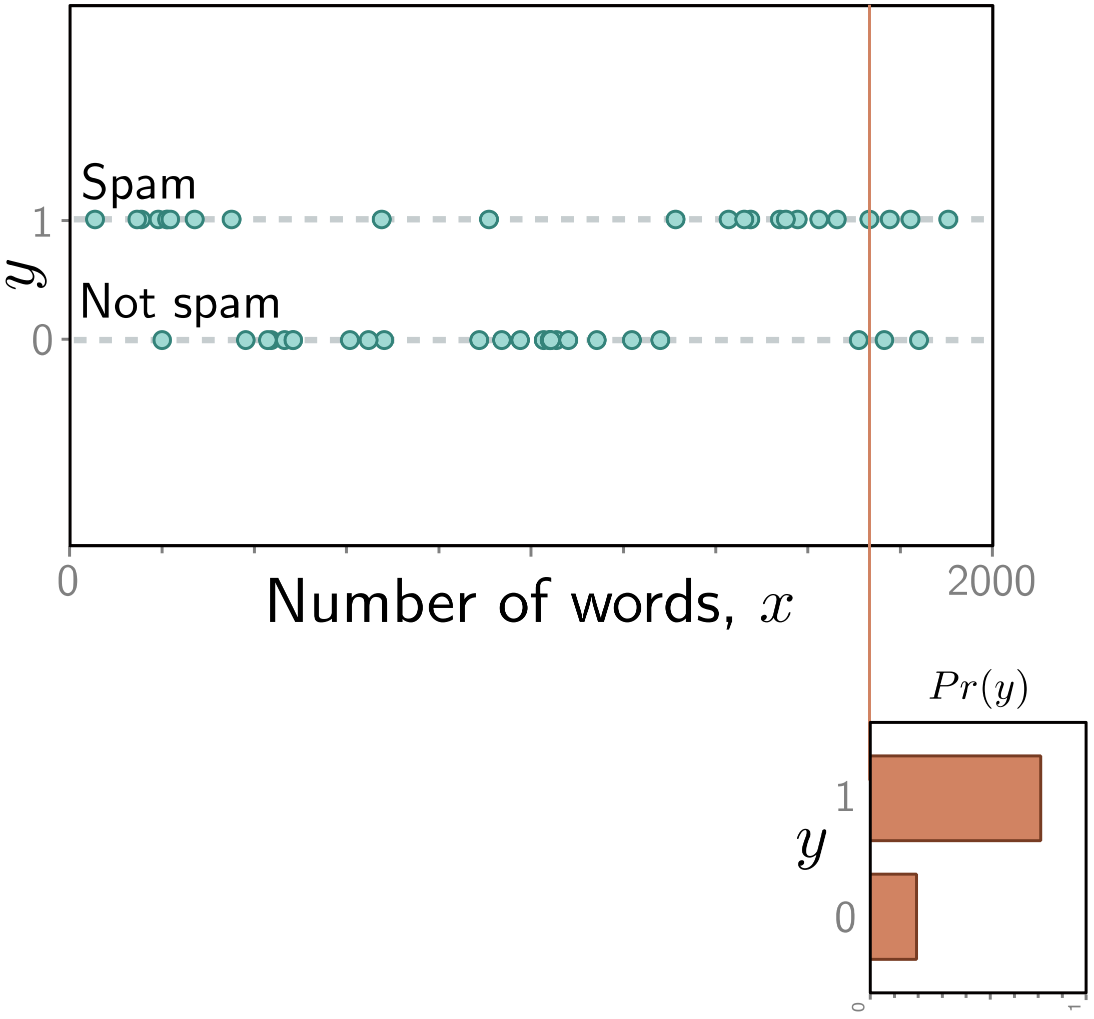

Spam
Not spam
1
0
y
0
2000
Number of words, x
$Pr(y)$
1
0
y
0
1

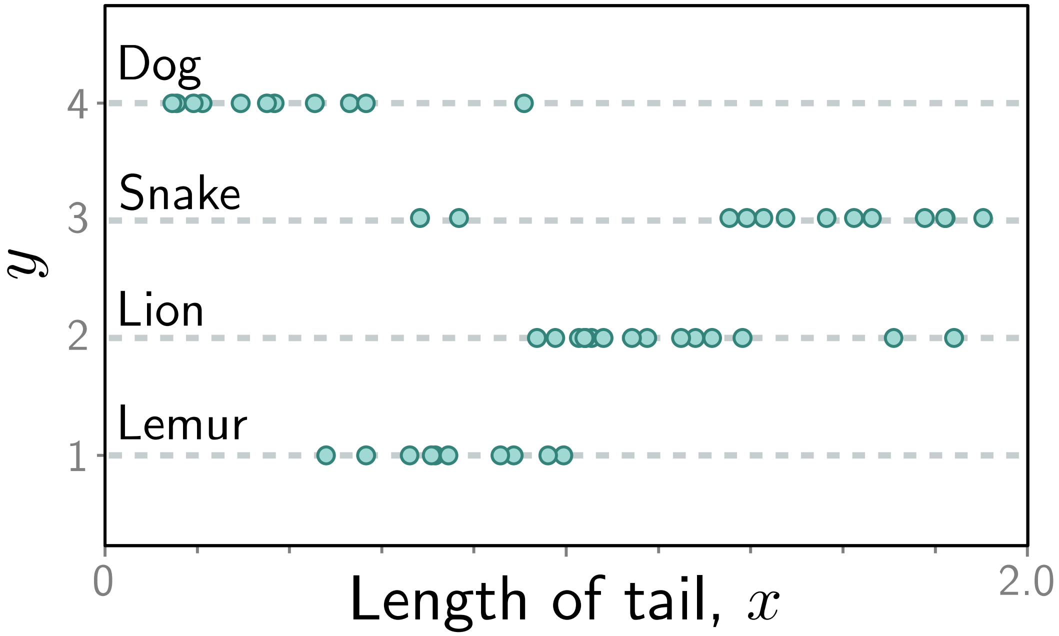


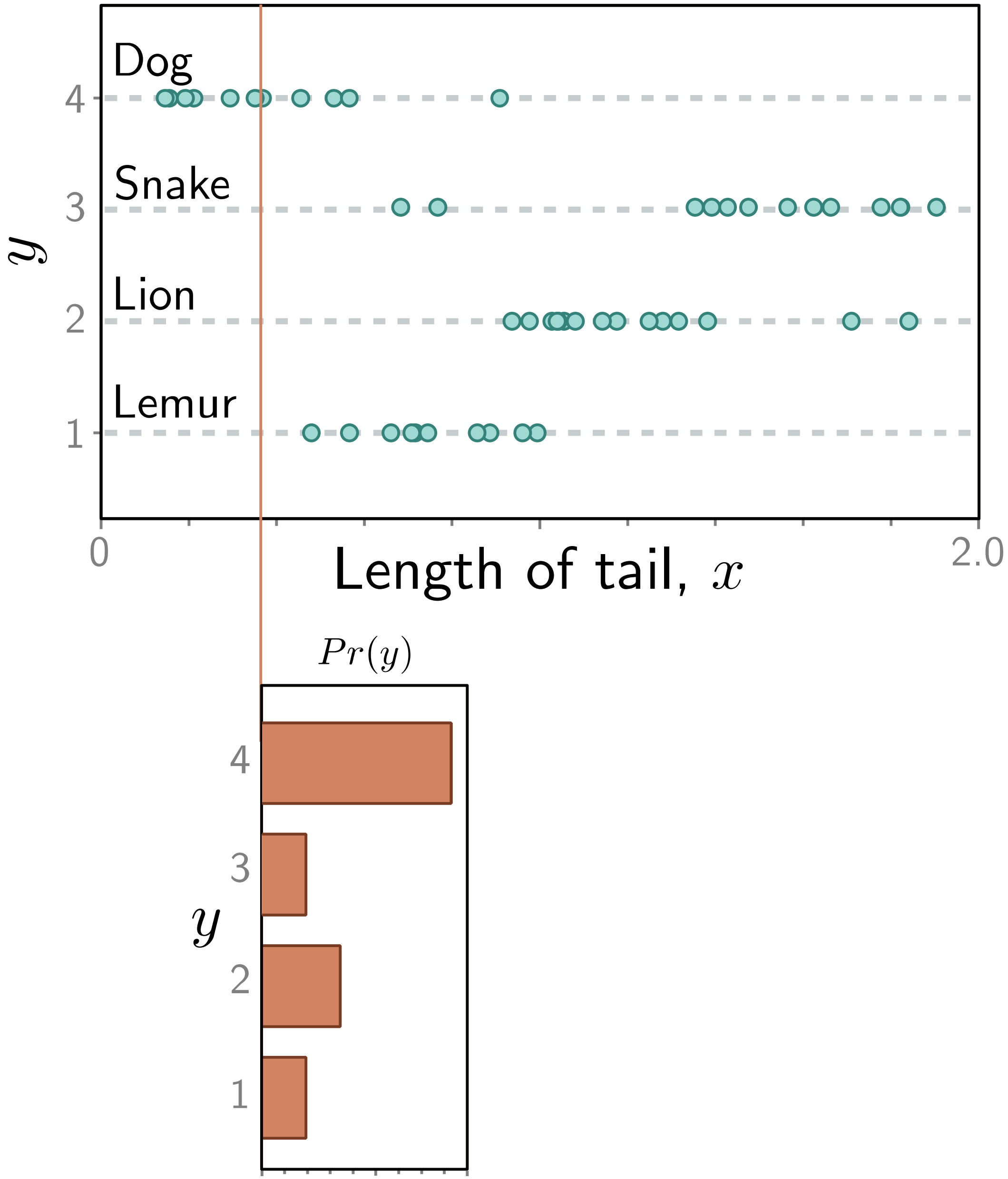
Spam
Not spam
1
0
y
0
2000
Number of words, x
$Pr(y)$
1
0
y
0
1

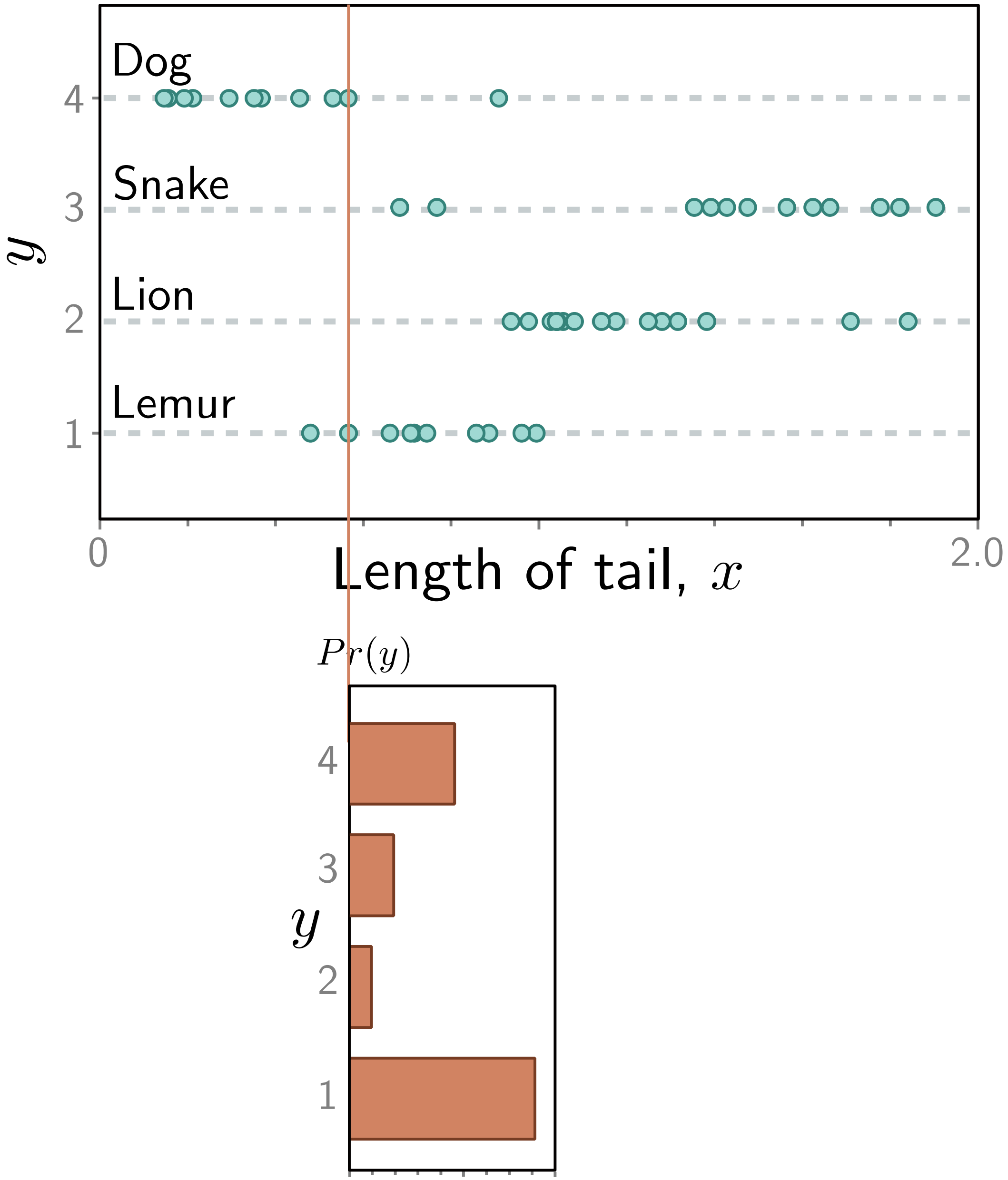


Spam
Not spam
1
0
y
0
2000
Number of words, x
Pr(y)
1
0
y
0
1

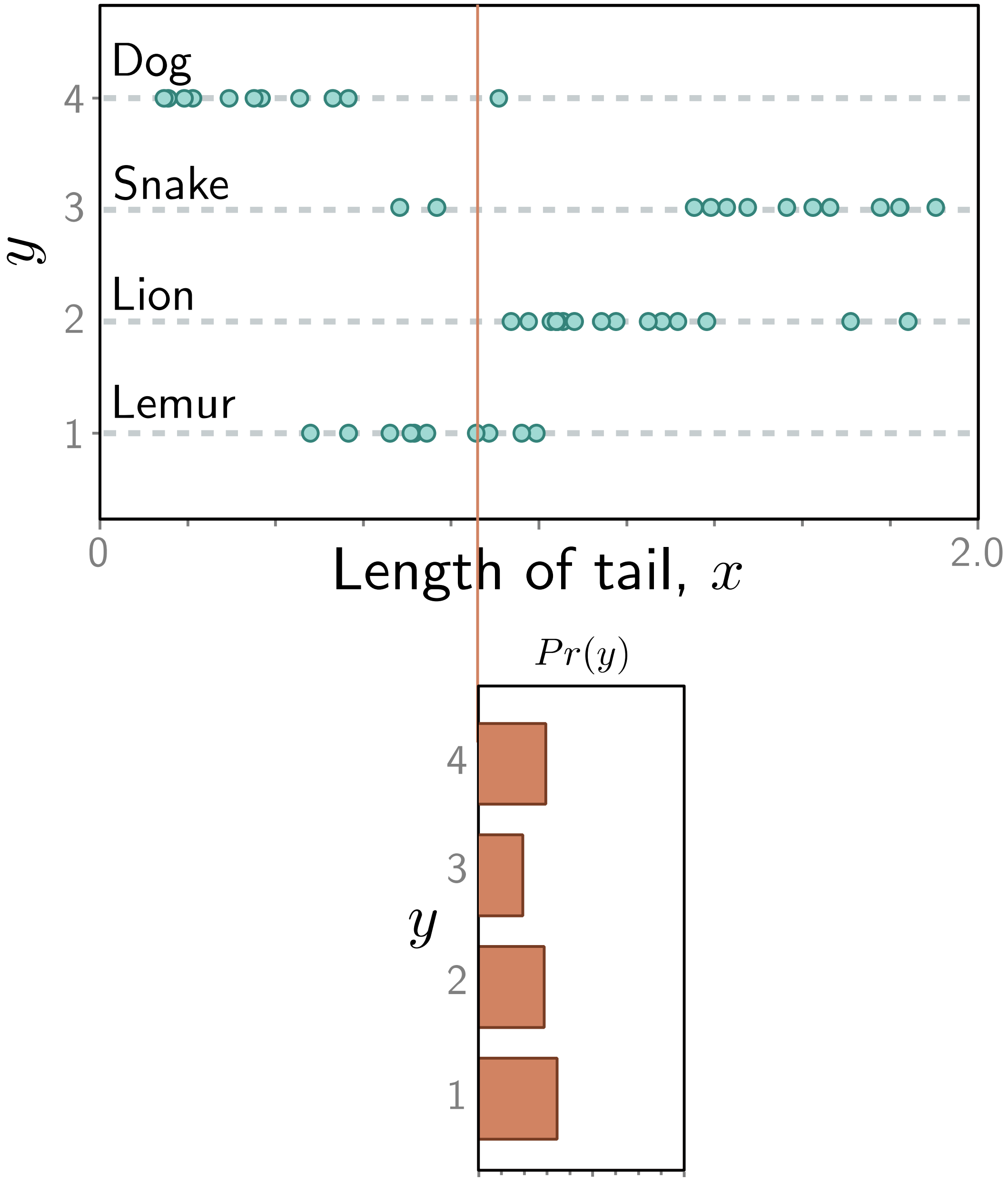

Spam
Not spam
1
0
y
0
2000
Number of words, x
$Pr(y)$
1
0
y
0
1


Dog
Snake
Lion
Lemur
4
3
2
1
y
0
2.0
Length of tail, x

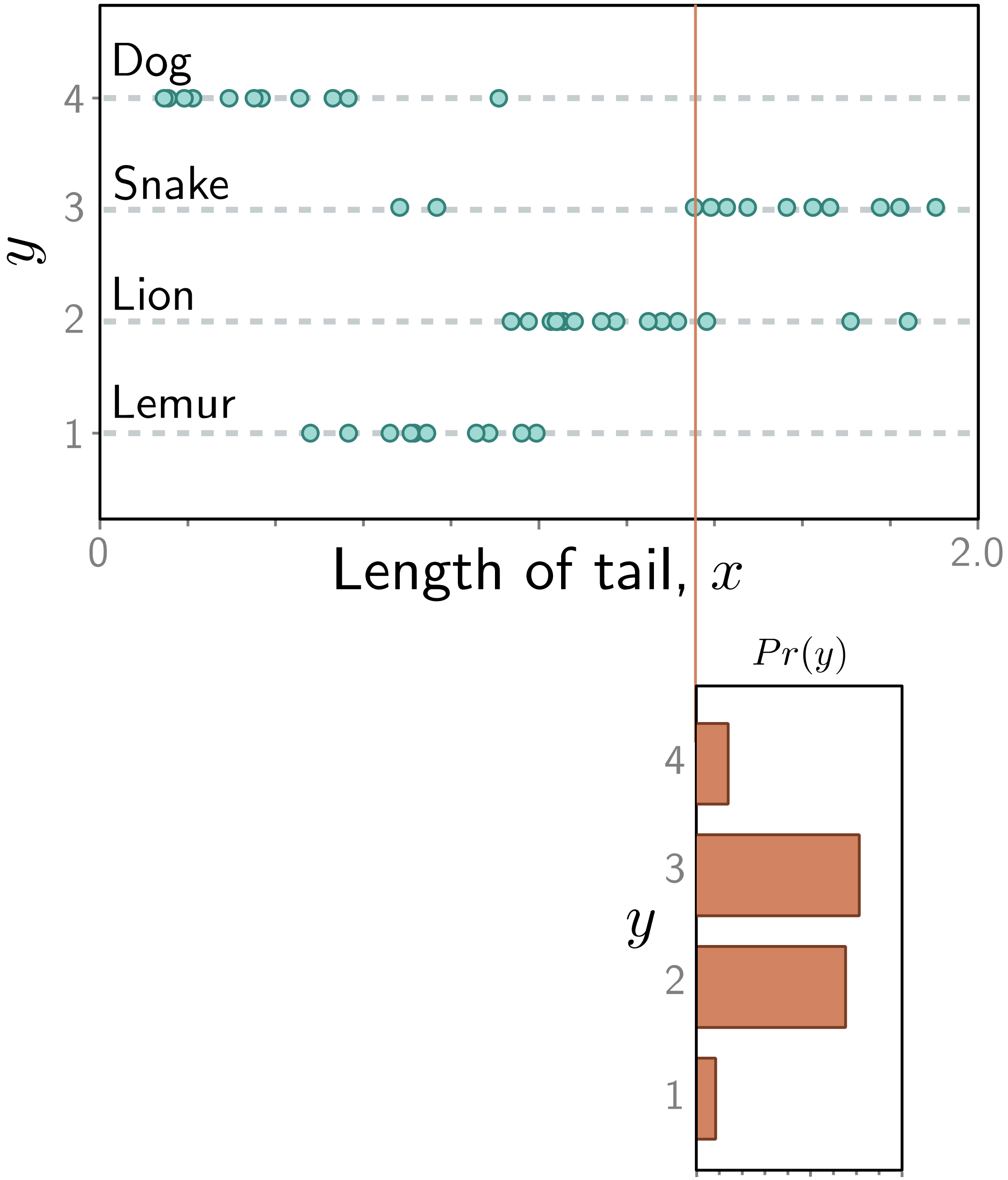

Dog
Snake
Lion
Lemur
4
3
2
1
y
0
2.0
Length of tail, x
$Pr(y)$
4
3
2
1
y



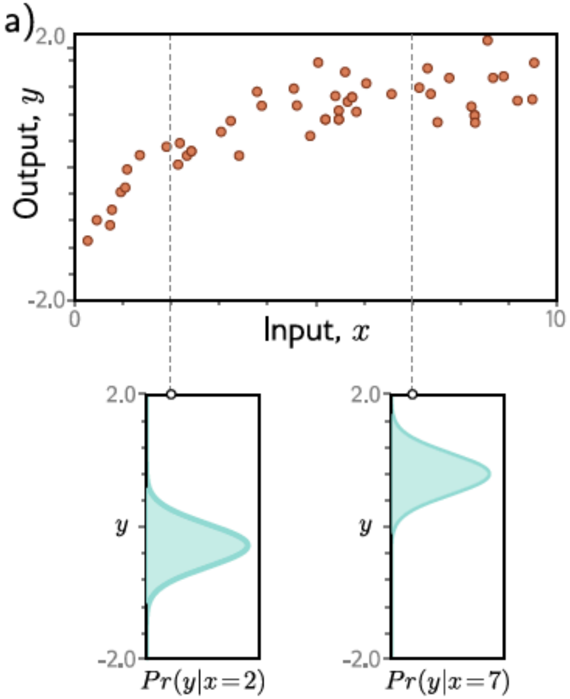
Dog
Snake
Lion
Lemur
4
3
2
1
y
0
2.0
Length of tail, x
Pr(y)
4
3
2
1
y

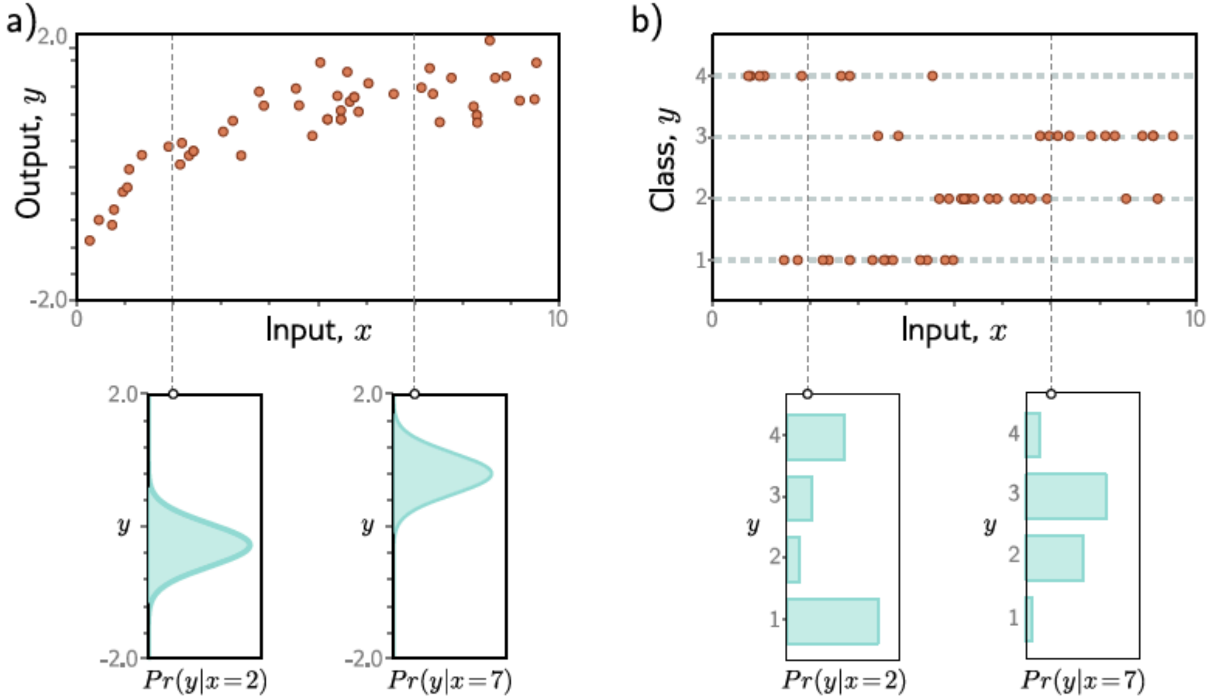


Dog
Snake
Lion
Lemur
4
3
2
1
y
0
2.0
Length of tail, x
$Pr(y)$
4
3
2
1
y

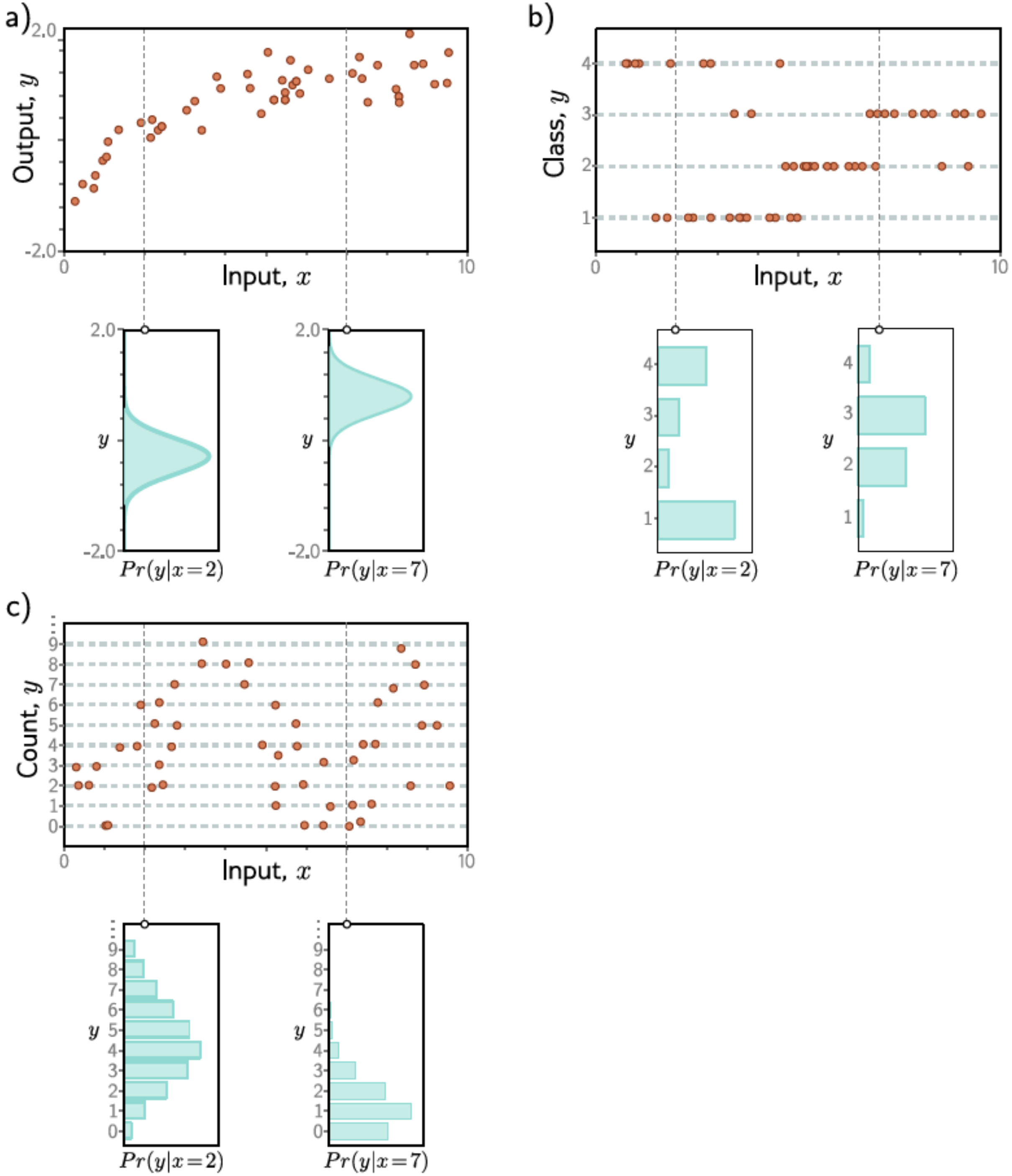


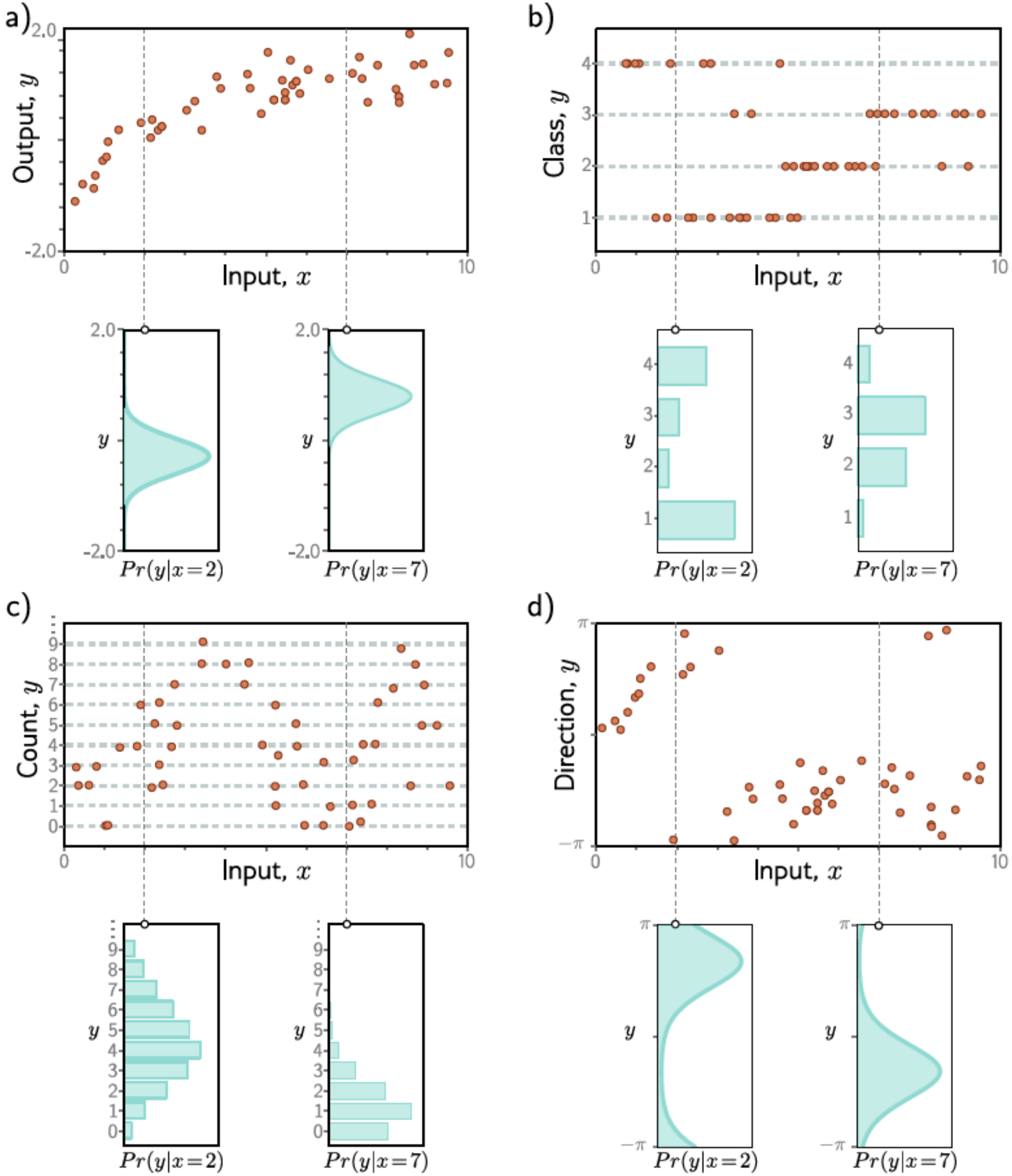
Dog
Snake
Lion
Lemur
4
3
2
1
y
0
2.0
Length of tail, x
$Pr(y)$
4
3
2
1
y


a)
2.0
-2.0
0
10
Output, y
Input, x
2.0
-2.0
y
$Pr(y|x=2)$
2.0
-2.0
y
$Pr(y|x=7)$



a)
Output, y
2.0
-2.0
0
10
Input, x
2.0
-2.0
y
$Pr(y|x=2)$
2.0
-2.0
y
$Pr(y|x=7)$
b)
Class, y
4
3
2
1
0
10
Input, x
4
3
2
1
y
$Pr(y|x=2)$
4
3
2
1
y
$Pr(y|x=7)$


a)
2.0
-2.0
Output, y
0
10
Input, x
y
$Pr(y|x=2)$
$Pr(y|x=7)$
b)
Class, y
Input, x
$Pr(y|x=2)$
$Pr(y|x=7)$
c)
Count, y
Input, x
$Pr(y|x=2)$
$Pr(y|x=7)$



a)
Output, y
Input, x
$Pr(y|x=2)$
$Pr(y|x=7)$
b)
Class, y
Input, x
$Pr(y|x=2)$
$Pr(y|x=7)$
c)
Count, y
Input, x
$Pr(y|x=2)$
$Pr(y|x=7)$
d)
Direction, y
Input, x
$Pr(y|x=2)$
$Pr(y|x=7)$

Loss function

- Training dataset of I pairs of input/output examples:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}$$

- Loss function or cost function measures how bad model is:

$$L\Big[\underbrace{\phi, \mathrm{f}[\mathbf{x}, \phi]}_{\text{model}}, \underbrace{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}}_{\text{train data}}\Big]$$

Loss function

- Training dataset of I pairs of input/output examples:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}$$

- Loss function or cost function measures how bad model is:

or for short:

$$L[\phi]$$

Returns a scalar that is smaller when model maps inputs to outputs better

Training

- Loss function:

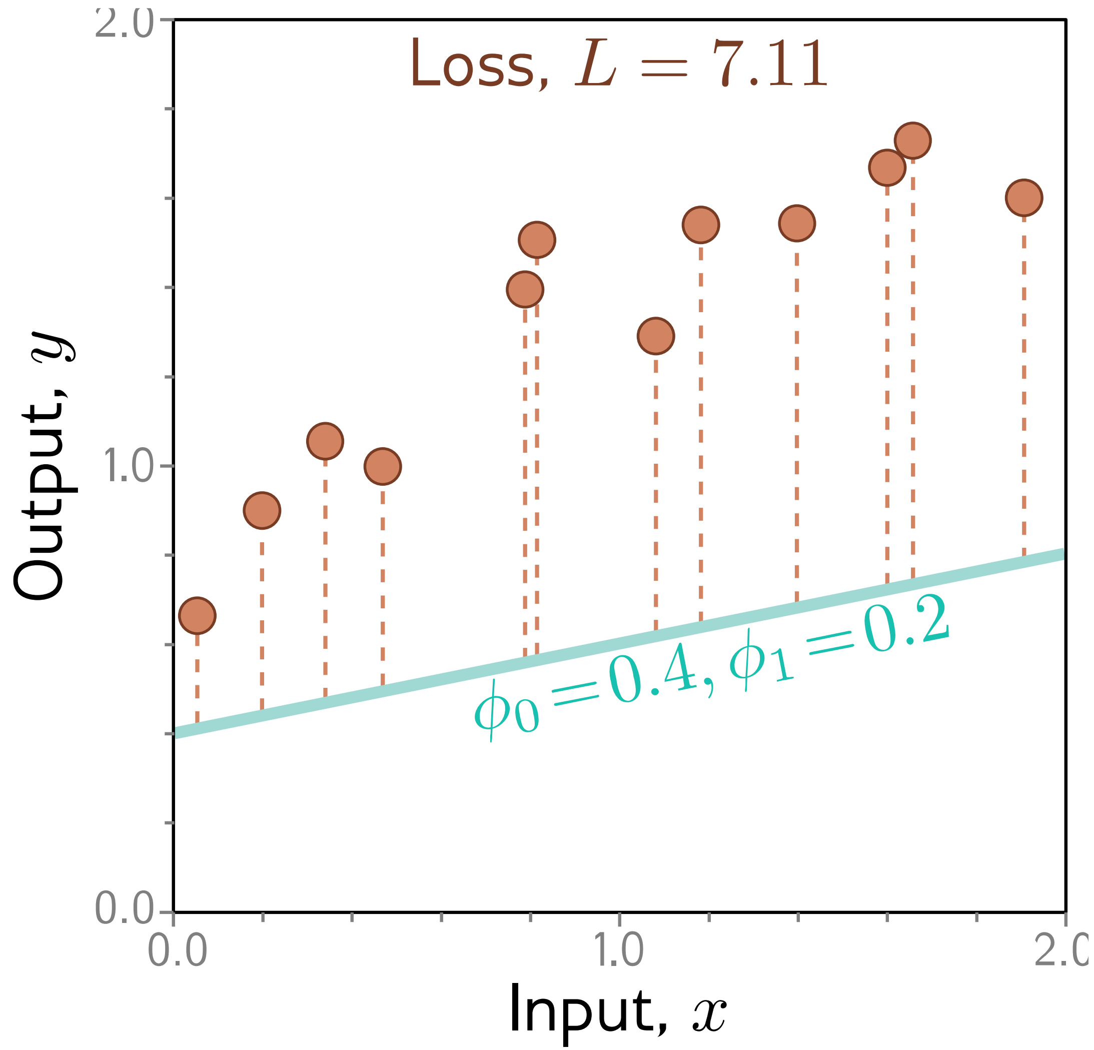
$$L\left[\phi\right]$$

Returns a scalar that is smaller when model maps inputs to outputs better

- Find the parameters that minimize the loss:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}}\Big[\mathrm{L}\left[\phi\right]\Big]$$

Example: 1D Linear regression loss function



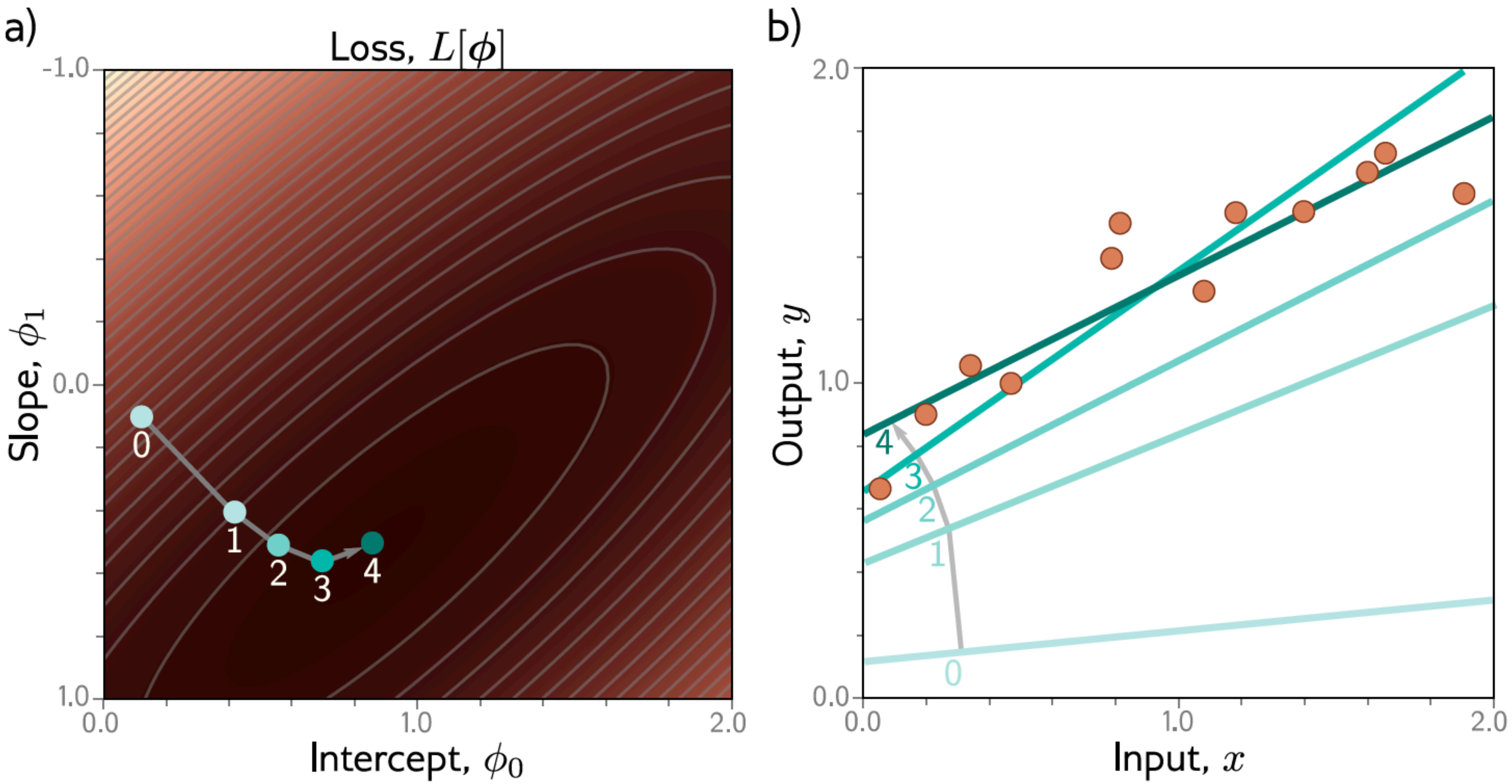
Loss function:

$$L[\phi] = \sum_{i=1}^{I} (\mathrm{f}[x_i, \phi] - y_i)^2$$

$$= \sum_{i=1}^{I} (\phi_0 + \phi_1 x_i - y_i)^2$$

"Least squares loss function"

Example: 1D Linear regression training



This technique is known as gradient descent

Loss functions

- Maximum likelihood
- Recipe for loss functions
- Example 1: univariate regression
- Example 2: binary classification
- Example 3: multiclass classification
- Other types of data
- Multiple outputs
- Cross entropy

How to construct loss functions

- Model predicts output y given input x

How to construct loss functions

- ~~Model predicts output y given input x~~

How to construct loss functions

- ~~Model predicts output *y* given input *x*~~
- Model predicts a conditional probability distribution:

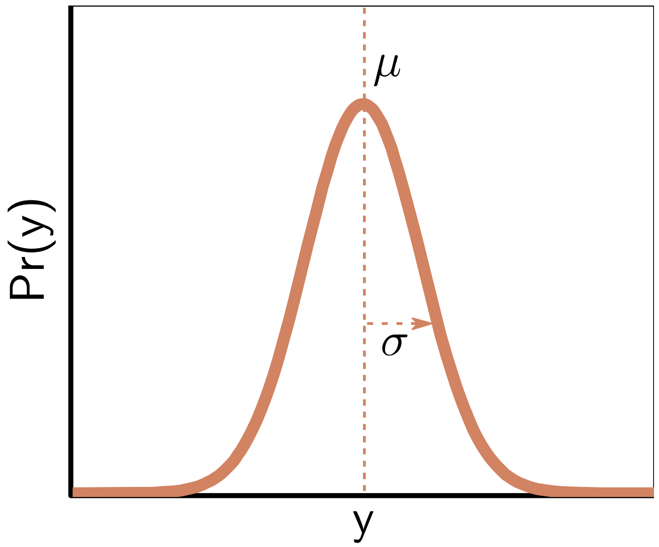
$$Pr(\mathbf{y}|\mathbf{x})$$

  over outputs **y** given inputs **x**.
- Loss function aims to make the outputs have high probability

How can a model predict a probability distribution?

1. Pick a known distribution (e.g., normal distribution) to model output y with parameters $\boldsymbol{\theta}$

   e.g., the normal distribution $\boldsymbol{\theta} = \{\mu, \sigma^2\}$



2. Use model to predict parameters $\boldsymbol{\theta}$ of probability distribution

Maximum likelihood criterion

$$\begin{aligned}\hat{\boldsymbol{\phi}} &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{x}_i)\right] \\ &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i|\boldsymbol{\theta}_i)\right] \\ &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i|\mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}])\right]\end{aligned}$$

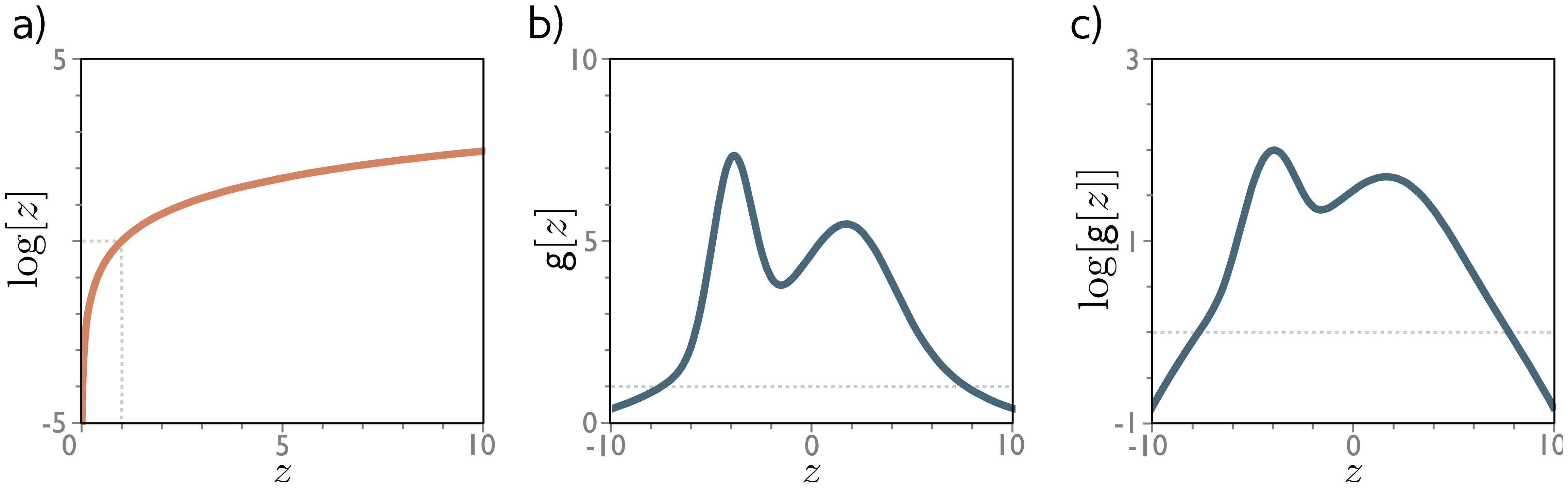
When we consider this probability as a function of the parameters ϕ, we call it a likelihood.

Problem:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}]) \right]$$

- The terms in this product might all be small
- The product might get so small that we can’t easily represent it

The log function is monotonic



Maximum of the logarithm of a function is in the same place as maximum of function

Maximum log likelihood

$$\begin{aligned}
\hat{\boldsymbol{\phi}} &= \underset{\boldsymbol{\phi}}{\mathrm{argmax}} \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}]) \right] \\
&= \underset{\boldsymbol{\phi}}{\mathrm{argmax}} \left[\log \left[\prod_{i=1}^{I} Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}]) \right] \right] \\
&= \underset{\boldsymbol{\phi}}{\mathrm{argmax}} \left[\sum_{i=1}^{I} \log \Big[Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}]) \Big] \right]
\end{aligned}$$

Now it’s a sum of terms, so doesn’t matter so much if the terms are small

Minimizing negative log likelihood

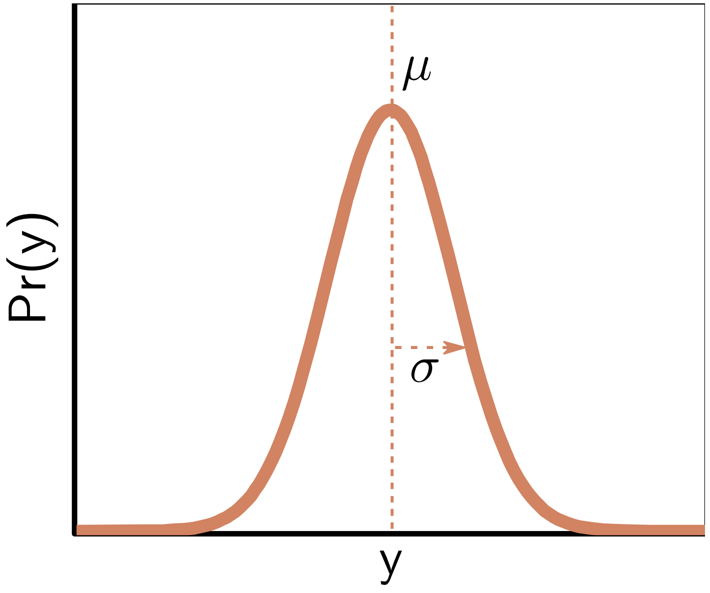
- By convention, we minimize things (i.e., a loss)

$$
\begin{aligned}
\hat{\boldsymbol{\phi}} &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\sum_{i=1}^{I} \log\Big[Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}]) \Big] \right] \\
&= \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \left[- \sum_{i=1}^{I} \log\Big[Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}]) \Big] \right] \\
&= \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \Big[\mathrm{L}[\boldsymbol{\phi}] \Big]
\end{aligned}
$$

Inference

- But now we predict a probability distribution
- We need an actual prediction (point estimate)
- Find the peak of the probability distribution (i.e., mean for normal)

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\mathrm{argmax}}\left[Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \phi])\right]$$



Recipe for loss functions

1. Choose a suitable probability distribution $Pr(\mathbf{y}|\boldsymbol{\theta})$ that is defined over the domain of the predictions $\mathbf{y}$ and has distribution parameters $\boldsymbol{\theta}$.
2. Set the machine learning model $\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}]$ to predict one or more of these parameters so $\boldsymbol{\theta} = \mathbf{f}[\mathbf{x}, \boldsymbol{\phi}]$ and $Pr(\mathbf{y}|\boldsymbol{\theta}) = Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}])$.
3. To train the model, find the network parameters $\hat{\boldsymbol{\phi}}$ that minimize the negative log-likelihood loss function over the training dataset pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[L[\boldsymbol{\phi}]\right] = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\left[-\sum_{i=1}^{I}\log\left[Pr(\mathbf{y}_i|\mathbf{f}[\mathbf{x}_i, \boldsymbol{\phi}])\right]\right]. \tag{5.7}$$

4. To perform inference for a new test example $\mathbf{x}$, return either the full distribution $Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \hat{\boldsymbol{\phi}}])$ or the maximum of this distribution.