Value of Information in Neural Networks

Roman V. Belavkin

Faculty of Science and Technology
Middlesex University, London NW4 4BT, UK

July 19, 2021
ACDL 2021

Value of Hartley's information

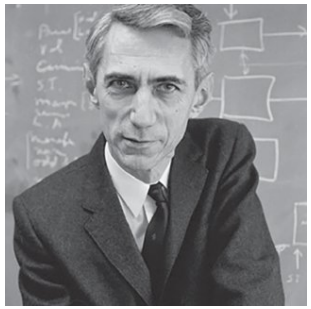
Feed-forward neural networks and VoI

Forecast and VoI

Values of Boltzmann's and Shannon's information

Solutions of VoI

Information and its Value

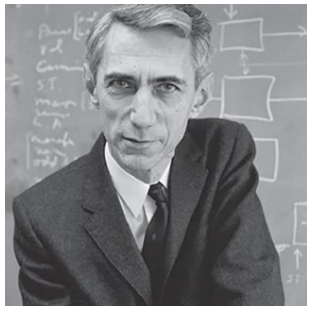


Claude Shannon

$$I_{xy} = \sum_{(x,y)} \left[\ln \frac{P(x \mid y)}{P(x)} \right] P(x,y)$$

(Shannon, 1948)

Information and its Value



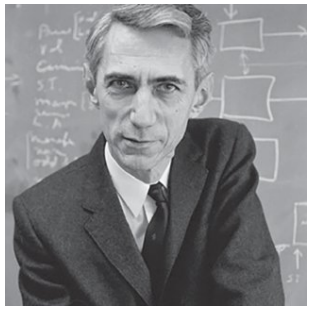
Claude Shannon

$$I_{xy} = \sum_{(x,y)} \left[\ln \frac{P(x \mid y)}{P(x)} \right] P(x,y)$$

(Shannon, 1948)

Ruslan Stratonovich

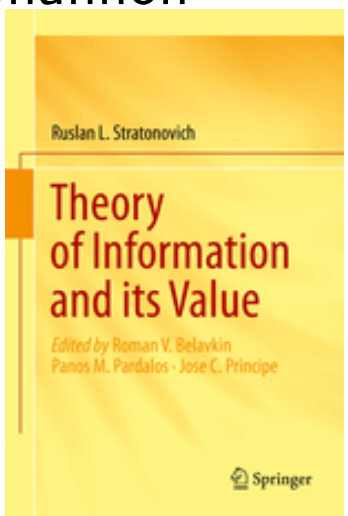
Information and its Value



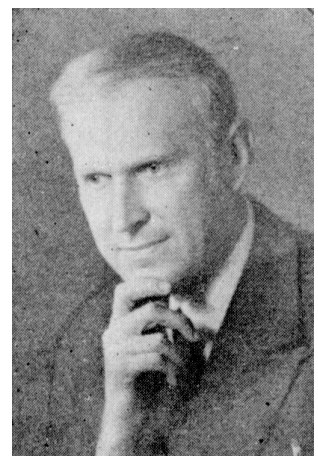
Claude Shannon

$$I_{xy} = \sum_{(x,y)} \left[\ln \frac{P(x \mid y)}{P(x)} \right] P(x,y)$$

(Shannon, 1948)

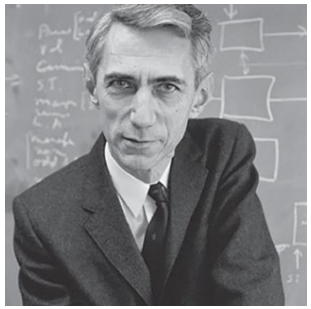


(Stratonovich, 1965, 1975):



Ruslan Stratonovich

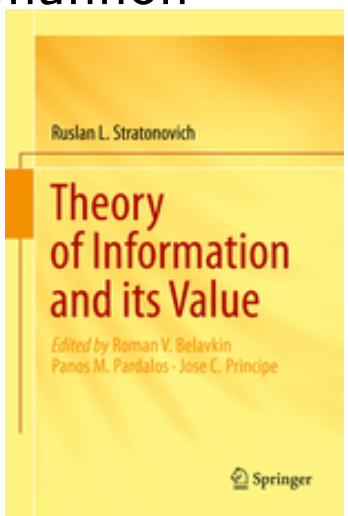
Information and its Value



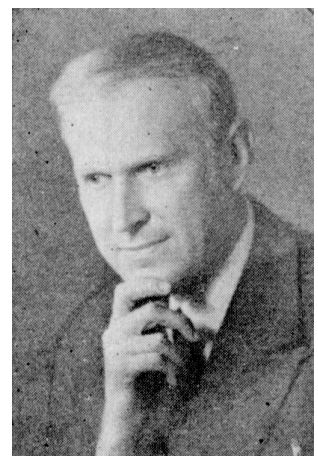
Claude Shannon

$$I_{xy} = \sum_{(x,y)} \left[\ln \frac{P(x \mid y)}{P(x)} \right] P(x,y)$$

(Shannon, 1948)



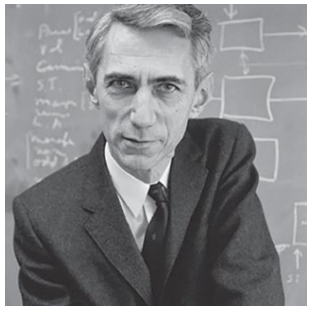
(Stratonovich, 1965, 1975):



Ruslan Stratonovich

Optimal algorithms for pattern recognition (Stratonovich, 1968)

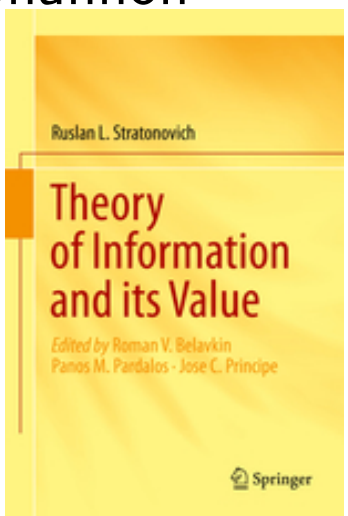
Information and its Value



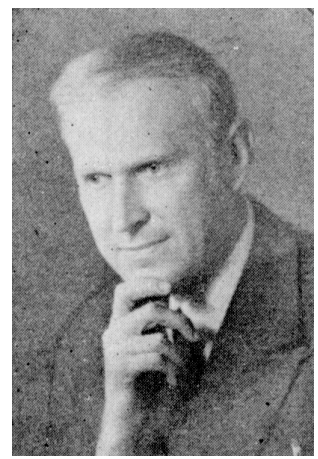
Claude Shannon

$$I_{xy} = \sum_{(x,y)} \left[\ln \frac{P(x \mid y)}{P(x)} \right] P(x,y)$$

(Shannon, 1948)



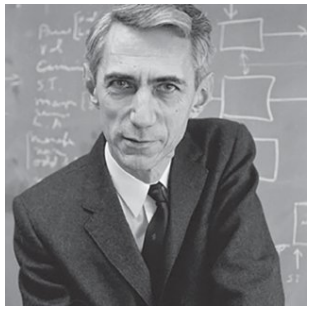
(Stratonovich, 1965, 1975):



Ruslan Stratonovich

Effectiveness of statistical methods in problems of synthesis of algorithms for function approximation (Stratonovich, 1969)

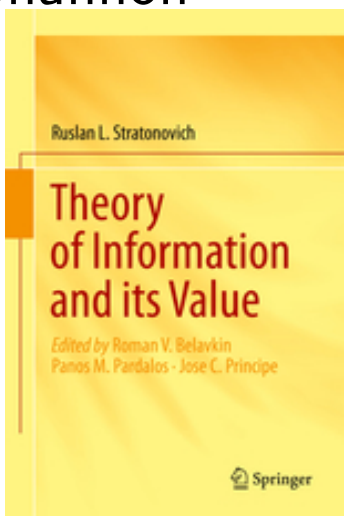
Information and its Value



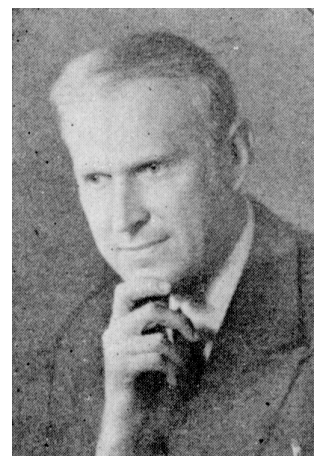
Claude Shannon

$$I_{xy} = \sum_{(x,y)} \left[\ln \frac{P(x \mid y)}{P(x)} \right] P(x,y)$$

(Shannon, 1948)



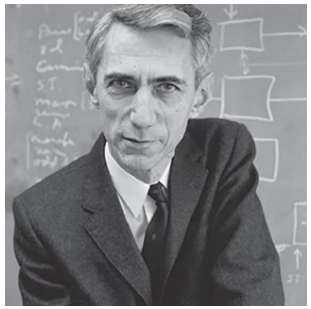
(Stratonovich, 1965, 1975):



Ruslan Stratonovich

Optimal expansion of functional subspace in algorithms for function and probability density approximation (Stratonovich, 1970b)

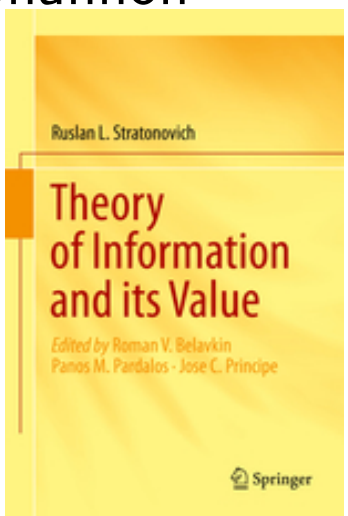
Information and its Value



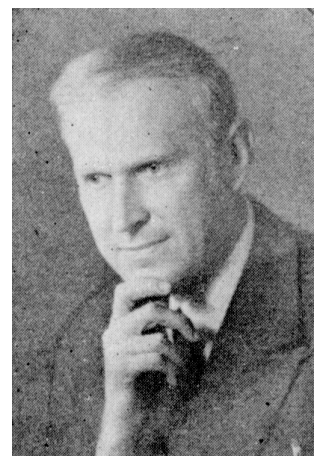
Claude Shannon

$$I_{xy} = \sum_{(x,y)} \left[\ln \frac{P(x \mid y)}{P(x)} \right] P(x,y)$$

(Shannon, 1948)



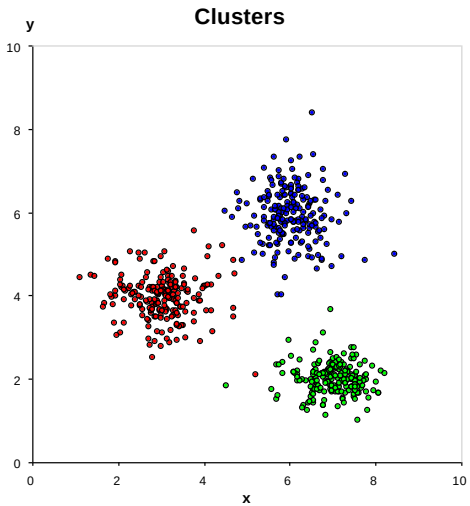
(Stratonovich, 1965, 1975):



Ruslan Stratonovich

Canonical recursive system of equations for optimal adaptive algorithms (Stratonovich, 1970a)

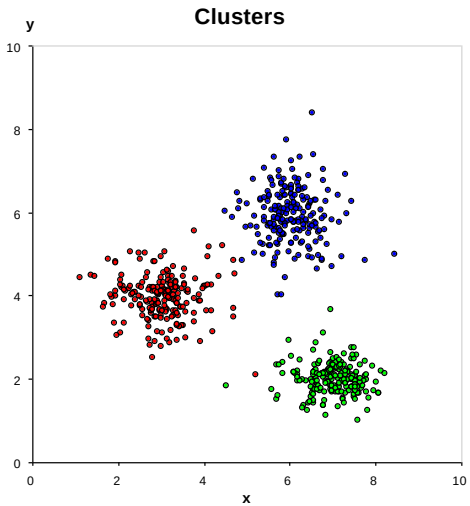
Information and Quality of Decisions



- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x c(x,z)\, P(x)$$

Information and Quality of Decisions



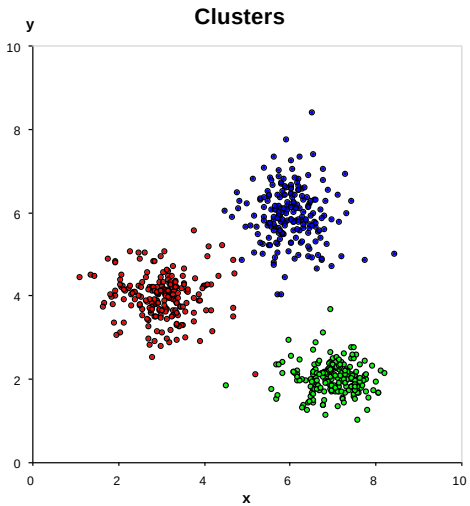
- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x c(x,z)\, P(x)$$

- Optimal $\hat{z}$ is defined by

$$\frac{\partial}{\partial z} \sum_x c(x,\hat{z})\, P(x) = 0$$

Information and Quality of Decisions



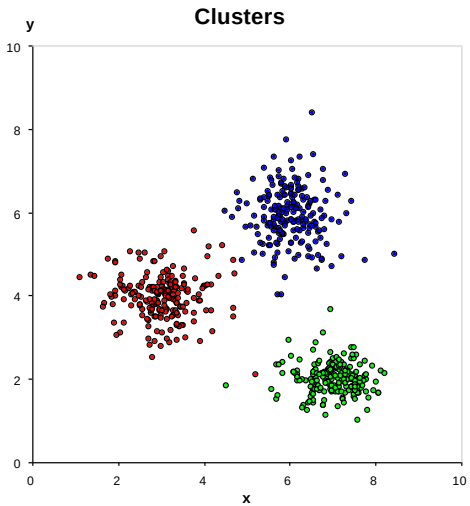
- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x c(x,z)\, P(x)$$

- Optimal $\hat{z}$ is defined by

$$\sum_x \frac{\partial}{\partial z} c(x,\hat{z})\, P(x) = 0$$

Information and Quality of Decisions



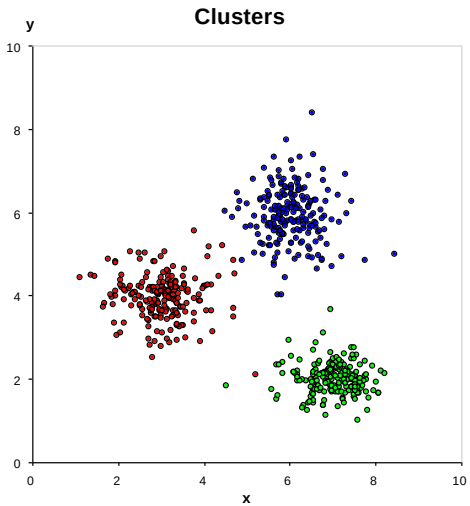
- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x \frac{1}{2}(x-z)^2\, P(x)$$

- Optimal $\hat{z}$ is defined by

$$\sum_x \frac{\partial}{\partial z}\frac{1}{2}(x-z)^2\, P(x) = 0$$

Information and Quality of Decisions



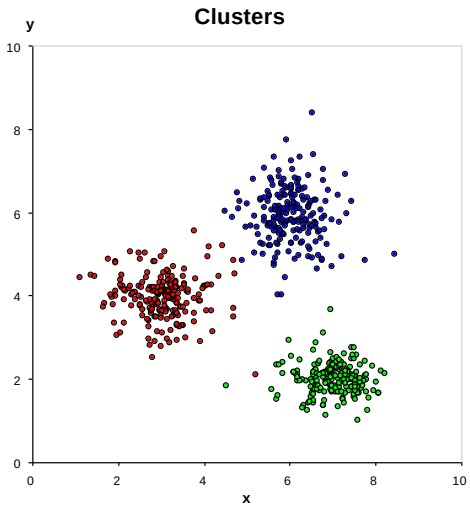
- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x \frac{1}{2}(x-z)^2\, P(x)$$

- Optimal $\hat{z}$ is defined by

$$\sum_x (x-\hat{z})\, P(x) = 0$$

Information and Quality of Decisions



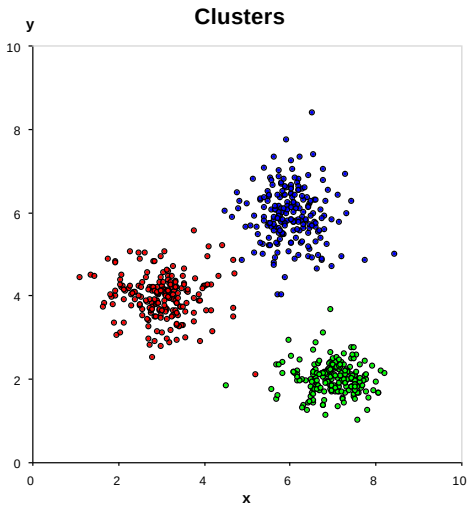
- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x \frac{1}{2}(x-z)^2\, P(x)$$

- Optimal $\hat{z}$ is defined by

$$\hat{z} = \sum_x x\, P(x)$$

Information and Quality of Decisions



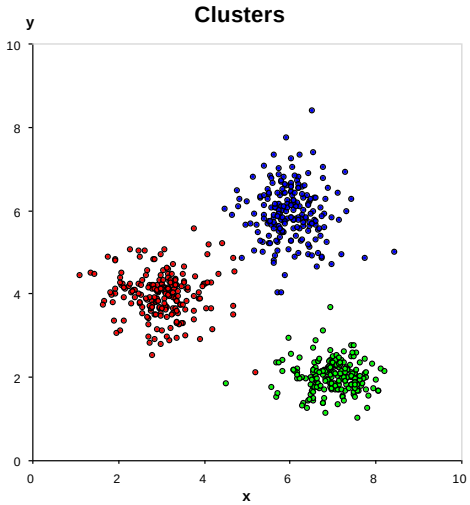
- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x \frac{1}{2}(x-z)^2 P(x)$$

- Optimal $\hat{z}$ is defined by

$$\hat{z} = \mathbb{E}\{x\}$$

Information and Quality of Decisions



- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x \frac{1}{2}(x-z)^2\, P(x)$$

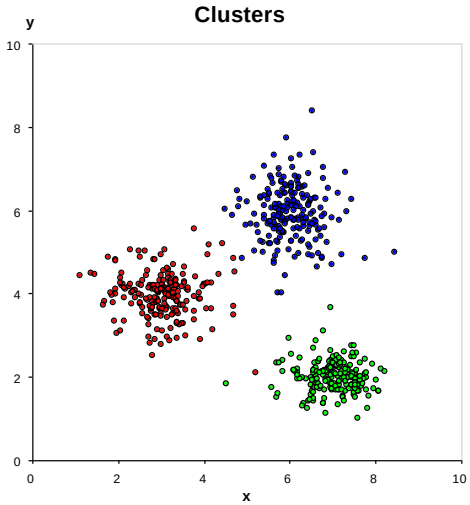
- Optimal $\hat{z}$ is defined by

$$\hat{z} = \mathbb{E}\{x\}$$

k-Means clustering

- Let us partion X into $k = 3$ subsets X_1, X_2, X_3

Information and Quality of Decisions



- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x \frac{1}{2}(x-z)^2 P(x)$$

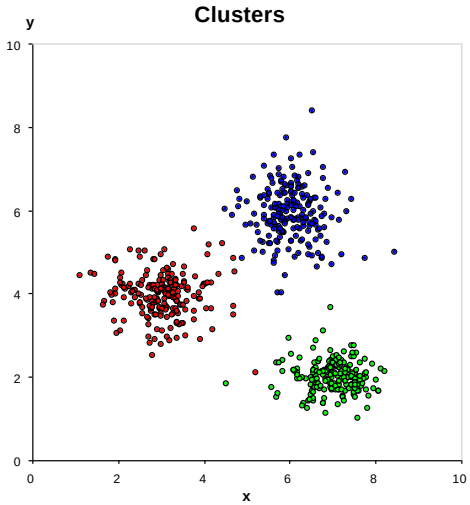
- Optimal $\hat{z}$ is defined by

$$\hat{z} = \mathbb{E}\{x\}$$

k-Means clustering

- Let us partion X into $k = 3$ subsets X_1, X_2, X_3
- This corresponds to some mapping $y(x)$ $(y : X \to \{y_1, y_2, y_3\})$

Information and Quality of Decisions



- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x \frac{1}{2}(x-z)^2\, P(x)$$

- Optimal $\hat{z}$ is defined by

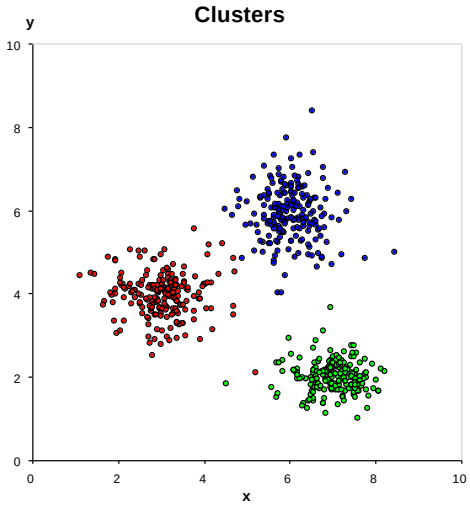
$$\hat{z} = \mathbb{E}\{x\}$$

k-Means clustering

- Let us partion X into $k = 3$ subsets X_1, X_2, X_3
- This corresponds to some mapping $y(x)$ $(y : X \to \{y_1, y_2, y_3\})$
- Find z_1, z_2, z_3 minimizing

$$\sum_y \mathbb{E}_{P(x|y)}\{c(x,z) \mid y\}\, P(y)$$

Information and Quality of Decisions



- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x \frac{1}{2}(x-z)^2\, P(x)$$

- Optimal $\hat{z}$ is defined by

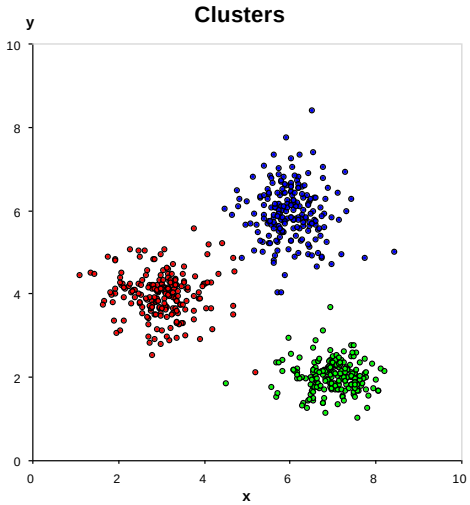
$$\hat{z} = \mathbb{E}\{x\}$$

k-Means clustering

- Let us partion X into $k = 3$ subsets X_1, X_2, X_3
- This corresponds to some mapping $y(x)$ $(y : X \to \{y_1, y_2, y_3\})$
- Find z_1, z_2, z_3 minimizing

$$\sum_y \mathbb{E}_{P(x|y)} \left\{ \frac{1}{2}(x-z)^2 \mid y \right\} P(y)\,, \qquad \hat{z}(y) = \sum_x x\, P(x|y)$$

Information and Quality of Decisions



- $P(x)$, $c : X \times X \to \mathbb{R}$, find $z \in X$ minimizing

$$\mathbb{E}_P\{c(x,z)\} = \sum_x \frac{1}{2}(x-z)^2\, P(x)$$

- Optimal $\hat{z}$ is defined by

$$\hat{z} = \mathbb{E}\{x\}$$

k-Means clustering

- Let us partion X into $k = 3$ subsets X_1, X_2, X_3
- This corresponds to some mapping $y(x)$ $(y : X \to \{y_1, y_2, y_3\})$
- Find z_1, z_2, z_3 minimizing

$$\sum_y \mathbb{E}_{P(x|y)} \left\{ \frac{1}{2}(x-z)^2 \mid y \right\} P(y)\,, \qquad \hat{z}(y) = \mathbb{E}\{x \mid y\}$$

Value of Hartley's information

- $H = \ln k$ is called *Hartley entropy*

Value of Hartley's information

- $H = \ln k$ is called *Hartley entropy*
- Define the following quantities:

$$R(0) := \inf_{z} \mathbb{E}_{P(x)}\{c(x, z)\}$$

Value of Hartley's information

- $H = \ln k$ is called *Hartley entropy*
- Define the following quantities:

$$R(0) := \inf_{z} \mathbb{E}_{P(x)}\{c(x,z)\}$$

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)}\{c(x,z) \mid y\} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

Value of Hartley's information

- $H = \ln k$ is called *Hartley entropy*
- Define the following quantities:

$$R(0) := \inf_{z} \mathbb{E}_{P(x)}\{c(x,z)\}$$

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)}\{c(x,z) \mid y\} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

- The *value of Harley information* (Stratonovich, 1965):

$$V(H) := R(0) - R(H)$$

Value of Hartley's information

- $H = \ln k$ is called *Hartley entropy*
- Define the following quantities:

$$U(0) := \sup_{z} \mathbb{E}_{P(x)}\{u(x,z)\}$$

$$U(H) := \sup_{y(x)} \mathbb{E}_{P(y)} \left\{ \sup_{z(y)} \mathbb{E}_{P(x|y)}\{u(x,z) \mid y\} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

- The *value of Harley information* (Stratonovich, 1965):

$$V(H) := U(H) - U(0)$$

Value of Hartley's information

- $H = \ln k$ is called *Hartley entropy*
- Define the following quantities:

$$U(0) := \sup_{z} \mathbb{E}_{P(x)}\{u(x,z)\}$$

$$U(H) := \sup_{y(x)} \mathbb{E}_{P(y)} \left\{ \sup_{z(y)} \mathbb{E}_{P(x|y)}\{u(x,z) \mid y\} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

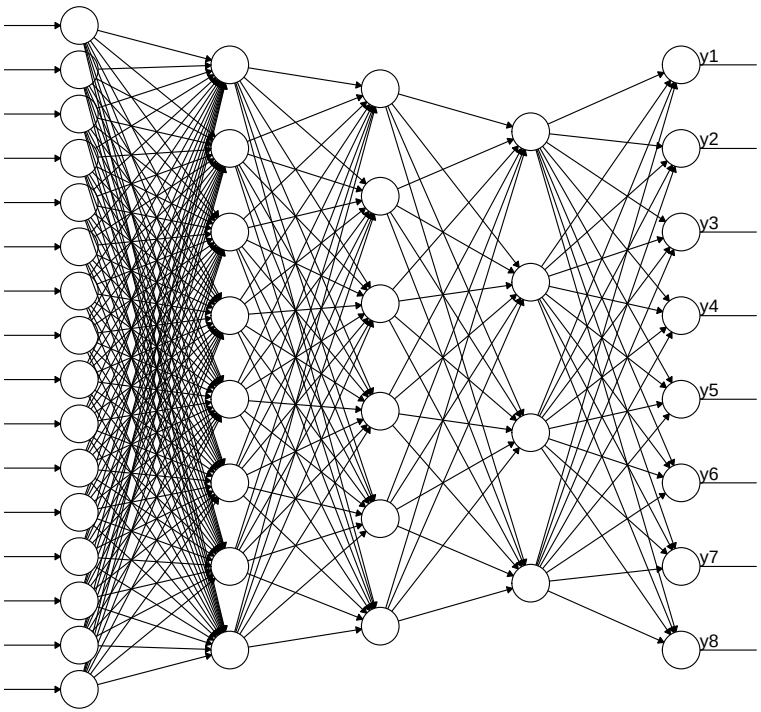
- The *value of Harley information* (Stratonovich, 1965):

$$V(H) := U(H) - U(0)$$

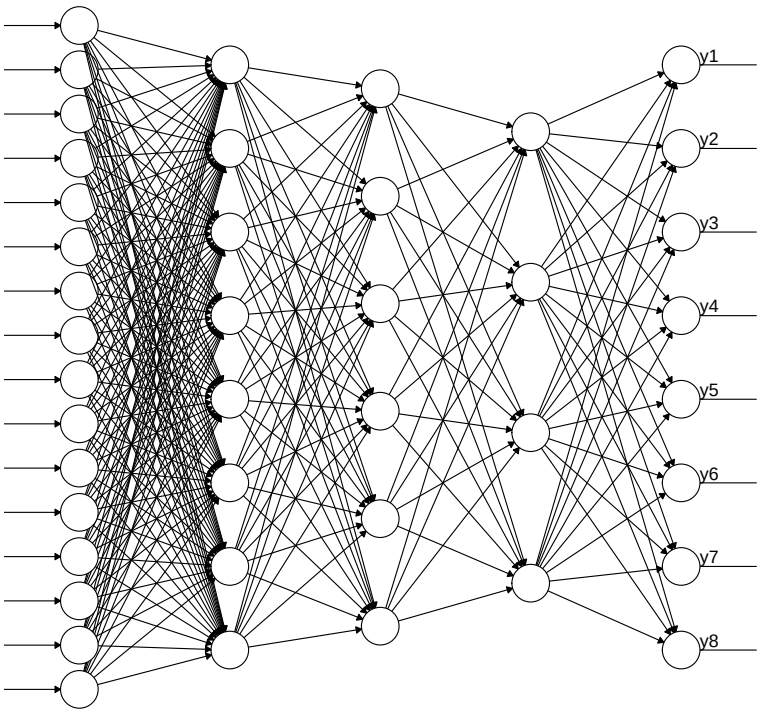
Remark

We are looking for optimal function $z(x) = z \circ y(x)$ *subject to* $|Z| \leq k$.

Feed-forward neural networks



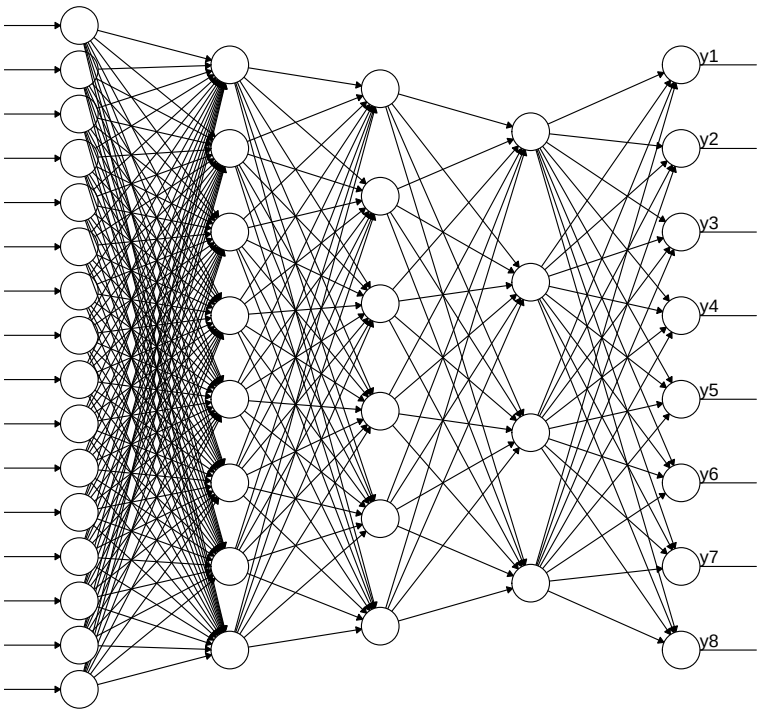
Feed-forward neural networks



Supervised training

1. Initialise the weights w_{ij} (e.g. at random).
2. Repeat

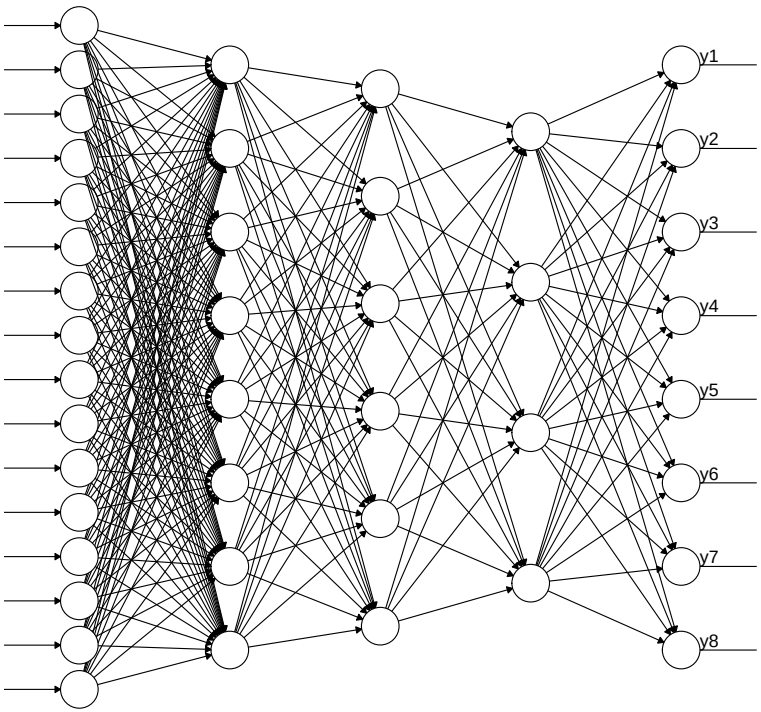
Feed-forward neural networks



Supervised training

1. Initialise the weights w_{ij} (e.g. at random).
2. Repeat
   1. Feed the network with an input y from a training set.

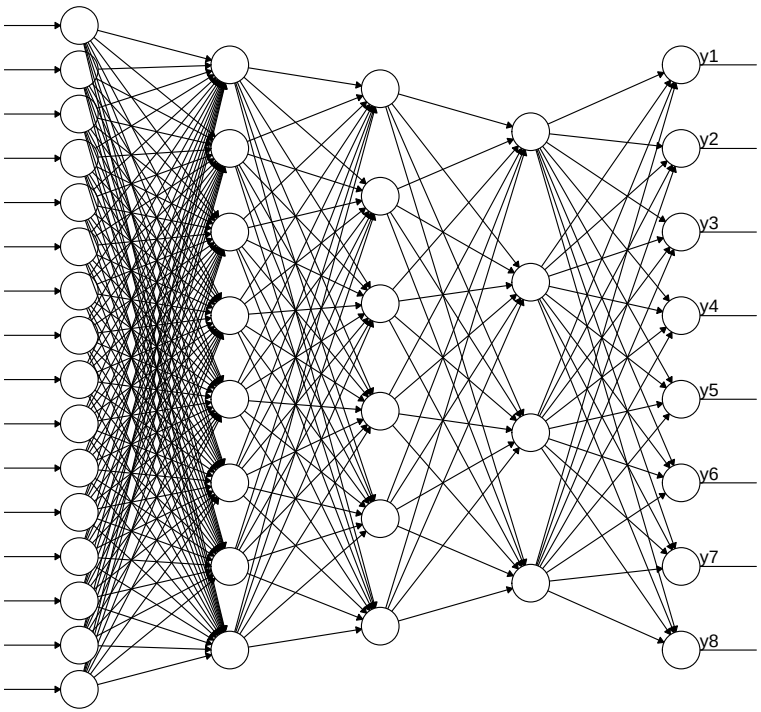
Feed-forward neural networks



Supervised training

1. Initialise the weights w_{ij} (e.g. at random).
2. Repeat
   1. Feed the network with an input y from a training set.
   2. Compute the network's output $z(y)$.

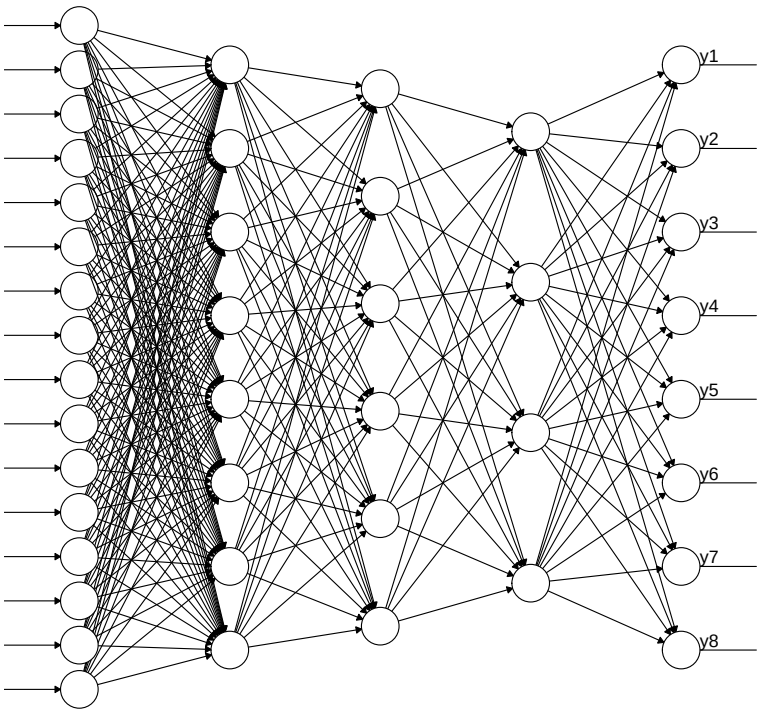
Feed-forward neural networks



Supervised training

1. Initialise the weights w_{ij} (e.g. at random).
2. Repeat
   1. Feed the network with an input y from a training set.
   2. Compute the network's output $z(y)$.
   3. Compute the error $x - z(y)$.

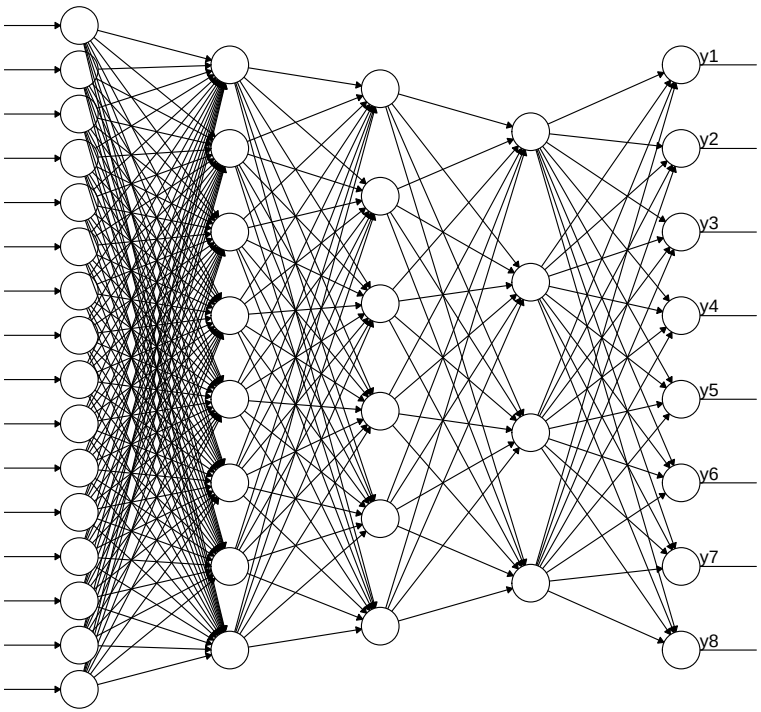
Feed-forward neural networks



Supervised training

1. Initialise the weights w_{ij} (e.g. at random).
2. Repeat
   1. Feed the network with an input y from a training set.
   2. Compute the network's output $z(y)$.
   3. Compute the error $x - z(y)$.
   4. Change the weights w_{ij} to minimise the error's cost $c(x, z(y))$.

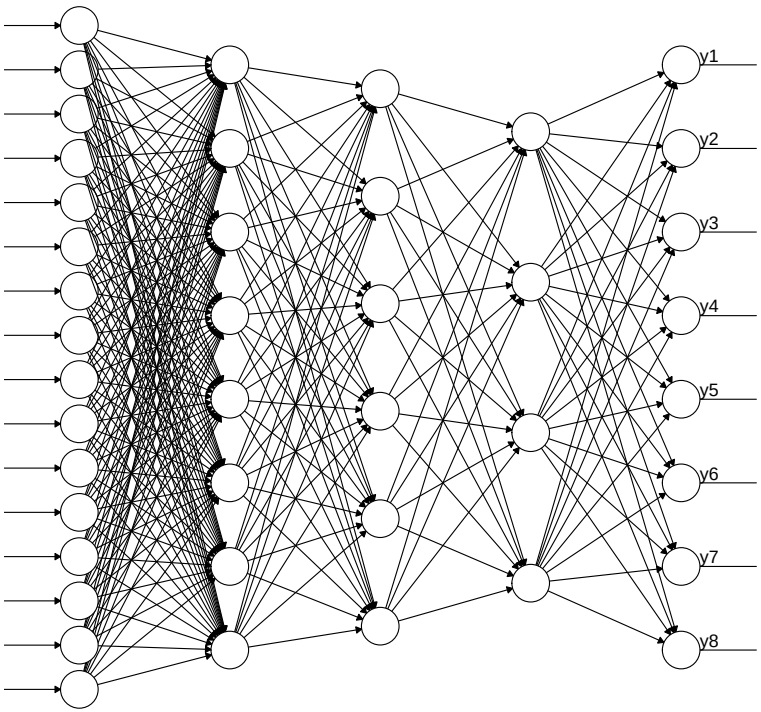
Feed-forward neural networks



Supervised training

1. Initialise the weights w_{ij} (e.g. at random).
2. Repeat
   1. Feed the network with an input y from a training set.
   2. Compute the network's output $z(y)$.
   3. Compute the error $x - z(y)$.
   4. Change the weights w_{ij} to minimise the error's cost $c(x, z(y))$.
3. Until the mean cost is small.

Feed-forward neural networks



Supervised training

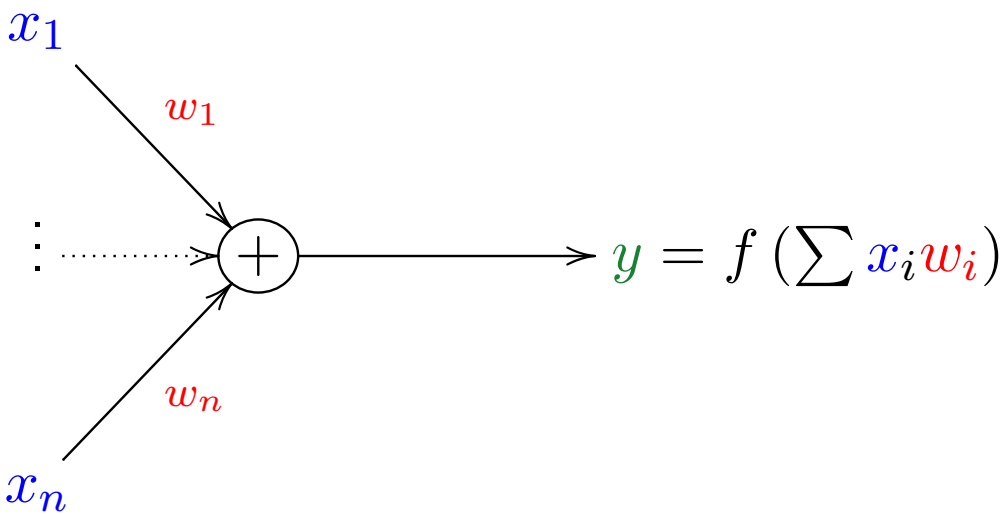
1. Initialise the weights w_{ij} (e.g. at random).
2. Repeat
   1. Feed the network with an input y from a training set.
   2. Compute the network's output $z(y)$.
   3. Compute the error $x - z(y)$.
   4. Change the weights w_{ij} to minimise the error's cost $c(x, z(y))$.
3. Until the mean cost is small.

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)} \{ c(x, z) \mid y \} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

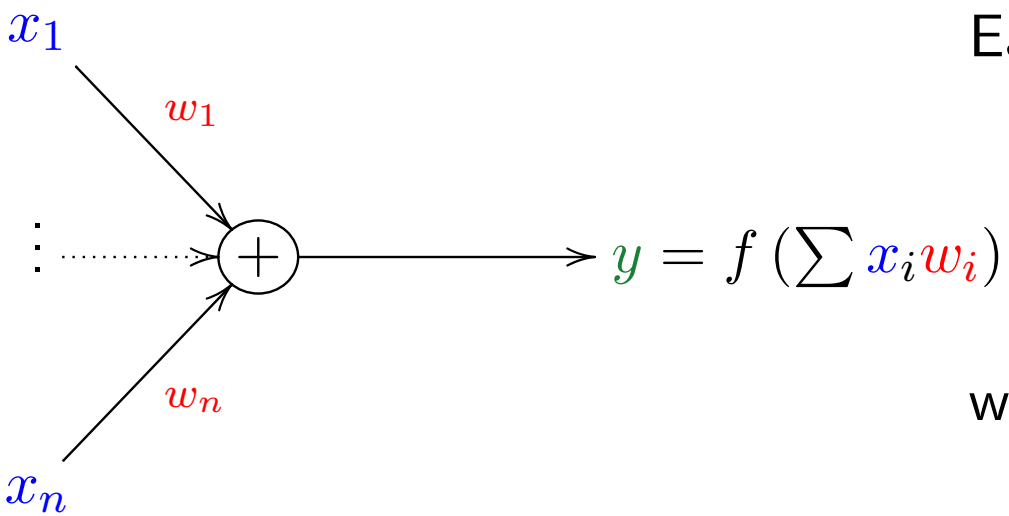
Integrate and Fire Neurons

McCulloch and Pitts (1943)



Integrate and Fire Neurons

McCulloch and Pitts (1943)



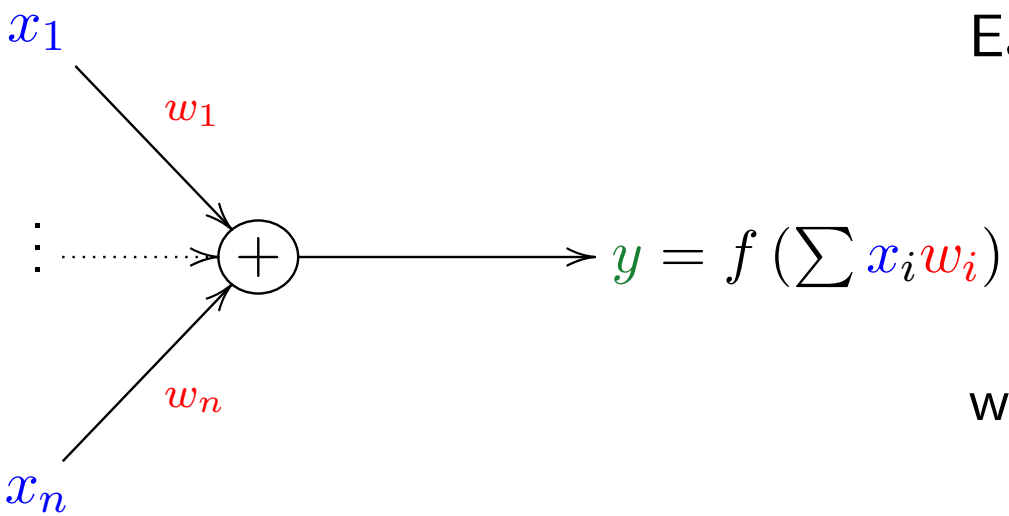
Each node computes a weighted sum:

$$v = \sum_{i=1}^{n} w_i x_i$$

which reminds us a linear model.

Integrate and Fire Neurons

McCulloch and Pitts (1943)



Each node computes a weighted sum:

$$v = \sum_{i=1}^{n} w_i x_i$$

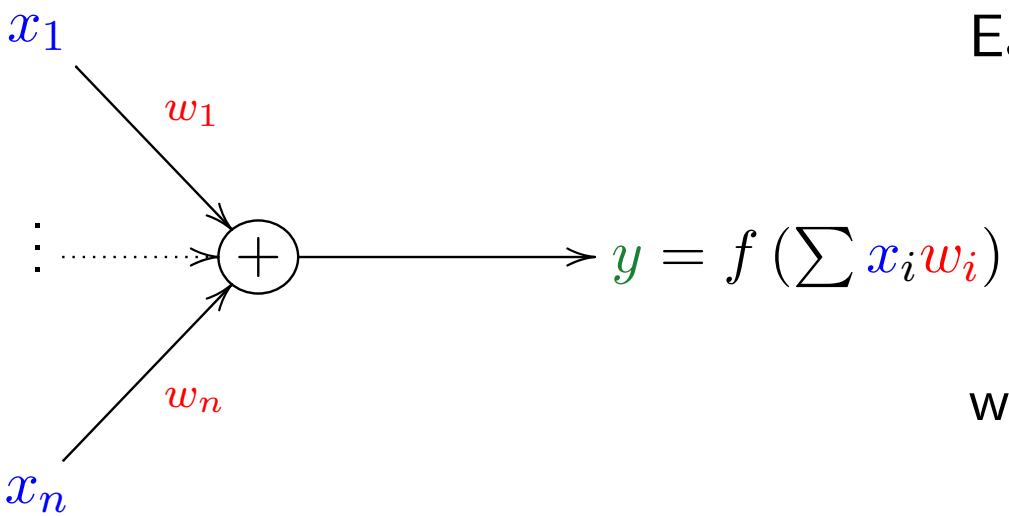
which reminds us a linear model.

Activation function:

$$f(v) = \begin{cases} 1 & \text{if } v \geq a \\ 0 & \text{otherwise} \end{cases}$$

Integrate and Fire Neurons

McCulloch and Pitts (1943)



Each node computes a weighted sum:

$$v = \sum_{i=1}^{n} w_i x_i$$

which reminds us a linear model.

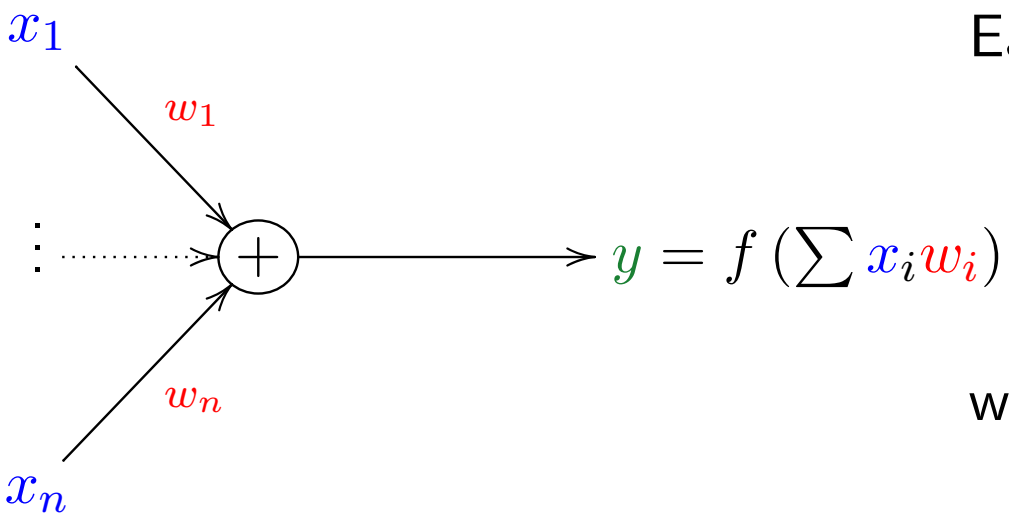
Activation function:

$$f(v) = \begin{cases} 1 & \text{if } v \geq a \\ 0 & \text{otherwise} \end{cases}$$

- Each node partitions the input space into two halves.

Integrate and Fire Neurons

McCulloch and Pitts (1943)



Each node computes a weighted sum:

$$v = \sum_{i=1}^{n} w_i x_i$$

which reminds us a linear model.

Activation function:

$$f(v) = \begin{cases} 1 & \text{if } v \geq a \\ 0 & \text{otherwise} \end{cases}$$

- Each node partitions the input space into two halves.
- With several nodes the perceptron acts as a classifier.

Single layer perceptrons

- The weighted sum for 2 inputs defines a line on (x_1, x_2)-plane:

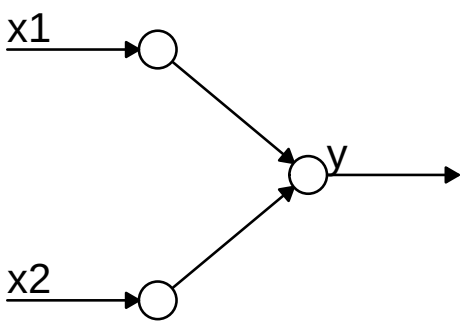
$$a = w_1 x_1 + w_2 x_2$$

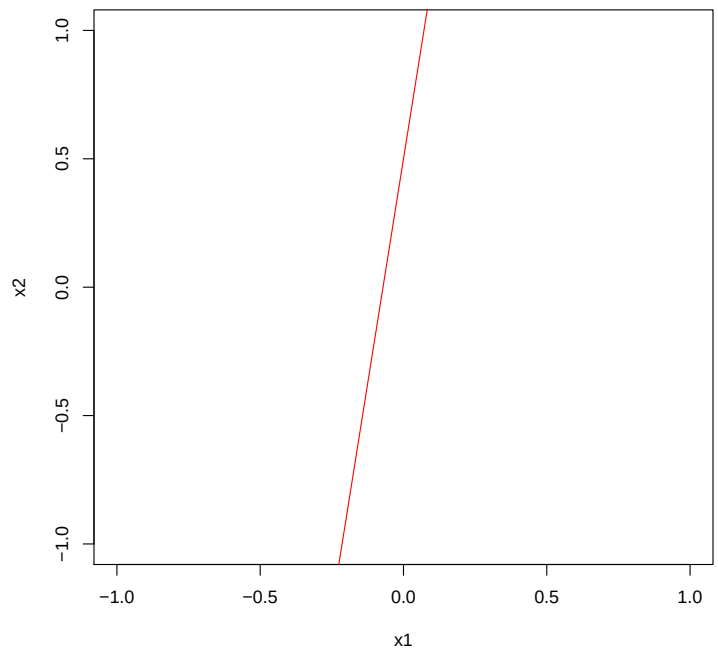
Single layer perceptrons

- The weighted sum for 2 inputs defines a line on (x_1, x_2)-plane:

$$a = w_1 x_1 + w_2 x_2 \quad \Longrightarrow \quad x_2 = \frac{a}{w_2} - \frac{w_1}{w_2} x_1$$

- The line splits the plane into 2 halfspaces.



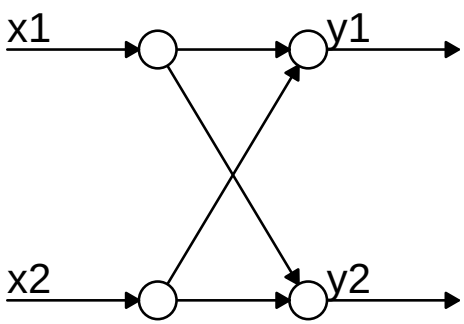


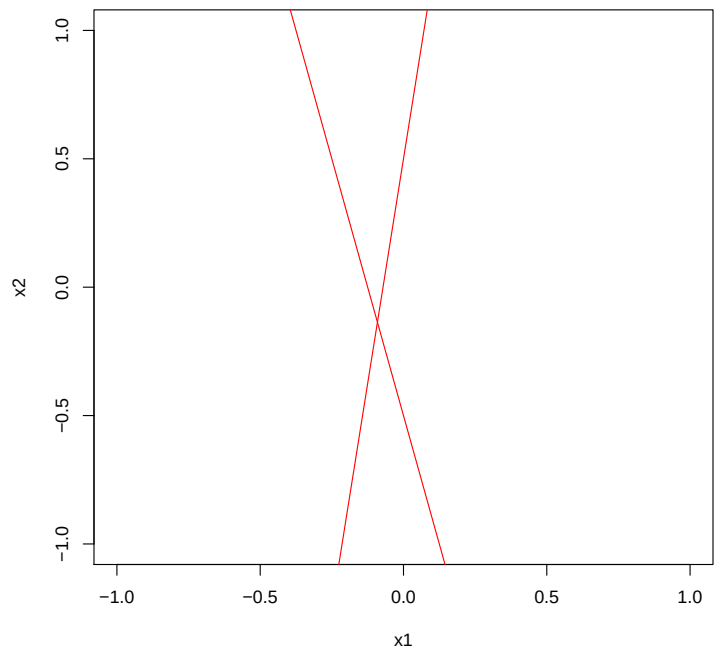
Single layer perceptrons

- The weighted sum for 2 inputs defines a line on (x_1, x_2)-plane:

$$a = w_1 x_1 + w_2 x_2 \quad \Longrightarrow \quad x_2 = \frac{a}{w_2} - \frac{w_1}{w_2} x_1$$

- The line splits the plane into 2 halfspaces.



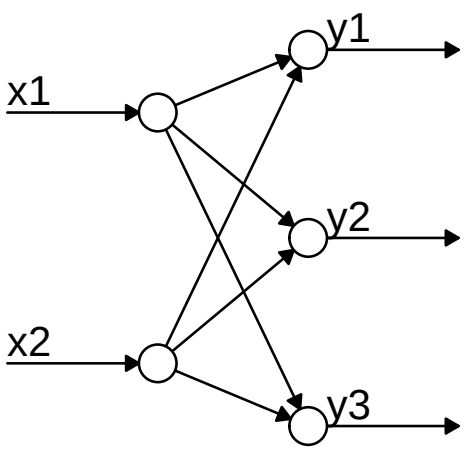


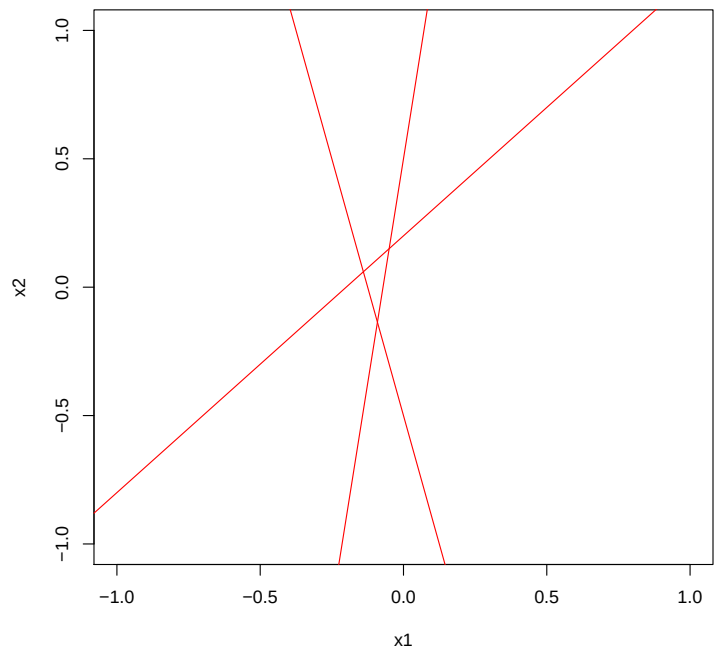
Single layer perceptrons

- The weighted sum for 2 inputs defines a line on (x_1, x_2)-plane:

$$a = w_1 x_1 + w_2 x_2 \quad \Longrightarrow \quad x_2 = \frac{a}{w_2} - \frac{w_1}{w_2} x_1$$

- The line splits the plane into 2 halfspaces.



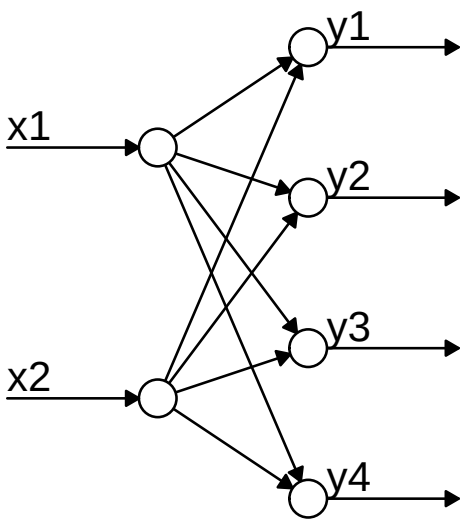


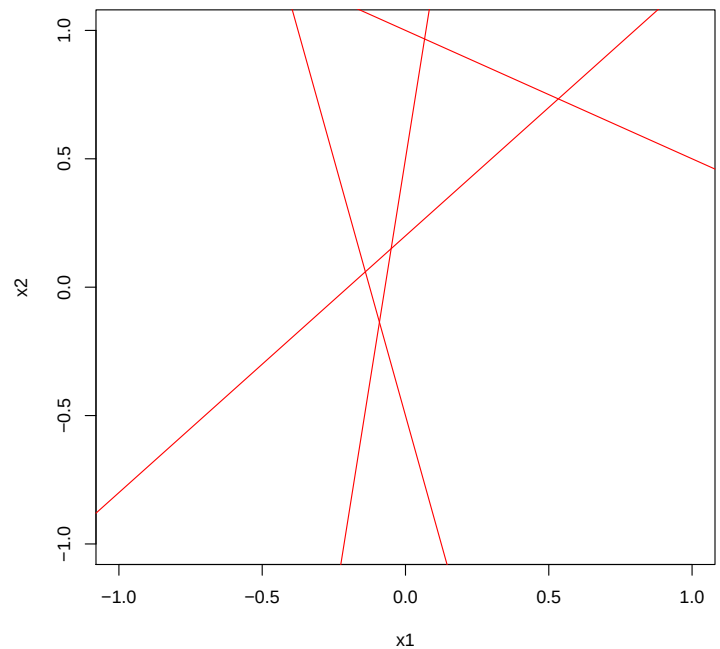
Single layer perceptrons

- The weighted sum for 2 inputs defines a line on (x_1, x_2)-plane:

$$a = w_1 x_1 + w_2 x_2 \quad \Longrightarrow \quad x_2 = \frac{a}{w_2} - \frac{w_1}{w_2} x_1$$

- The line splits the plane into 2 halfspaces.



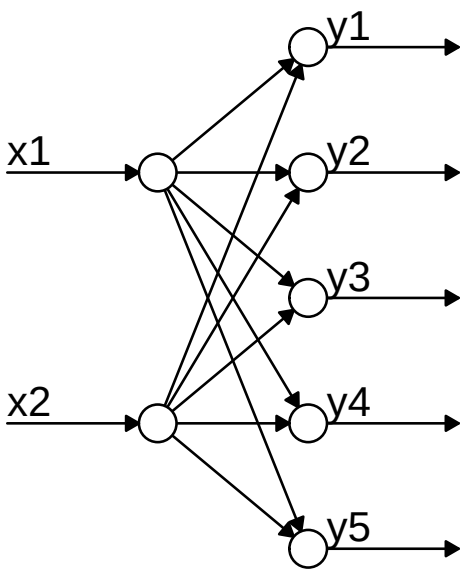


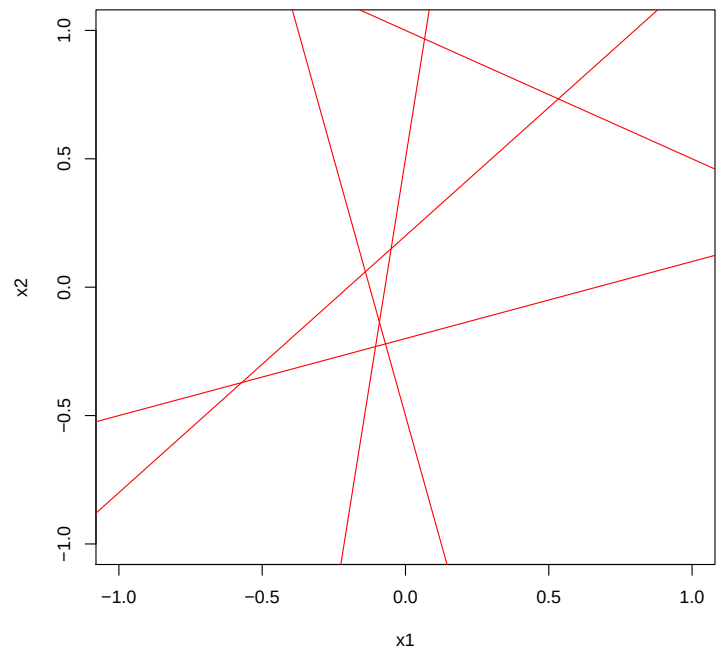
Single layer perceptrons

- The weighted sum for 2 inputs defines a line on (x_1, x_2)-plane:

$$a = w_1 x_1 + w_2 x_2 \quad \Longrightarrow \quad x_2 = \frac{a}{w_2} - \frac{w_1}{w_2} x_1$$

- The line splits the plane into 2 halfspaces.



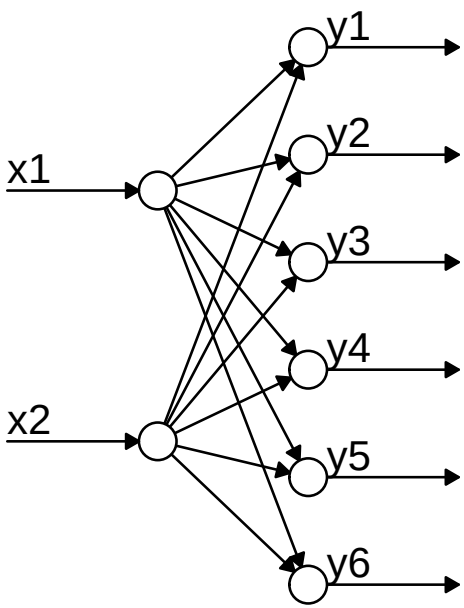


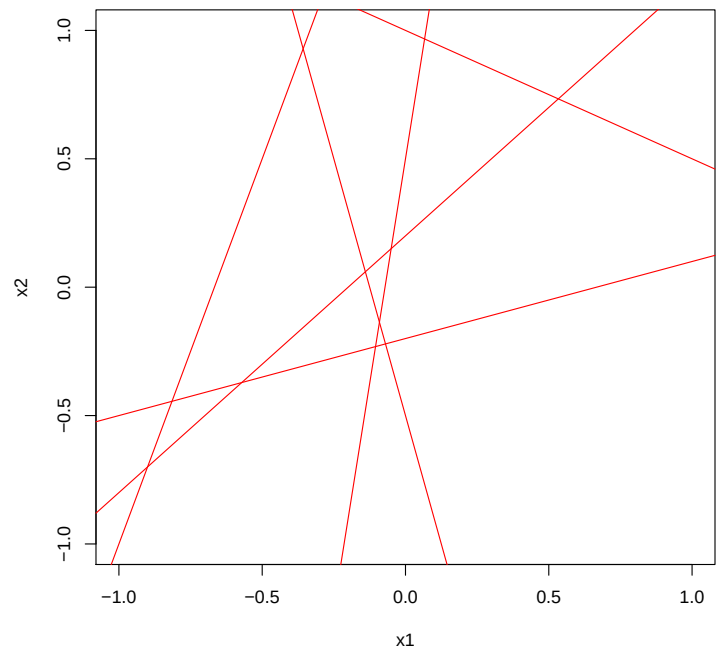
Single layer perceptrons

- The weighted sum for 2 inputs defines a line on (x_1, x_2)-plane:

$$a = w_1 x_1 + w_2 x_2 \quad \Longrightarrow \quad x_2 = \frac{a}{w_2} - \frac{w_1}{w_2} x_1$$

- The line splits the plane into 2 halfspaces.



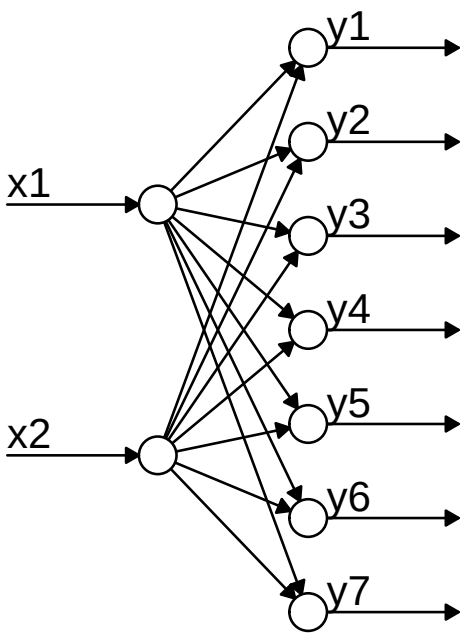


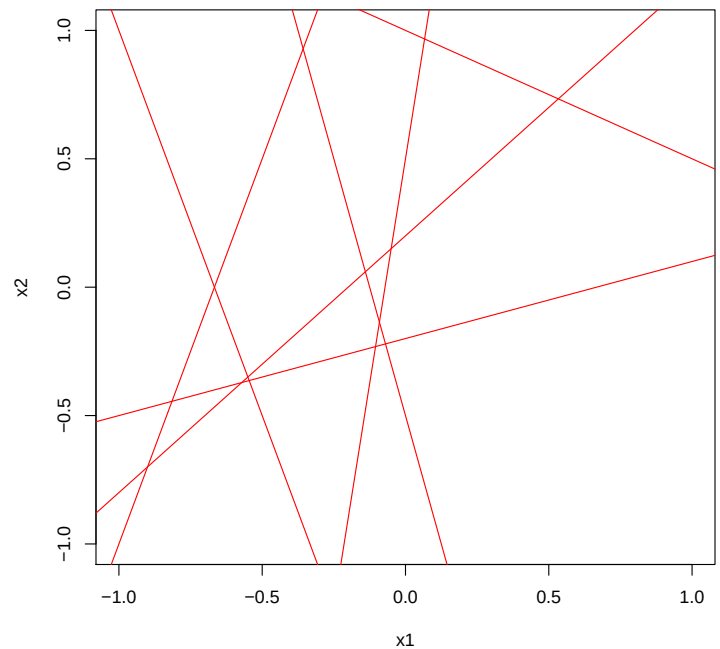
Single layer perceptrons

- The weighted sum for 2 inputs defines a line on (x_1, x_2)-plane:

$$a = w_1 x_1 + w_2 x_2 \quad \Longrightarrow \quad x_2 = \frac{a}{w_2} - \frac{w_1}{w_2} x_1$$

- The line splits the plane into 2 halfspaces.



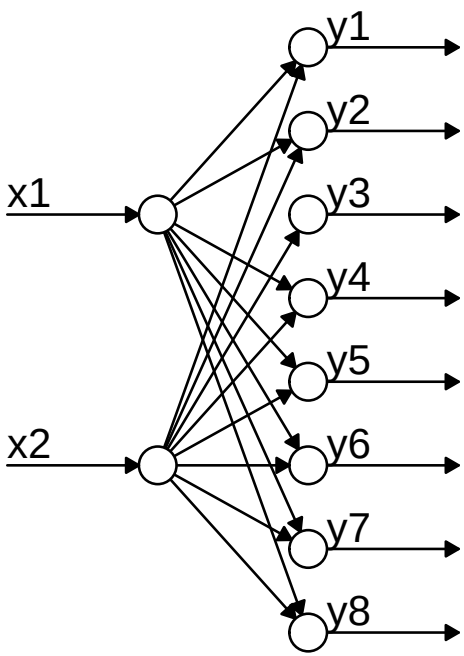


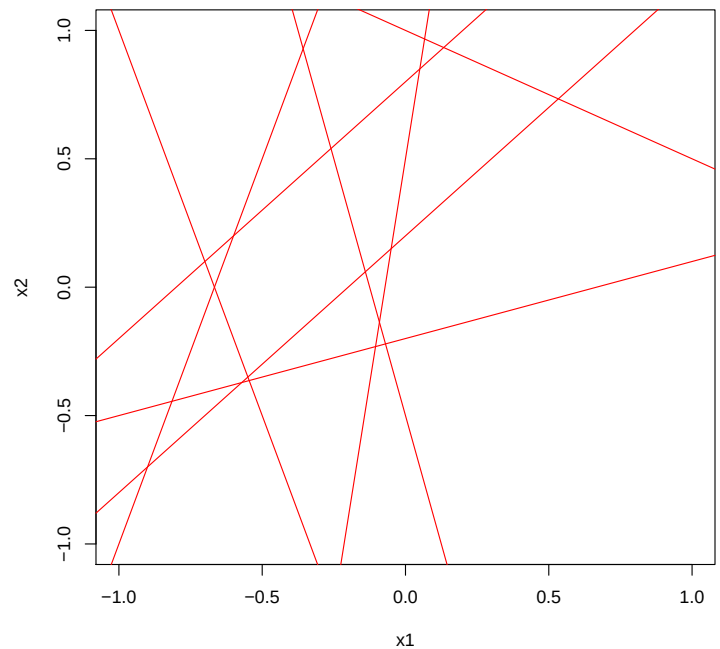
Single layer perceptrons

- The weighted sum for 2 inputs defines a line on (x_1, x_2)-plane:

$$a = w_1 x_1 + w_2 x_2 \quad \Longrightarrow \quad x_2 = \frac{a}{w_2} - \frac{w_1}{w_2} x_1$$

- The line splits the plane into 2 halfspaces.



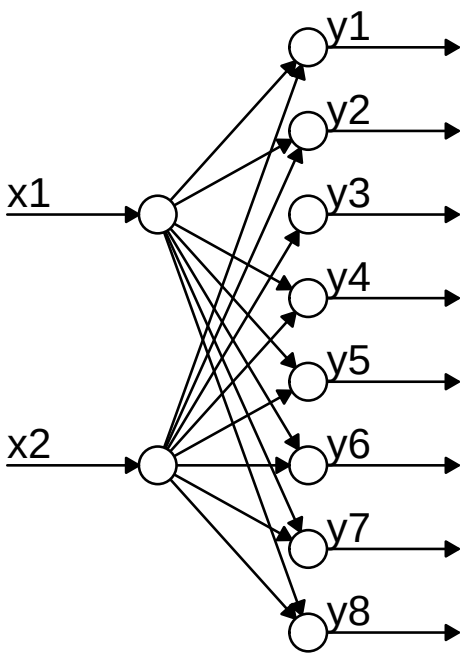


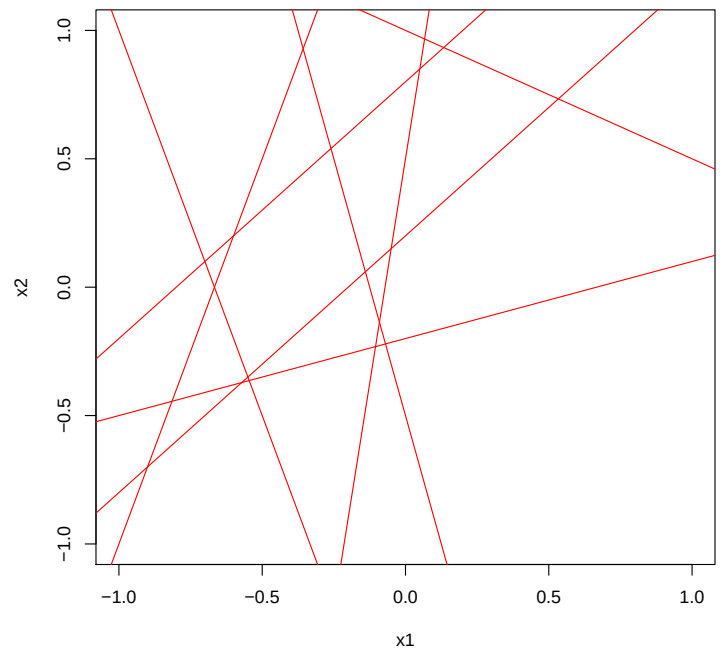
Single layer perceptrons

- The weighted sum for 2 inputs defines a line on (x_1, x_2)-plane:

$$a = w_1 x_1 + w_2 x_2 \quad \Longrightarrow \quad x_2 = \frac{a}{w_2} - \frac{w_1}{w_2} x_1$$

- The line splits the plane into 2 halfspaces.



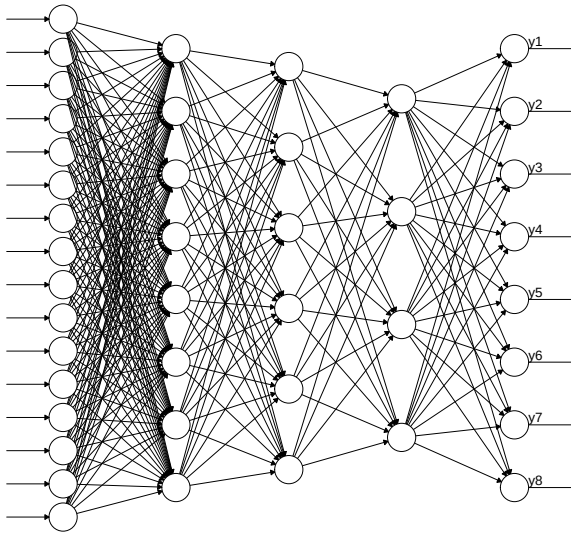


- k nodes partition the space into 2^k subsets.

Learning neural models and VoI

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)} \{ c(x,z) \mid y \} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

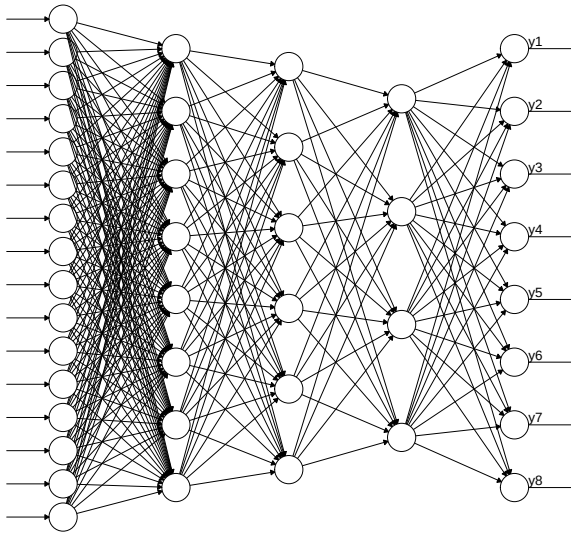


- Some features of deep NNs training including:

Learning neural models and VoI

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)} \{ c(x, z) \mid y \} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

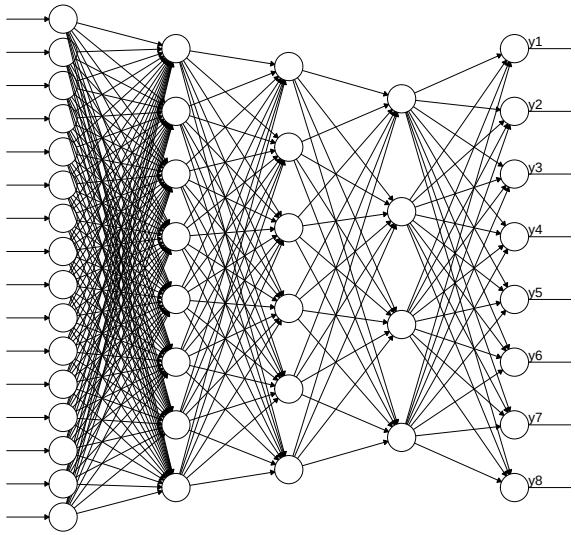


- Some features of deep NNs training including:
  - Partial connectivity.

Learning neural models and VoI

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)} \{ c(x, z) \mid y \} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

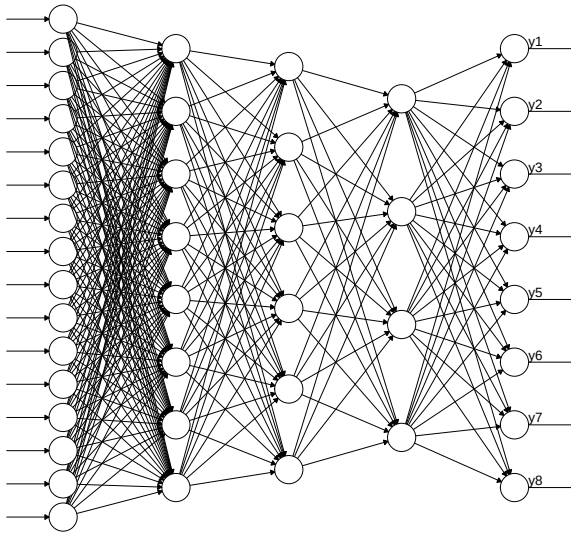


- Some features of deep NNs training including:
  - Partial connectivity.
  - Randomization techniques, such as dropout.

Learning neural models and VoI

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)} \{ c(x,z) \mid y \} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

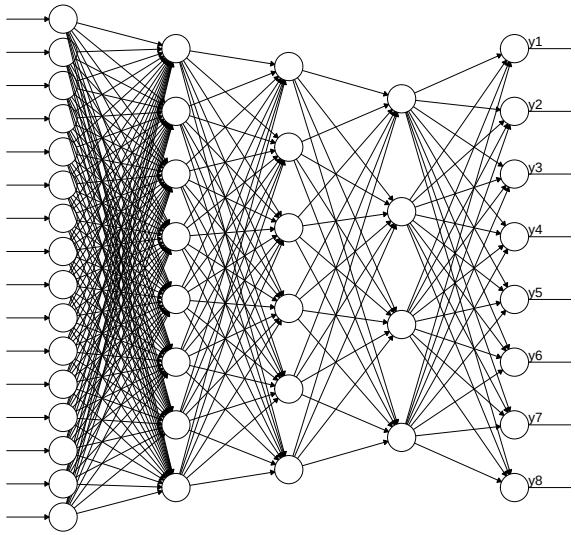


- Some features of deep NNs training including:
  - Partial connectivity.
  - Randomization techniques, such as dropout.
  - Specialized layers (e.g. convolution, $\max$-pooling).

Learning neural models and VoI

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)} \{ c(x,z) \mid y \} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

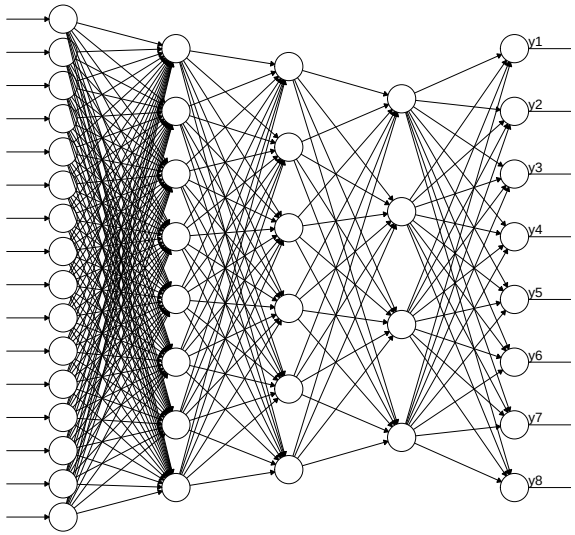


- Some features of deep NNs training including:
  - Partial connectivity.
  - Randomization techniques, such as dropout.
  - Specialized layers (e.g. convolution, $\max$-pooling).
  - Weight regularization.

Learning neural models and VoI

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)} \{ c(x,z) \mid y \} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

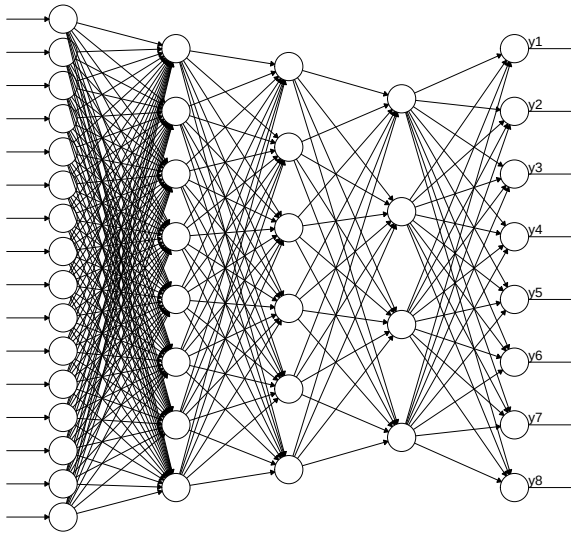


- Some features of deep NNs training including:
  - Partial connectivity.
  - Randomization techniques, such as dropout.
  - Specialized layers (e.g. convolution, $\max$-pooling).
  - Weight regularization.
  - Sublinear activation functions (e.g. ReLU $= \max\{0, v\}$).

Learning neural models and VoI

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)} \{ c(x,z) \mid y \} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$

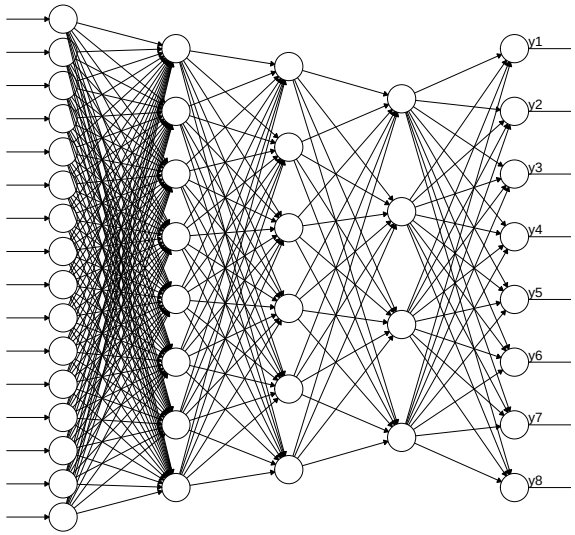


- Some features of deep NNs training including:
  - Partial connectivity.
  - Randomization techniques, such as dropout.
  - Specialized layers (e.g. convolution, $\max$-pooling).
  - Weight regularization.
  - Sublinear activation functions (e.g. ReLU $= \max\{0, v\}$).
  - Crossentropy as cost function.

Learning neural models and VoI

$$R(H) := \inf_{y(x)} \mathbb{E}_{P(y)} \left\{ \inf_{z(y)} \mathbb{E}_{P(x|y)} \{ c(x,z) \mid y \} \right\}$$

$$\text{Subject to } H = \ln |Y| \leq \ln k$$



- Some features of deep NNs training including:
  - Partial connectivity.
  - Randomization techniques, such as dropout.
  - Specialized layers (e.g. convolution, $\max$-pooling).
  - Weight regularization.
  - Sublinear activation functions (e.g. ReLU $= \max\{0, v\}$).
  - Crossentropy as cost function.
- Learning weights in neural networks can be seens as a process of maximization of the value of information.

Linear trend and forecast

Table: TSLA.Close price Aug 2020

Date	Price(t)	
2020-08-03	297.000	
2020-08-04	297.400	
2020-08-05	297.004	
2020-08-06	297.916	
2020-08-07		?

Linear trend and forecast

Table: TSLA.Close price Aug 2020

Date	Price(t)	
2020-08-03	297.000	
2020-08-04	297.400	
2020-08-05	297.004	
2020-08-06	297.916	
2020-08-07	?	?

- We can compute linear trend and use it for forecast.
- We can use correlations between prices on different days.
- Can we predict tomorrow's price using the price today?

Linear trend and forecast

Table: TSLA.Close price Aug 2020

Date	Price(t)	Price(t+1)
2020-08-03	297.000	297.400
2020-08-04	297.400	297.004
2020-08-05	297.004	297.916
2020-08-06	297.916	290.542
2020-08-07	290.542	?

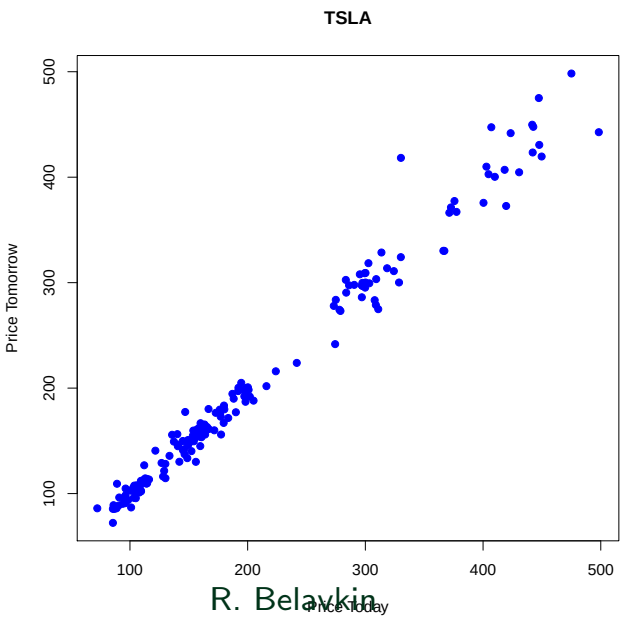
- We can compute linear trend and use it for forecast.
- We can use correlations between prices on different days.
- Can we predict tomorrow's price using the price today?

Linear trend and forecast

Table: TSLA.Close price Aug 2020

Date	Price(t)	Price(t+1)
2020-08-03	297.000	297.400
2020-08-04	297.400	297.004
2020-08-05	297.004	297.916
2020-08-06	297.916	290.542
2020-08-07	290.542	?

- We can compute linear trend and use it for forecast.
- We can use correlations between prices on different days.
- Can we predict tomorrow's price using the price today?

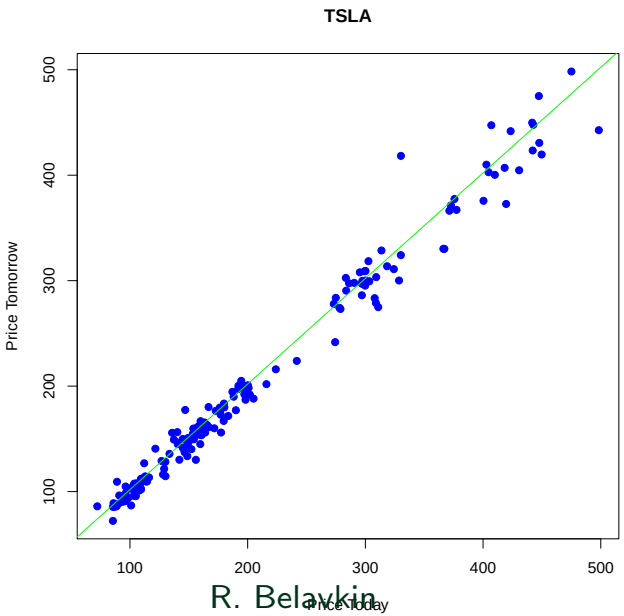


Linear trend and forecast

Table: TSLA.Close price Aug 2020

Date	Price(t)	Price(t+1)
2020-08-03	297.000	297.400
2020-08-04	297.400	297.004
2020-08-05	297.004	297.916
2020-08-06	297.916	290.542
2020-08-07	290.542	?

- We can compute linear trend and use it for forecast.
- We can use correlations between prices on different days.
- Can we predict tomorrow's price using the price today?

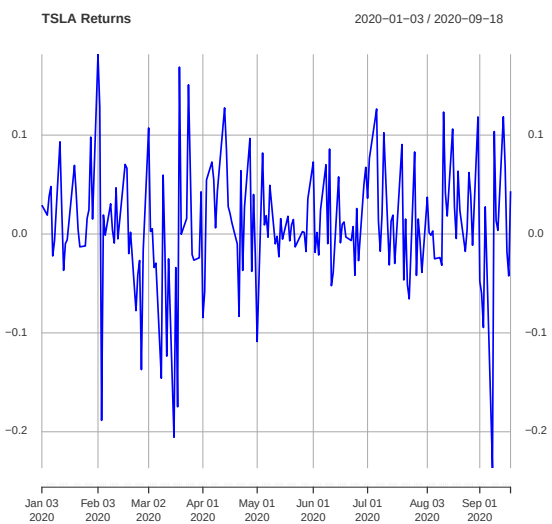


Linear trend and forecast

Table: TSLA.Close price Aug 2020

Date	Price(t)	Price(t+1)
2020-08-03	297.000	297.400
2020-08-04	297.400	297.004
2020-08-05	297.004	297.916
2020-08-06	297.916	290.542
2020-08-07	290.542	?

- We can compute linear trend and use it for forecast.
- We can use correlations between prices on different days.
- Can we predict tomorrow's price using the price today?



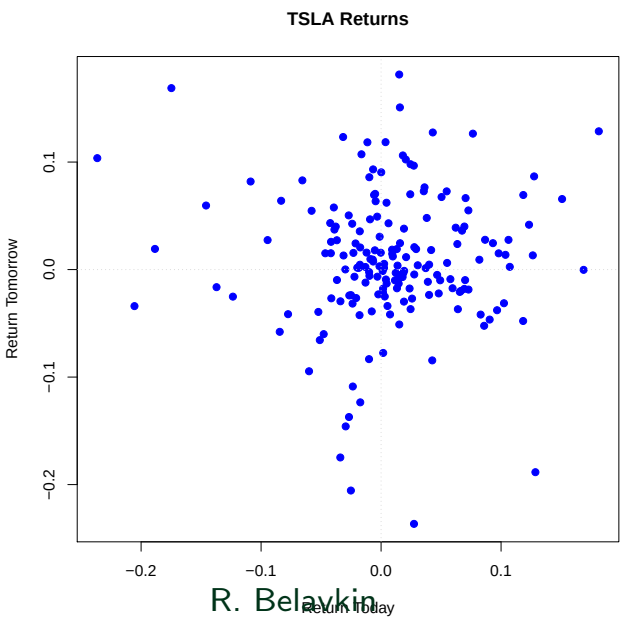
Can we predict log-returns? (i.e. changes in price)

Linear trend and forecast

Table: TSLA.Close price Aug 2020

Date	Price(t)	Price(t+1)
2020-08-03	297.000	297.400
2020-08-04	297.400	297.004
2020-08-05	297.004	297.916
2020-08-06	297.916	290.542
2020-08-07	290.542	?

- We can compute linear trend and use it for forecast.
- We can use correlations between prices on different days.
- Can we predict tomorrow's price using the price today?



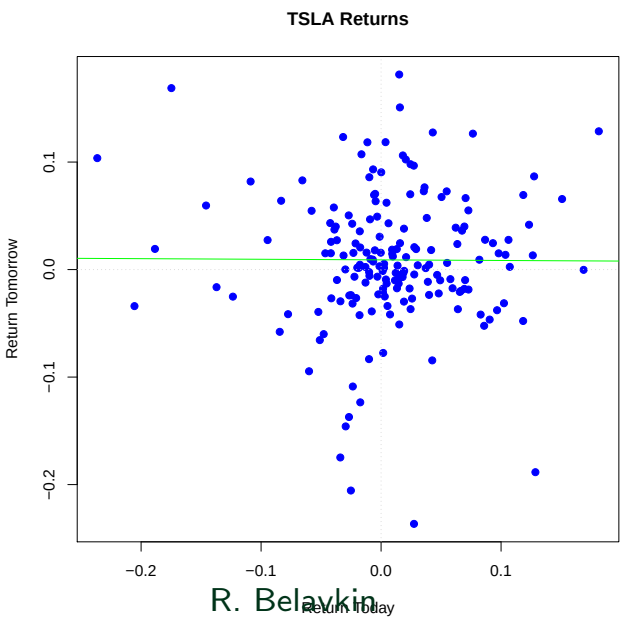
Can we predict log-returns? (i.e. changes in price)

Linear trend and forecast

Table: TSLA.Close price Aug 2020

Date	Price(t)	Price(t+1)
2020-08-03	297.000	297.400
2020-08-04	297.400	297.004
2020-08-05	297.004	297.916
2020-08-06	297.916	290.542
2020-08-07	290.542	?

- We can compute linear trend and use it for forecast.
- We can use correlations between prices on different days.
- Can we predict tomorrow's price using the price today?

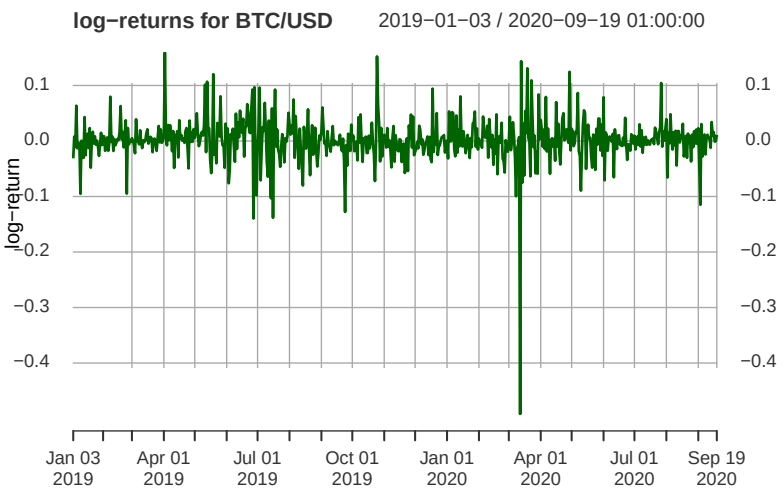


Can we predict log-returns? (i.e. changes in price)

Predicting log-returns

Table: BTCUSD log-returns Jan 2019

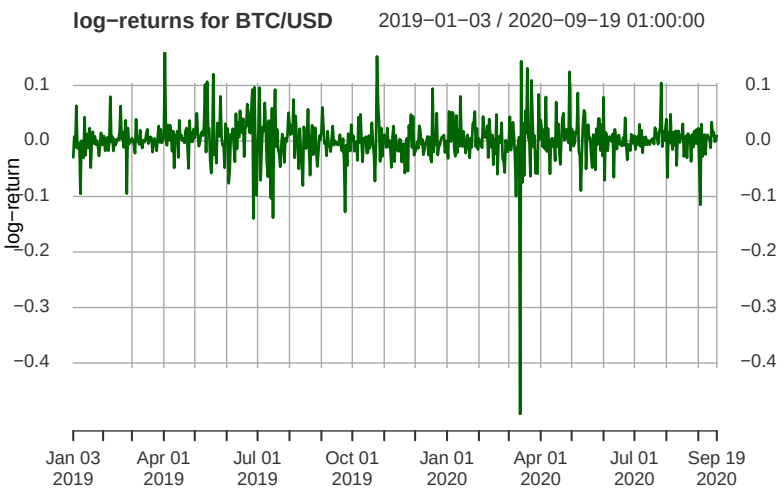
Date	log-return(t)
2019-01-03	-0.031232550
2019-01-04	0.007767325
2019-01-05	-0.010932127
2019-01-06	0.063509080
2019-01-07	-0.013160785
2019-01-08	-0.003384510



Predicting log-returns

Table: BTCUSD log-returns Jan 2019

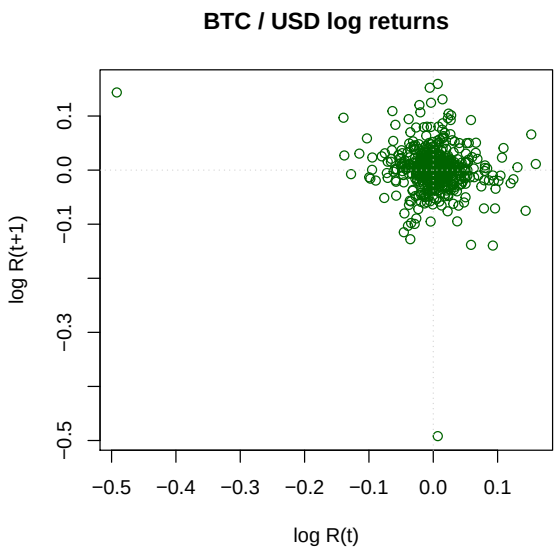
Date	log-return(t)	log-return(t+1)
2019-01-03	-0.031232550	0.007767325
2019-01-04	0.007767325	-0.010932127
2019-01-05	-0.010932127	0.063509080
2019-01-06	0.063509080	-0.013160785
2019-01-07	-0.013160785	-0.003384510
2019-01-08	-0.003384510	-0.004081489



Predicting log-returns

Table: BTCUSD log-returns Jan 2019

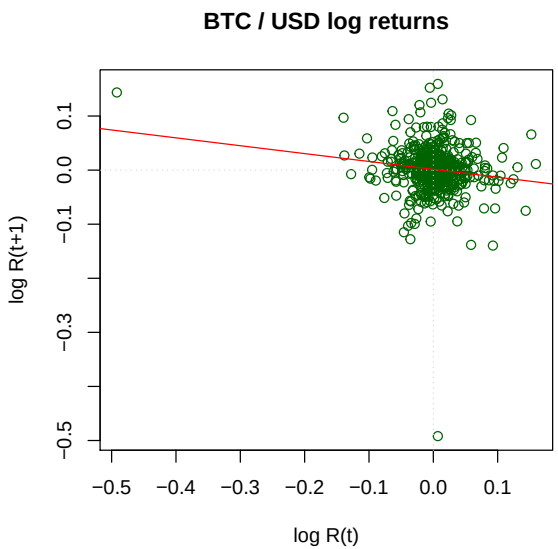
Date	log-return(t)	log-return(t+1)
2019-01-03	-0.031232550	0.007767325
2019-01-04	0.007767325	-0.010932127
2019-01-05	-0.010932127	0.063509080
2019-01-06	0.063509080	-0.013160785
2019-01-07	-0.013160785	-0.003384510
2019-01-08	-0.003384510	-0.004081489



Predicting log-returns

Table: BTCUSD log-returns Jan 2019

Date	log-return(t)	log-return(t+1)
2019-01-03	-0.031232550	0.007767325
2019-01-04	0.007767325	-0.010932127
2019-01-05	-0.010932127	0.063509080
2019-01-06	0.063509080	-0.013160785
2019-01-07	-0.013160785	-0.003384510
2019-01-08	-0.003384510	-0.004081489



- We can try to model log-returns for two consecutive days:

$$\log R(t+1) \approx a + b \log R(t)$$

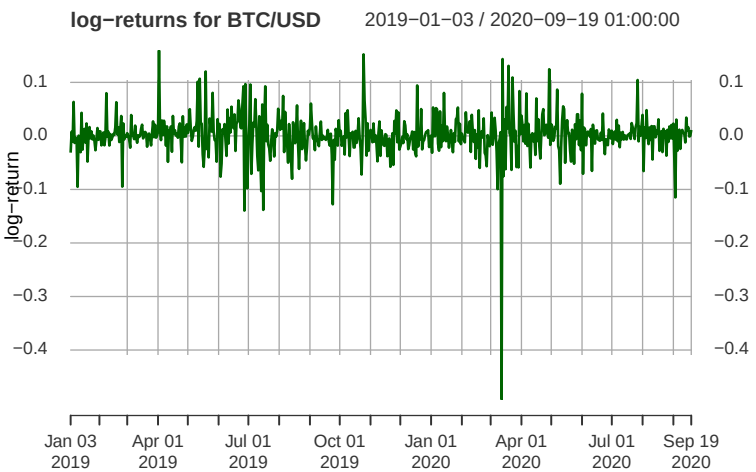
where $R(t) = S(t)/S(t-1)$ ratio of prices.

- The performance may be poor, as the correlation is small.

Predicting log-returns (cont.)

Table: BTCUSD log-returns Jan 2019

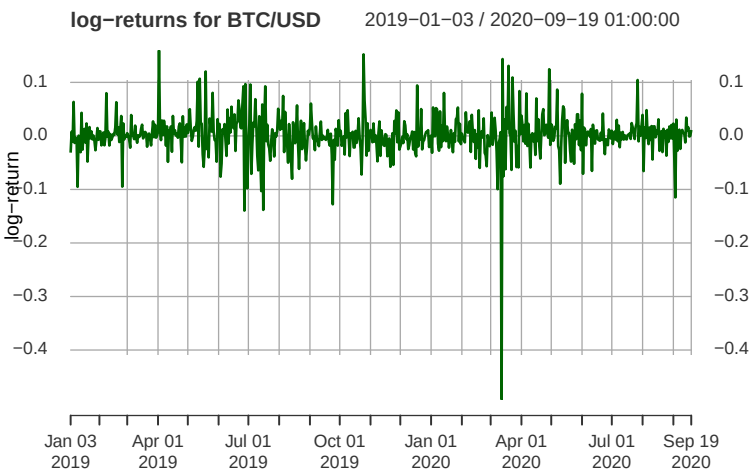
Date	log-return(t-2)	log-return(t-1)	log-return(t)	
2019-01-06	-0.031232550	0.007767325	-0.010932127	
2019-01-07	0.007767325	-0.010932127	0.063509080	
2019-01-08	-0.010932127	0.063509080	-0.013160785	
2019-01-09	0.063509080	-0.013160785	-0.003384510	



Predicting log-returns (cont.)

Table: BTCUSD log-returns Jan 2019

Date	log-return(t-2)	log-return(t-1)	log-return(t)	log-return(t+1)
2019-01-06	-0.031232550	0.007767325	-0.010932127	0.063509080
2019-01-07	0.007767325	-0.010932127	0.063509080	-0.013160785
2019-01-08	-0.010932127	0.063509080	-0.013160785	-0.003384510
2019-01-09	0.063509080	-0.013160785	-0.003384510	-0.004081489



Predicting log-returns (cont.)

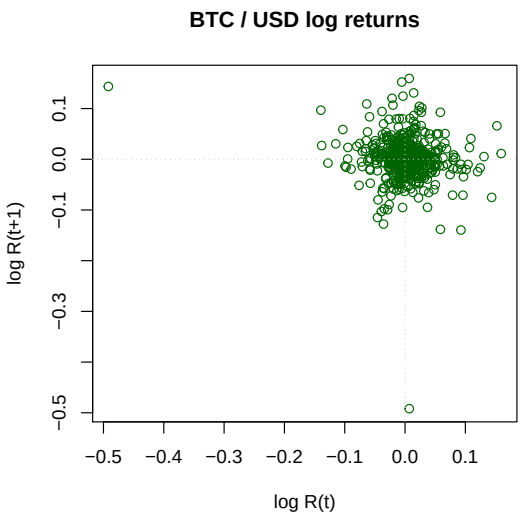
Table: BTCUSD log-returns Jan 2019

Date	log-return(t-2)	log-return(t-1)	log-return(t)	log-return(t+1)
2019-01-06	-0.031232550	0.007767325	-0.010932127	0.063509080
2019-01-07	0.007767325	-0.010932127	0.063509080	-0.013160785
2019-01-08	-0.010932127	0.063509080	-0.013160785	-0.003384510
2019-01-09	0.063509080	-0.013160785	-0.003384510	-0.004081489

Predict next day log-return based on n previous day's log-returns:

$$\log R(t+1) \approx a + b_1 \log R(t) + b_2 \log R(t-1) + \cdots + b_n \log R(t-n)$$

where $R(t) = S(t)/S(t-1)$ ratio of prices.



Predicting log-returns (cont.)

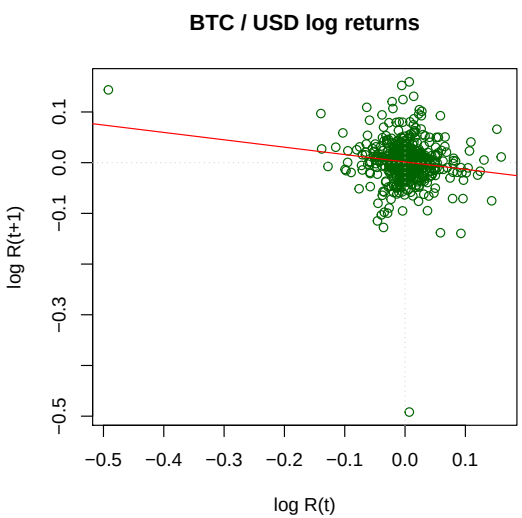
Table: BTCUSD log-returns Jan 2019

Date	log-return(t-2)	log-return(t-1)	log-return(t)	log-return(t+1)
2019-01-06	-0.031232550	0.007767325	-0.010932127	0.063509080
2019-01-07	0.007767325	-0.010932127	0.063509080	-0.013160785
2019-01-08	-0.010932127	0.063509080	-0.013160785	-0.003384510
2019-01-09	0.063509080	-0.013160785	-0.003384510	-0.004081489

Predict next day log-return based on n previous day's log-returns:

$$\log R(t+1) \approx a + b_1 \log R(t) + b_2 \log R(t-1) + \cdots + b_n \log R(t-n)$$

where $R(t) = S(t)/S(t-1)$ ratio of prices.



We may also include past log-returns of other cryptocurrencies as predictors.

Application to trading

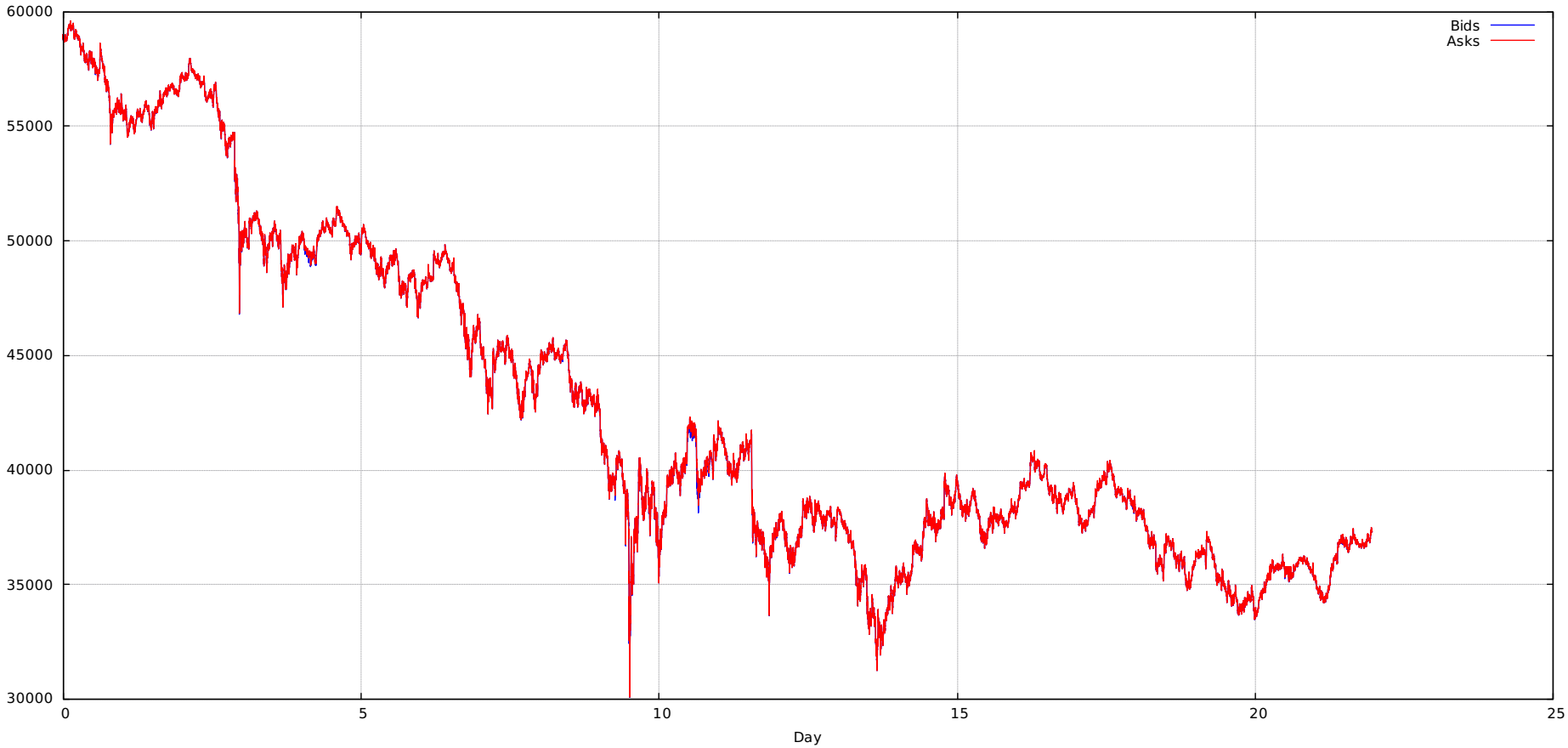


Figure: BTC/USD, 10–22 May, 2021

Application to trading

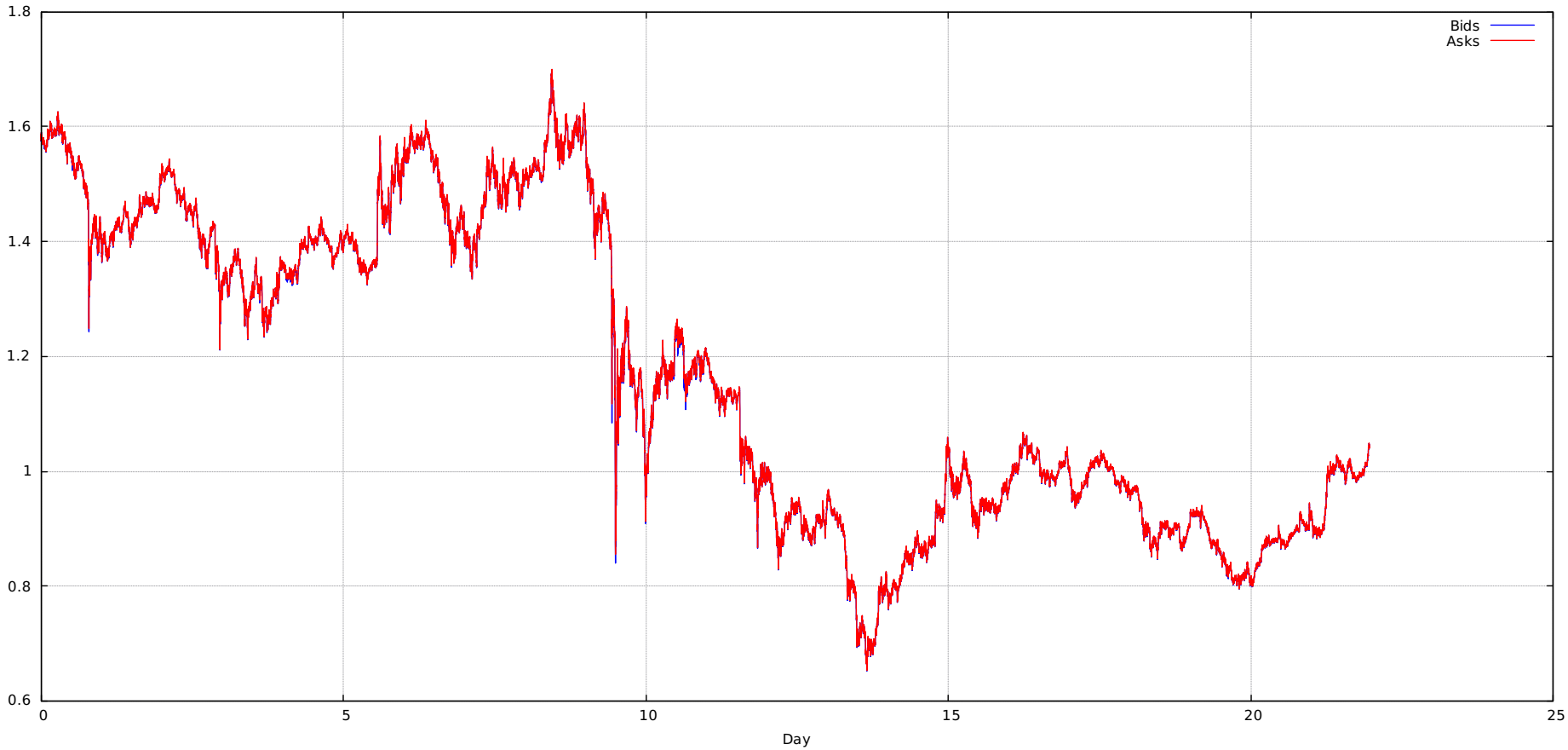


Figure: XRP/USD, 10–22 May, 2021

Application to trading

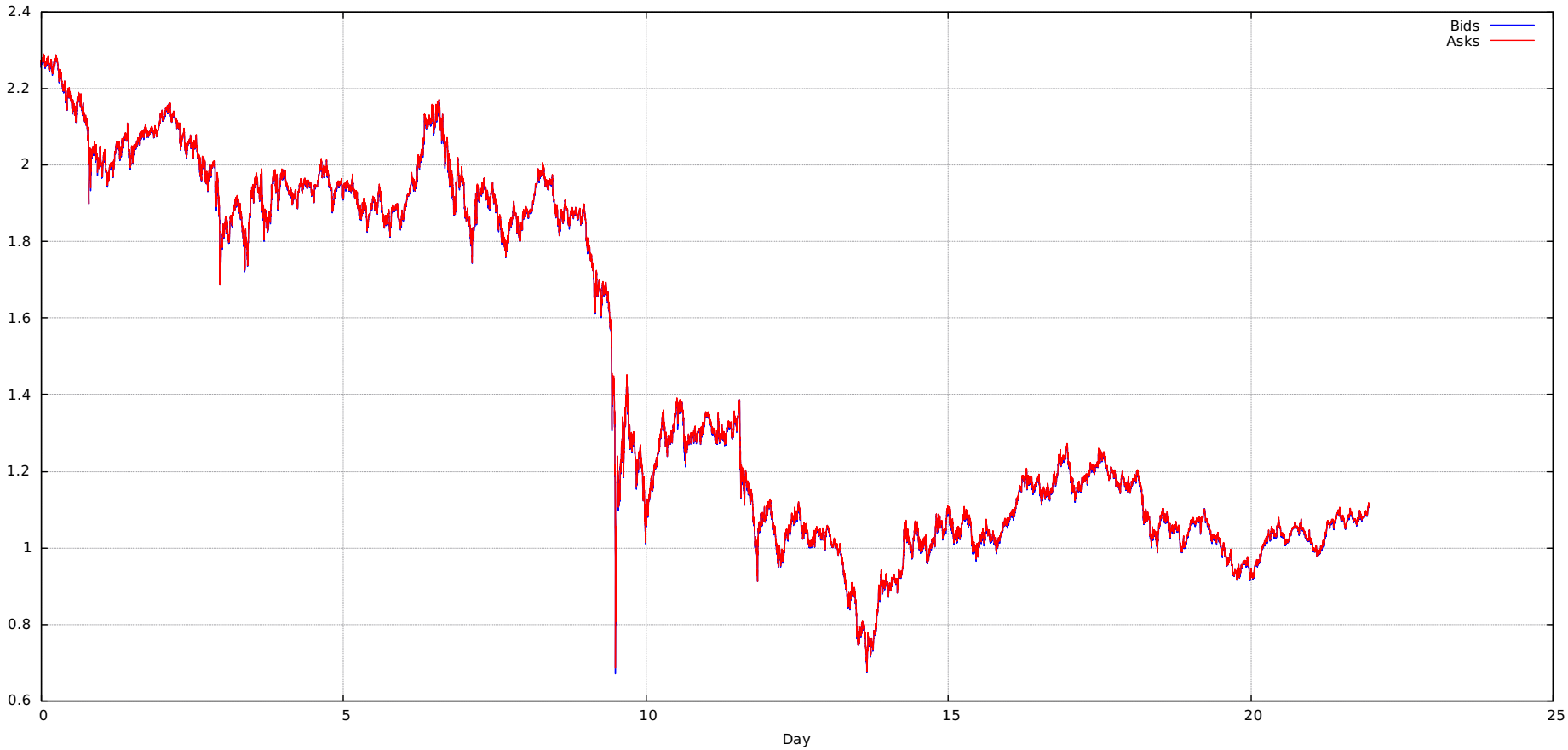


Figure: IOT/USD, 10–22 May, 2021

Application to trading

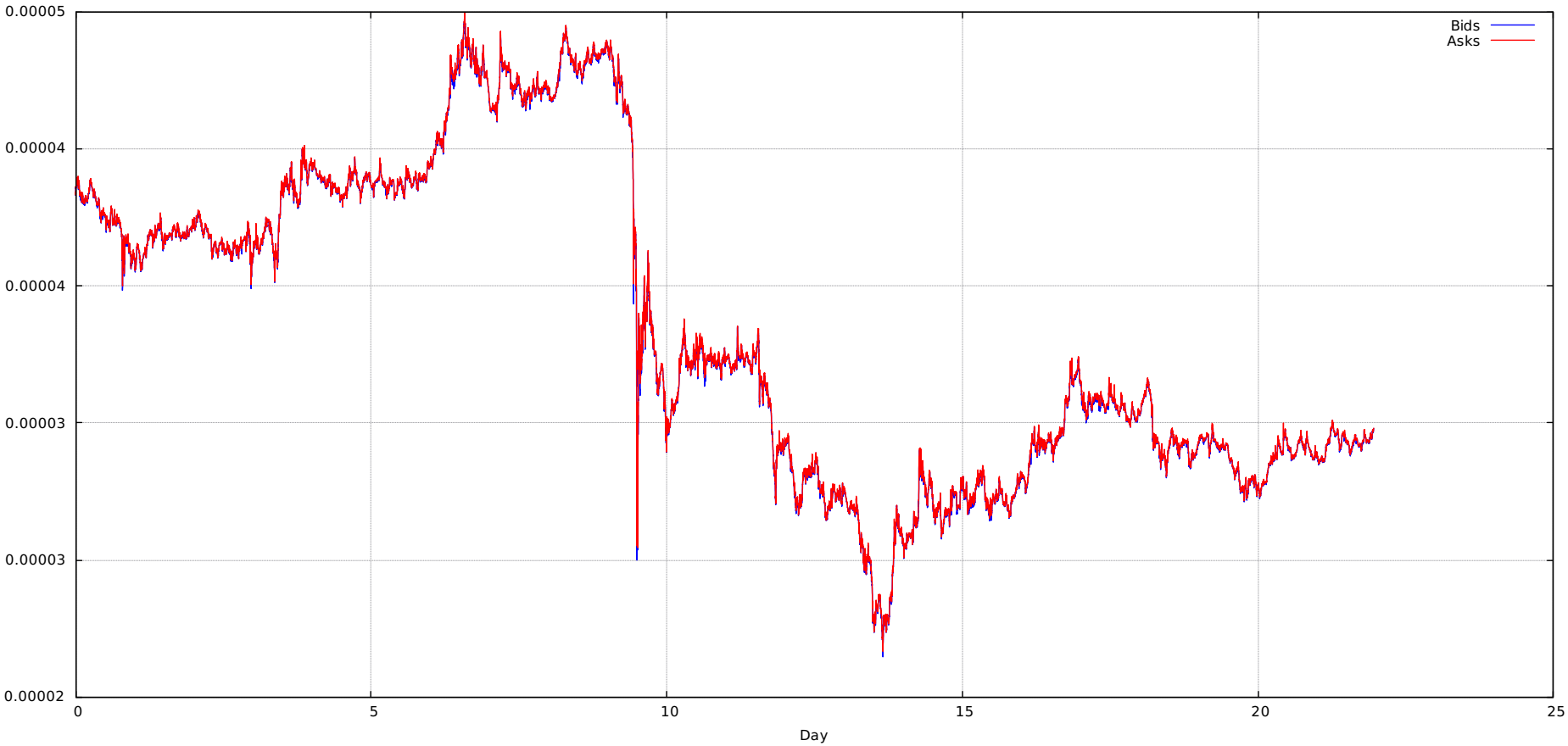


Figure: IOT/BTC, 10–22 May, 2021

Application to trading

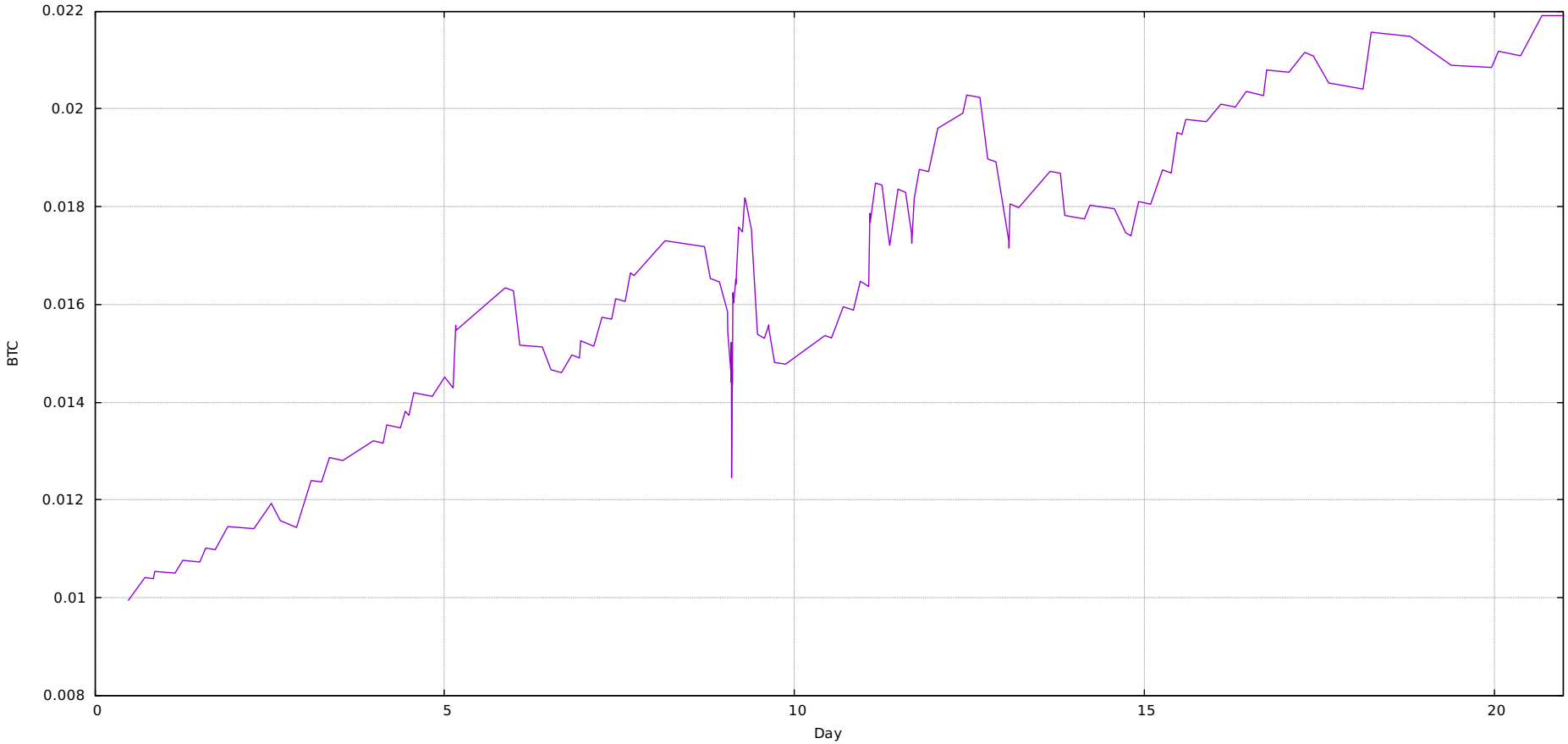


Figure: Amount of BTC, trading during 10–22 May, 2021, using array of forecasts for ≈ 300 trade pairs and ranging from 30 min to 24h into the future.

Application to trading

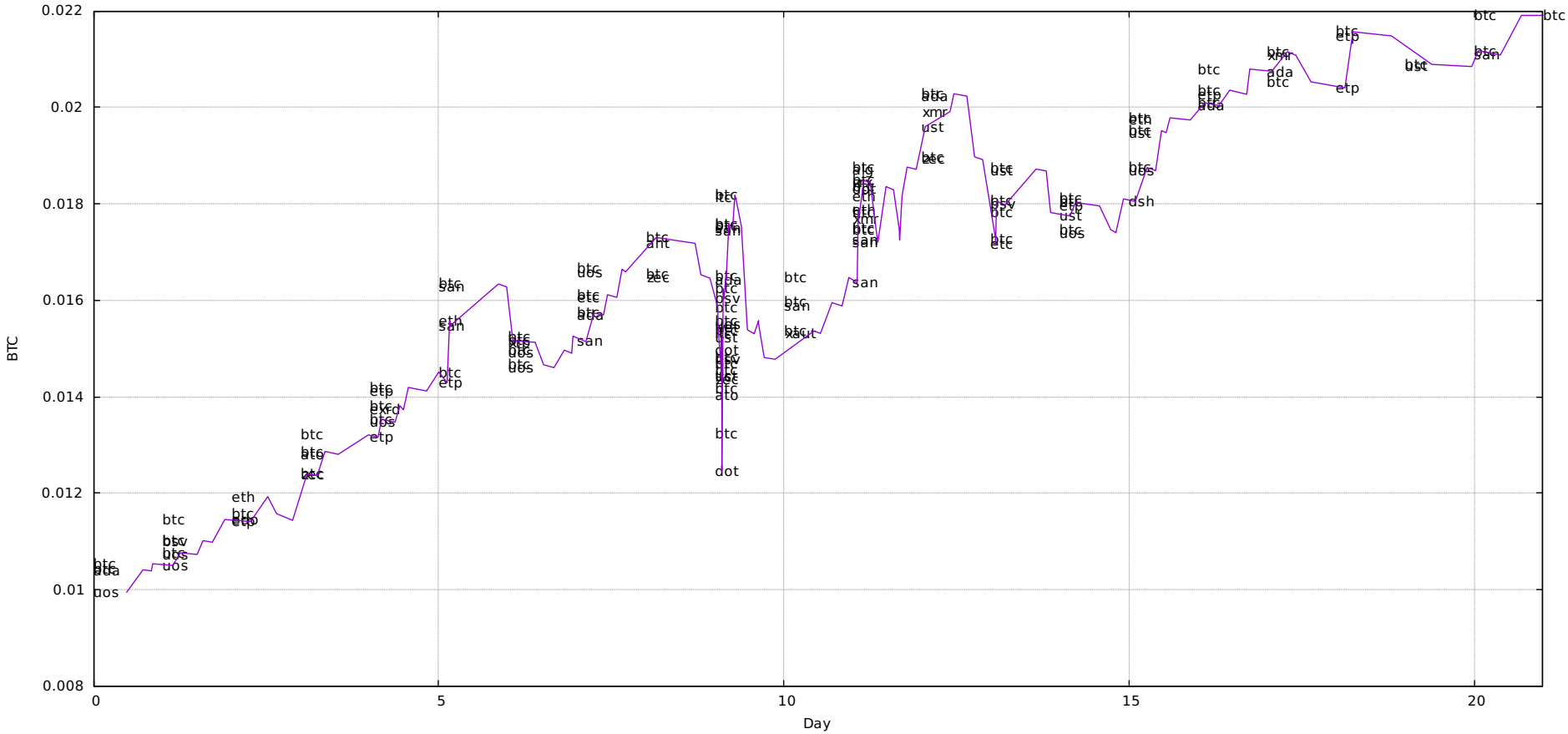


Figure: Amount of BTC, trading during 10-22 May, 2021, using array of forecasts for ≈ 300 trade pairs and ranging from 30 min to 24h into the future.

Value of Boltzmann's information

- *Entropy* as the expected 'surprise' $-\ln P(z)$:

$$H(Z) := \mathbb{E}_{P(z)}\{-\ln P(z)\}$$

Value of Boltzmann's information

- *Entropy* as the expected 'surprise' $-\ln P(z)$:

$$H(Z) := -\sum_z [\ln P(z)]\, P(z)$$

Value of Boltzmann's information

- *Entropy* as the expected 'surprise' $-\ln P(z)$:

$$H(Z) := -\sum_z [\ln P(z)]\, P(z) \leq \ln |Z|$$

Value of Boltzmann's information

- *Entropy* as the expected 'surprise' $-\ln P(z)$:

$$H(Z) := -\sum_z [\ln P(z)]\, P(z) \le \ln |Z|$$

- (here $P(z) = P\{x : z \circ y(x) = z\}$)

Value of Boltzmann's information

- *Entropy* as the expected 'surprise' $-\ln P(z)$:

$$H(Z) := -\sum_z [\ln P(z)]\, P(z) \le \ln |Z|$$

- (here $P(z) = P\{x : z \circ y(x) = z\}$)
- Consider the quantity

$$U(H) := \sup_{y(x)} \mathbb{E}_{P(y)} \left\{ \sup_{z(y)} \mathbb{E}_{P(x|y)} \{u(x,z) \mid y\} \right\}$$

$$\text{Subject to } H(Z) \le \ln k$$

Value of Boltzmann's information

- *Entropy* as the expected 'surprise' $-\ln P(z)$:

$$H(Z) := -\sum_{z} [\ln P(z)]\, P(z) \leq \ln |Z|$$

- (here $P(z) = P\{x : z \circ y(x) = z\}$)
- Consider the quantity

$$U(H) := \sup_{y(x)} \mathbb{E}_{P(y)} \left\{ \sup_{z(y)} \mathbb{E}_{P(x|y)} \{u(x, z) \mid y\} \right\}$$

$$\text{Subject to } H(Z) \leq \ln k$$

- $V(H) = U(H) - U(0)$ is now called the *value of Boltzmann information* (Stratonovich, 1965).

Value of Boltzmann's information

- *Entropy* as the expected 'surprise' $-\ln P(z)$:

$$H(Z) := -\sum_{z} [\ln P(z)]\, P(z) \leq \ln |Z|$$

- (here $P(z) = P\{x : z \circ y(x) = z\}$)
- Consider the quantity

$$U(H) := \sup_{y(x)} \mathbb{E}_{P(y)} \left\{ \sup_{z(y)} \mathbb{E}_{P(x|y)} \{u(x, z) \mid y\} \right\}$$

$$\text{Subject to } H(Z) \leq \ln k$$

- $V(H) = U(H) - U(0)$ is now called the *value of Boltzmann information* (Stratonovich. 1965).

Remark

We are looking for optimal function $z(x) = z \circ y(x)$ *subject to* $H(Z) \leq \ln k$.

Value of Shannon's information

- Shannon's *mutual information*:

$$I(X,Z) := \mathbb{E}_{P(x,z)} \left\{ \ln \frac{P(z \mid x)}{P(z)} \right\}$$

Value of Shannon's information

- Shannon's *mutual information*:

$$I(X, Z) := H(Z) - H(Z \mid X)$$

Value of Shannon's information

- Shannon's *mutual information*:

$$I(X, Z) := H(X) - H(X \mid Z)$$

Value of Shannon's information

- Shannon's *mutual information*:

$$I(X, Z) := H(X) - H(X \mid Z) \leq \min[H(X), H(Z)]$$

Value of Shannon's information

- Shannon's *mutual information*:

$$I(X,Z) := H(X) - H(X \mid Z) \leq \min[H(X), H(Z)]$$

- Consider the quantity

$$U(I) := \sup_{P(z|x)} \mathbb{E}_{P(x,z)}\{u(x,z)\}$$

$$\text{Subject to } I(X,Z) \leq I$$

where $P(z \mid x) = \sum_Y P(z \mid y)P(y \mid x)$ and $I(X,Y) \leq I$.

Value of Shannon's information

- Shannon's *mutual information*:

$$I(X,Z) := H(X) - H(X \mid Z) \leq \min[H(X), H(Z)]$$

- Consider the quantity

$$U(I) := \sup_{P(z|x)} \mathbb{E}_{P(x,z)}\{u(x,z)\}$$

$$\text{Subject to } I(X,Z) \leq I$$

  where $P(z \mid x) = \sum_Y P(z \mid y)P(y \mid x)$ and $I(X,Y) \leq I$.
- $V(I) = U(I) - U(0)$ is called the *value of Shannon's information* (Stratonovich, 1965).

Value of Shannon's information

- Shannon's *mutual information*:

$$I(X,Z) := H(X) - H(X \mid Z) \leq \min[H(X), H(Z)]$$

- Consider the quantity

$$U(I) := \sup_{P(z|x)} \mathbb{E}_{P(x,z)}\{u(x,z)\}$$

$$\text{Subject to } I(X,Z) \leq I$$

  where $P(z \mid x) = \sum_Y P(z \mid y)P(y \mid x)$ and $I(X,Y) \leq I$.
- $V(I) = U(I) - U(0)$ is called the *value of Shannon's information* (Stratonovich, 1965).

Remark

Instead of functions $z(x) = z \circ y(x)$, we are looking for optimal $P(z \mid x)$ subject to $I(X,Z) \leq I$.

Asymptotic Equivalence of Different VoIs

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$P(x) \qquad \mathbb{E}\{u(x, z)\}$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$P(x) \qquad\qquad \max_z \mathbb{E}\{u(x, z)\}$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$P(x) \qquad \max_z \mathbb{E}\{u(x, z)\} =: U(0)$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$x = f^{-1}(y)$$

$$P(x) \qquad \max_z \mathbb{E}\{u(x, z)\} =: U(0)$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$x = f^{-1}(y) \qquad u(x, z)$$

$$P(x) \qquad \max_z \mathbb{E}\{u(x, z)\} =: U(0)$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$x = f^{-1}(y) \qquad \max_{z(x)} u(x, z)$$

$$P(x) \qquad \max_{z} \mathbb{E}\{u(x, z)\} =: U(0)$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$x = f^{-1}(y) \qquad \max_{z(x)} u(x, z) =: U(\infty)$$

$$P(x) \qquad \max_{z} \mathbb{E}\{u(x, z)\} =: U(0)$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$x = f^{-1}(y) \qquad \max_{z(x)} u(x, z) =: U(\infty)$$

$$P(x \mid y)$$

$$P(x) \qquad \max_{z} \mathbb{E}\{u(x, z)\} =: U(0)$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$x = f^{-1}(y) \qquad \max_{z(x)} u(x, z) =: U(\infty)$$

$$P(x \mid y) \qquad \mathbb{E}\{u(x, z) \mid y\}$$

$$P(x) \qquad \max_{z} \mathbb{E}\{u(x, z)\} =: U(0)$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$x = f^{-1}(y) \qquad \max_{z(x)} u(x, z) =: U(\infty)$$

$$P(x \mid y) \qquad \max_{P(z|x):I\{x,z\}\leq\lambda} \mathbb{E}\{u(x, z) \mid y\}$$

$$P(x) \qquad \max_{z} \mathbb{E}\{u(x, z)\} =: U(0)$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x, z)$ — utility.
-

$$x = f^{-1}(y) \qquad \max_{z(x)} u(x, z) =: U(\infty)$$

$$P(x \mid y) \qquad \max_{P(z|x):I\{x,z\}\leq\lambda} \mathbb{E}\{u(x, z) \mid y\} =: U(\lambda)$$

$$P(x) \qquad \max_{z} \mathbb{E}\{u(x, z)\} =: U(0)$$

Asymptotic Equivalence of Different Vols

- x — hidden, y — observed, z — control, $u(x,z)$ — utility.
-

$$x = f^{-1}(y) \qquad \max_{z(x)} u(x,z) =: U(\infty)$$

$$P(x \mid y) \qquad \max_{P(z|x):I\{x,z\}\leq\lambda} \mathbb{E}\{u(x,z) \mid y\} =: U(\lambda)$$

$$P(x) \qquad \max_{z} \mathbb{E}\{u(x,z)\} =: U(0)$$

- For equal constraints $\ln|Z| \leq \ln k$, $H(Z) \leq \ln k$, $I(X,Z) \leq I = lnk$, the values of Hartley, Boltzmann and Shannon's information satisfy

$$V(\ln k) \leq V(H) \leq V(I)$$

Asymptotic Equivalence of Different VoIs

- x — hidden, y — observed, z — control, $u(x,z)$ — utility.
-

$$x = f^{-1}(y) \qquad \max_{z(x)} u(x,z) =: U(\infty)$$

$$P(x \mid y) \qquad \max_{P(z|x):I\{x,z\}\leq\lambda} \mathbb{E}\{u(x,z) \mid y\} =: U(\lambda)$$

$$P(x) \qquad \max_{z} \mathbb{E}\{u(x,z)\} =: U(0)$$

- For equal constraints $\ln|Z| \leq \ln k$, $H(Z) \leq \ln k$, $I(X,Z) \leq I = lnk$, the values of Hartley, Boltzmann and Shannon's information satisfy

$$V(\ln k) \leq V(H) \leq V(I)$$

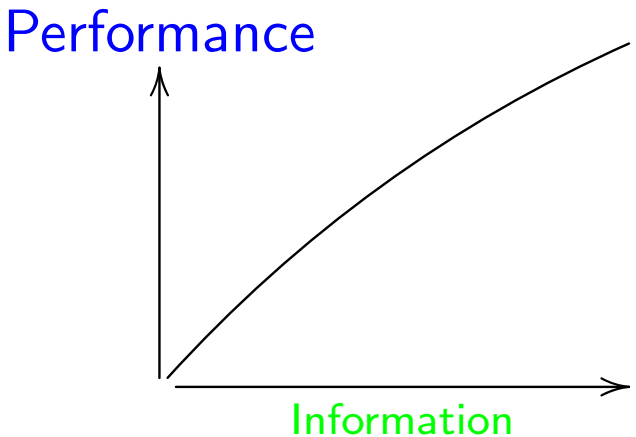
- One of the main results of the theory is that all types of VoI are asymptotically equivalent (Stratonovich, 1975).

The Maximum Entropy (or first) variational problem

Problem I

- Linear programming problem:

$$\text{maximize (minimize)} \quad \mathbb{E}_p\{u\} \quad \text{subject to} \quad \mathbb{E}_p\{\ln(p/q)\} \leq \lambda$$



Maximize performance

s.t. information $\leq \lambda$

The Maximum Entropy (or first) variational problem

Problem I

- Linear programming problem:

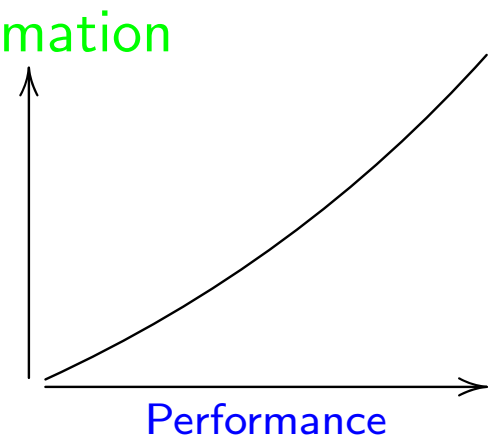
$$\text{maximize (minimize)} \quad \mathbb{E}_p\{u\} \quad \text{subject to} \quad \mathbb{E}_p\{\ln(p/q)\} \le \lambda$$

- The inverse convex programming problem:

$$\text{minimize} \quad \mathbb{E}_p\{\ln(p/q)\} \quad \text{subject to} \quad \mathbb{E}_p\{u\} \ge \upsilon \quad \left(\mathbb{E}_p\{u\} \le \upsilon\right)$$

Minimize information

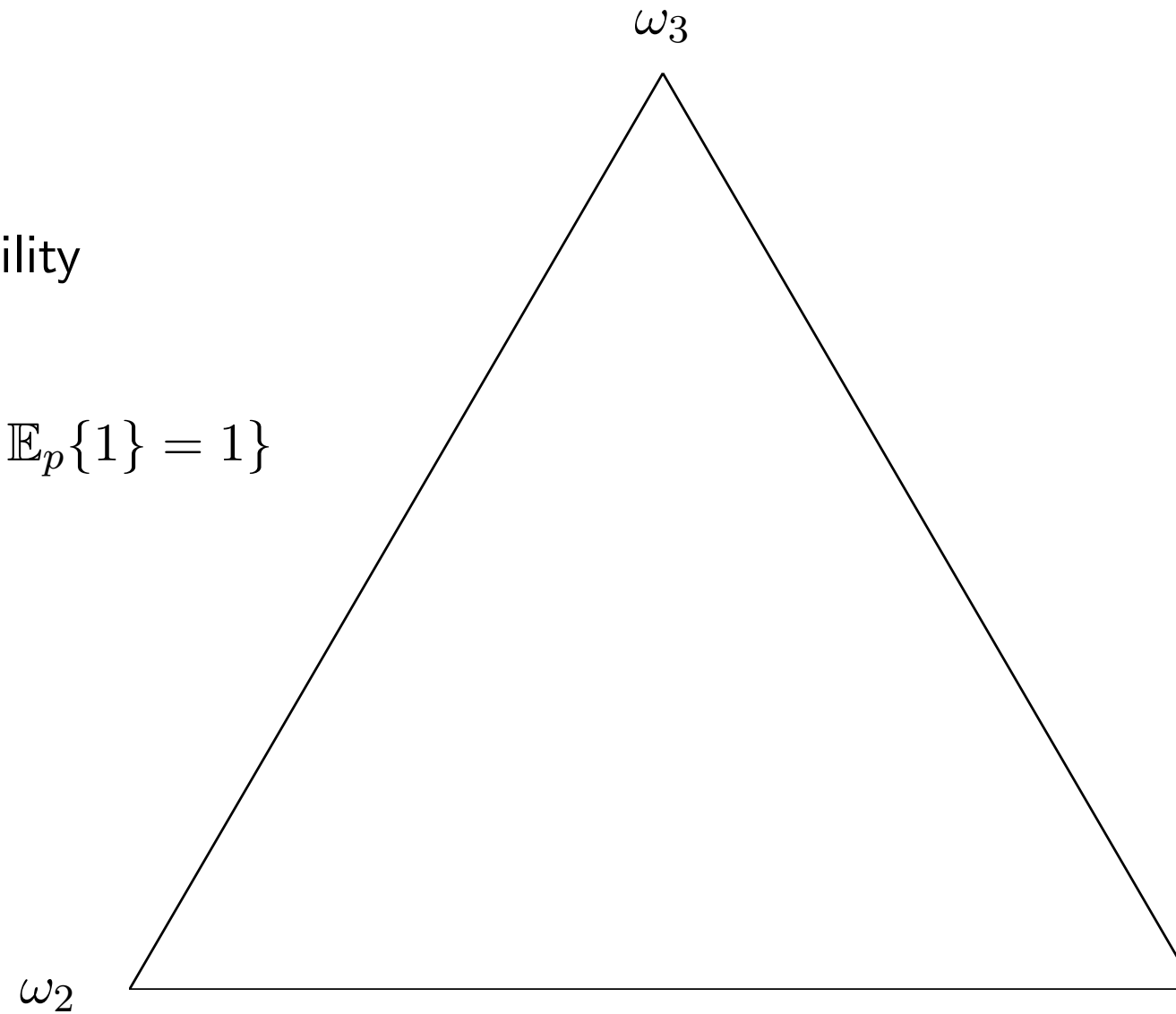
s.t. performance $\ge \upsilon$



Information-geometruc view

- The set of all probability measures

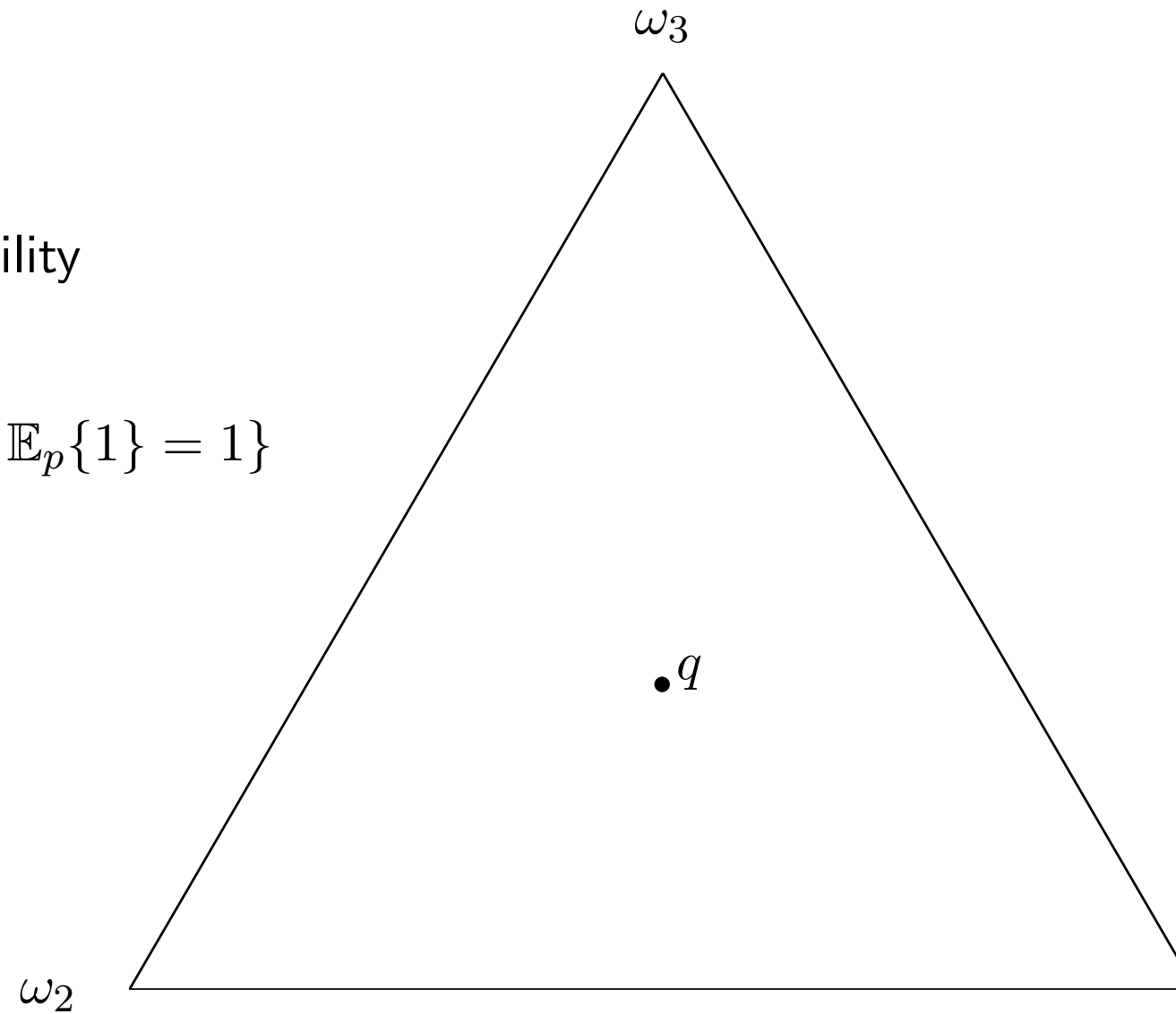
$$\mathcal{P}(\Omega) := \{p : p \geq 0\,,\ \mathbb{E}_p\{1\} = 1\}$$



Information-geometruc view

- The set of all probability measures

$$\mathcal{P}(\Omega) := \{p : p \geq 0\,,\ \mathbb{E}_p\{1\} = 1\}$$

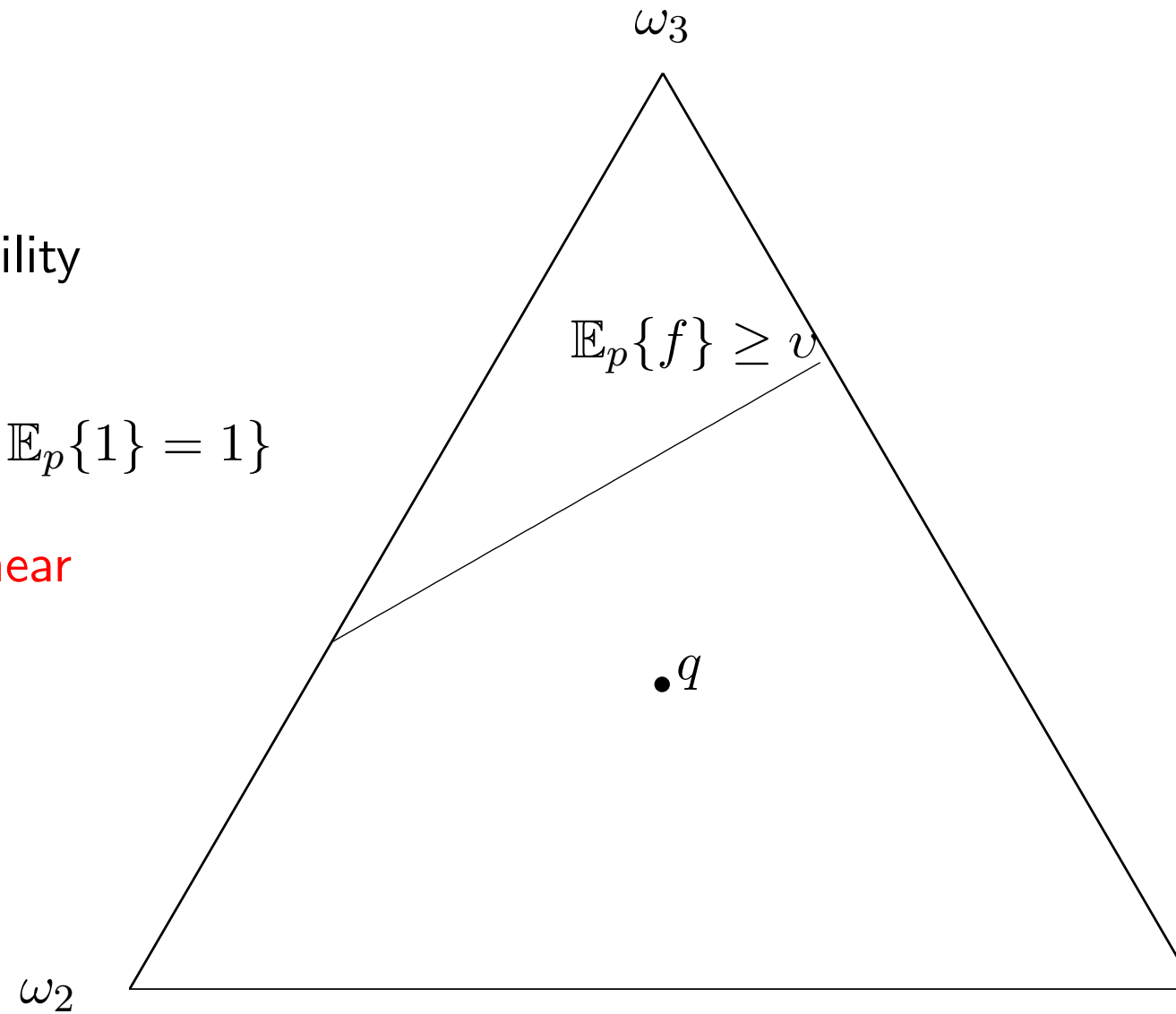


Information-geometruc view

- The set of all probability measures

  $\mathcal{P}(\Omega) := \{p : p \geq 0\,,\ \mathbb{E}_p\{1\} = 1\}$

- $\mathbb{E}_p\{u\} := \langle u, p\rangle$ is linear

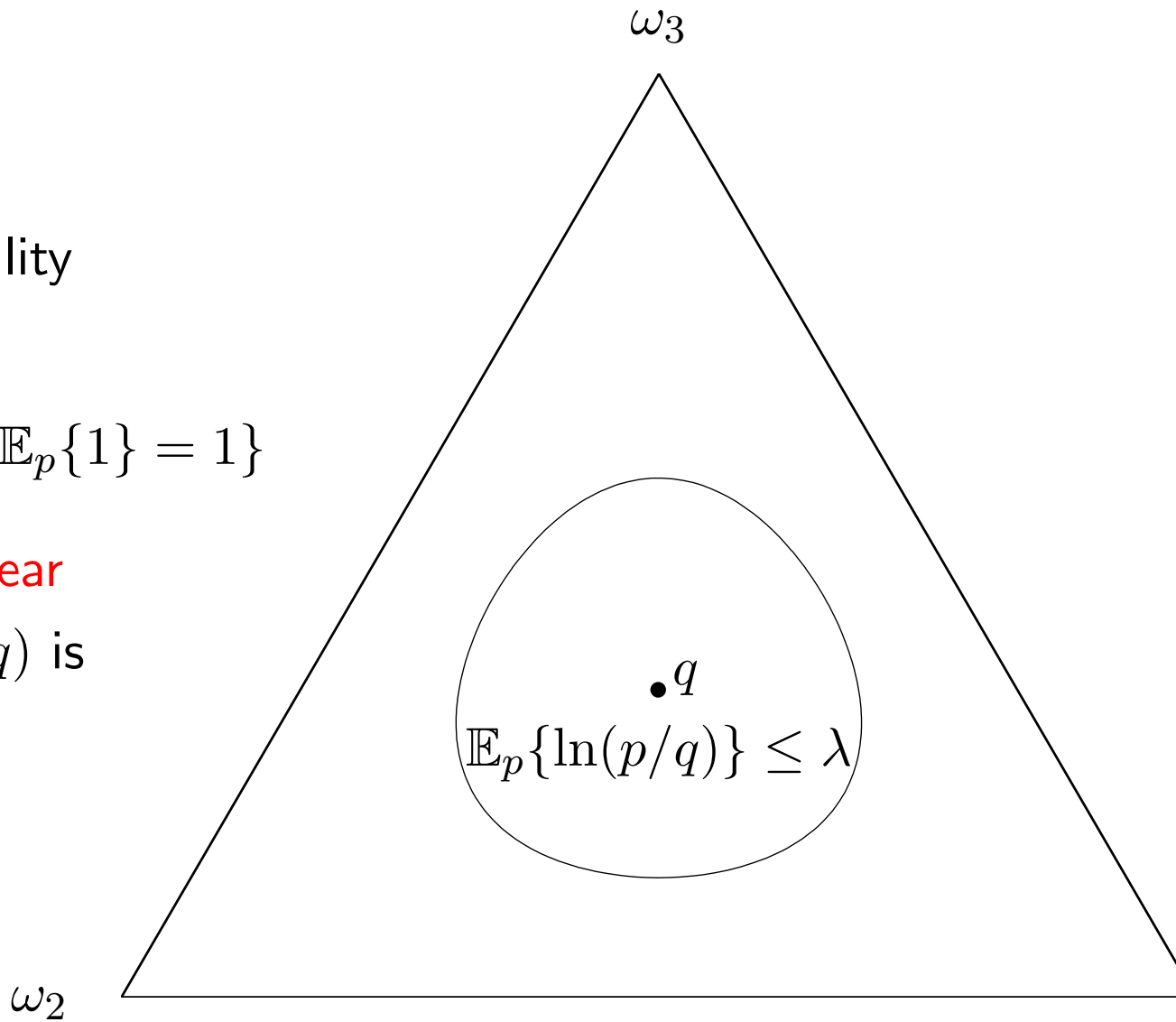


Information-geometruc view

- The set of all probability measures

$$\mathcal{P}(\Omega) := \{p : p \geq 0\,,\ \mathbb{E}_p\{1\} = 1\}$$

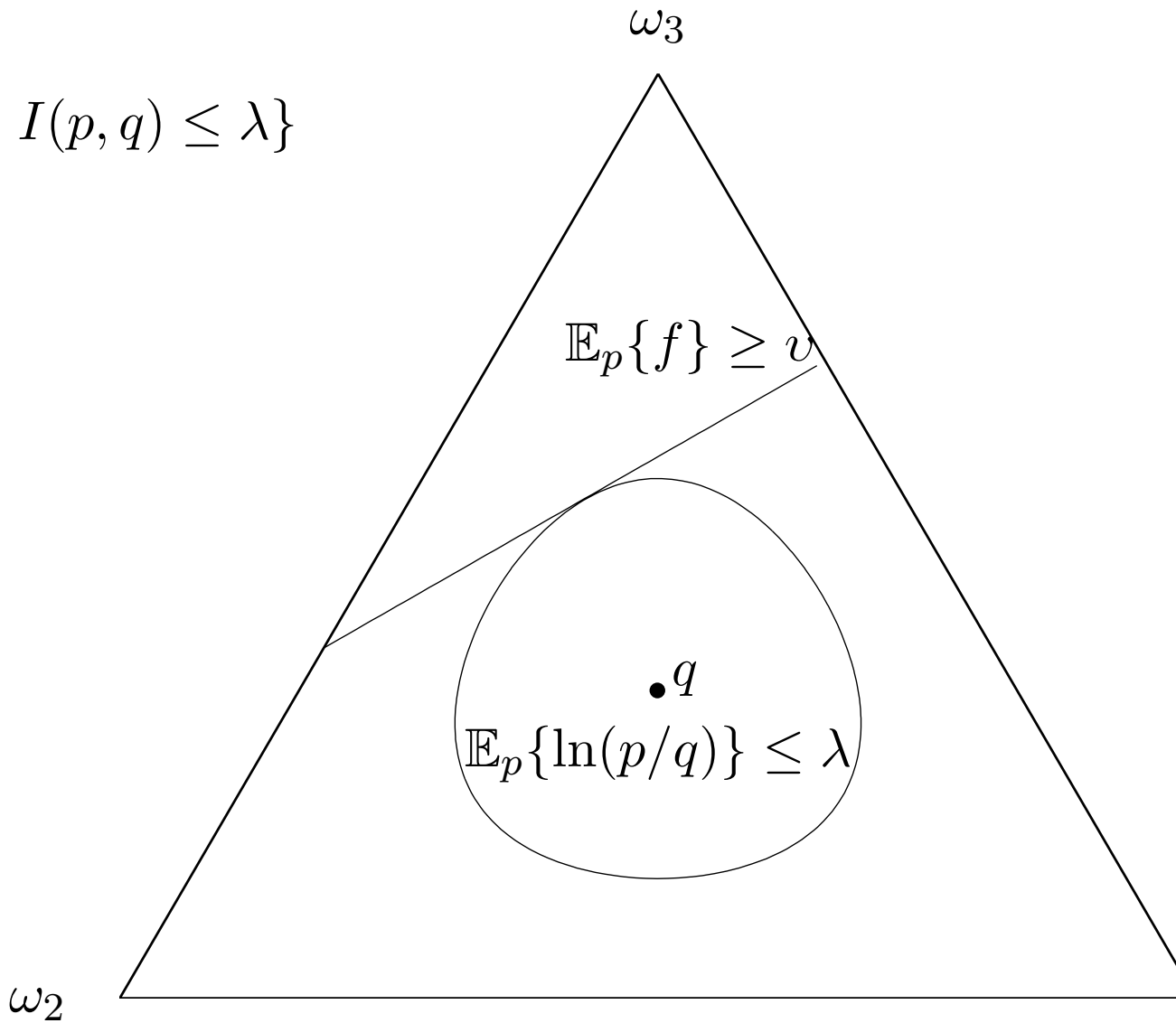
- $\mathbb{E}_p\{u\} := \langle u, p\rangle$ is linear
- $\mathbb{E}_p\{\ln(p/q)\} =: I(p,q)$ is convex



Information-geometruc view (cont)

- Maximize $\mathbb{E}_p\{u\}$

$$V(\lambda) := \sup\{\mathbb{E}_p\{u\} : I(p,q) \leq \lambda\}$$



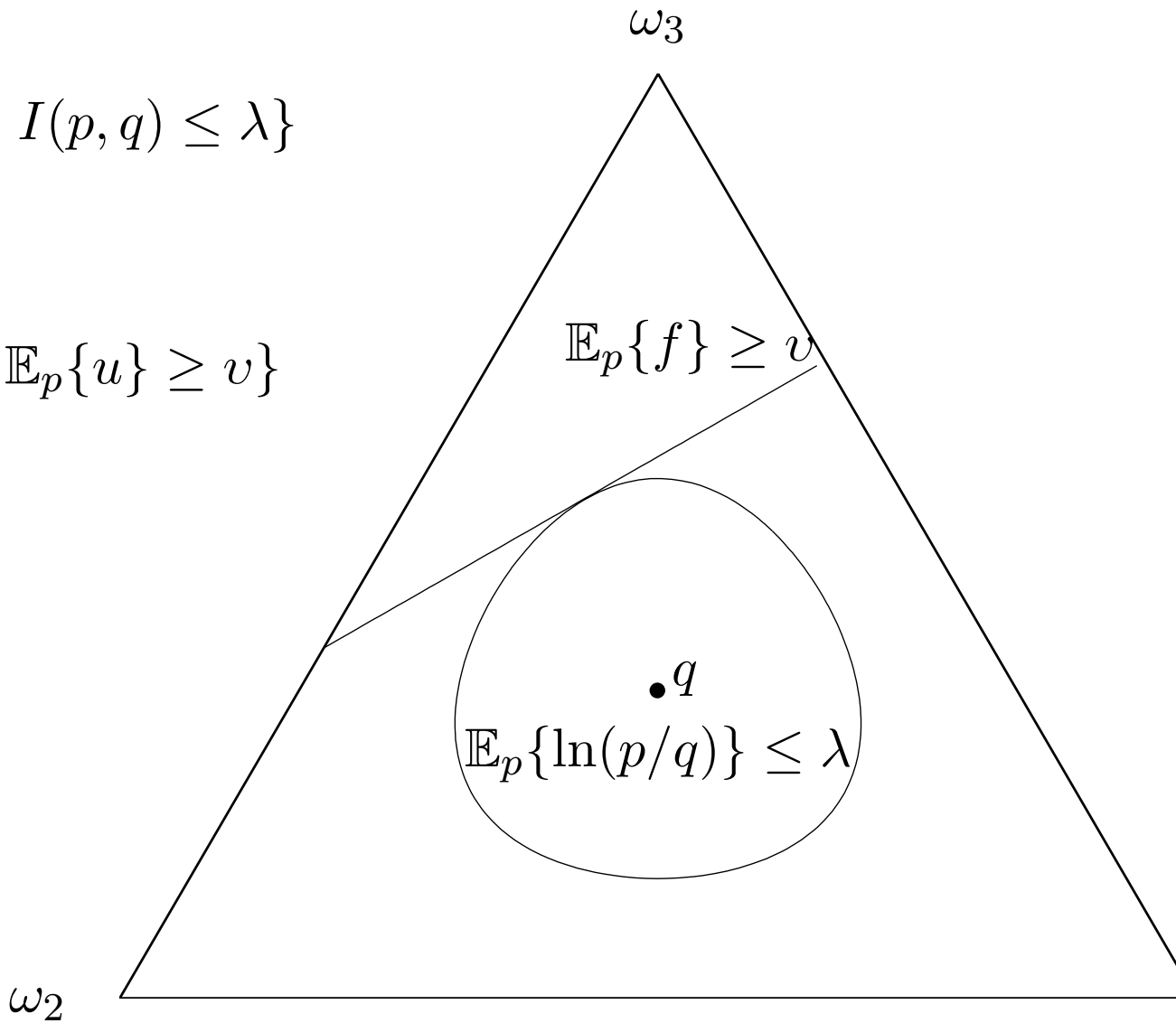
Information-geometruc view (cont)

- Maximize $\mathbb{E}_p\{u\}$

  $V(\lambda) := \sup\{\mathbb{E}_p\{u\} : I(p,q) \leq \lambda\}$

- Minimize $I(p,q)$:

  $\lambda(V) := \inf\{I(p,q) : \mathbb{E}_p\{u\} \geq \upsilon\}$



Information-geometruc view (cont)

- Maximize $\mathbb{E}_p\{u\}$

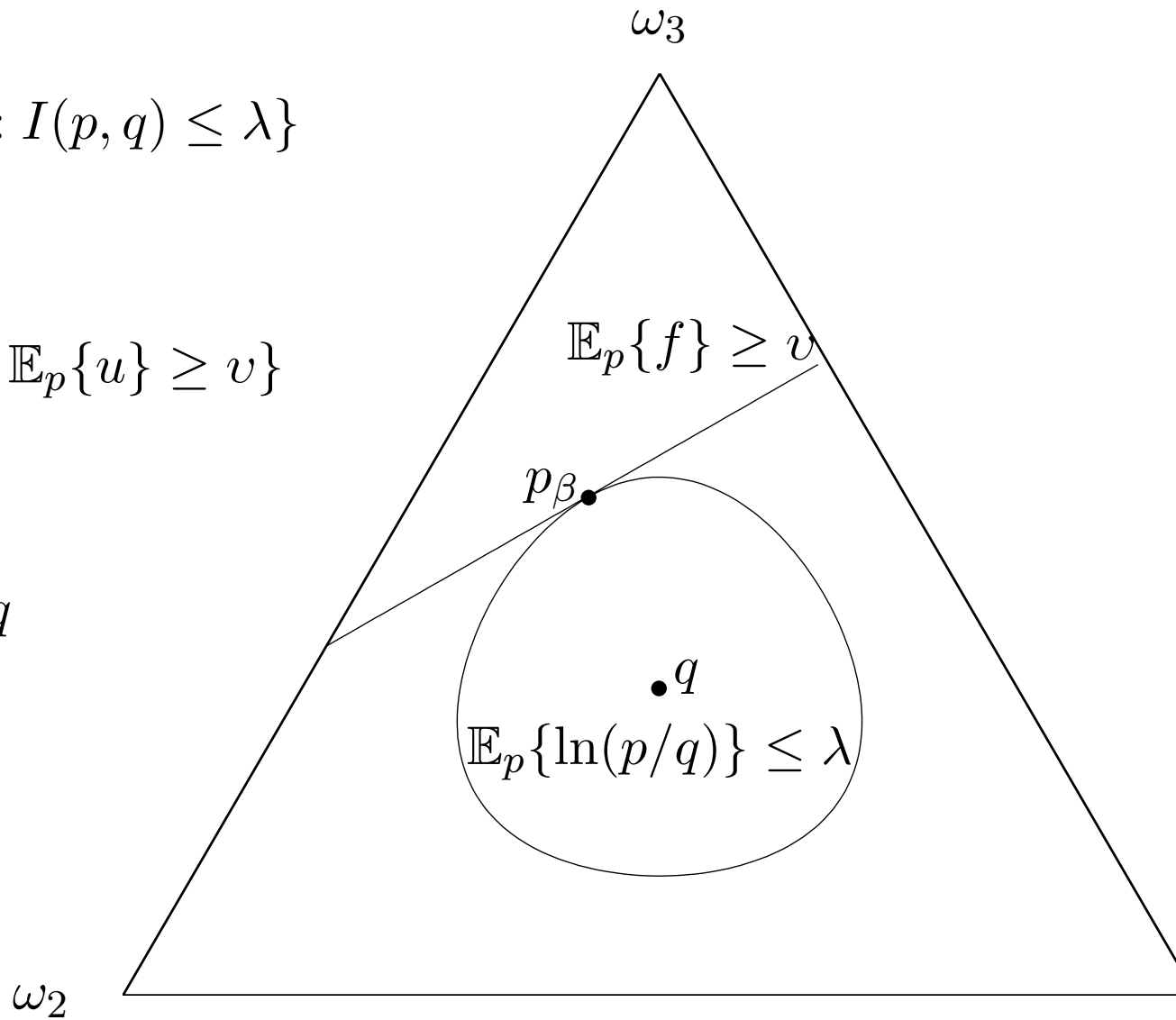
  $V(\lambda) := \sup\{\mathbb{E}_p\{u\} : I(p,q) \le \lambda\}$

- Minimize $I(p,q)$:

  $\lambda(V) := \inf\{I(p,q) : \mathbb{E}_p\{u\} \ge \upsilon\}$

- Optimal solutions:

  $p(\beta) = e^{\beta\, u - \Psi(\beta)}\, q$



Information-geometruc view (cont)

- Maximize $\mathbb{E}_p\{u\}$

  $V(\lambda) := \sup\{\mathbb{E}_p\{u\} : I(p,q) \le \lambda\}$

- Minimize $I(p,q)$:

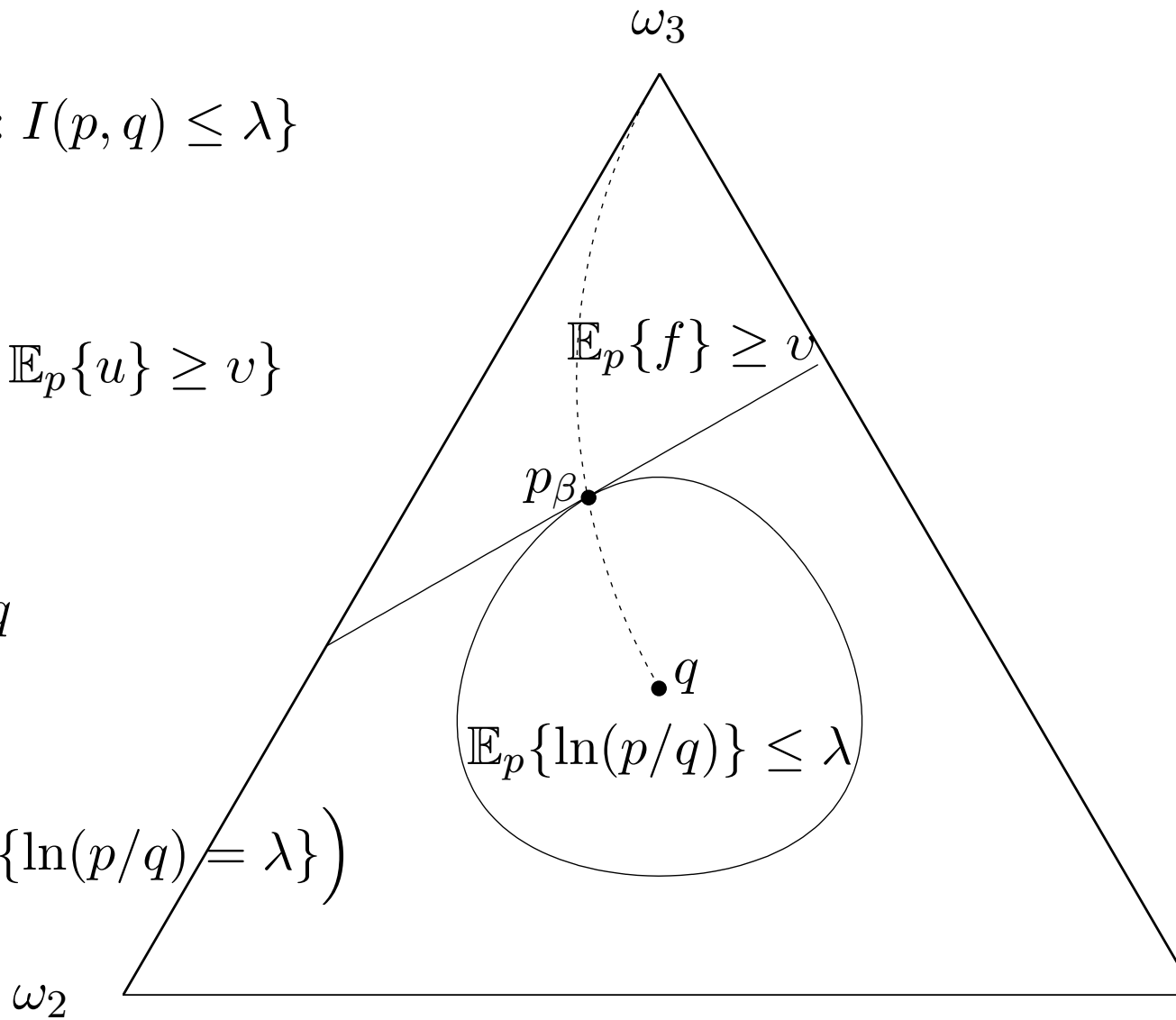
  $\lambda(V) := \inf\{I(p,q) : \mathbb{E}_p\{u\} \ge v\}$

- Optimal solutions:

  $$p(\beta) = e^{\beta\, u - \Psi(\beta)}\, q$$

- Constraints

  $\mathbb{E}_{p(\beta)}\{u\} = v, \quad \Big(\mathbb{E}_p\{\ln(p/q) = \lambda\}\Big)$



Solution

- Lagrange function

$$K(p, \beta) = \mathbb{E}_p\{\ln(p/q)\} + \beta[\upsilon - \mathbb{E}_p\{u\}]$$

Solution

- Lagrange function

$$K(p,\beta) = \mathbb{E}_p\{\ln(p/q)\} + \beta[\upsilon - \mathbb{E}_p\{u\}]$$

- Necessary and sufficient conditions $\nabla K(p,\beta) = 0$:

$$\begin{aligned} \nabla_p K(p,\beta) &= \ln(p/q) + 1 - \beta u = 0 \\ \nabla_\beta K(p,\beta) &= \upsilon - \mathbb{E}_p\{u\} = 0 \end{aligned}$$

Solution

- Lagrange function

$$K(p,\beta) = \mathbb{E}_p\{\ln(p/q)\} + \beta[\upsilon - \mathbb{E}_p\{u\}]$$

- Necessary and sufficient conditions $\nabla K(p,\beta) = 0$:

$$\begin{aligned} \nabla_p K(p,\beta) &= \ln(p/q) + 1 - \beta u = 0 \\ \nabla_\beta K(p,\beta) &= \upsilon - \mathbb{E}_p\{u\} = 0 \end{aligned}$$

- Optimal solutions:

$$p(\beta) = e^{\beta\, u - \Psi(\beta)}\, q\,, \qquad \mathbb{E}_{p(\beta)}\{u\} = \upsilon \quad \Big(\mathbb{E}_p\{\ln(p/q) = \lambda\}\Big)$$

Solution

- Lagrange function

$$K(p,\beta) = \mathbb{E}_p\{\ln(p/q)\} + \beta[\upsilon - \mathbb{E}_p\{u\}]$$

- Necessary and sufficient conditions $\nabla K(p,\beta) = 0$:

$$\begin{aligned} \nabla_p K(p,\beta) &= \ln(p/q) + 1 - \beta u = 0 \\ \nabla_\beta K(p,\beta) &= \upsilon - \mathbb{E}_p\{u\} = 0 \end{aligned}$$

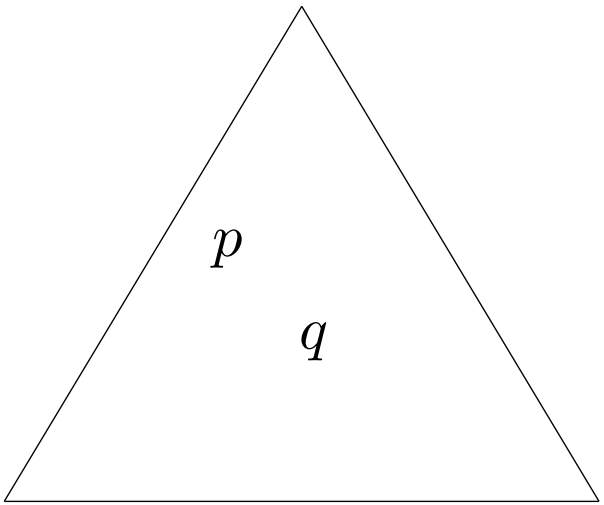
- Optimal solutions:

$$p(\beta) = e^{\beta\, u - \Psi(\beta)}\, q\,, \qquad \mathbb{E}_{p(\beta)}\{u\} = \upsilon \quad \Big(\mathbb{E}_p\{\ln(p/q) = \lambda\}\Big)$$

- Optimal *inverse temperature* β:

$$\beta = \frac{dK(\upsilon)}{d\upsilon} \quad \text{or} \quad \beta^{-1} = \frac{dV(\lambda)}{d\lambda}$$

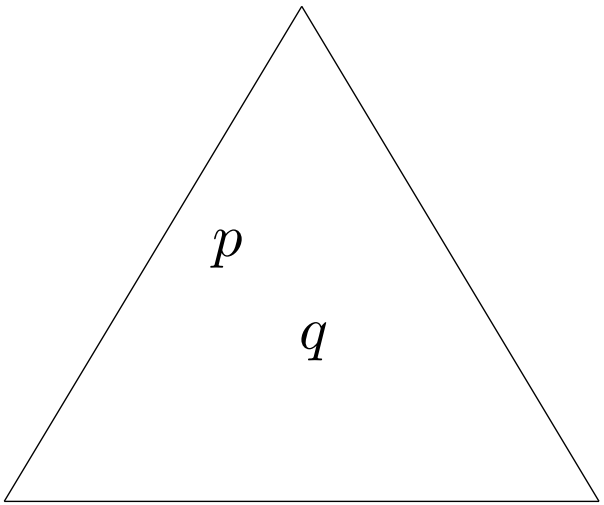
Variational problems for composite systems



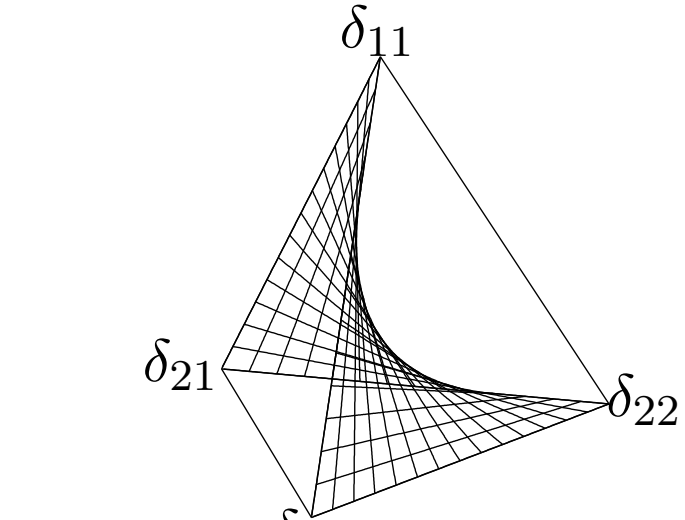
$\mathcal{P}(X)$

type I : Find optimal $p \in \mathcal{P}(X)$

Variational problems for composite systems



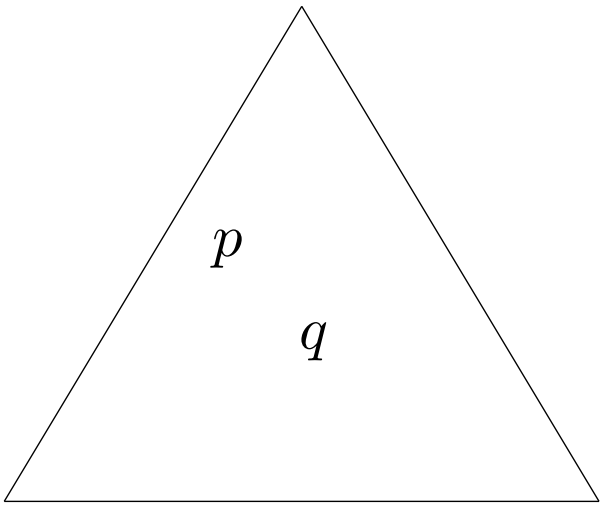
$\mathcal{P}(X)$



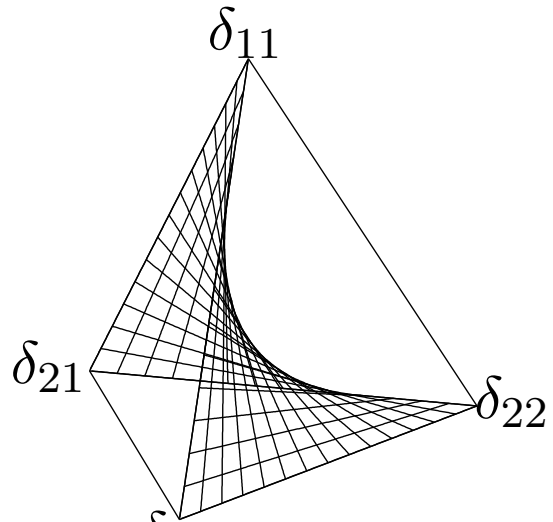
$\mathcal{P}(X \otimes Z)$

type I : Find optimal $p \in \mathcal{P}(X)$

Variational problems for composite systems



$\mathcal{P}(X)$

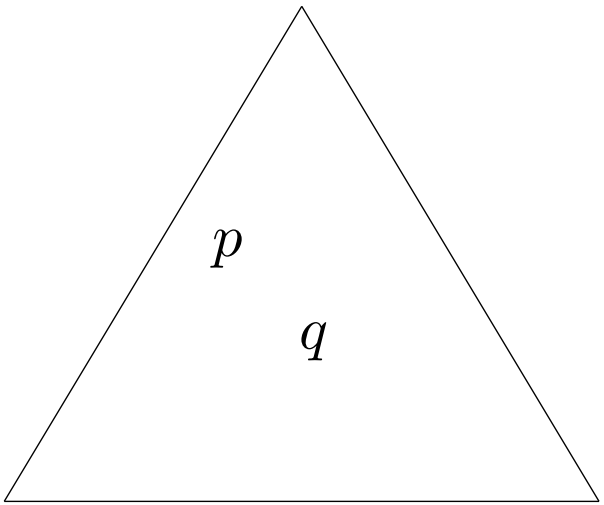


$\mathcal{P}(X \otimes Z)$

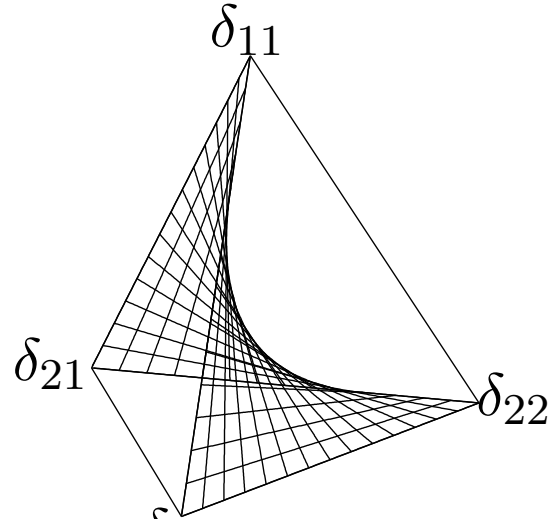
type I : Find optimal $p \in \mathcal{P}(X)$

type II : Find optimal input (marginal) $q \in \mathcal{P}(X)$ for fixed channel $\Lambda : \mathcal{P}(X) \to \mathcal{P}(Z)$.

Variational problems for composite systems



$\mathcal{P}(X)$



$\mathcal{P}(X \otimes Z)$

type I : Find optimal $p \in \mathcal{P}(X)$

type II : Find optimal input (marginal) $q \in \mathcal{P}(X)$ for fixed channel $\Lambda : \mathcal{P}(X) \to \mathcal{P}(Z)$.

type III : Find optimal channel $\Lambda : \mathcal{P}(X) \to \mathcal{P}(Z)$ for fixed marginal $q \in \mathcal{P}(X)$.

Third variational problem

- Linear programming problem:

$$\text{maximize} \quad \mathbb{E}_{P(z|x)}\{u(x,z)\} \quad \text{subject to} \quad I(X,Z) \leq \lambda$$

Third variational problem

- Linear programming problem:

$$\text{maximize} \quad \mathbb{E}_{P(z|x)}\{u(x,z)\} \quad \text{subject to} \quad I(X,Z) \leq \lambda$$

- The inverse convex programming problem:

$$\text{minimize} \quad I(X,Z) \quad \text{subject to} \quad \mathbb{E}_{P(z|x)}\{u(x,z)\} \geq \upsilon$$

Third variational problem

- Linear programming problem:

$$\text{maximize} \quad \mathbb{E}_{P(z|x)}\{u(x,z)\} \quad \text{subject to} \quad I(X,Z) \leq \lambda$$

- The inverse convex programming problem:

$$\text{minimize} \quad I(X,Z) \quad \text{subject to} \quad \mathbb{E}_{P(z|x)}\{u(x,z)\} \geq v$$

- Information:

$$I(X,Z) = H(Z) - H(Z \mid X) \leq H(Z) \leq \ln |Z|$$

Optimal transition kernels

- Optimal solutions achieving $V(\lambda)$ have exponential form, such as:

$$P(z \mid x) = \frac{P(z)\, e^{\beta u(x,z)}}{\sum_z P(z)\, e^{\beta u(x,z)}}$$

Optimal transition kernels

- Optimal solutions achieving $V(\lambda)$ have exponential form, such as:

$$P(z \mid x) = \frac{e^{\beta u(x,z)}}{\sum_z e^{\beta u(x,z)}}$$

- $\beta >$ is called *inverse temperature*, and it is the Lagrange multiplier related to the information constraint:

$$I\{x, z\} \leq \lambda$$

Optimal transition kernels

- Optimal solutions achieving $V(\lambda)$ have exponential form, such as:

$$P(z \mid x) = \frac{e^{\beta u(x,z)}}{\sum_z e^{\beta u(x,z)}}$$

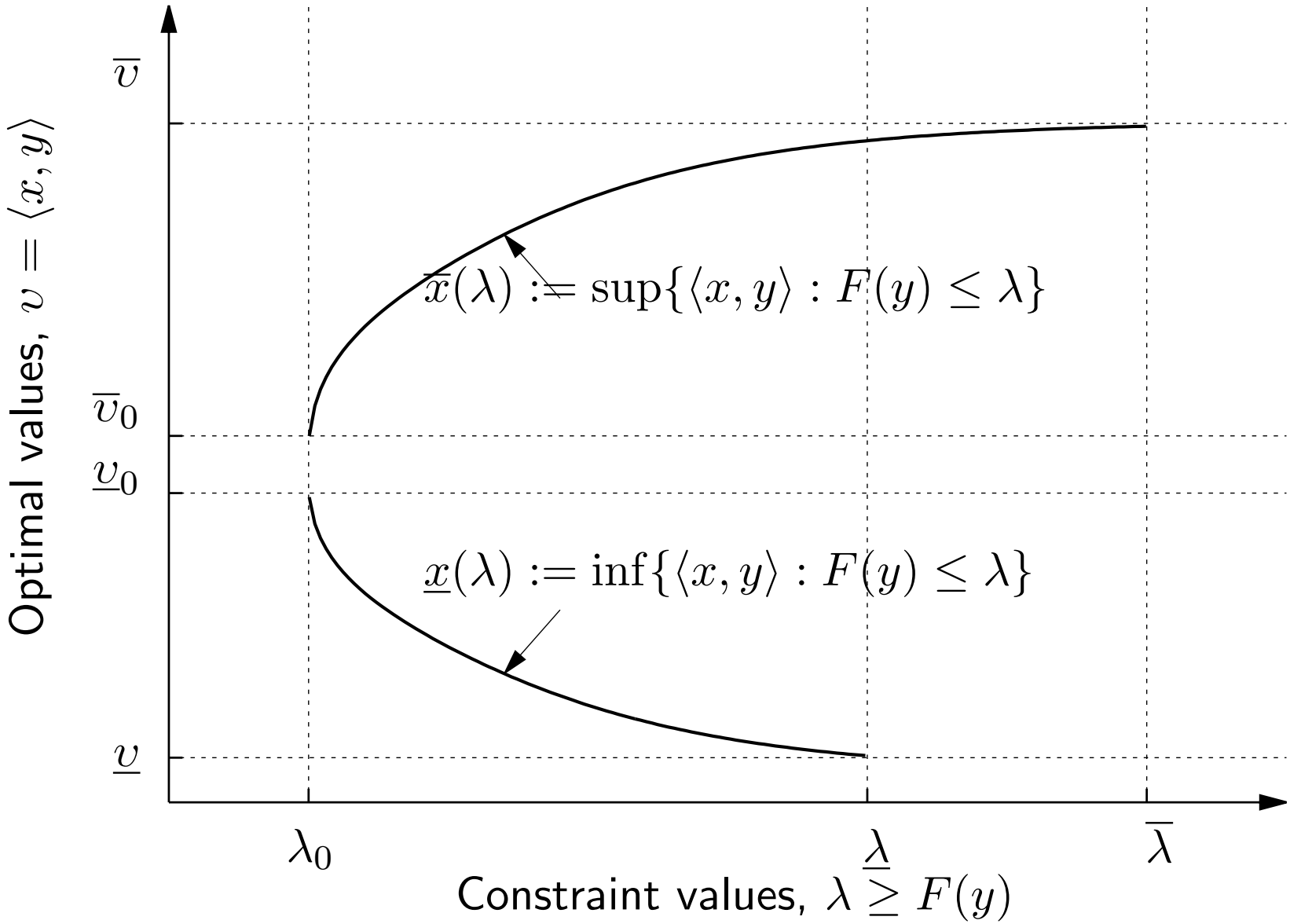
- $\beta >$ is called *inverse temperature*, and it is the Lagrange multiplier related to the information constraint:

$$I\{x, z\} \leq \lambda$$

- One can show that temperature β^{-1} is the slope of $V(\lambda)$:

$$\beta^{-1} = \frac{dV(\lambda)}{d\lambda}$$

Concave and convex value functions



Value of Hartley's information

Feed-forward neural networks and VoI

Forecast and VoI

Values of Boltzmann's and Shannon's information

Solutions of VoI

McCulloch, W., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115-133.

Shannon, C. E. (1948, July and October). A mathematical theory of communication. *Bell System Technical Journal*, *27*, 379–423 and 623–656.

Stratonovich, R. L. (1965). On value of information. *Izvestiya of USSR Academy of Sciences, Technical Cybernetics*, *5*, 3–12. (In Russian)

Stratonovich, R. L. (1968). Optimal algorithms for pattern recognition. *Automatics and Telemechanics*, *2*. (In Russian)

Stratonovich, R. L. (1969). Effectiveness of statistical methods in problems of synthesis of algorithms for function approximation. *Izvestiya of USSR Academy of Sciences, Technical Cybernetics*, *1*. (In Russian)

Stratonovich, R. L. (1970a). Canonical recursive system of equations for optimal adaptive algorithms. *Automatics and Telemechanics*, *5*. (In Russian)

Stratonovich, R. L. (1970b). Optimal expansion of functional subspace in algorithms for function and probability density approximation.

Izvestiya of USSR Academy of Sciences, Technical Cybernetics, *2*. (In Russian)

Stratonovich, R. L. (1975). *Theory of information*. Moscow, USSR: Sovetskoe Radio. (In Russian)