Business Analytics

1. Prediction, 1.1 Tasks and Error Measures

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim, Germany

Outline

Outline

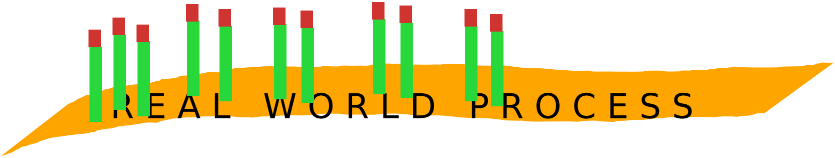
The Prediction Problem Informally

REAL WORLD PROCESS

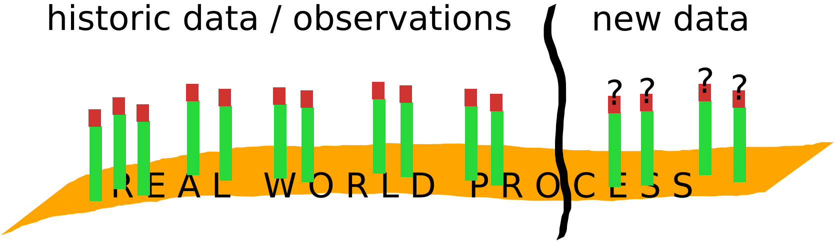
The Prediction Problem Informally

The Prediction Problem Informally

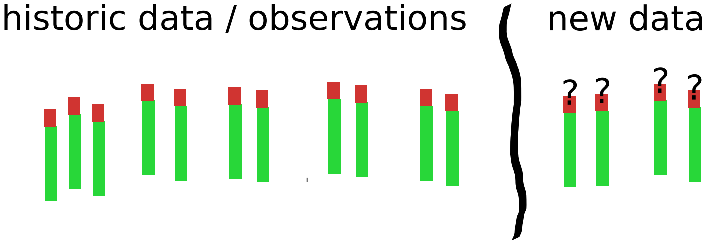
historic data / observations



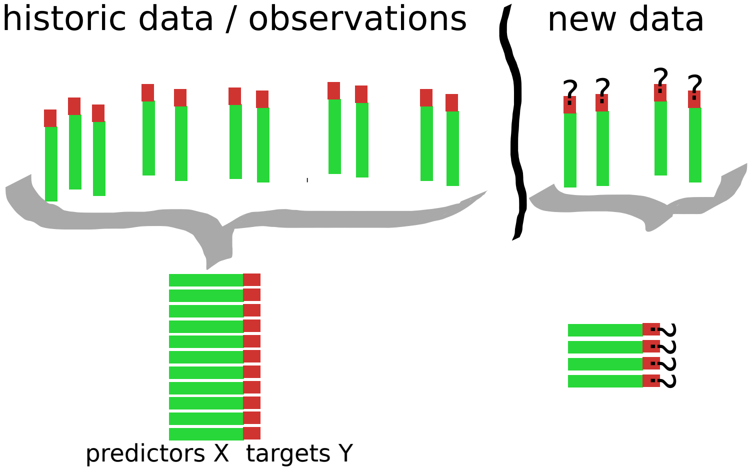
The Prediction Problem Informally



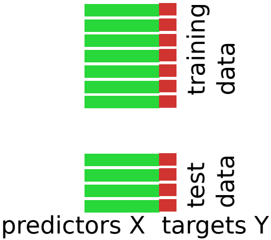
The Prediction Problem Informally



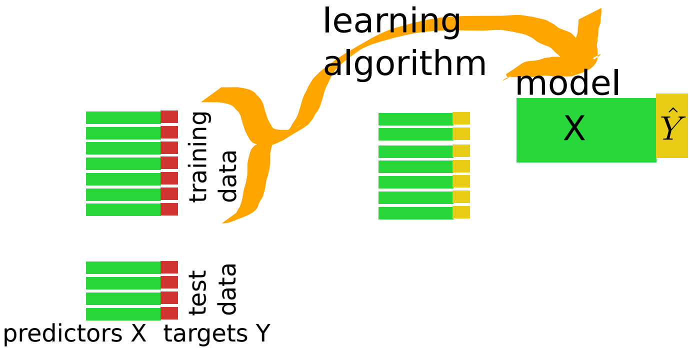
The Prediction Problem Informally



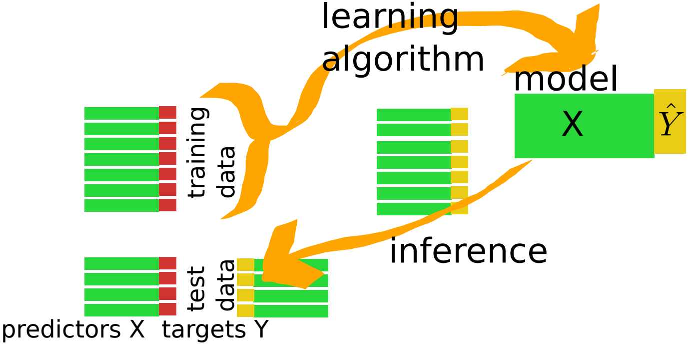
The Prediction Problem Informally



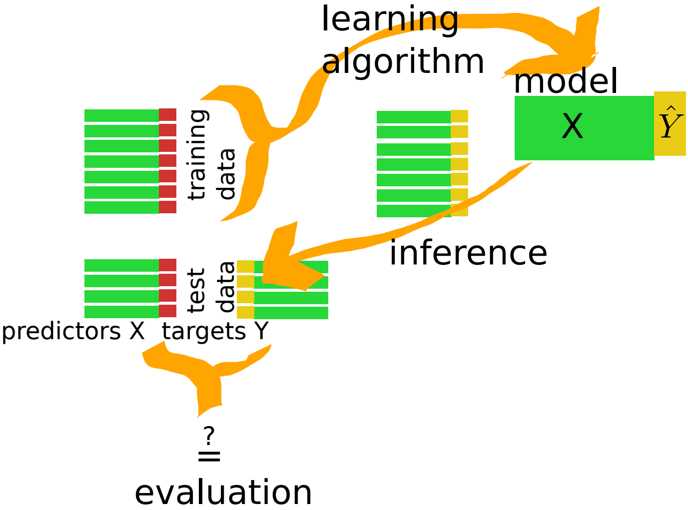
The Prediction Problem Informally



The Prediction Problem Informally



The Prediction Problem Informally



The Prediction Problem Formally

Let $\mathcal{X}$ be any set (called **predictor space**),
$\mathcal{Y}$ be any set (called **target space**), and
$p : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_0^+$ be an unknown joint distribution / density.

Given

- a sample $\mathcal{D}^{\text{train}} \subseteq \mathcal{X} \times \mathcal{Y}$ (called **training set**), drawn from p,
- a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ that measures how bad it is to predict value $\hat{y}$ if the true value is y,

compute a **prediction function**

$$\hat{y} : \mathcal{X} \rightarrow \mathcal{Y}$$

with minimal risk

$$\text{risk}(\hat{y}; p) := \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, \hat{y}(x))\, p(x, y)\, d(x, y)$$

Explanation: $\text{risk}(\hat{y}; p)$ can be estimated by the **empirical risk**

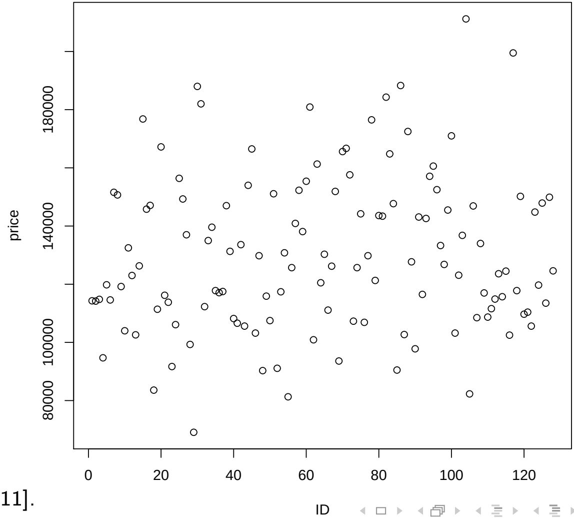
$$\text{risk}(\hat{y}; \mathcal{D}^{\text{test}}) := \frac{1}{|D^{\text{test}}|} \sum_{(x,y) \in \mathcal{D}^{\text{test}}} \ell(y, \hat{y}(x))$$

Outline

Example: House Prices

train set $\mathcal{D}^{\text{train}}$:

price [$]
114,300
114,200
114,800
94,700
119,800
114,600
151,600
150,700
119,200
104,000
⋮

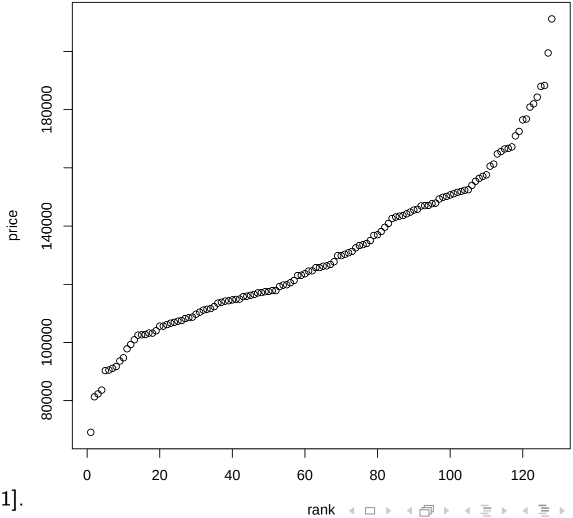


Note: Data set from [Jan11].

Example: House Prices

train set $\mathcal{D}^{\text{train}}$:

price [$]
114,300
114,200
114,800
94,700
119,800
114,600
151,600
150,700
119,200
104,000
⋮

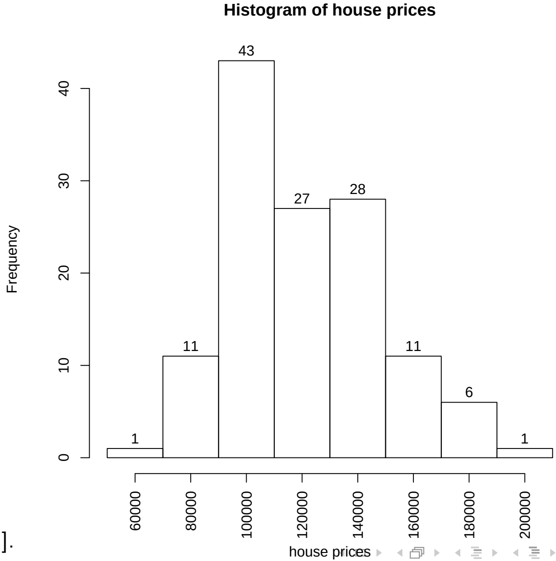


Note: Data set from [Jan11].

Example: House Prices

train set $\mathcal{D}^{\text{train}}$:

price [$]
114,300
114,200
114,800
94,700
119,800
114,600
151,600
150,700
119,200
104,000
⋮



Note: Data set from [Jan11].

Prediction (without Predictors)

Let $\mathcal{Y}$ be any set (called **target space**), and

$p : \mathcal{Y} \rightarrow \mathbb{R}_0^+$ be a distribution / density.

Given

- a sample $\mathcal{D}^{\text{train}} \subseteq \mathcal{Y}$ (called **training set**), drawn from p,
- a **loss** function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ that measures how bad it is to predict value $\hat{y}$ if the true value is y,

compute a **predicted value**

$$\hat{y} \in \mathcal{Y}$$

with minimal **risk**

$$\text{risk}(\hat{y}; p) := \int_{\mathcal{Y}} \ell(y, \hat{y})\, p(y)\, dy$$

Explanation: $\text{risk}(\hat{y}; p)$ can be estimated by the **empirical risk**

$$\text{risk}(\hat{y}; \mathcal{D}^{\text{test}}) := \frac{1}{|D^{\text{test}}|} \sum_{y \in \mathcal{D}^{\text{test}}} \ell(y, \hat{y})$$

Example: House Prices

- Target space: $\mathcal{Y} := \mathbb{R}_0^+$
- Loss: $\ell(y, \hat{y}) := (y - \hat{y})^2$
- Training set: $\mathcal{D}^{\text{train}} := \{114300, 114200, 114800, 94700, 119800, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{188300, 102700, 172500, 127700, \ldots\}$

Given some **sample house prices** $\mathcal{D}^{\text{train}}$, compute$^{*)}$ a **predicted house price** $\hat{y}$ with minimal **Root Mean Squared Error (RMSE)**:

$$\text{RMSE}(\mathcal{D}^{\text{test}}, \hat{y}) := \sqrt{\frac{1}{|\mathcal{D}^{\text{test}}|} \sum_{y \in \mathcal{D}^{\text{test}}} (y - \hat{y})^2}$$

for house prices $\mathcal{D}^{\text{test}}$ observed in the future.

Note: $^{*)}$ without using $\mathcal{D}^{\text{test}}$.

Prediction with Squared Loss

The **prediction problem with squared loss** $\ell(y, \hat{y}) := (y - \hat{y})^2$ minimizes **Mean Squared Error (MSE)** / **Root Mean Squared Error (RMSE)**:

$$\text{MSE}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{|\mathcal{D}^{\text{test}}|} \sum_{y \in \mathcal{D}^{\text{test}}} (y - \hat{y})^2$$

$$\text{RMSE}(\mathcal{D}^{\text{test}}, \hat{y}) := \sqrt{\frac{1}{|\mathcal{D}^{\text{test}}|} \sum_{y \in \mathcal{D}^{\text{test}}} (y - \hat{y})^2}$$

Lemma

The predicted value with minimal squared loss / RMSE is the mean:

$$\hat{y} := \frac{1}{|\mathcal{D}^{train}|} \sum_{y \in \mathcal{D}^{train}} y$$

Prediction with Squared Loss

Lemma

The predicted value with minimal squared loss / RMSE is the mean:

$$\hat{y} := \frac{1}{|\mathcal{D}^{train}|} \sum_{y \in \mathcal{D}^{train}} y$$

Proof.

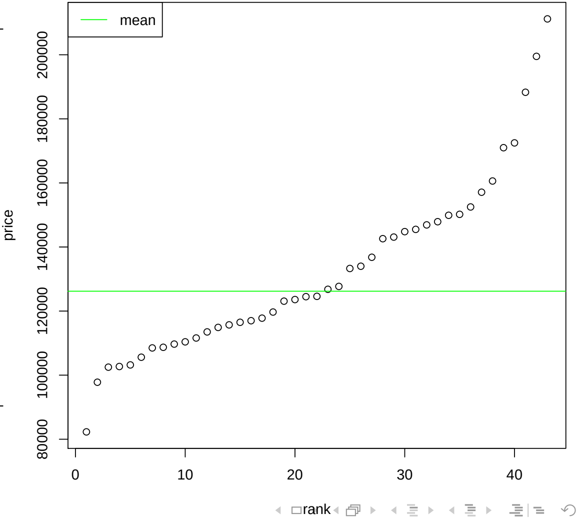
$$\frac{\partial \text{MSE}}{\partial \hat{y}} = \frac{1}{|\mathcal{D}^{\text{train}}|} \sum_{y \in \mathcal{D}^{\text{train}}} -2(y - \hat{y}) \stackrel{!}{=} 0$$

$$\rightsquigarrow \frac{1}{|\mathcal{D}^{\text{train}}|} \sum_{y \in \mathcal{D}^{\text{train}}} y - \frac{1}{|\mathcal{D}^{\text{train}}|} |\mathcal{D}^{\text{train}}| \hat{y} = 0$$

Evaluation: House Prices

test set $\mathcal{D}^{\text{test}}$:

y (price [\$])	$\hat{y}$
188,300	129,395.3
102,700	129,395.3
172,500	129,395.3
127,700	129,395.3
97,800	129,395.3
143,100	129,395.3
116,500	129,395.3
142,600	129,395.3
157,100	129,395.3
⋮	⋮



RMSE: 27,515.84
MAE: 21,541.31

Example: House Prices II

- Target space: $\mathcal{Y} := \mathbb{R}_0^+$
- Loss: $\ell(y, \hat{y}) := |y - \hat{y}|$
- Training set: $\mathcal{D}^{\text{train}} := \{114300, 114200, 114800, 94700, 119800, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{188300, 102700, 172500, 127700, \ldots\}$

Given some **sample house prices** $\mathcal{D}^{\text{train}}$, compute$^{*)}$ a **predicted house price** $\hat{y}$ with minimal **Mean Absolute Error (MAE)**:

$$\text{MAE}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} |y - \hat{y}|$$

for house prices $\mathcal{D}^{\text{test}}$ observed in the future.

Note: $^{*)}$ without using $\mathcal{D}^{\text{test}}$.

Prediction with Absolute Loss

The **prediction problem with absolute loss** $\ell(y, \hat{y}) := |y - \hat{y}|$ minimizes **Mean Absolute Error (MAE)**:

$$\text{MAE}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{|\mathcal{D}^{\text{train}}|} \sum_{y \in \mathcal{D}^{\text{test}}} |y - \hat{y}|$$

Lemma

The predicted value with minimal absolute error / MAE is the median:

$$\hat{y} := \textit{median}\ \mathcal{D}^{\textit{train}} := \begin{cases} y_{((n+1)/2)}, & \textit{for } n \textit{ odd} \\ \frac{1}{2}(y_{(n/2)} + y_{(n/2+1)}), & \textit{for } n \textit{ even} \end{cases}$$

with $\mathcal{D}^{\textit{train}} = \{y_{(1)}, \ldots, y_{(n)}\}$ *and* $y_{(i)}$ *sorted increasingly.*

Prediction with Absolute Loss

Lemma

The predicted value with minimal absolute error / MAE is the median:

$$\hat{y} := \textit{median}\ \mathcal{D}^{\textit{train}} := \begin{cases} y_{((n+1)/2)}, & \textit{for } n \textit{ odd} \\ \frac{1}{2}(y_{(n/2)} + y_{(n/2+1)}), & \textit{for } n \textit{ even} \end{cases}$$

with $\mathcal{D}^{\textit{train}} = \{y_{(1)}, \ldots, y_{(n)}\}$ *and* $y_{(i)}$ *sorted increasingly.*

Proof.

$$\frac{\partial \text{MAE}}{\partial \hat{y}} = \frac{1}{|\mathcal{D}^{\text{train}}|} \left(\sum_{y \in \mathcal{D}^{\text{train}}: y > \hat{y}} -1 + \sum_{y \in \mathcal{D}^{\text{train}}: y < \hat{y}} 1 \right) \stackrel{!}{=} 0$$

$\rightsquigarrow$ there have to be as many y's smaller than $\hat{y}$ as larger than $\hat{y}$. □

Evaluation: House Prices II

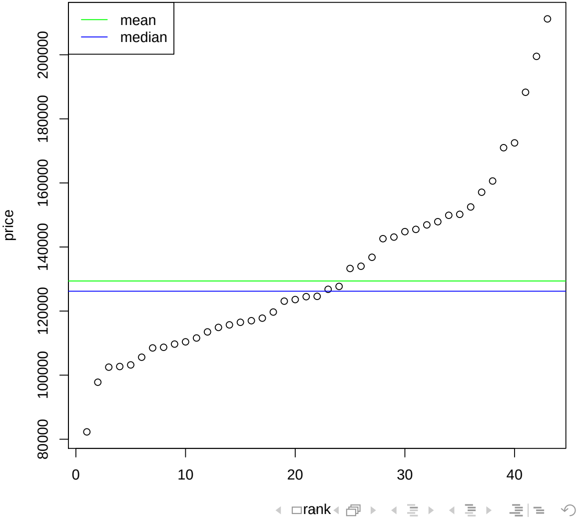
test set $\mathcal{D}^{\text{test}}$:

y (price [$])	$\hat{y}$
188,300	126,200
102,700	126,200
172,500	126,200
127,700	126,200
97,800	126,200
143,100	126,200
116,500	126,200
142,600	126,200
157,100	126,200
⋮	⋮

MAE: 21,267.44

RMSE: 28,052.88

$\text{MAE}_{\epsilon=5000}$: 16,706.98



Example: House Prices III

- Target space: $\mathcal{Y} := \mathbb{R}_0^+$
- Loss: $\ell(y, \hat{y}) := [|y - \hat{y}| - \epsilon]_0$ for $\epsilon := 5000$
- Training set: $\mathcal{D}^{\text{train}} := \{114300, 114200, 114800, 94700, 119800, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{188300, 102700, 172500, 127700, \ldots\}$

Given some **sample house prices** $\mathcal{D}^{\text{train}}$, compute*) a **predicted house price** $\hat{y}$ with minimal **ϵ-insensitive error**:

$$\text{MAE}_\epsilon(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} [|y - \hat{y}| - \epsilon]_0$$

for house prices $\mathcal{D}^{\text{test}}$ observed in the future.

Note: *) without using $\mathcal{D}^{\text{test}}$. $[x]_0 := \max(x, 0)$.

Prediction with ϵ-insensitive error

For given $\epsilon \in \mathbb{R}_0^+$, the **prediction problem with ϵ-insensitive error** $\ell(y, \hat{y}) := [|y - \hat{y}| - \epsilon]_0$ minimizes the **ϵ-insensitive error**:

$$\text{MAE}_\epsilon(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} [|y - \hat{y}| - \epsilon]_0$$

Lemma

The predicted value with minimal ϵ-insensitive error is:

$$\hat{y} := \frac{1}{2}(y_{(i)} + y_{(n-i+1)}) \quad \textit{with } i := \max\{i = 1, \ldots, n \mid y_{(n-i+1)} - y_{(i)} > 2\epsilon\}$$

with $\mathcal{D}^{train} = \{y_{(1)}, \ldots, y_{(n)}\}$ and $y_{(i)}$ sorted increasingly.

Prediction with ϵ-insensitive error

Lemma

The predicted value with minimal ϵ-insensitive error is:

$$\hat{y} := \frac{1}{2}(y_{(i)} + y_{(n-i+1)}) \quad \text{with } i := \max\{i = 1, \ldots, n \mid y_{(n-i+1)} - y_{(i)} > 2\epsilon\}$$

with $\mathcal{D}^{train} = \{y_{(1)}, \ldots, y_{(n)}\}$ and $y_{(i)}$ sorted increasingly.

Proof.

$$\frac{\partial \text{MAE}_\epsilon}{\partial \hat{y}} = \frac{1}{|\mathcal{D}^{\text{train}}|} \left(\sum_{y \in \mathcal{D}^{\text{train}}: y > \hat{y}+\epsilon} -1 + \sum_{y \in \mathcal{D}^{\text{train}}: y < \hat{y}-\epsilon} 1 \right) \stackrel{!}{=} 0$$

$\rightsquigarrow$ there have to be as many y's smaller than $\hat{y} - \epsilon$ as larger than $\hat{y} + \epsilon$. □

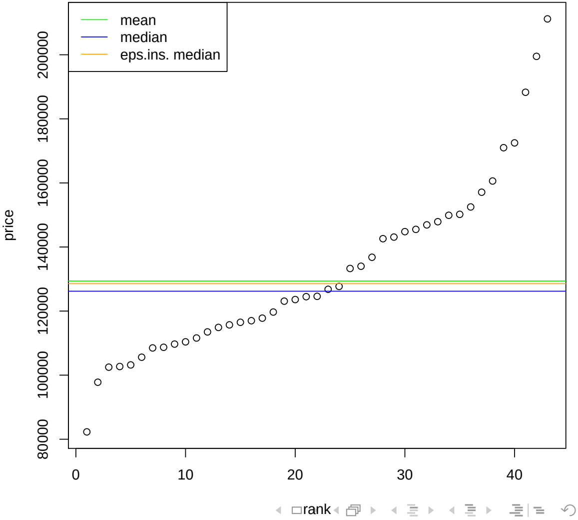
Evaluation: House Prices III

test set $\mathcal{D}^{\text{test}}$:

y (price [\$])	$\hat{y}$
188,300	128,550
102,700	128,550
172,500	128,550
127,700	128,550
97,800	128,550
143,100	128,550
116,500	128,550
142,600	128,550
157,100	128,550
⋮	⋮

MAE: 21,443.02

$\text{MAE}_{\epsilon=5000}$: 16,668.60

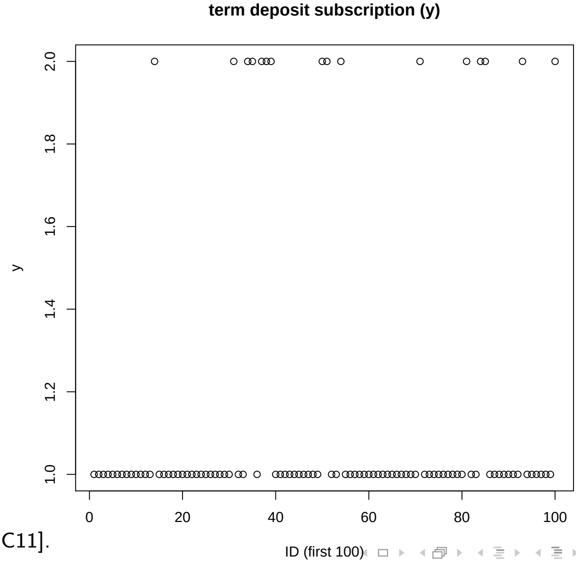


Outline

Example: Direct Bank Marketing

train set $\mathcal{D}^{\text{train}}$: subscription
no
no
no
⋮
no
yes
no
no
no
no
⋮

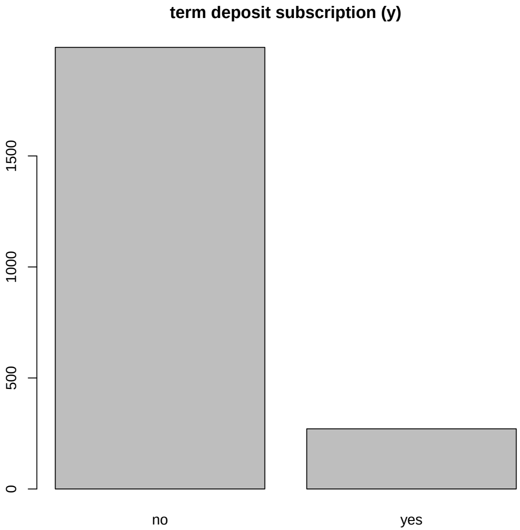
Note: Data set from [MLC11].



Example: Direct Bank Marketing

train set $\mathcal{D}^{\text{train}}$:

subscription
no
no
no
⋮
no
yes
no
no
no
no
⋮



Note: Data set from [MLC11].

Example: Direct Bank Marketing I

- Target space: $\mathcal{Y} := \{\text{no}, \text{yes}\} = \{0, 1\}$
- Loss: $\ell(y, \hat{y}) := \delta(y \neq \hat{y})$
- Training set: $\mathcal{D}^{\text{train}} := \{0, 0, 0, \ldots, 0, 1, 0, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{0, 0, 0, \ldots, 0, 1, 0, \ldots\}$

Given some **customer responses** $\mathcal{D}^{\text{train}}$, compute$^{*)}$ a **predicted customer response** $\hat{y}$ with minimal **misclassification rate**:

$$\text{MR}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} \delta(y \neq \hat{y})$$

for customer responses $\mathcal{D}^{\text{test}}$ observed in the future.

Note: $^{*)}$ without using $\mathcal{D}^{\text{test}}$. $\delta(A) := \begin{cases} 1, & \text{if } A \text{ is true} \\ 0, & \text{else} \end{cases}$

Prediction with 0/1 loss (binary classification)

The **prediction problem with 0/1 loss** (**binary classification**) $\ell(y, \hat{y}) := \delta(y \neq \hat{y})$ minimizes the **misclassification rate**:

$$\text{MR}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} \delta(y \neq \hat{y})$$

Lemma

The predicted value with minimal misclassification rate is the **majority class***:*

$$\hat{y} := \begin{cases} 1, & \textit{if } \hat{n}_1 > \hat{n}_0 \\ 0, & \textit{else} \end{cases}$$

$$\textit{with } \hat{n}_y := |\{y' \in \mathcal{D}^{train} \mid y' = y\}|, \quad y \in \mathcal{Y} := \{0, 1\}$$

Note: Equivalent to minimizing MR is maximizing **accuracy** $\text{acc}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} \delta(y = \hat{y})$.

Prediction with 0/1 loss (binary classification)

Lemma

The predicted value with minimal misclassification rate is the **majority class***:*

$$\hat{y} := \begin{cases} 1, & \text{if } \hat{n}_1 > \hat{n}_0 \\ 0, & \text{else} \end{cases}$$

$$\text{with } \hat{n}_y := |\{y' \in \mathcal{D}^{train} \mid y' = y\}|, \quad y \in \mathcal{Y} := \{0, 1\}$$

Proof.

$$\text{MR}(\mathcal{D}^{\text{train}}, \hat{y} = 0) = \frac{|\mathcal{D}^{\text{train}}| - \hat{n}_0}{|\mathcal{D}^{\text{train}}|}$$

$$\text{MR}(\mathcal{D}^{\text{train}}, \hat{y} = 1) = \frac{|\mathcal{D}^{\text{train}}| - \hat{n}_1}{|\mathcal{D}^{\text{train}}|}$$

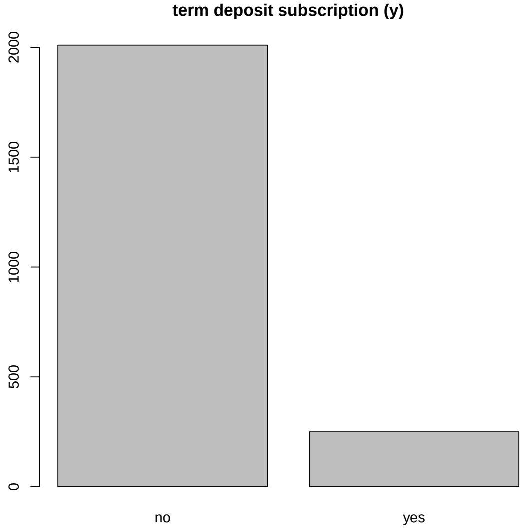
$\rightsquigarrow$ minimal for $\hat{y}$ with maximal $\hat{n}_{\hat{y}}$. □

Evaluation: Direct Bank Marketing

test set $\mathcal{D}^{\text{test}}$:

y	$\hat{y}$
no	no
no	no
no	no
⋮	⋮
no	no
yes	no
no	no
no	no
no	no
⋮	⋮

MR: 0.111



Example: Direct Bank Marketing II

- Target space: $\mathcal{Y} := \{\text{no}, \text{yes}\} = \{0, 1\}$
- Loss: $\ell(y, \hat{y}) := \delta(y \neq \hat{y}) c_{y,\hat{y}}$, for $c_{0,1} := 1, c_{1,0} := 20$.
- Training set: $\mathcal{D}^{\text{train}} := \{0, 0, 0, \ldots, 0, 1, 0, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{0, 0, 0, \ldots, 0, 1, 0, \ldots\}$

Given some **customer responses** $\mathcal{D}^{\text{train}}$, compute$^{*)}$ a **predicted customer response** $\hat{y}$ with minimal **misclassification cost**:

$$\text{cost}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} \delta(y \neq \hat{y}) c_{y,\hat{y}}$$

for customer responses $\mathcal{D}^{\text{test}}$ observed in the future.

Note: $^{*)}$ without using $\mathcal{D}^{\text{test}}$.

Prediction with misclassification cost

Given misclassification costs $c_{0,1}, c_{1,0} \in \mathbb{R}$, the **prediction problem with misclassification cost** (**cost-sensitive binary classification**) $\ell(y, \hat{y}) := \delta(y \neq \hat{y}) c_{y,\hat{y}}$ minimizes the **misclassification cost**:

$$\text{cost}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} \delta(y \neq \hat{y}) c_{y,\hat{y}}$$

Lemma

The predicted value with minimal misclassification cost is:

$$\hat{y} := \begin{cases} 1, & \textit{if } \hat{n}_1 c_{1,0} > \hat{n}_0 c_{0,1} \\ 0, & \textit{else} \end{cases}$$

$$\textit{with } \hat{n}_y := |\{y' \in \mathcal{D}^{\textit{train}} \mid y' = y\}|, \quad y \in \mathcal{Y} := \{0, 1\}$$

Note: The problem depends only on the **cost ratio** $c_{0,1}/c_{1,0}$.

Prediction with misclassification cost

Lemma

The predicted value with minimal misclassification cost is:

$$\hat{y} := \begin{cases} 1, & \textit{if } \hat{n}_1 c_{1,0} > \hat{n}_0 c_{0,1} \\ 0, & \textit{else} \end{cases}$$

$$\textit{with } \hat{n}_y := |\{y' \in \mathcal{D}^{train} \mid y' = y\}|, \quad y \in \mathcal{Y} := \{0, 1\}$$

Proof.

$$\mathrm{cost}(\mathcal{D}^{\mathrm{train}}, \hat{y} = 0) = \frac{\hat{n}_1 c_{1,0}}{|\mathcal{D}^{\mathrm{train}}|}$$

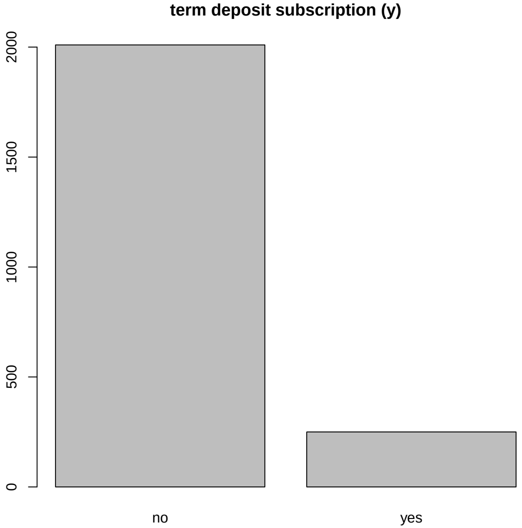
$$\mathrm{cost}(\mathcal{D}^{\mathrm{train}}, \hat{y} = 1) = \frac{\hat{n}_0 c_{0,1}}{|\mathcal{D}^{\mathrm{train}}|}$$

$\rightsquigarrow$ minimal for $\hat{y}$ with maximal $\hat{n}_{\hat{y}} c_{\hat{y},1-\hat{y}}$. $\square$

Evaluation: Direct Bank Marketing II

test set $\mathcal{D}^{\text{test}}$:

y	$\hat{y}$
no	yes
no	yes
no	yes
⋮	⋮
no	yes
yes	yes
no	yes
no	yes
no	yes
⋮	⋮



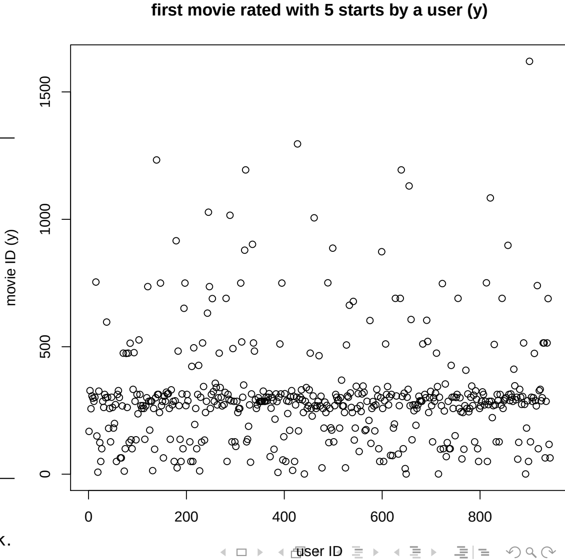
MR: 0.889
cost: 0.889
cost ("no"): 2.21

Outline

Example: First Highly-Rated Product

train set $\mathcal{D}^{\text{train}}$:

product name (movie)	ID
Monty Python and the Holy Grail (1974)	168
Conspiracy Theory (1997)	328
Men in Black (1997)	257
Devil's Advocate, The (1997)	307
Face/Off (1997)	298
English Patient, The (1996)	286
L.A. Confidential (1997)	302
Red Corner (1997)	754
⋮	



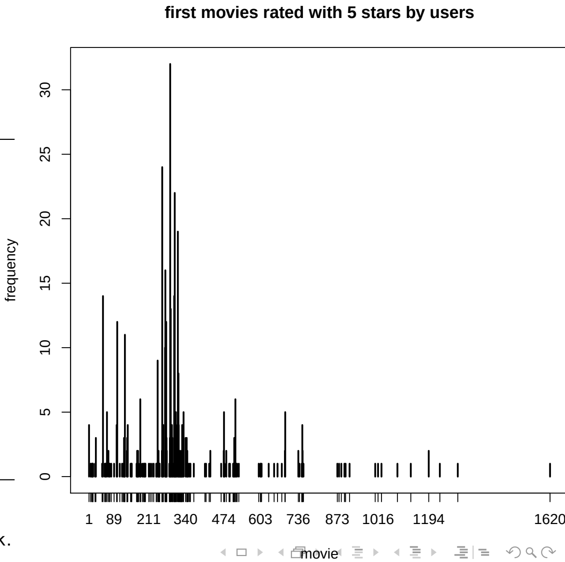
Note: Data derived from Movielens 100k.

Example: First Highly-Rated Product

train set $\mathcal{D}^{\text{train}}$:

product name (movie)	ID
Monty Python and the Holy Grail (1974)	168
Conspiracy Theory (1997)	328
Men in Black (1997)	257
Devil's Advocate, The (1997)	307
Face/Off (1997)	298
English Patient, The (1996)	286
L.A. Confidential (1997)	302
Red Corner (1997)	754
⋮	

Note: Data derived from Movielens 100k.



Example: First Highly-Rated Product

- Target space: $\mathcal{Y} := \{1, 2, \ldots, 1682\}$
- Loss: $\ell(y, \hat{y}) := \delta(y \neq \hat{y})$
- Training set: $\mathcal{D}^{\text{train}} := \{168, 328, 257, 307, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{275, 258, 127, 258, 654, \ldots\}$

Given some **first highly-rated products** $\mathcal{D}^{\text{train}}$, compute$^{*)}$ a **predicted first highly-rated products** $\hat{y}$ with minimal **misclassification rate**:

$$\text{MR}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} \delta(y \neq \hat{y})$$

for first highly-rated products $\mathcal{D}^{\text{test}}$ observed in the future.

Note: $^{*)}$ without using $\mathcal{D}^{\text{test}}$.

Prediction with 0/1 loss (multiclass classification)

The **prediction problem with 0/1 loss** (**multiclass classification**) $\ell(y, \hat{y}) := \delta(y \neq \hat{y})$ minimizes the **misclassification rate**:

$$\mathrm{MR}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} \delta(y \neq \hat{y})$$

Lemma

The predicted value with minimal misclassification rate is the **majority class***:*

$$\hat{y} := \arg\max_{y \in \mathcal{Y}} n_y$$
$$\textit{with } \hat{n}_y := |\{y' \in \mathcal{D}^{train} \mid y' = y\}|, \quad y \in \mathcal{Y}$$

Note: Equivalent to minimizing MR is maximizing **accuracy** $\mathrm{acc}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} \delta(y = \hat{y})$.

Prediction with 0/1 loss (multiclass classification)

Lemma
The predicted value with minimal misclassification rate is the **majority class***:*

$$\hat{y} := \arg\max_{y \in \mathcal{Y}} n_y$$

$$\textit{with } \hat{n}_y := |\{y' \in \mathcal{D}^{\textit{train}} \mid y' = y\}|, \quad y \in \mathcal{Y}$$

Proof.

$$\mathrm{MR}(\mathcal{D}^{\text{train}}, \hat{y}) = \frac{|\mathcal{D}^{\text{train}}| - \hat{n}_{\hat{y}}}{|\mathcal{D}^{\text{train}}|}$$

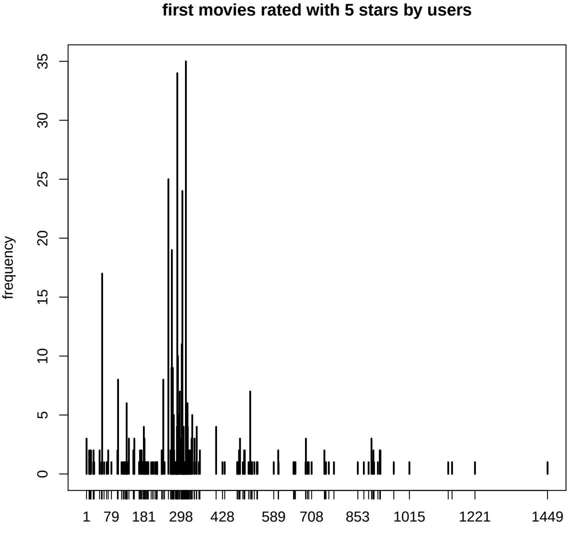
$\rightsquigarrow$ minimal for $\hat{y}$ with maximal $\hat{n}_{\hat{y}}$. □

Evaluation: First Highly-Rated Product

test set $\mathcal{D}^{\text{train}}$:

y	$\hat{y}$
275	286
258	286
127	286
258	286
654	286
69	286
151	286
302	286
286	286
148	286
⋮	

MR: 0.986



Example: First Highly-Rated Product II

- Target space: $\mathcal{Y} := \{1, 2, \ldots, 1682\}$
- Loss: $\ell(y, \hat{y}) := \delta(y \neq \hat{y}) c_{y,\hat{y}}$, for given $c_{y,\hat{y}}$.
- Training set: $\mathcal{D}^{\text{train}} := \{168, 328, 257, 307, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{275, 258, 127, 258, 654, \ldots\}$

Given some **first highly-rated products** $\mathcal{D}^{\text{train}}$, compute$^{*)}$ a **predicted first highly-rated product** $\hat{y}$ with minimal **misclassification cost**:

$$\text{cost}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} \delta(y \neq \hat{y}) c_{y,\hat{y}}$$

for first highly-rated products $\mathcal{D}^{\text{test}}$ observed in the future.

Note: $^{*)}$ without using $\mathcal{D}^{\text{test}}$.

Prediction with misclassification cost

Given a misclassification cost matrix $c \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$, the **prediction problem with misclassification cost** (**cost-sensitive classification**) $\ell(y,\hat{y}) := \delta(y \neq \hat{y}) c_{y,\hat{y}}$ minimizes the **misclassification cost**:

$$\text{cost}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} \delta(y \neq \hat{y}) c_{y,\hat{y}}$$

Lemma

The predicted value with minimal misclassification cost is:

$$\hat{y} := \arg\min_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}, y\neq\hat{y}} \hat{n}_y c_{y,\hat{y}}$$

$$\textit{with } \hat{n}_y := |\{y' \in \mathcal{D}^{train} \mid y' = y\}|, \quad y \in \mathcal{Y}$$

Note: The diagonal $c_{y,y} := 0$ for all $y \in \mathcal{Y}$.

Prediction with misclassification cost

Lemma

The predicted value with minimal misclassification cost is:

$$\hat{y} := \arg\min_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}, y\neq\hat{y}} \hat{n}_y c_{y,\hat{y}}$$

$$\textit{with } \hat{n}_y := |\{y' \in \mathcal{D}^{\textit{train}} \mid y' = y\}|, \quad y \in \mathcal{Y}$$

Proof.

$$\text{cost}(\mathcal{D}^{\text{train}}, \hat{y}) = \frac{1}{|\mathcal{D}^{\text{train}}|} \sum_{y\in\mathcal{Y}, y\neq\hat{y}} \hat{n}_y c_{y,\hat{y}}$$

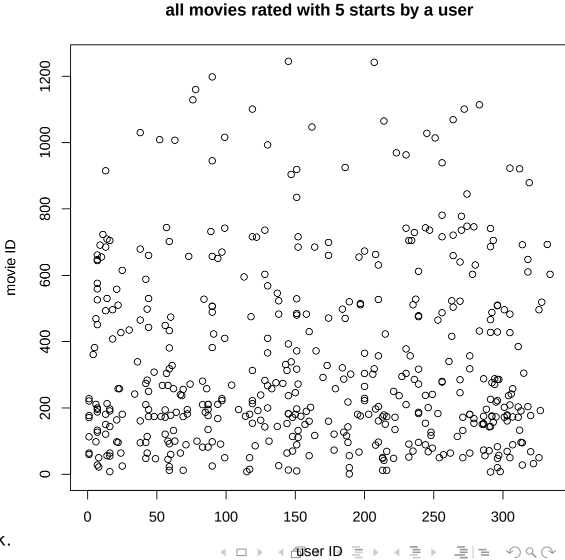
□

Outline

Example: All Highly-Rated Products

train set $\mathcal{D}^{\text{train}}$:

movie IDs
1, 6, 9, 12, 13, ...
320, 321, 328, 340, 346, 347
42, 89,100,101,109, ...
4, 7, 8, 9, 12, ...
6, 50, 201, 286, 298, ...
9, 15, 28, 83, 173, ...
4, 12, 13, 23, 28, ...
50, 125, 181, 255, 269, ...
⋮

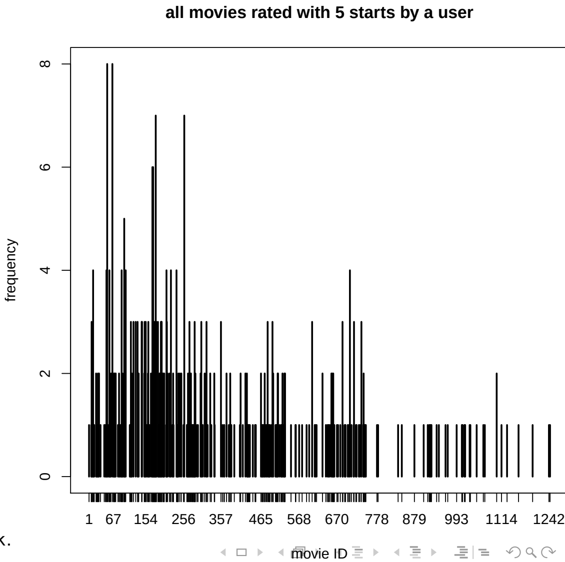


Note: Data derived from Movielens 100k.

Example: All Highly-Rated Products

train set $\mathcal{D}^{\text{train}}$:

movie IDs
1, 6, 9, 12, 13, ...
320, 321, 328, 340, 346, 347
42, 89,100,101,109, ...
4, 7, 8, 9, 12, ...
6, 50, 201, 286, 298, ...
9, 15, 28, 83, 173, ...
4, 12, 13, 23, 28, ...
50, 125, 181, 255, 269, ...
⋮



Note: Data derived from Movielens 100k.

Example: All Highly-Rated Products

- Target space: $\mathcal{Y} := \mathcal{P}(I) := \{\emptyset, \{1\}, \ldots, \{1682\}, \{1,2\}, \{1,3\}, \ldots\}$ with $I := \{1, 2, \ldots, 1682\}$.
- Training set: $\mathcal{D}^{\text{train}} := \{\{1, 6, 9, \ldots\}, \{320, 321, 328, \ldots\}, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{\{50, 100, 127, \ldots\}, \{50, 258, 294, \ldots\}, \ldots\}$

What is a good quality measure?

- **Recall**: $\text{recall}(y, \hat{y}) := \frac{|y \cap \hat{y}|}{|y|}$

Note: Quality measures are maximized, losses are minimized, thus the negative of a quality measure is a loss.

Example: All Highly-Rated Products

- Target space: $\mathcal{Y} := \mathcal{P}(I) := \{\emptyset, \{1\}, \ldots, \{1682\}, \{1,2\}, \{1,3\}, \ldots\}$ with $I := \{1, 2, \ldots, 1682\}$.
- Training set: $\mathcal{D}^{\text{train}} := \{\{1, 6, 9, \ldots\}, \{320, 321, 328, \ldots\}, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{\{50, 100, 127, \ldots\}, \{50, 258, 294, \ldots\}, \ldots\}$

What is a good quality measure?

- **Recall**: $\text{recall}(y, \hat{y}) := \frac{|y \cap \hat{y}|}{|y|}$
  — but recall is maximized trivially for $\hat{y} := I$.

Note: Quality measures are maximized, losses are minimized, thus the negative of a quality measure is a loss.

Example: All Highly-Rated Products

- Target space: $\mathcal{Y} := \mathcal{P}(I) := \{\emptyset, \{1\}, \ldots, \{1682\}, \{1,2\}, \{1,3\}, \ldots\}$ with $I := \{1, 2, \ldots, 1682\}$.
- Training set: $\mathcal{D}^{\text{train}} := \{\{1, 6, 9, \ldots\}, \{320, 321, 328, \ldots\}, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{\{50, 100, 127, \ldots\}, \{50, 258, 294, \ldots\}, \ldots\}$

What is a good quality measure?

- **Recall**: $\text{recall}(y, \hat{y}) := \frac{|y \cap \hat{y}|}{|y|}$
  — but recall is maximized trivially for $\hat{y} := I$.
- **Precision**: $\text{precision}(y, \hat{y}) := \frac{|y \cap \hat{y}|}{|\hat{y}|}$

Note: Quality measures are maximized, losses are minimized, thus the negative of a quality measure is a loss.

Example: All Highly-Rated Products

- Target space: $\mathcal{Y} := \mathcal{P}(I) := \{\emptyset, \{1\}, \ldots, \{1682\}, \{1,2\}, \{1,3\}, \ldots\}$ with $I := \{1, 2, \ldots, 1682\}$.
- Training set: $\mathcal{D}^{\text{train}} := \{\{1, 6, 9, \ldots\}, \{320, 321, 328, \ldots\}, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{\{50, 100, 127, \ldots\}, \{50, 258, 294, \ldots\}, \ldots\}$

What is a good quality measure?

- **Recall**: $\text{recall}(y, \hat{y}) := \frac{|y \cap \hat{y}|}{|y|}$
  — but recall is maximized trivially for $\hat{y} := I$.
- **Precision**: $\text{precision}(y, \hat{y}) := \frac{|y \cap \hat{y}|}{|\hat{y}|}$
  — but precision is maximized trivially for $\hat{y} := \emptyset$.

Note: Quality measures are maximized, losses are minimized, thus the negative of a quality measure is a loss.

Example: All Highly-Rated Products

- Target space: $\mathcal{Y} := \mathcal{P}(I) := \{\emptyset, \{1\}, \ldots, \{1682\}, \{1,2\}, \{1,3\}, \ldots\}$ with $I := \{1, 2, \ldots, 1682\}$.
- Training set: $\mathcal{D}^{\text{train}} := \{\{1, 6, 9, \ldots\}, \{320, 321, 328, \ldots\}, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{\{50, 100, 127, \ldots\}, \{50, 258, 294, \ldots\}, \ldots\}$

What is a good quality measure?

- **Recall**: $\text{recall}(y, \hat{y}) := \frac{|y \cap \hat{y}|}{|y|}$
  — but recall is maximized trivially for $\hat{y} := I$.
- **Precision**: $\text{precision}(y, \hat{y}) := \frac{|y \cap \hat{y}|}{|\hat{y}|}$
  — but precision is maximized trivially for $\hat{y} := \emptyset$.
- **F_1 measure**: $F_1(y, \hat{y}) := 2\frac{\text{recall}(y,\hat{y})\text{precision}(y,\hat{y})}{\text{recall}(y,\hat{y})+\text{precision}(y,\hat{y})} = \frac{2|y \cap \hat{y}|}{|y|+|\hat{y}|}$

Note: Quality measures are maximized, losses are minimized, thus the negative of a quality measure is a loss.

Example: All Highly-Rated Products

- Target space: $\mathcal{Y} := \mathcal{P}(I) := \{\emptyset, \{1\}, \ldots, \{1682\}, \{1,2\}, \{1,3\}, \ldots\}$ with $I := \{1, 2, \ldots, 1682\}$.
- Loss: $\ell(y, \hat{y}) := 1 - F_1(y, \hat{y}) = 1 - \frac{2|y \cap \hat{y}|}{|y| + |\hat{y}|}$ (negative F_1)
- Training set: $\mathcal{D}^{\text{train}} := \{\{1, 6, 9, \ldots\}, \{320, 321, 328, \ldots\}, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{\{50, 100, 127, \ldots\}, \{50, 258, 294, \ldots\}, \ldots\}$

Example: All Highly-Rated Products

- Target space: $\mathcal{Y} := \mathcal{P}(I) := \{\emptyset, \{1\}, \ldots, \{1682\}, \{1,2\}, \{1,3\}, \ldots\}$ with $I := \{1, 2, \ldots, 1682\}$.
- Loss: $\ell(y, \hat{y}) := 1 - F_1(y, \hat{y}) = 1 - \frac{2|y \cap \hat{y}|}{|y| + |\hat{y}|}$ (negative F_1)
- Training set: $\mathcal{D}^{\text{train}} := \{\{1, 6, 9, \ldots\}, \{320, 321, 328, \ldots\}, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{\{50, 100, 127, \ldots\}, \{50, 258, 294, \ldots\}, \ldots\}$

Given some **sets of highly-rated products** $\mathcal{D}^{\text{train}}$, compute$^{*)}$ a **predicted sets of highly-rated products** $\hat{y}$ with minimal **negative $\mathbf{F}_1$ error**:

$$\mathrm{F}_1(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} 1 - F_1(y, \hat{y})$$

for sets of highly-rated products $\mathcal{D}^{\text{test}}$ observed in the future.

Note: $^{*)}$ without using $\mathcal{D}^{\text{test}}$.

Prediction with Negative F_1 loss (multi-label classification)

The **prediction problem with negative F_1 loss** (**multi-label classification**) $\ell(y, \hat{y}) := 1 - F_1(y, \hat{y}) = 1 - \frac{2|y \cap \hat{y}|}{|y|+|\hat{y}|}$ minimizes the **negative F_1 error**:

$$\text{1-F}_1(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} 1 - \frac{2|y \cap \hat{y}|}{|y| + |\hat{y}|}$$

Outline

Example: Product Preferences

Customer Alice:

- product A is better than B
- product C is better than D
- products A/B and C/D are not comparable.

Example: Product Preferences

Customer Alice:

- product A is better than B
- product C is better than D
- products A/B and C/D are not comparable.

Customer Bob:

- product A is better than B, B is better than C
- product D is better than C
- products A/B and D are not comparable.

Example: Product Preferences

Customer Alice:

- product A is better than B
- product C is better than D
- products A/B and C/D are not comparable.

Customer Bob:

- product A is better than B, B is better than C
- product D is better than C
- products A/B and D are not comparable.

Avoid:

- product A is better than A.
- product A is better than B, B better than C, but A is not better than C.

Example: Product Preferences

Avoid:

- product A is better than A.
- product A is better than B, B better than C,
  but A is not better than C.

For a set I:

$$\begin{aligned}\text{ranking}(I) := \{y \subseteq I \times I \mid & \forall i \in I : (i,i) \notin y, \\ & \forall i,j,k \in I : (i,j),(j,k) \in y \Rightarrow (i,k) \in y\}\end{aligned}$$

Example: Product Preferences

Customer Alice:

- product A is better than B
- product C is better than D
- products A/B and C/D are not comparable.

$$y_{\text{Alice}} := \{(A, B), (C, D)\}$$

Customer Bob:

- product A is better than B, B is better than C
- product D is better than C
- products A/B and D are not comparable.

$$y_{\text{Bob}} := \{(A, B), (B, C), (A, C), (D, C)\}$$

Example: Product Preferences

- Target space: $\mathcal{Y} := \text{ranking}(\{A, B, C, D\})$
- Training set:
  $\mathcal{D}^{\text{train}} := \{\{(A, B), (C, D)\}, \{(A, B), (B, C), (A, C), (D, C)\}, \ldots\}$
- Test set: $\mathcal{D}^{\text{test}} := \{\{(A, B), (A, C), (A, D)\}, \ldots\}$

How to measure error for rankings?

- 0/1 loss: $\ell(y, \hat{y}) := \delta(y \neq \hat{y})$.
  — very rough, e.g., $\hat{y}_1 := \{(A, B)\}$ as bad as $\hat{y}_2 := \{(B, A), (D, C)\}$ for y_{Alice}.
- 1 - Area under the Curve (1-AUC):

$$\text{AUC}(y, \hat{y}) := \frac{1}{|y|} \sum_{(i,j) \in y} \delta((i, j) \in \hat{y})$$

Prediction with 1-AUC loss (ranking)

The **prediction problem with 1-AUC loss** (**ranking**) $\ell(y, \hat{y}) := 1 - \text{AUC}(y, \hat{y}) = 1 - \frac{1}{|y|} \sum_{(i,j) \in y} \delta((i,j) \in \hat{y})$ minimizes the **1-AUC error**:

$$\text{1-AUC}(\mathcal{D}^{\text{test}}, \hat{y}) := \frac{1}{n} \sum_{y \in \mathcal{D}^{\text{test}}} 1 - \frac{1}{|y|} \sum_{(i,j) \in y} \delta((i,j) \in \hat{y})$$

Outline

Example: House Prices Again

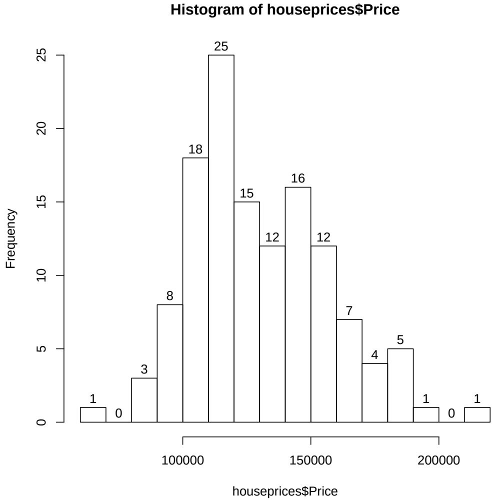
train set $\mathcal{D}^{\text{train}}$:

price [$]
114,300
114,200
114,800
94,700
119,800
114,600
151,600
150,700
119,200
104,000
⋮

Example: House Prices Again

train set $\mathcal{D}^{\text{train}}$:

price [$]
114,300
114,200
114,800
94,700
119,800
114,600
151,600
150,700
119,200
104,000
⋮



How useful is an average price of ca. 130.000$ if it is untypical?

Example: House Prices Again

How to predict the certainty? How prices may vary?

- predict minimum and maximum prices?
  — OK, but does not tell about typical prices either.

Example: House Prices Again

How to predict the certainty? How prices may vary?

- predict minimum and maximum prices?
  — OK, but does not tell about typical prices either.
- predict average price plus price range that contains 25%, 50% of all prices?
  — OK, but will also be off for bimodal distributions.

Example: House Prices Again

How to predict the certainty? How prices may vary?

- predict minimum and maximum prices?
  — OK, but does not tell about typical prices either.
- predict average price plus price range that contains 25%, 50% of all prices?
  — OK, but will also be off for bimodal distributions.
- predict for every possible price a score how likely it is.

Density

Let $\mathcal{Y}$ be a set. A function

$$p : \mathcal{Y} \to \mathbb{R}_0^+$$

with

$$\int_{\mathcal{Y}} p(y) dy = 1$$

is called **density**.

For $y \in \mathcal{Y}$, $p(y)$ measures how likely y is.

Density

Let $\mathcal{Y}$ be a set. A function

$$p : \mathcal{Y} \rightarrow \mathbb{R}_0^+$$

with

$$\int_{\mathcal{Y}} p(y) dy = 1$$

is called **density**.

For $y \in \mathcal{Y}$, $p(y)$ measures how likely y is.

Example:

$$p(y; a, b) := \frac{1}{b-a} \delta(y \in [a, b]), \quad a, b \in \mathbb{R}, a < b \qquad \text{(uniform density)}$$

Density

Let $\mathcal{Y}$ be a set. A function

$$p : \mathcal{Y} \to \mathbb{R}_0^+$$

with

$$\int_{\mathcal{Y}} p(y) dy = 1$$

is called **density**.

For $y \in \mathcal{Y}$, $p(y)$ measures how likely y is.

Example:

$$p(y; a, b) := \frac{1}{b-a} \delta(y \in [a, b]), \quad a, b \in \mathbb{R}, a < b \qquad \text{(uniform density)}$$

$$p(y; \mu, \sigma^2) := \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\mu)^2}{2\sigma^2}}, \qquad \mu, \sigma^2 \in \mathbb{R}, \sigma^2 > 0 \qquad \text{(normal density)}$$

Likelihood

For a set $\mathcal{D} \subseteq \mathcal{Y}$

$$L(\mathcal{D}; p) := \prod_{y \in \mathcal{D}} p(y)$$

is called **likelihood** and

$$\ell(\mathcal{D}; p) := -\log L(\mathcal{D}; p) = \sum_{y \in \mathcal{D}} \log p(y)$$

is called **negative log-likelihood**.

The better p models $\mathcal{D}$,

- the higher the likelihood,
- the smaller the negative log-likelihood.
  (The negative log-likelihood is a loss.)

Best Uniform Density for a Data Set?

Let $\mathcal{D} \subseteq \mathcal{Y}$ be a set. What is the uniform density

$$p(y; a, b) := \frac{1}{b-a}\delta(y \in [a, b]), \qquad a, b \in \mathbb{R}, a < b$$

that best models $\mathcal{D}$, i.e., with maximal likelihood?

Best Uniform Density for a Data Set?

Let $\mathcal{D} \subseteq \mathcal{Y}$ be a set. What is the uniform density

$$p(y; a, b) := \frac{1}{b-a}\delta(y \in [a, b]), \qquad a, b \in \mathbb{R}, a < b$$

that best models $\mathcal{D}$, i.e., with maximal likelihood?

For any $y_0 \in \mathcal{D}$, let:

$$a := y_0 - \frac{1}{n}, \quad b := y_0 + \frac{1}{n}, \quad n \in \mathbb{N}$$

$$\rightsquigarrow L(\mathcal{D}; p) \geq \frac{n}{2}$$

i.e., the likelihood is unbounded: there is no best uniform density.

Best Normal Density for a Data Set?

The same is true for the normal density with $\mu = y_0 \in \mathcal{D}$.

Best Normal Density for a Data Set?

The same is true for the normal density with $\mu = y_0 \in \mathcal{D}$.

If we exclude such μ:

$$
\begin{aligned}
-\log L(p;\mathcal{D}) &= -\sum_{y\in\mathcal{D}} \log p(y) \\
&= -\sum_{y\in\mathcal{D}} \log \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\mu)^2}{2\sigma^2}} \\
&= \sum_{y\in\mathcal{D}} \frac{1}{2}\log(2\pi) + \sum_{y\in\mathcal{D}} \frac{1}{2}\log\sigma^2 + \sum_{y\in\mathcal{D}} \frac{(y-\mu)^2}{2\sigma^2} \\
&= |D|\frac{1}{2}\log(2\pi) + |D|\frac{1}{2}\log\sigma^2 + \sum_{y\in\mathcal{D}} \frac{(y-\mu)^2}{2\sigma^2}
\end{aligned}
$$

Best Normal Density for a Data Set?

The same is true for the normal density with $\mu = y_0 \in \mathcal{D}$.

If we exclude such μ:

$$-\log L(p; \mathcal{D}) = |D|\frac{1}{2}\log(2\pi) + |D|\frac{1}{2}\log\sigma^2 + \sum_{y\in\mathcal{D}} \frac{(y-\mu)^2}{2\sigma^2}$$

$$\frac{\partial(-\log L)}{\partial\mu} = \sum_{y\in\mathcal{D}} -2\frac{y-\mu}{2\sigma^2} \stackrel{!}{=} 0$$

$$\rightsquigarrow \mu = \frac{1}{|\mathcal{D}|}\sum_{y\in\mathcal{D}} y$$

Best Normal Density for a Data Set?

The same is true for the normal density with $\mu = y_0 \in \mathcal{D}$.

If we exclude such μ:

$$-\log L(p; \mathcal{D}) = |D|\frac{1}{2}\log(2\pi) + |D|\frac{1}{2}\log\sigma^2 + \sum_{y\in\mathcal{D}} \frac{(y-\mu)^2}{2\sigma^2}$$

$$\frac{\partial(-\log L)}{\partial\mu} = \sum_{y\in\mathcal{D}} -2\frac{y-\mu}{2\sigma^2} \stackrel{!}{=} 0$$

$$\rightsquigarrow \mu = \frac{1}{|\mathcal{D}|}\sum_{y\in\mathcal{D}} y$$

$$\frac{\partial(-\log L)}{\partial\sigma^2} = \frac{1}{2}|\mathcal{D}|\frac{1}{\sigma^2} - \sum_{y\in\mathcal{D}} \frac{(y-\mu)^2}{2(\sigma^2)^2} \stackrel{!}{=} 0$$

$$\rightsquigarrow \sigma^2 = \frac{1}{|\mathcal{D}|}\sum_{y\in\mathcal{D}} (y-\mu)^2$$

Example: House Prices Again

$$a := \min \mathcal{D}^{\text{train}} = 69100, \quad b := \max \mathcal{D}^{\text{train}} = 188000$$

$$\rightsquigarrow -\log L(\mathcal{D}^{\text{train}}; p_{\text{uniform}}) = 11.686$$

$$-\log L(\mathcal{D}^{\text{test}}; p_{\text{uniform}}) = \infty$$

as $\mathcal{D}^{\text{test}}$ contains a price $y = 211200$ outside the training range, thus with $p_{\text{uniform}}(y) = 0$.

Example: House Prices Again

$$a := \min \mathcal{D}^{\text{train}} = 69100, \quad b := \max \mathcal{D}^{\text{train}} = 188000$$

$$\rightsquigarrow -\log L(\mathcal{D}^{\text{train}}; p_{\text{uniform}}) = 11.686$$

$$-\log L(\mathcal{D}^{\text{test}}; p_{\text{uniform}}) = \infty$$

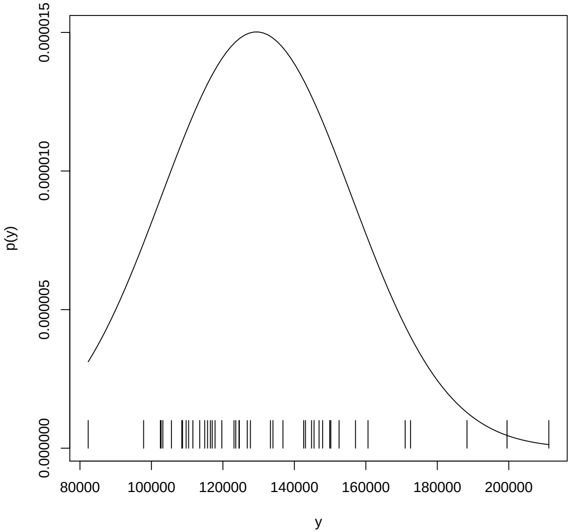
as $\mathcal{D}^{\text{test}}$ contains a price $y = 211200$ outside the training range, thus with $p_{\text{uniform}}(y) = 0$.

$$\mu = \mu\mathcal{D}^{\text{train}} = 129395.3, \quad \sigma = \sigma\mathcal{D}^{\text{train}} = 26562$$

$$\rightsquigarrow -\log L(\mathcal{D}^{\text{train}}; p_{\text{normal}}) = 11.600$$

$$-\log L(\mathcal{D}^{\text{test}}; p_{\text{normal}}) = 11.643$$

Example: House Prices Again



Outline

Certainty for Binary Targets

Predict not just the class/label $y \in \mathcal{Y}$, but provide

- a **probability** / **certainty factor** $\hat{y} \in [0, 1]$ and then predict

$$\hat{y}' := \delta(\hat{y} > 0.5)$$

- an unbounded certainty factor / **score** $\hat{y} \in \mathbb{R}$ and then predict

$$\hat{y}' := \delta(\hat{y} > 0)$$

Binary Targets / Losses for Probabilities

- treat y like a continuous target and use any regression loss, e.g.,

$$\ell(y, \hat{y}) := (y - \hat{y})^2$$

Binary Targets / Losses for Probabilities

- treat y like a continuous target and use any regression loss, e.g.,

$$\ell(y, \hat{y}) := (y - \hat{y})^2$$

- **binomial negative log-likelihood**:

$$L(y, \hat{y}) := \hat{y}^y (1 - \hat{y})^{(1-y)}$$
$$\ell(y, \hat{y}) := -\log L(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

Binary Targets / Losses for Probabilities

- treat y like a continuous target and use any regression loss, e.g.,

$$\ell(y, \hat{y}) := (y - \hat{y})^2$$

- **binomial negative log-likelihood**:

$$L(y, \hat{y}) := \hat{y}^y (1 - \hat{y})^{(1-y)}$$
$$\ell(y, \hat{y}) := -\log L(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

Lemma

Both, squared error and binomial negative log-likelihood, are minimized by the ***relative positive class frequency****:*

$$\hat{y} := \frac{\hat{n}_1}{|\mathcal{D}^{train}|} = \frac{1}{|\mathcal{D}^{train}|} \sum_{y \in \mathcal{D}^{train}} y$$

Binary Targets / Losses for Scores

- treat y like a continuous target and use any regression loss, e.g.,

$$\ell(y, \hat{y}) := (y - \hat{y})^2$$

Binary Targets / Losses for Scores

- treat y like a continuous target and use any regression loss, e.g.,

$$\ell(y, \hat{y}) := (y - \hat{y})^2$$

— but this does also penalize $\hat{y} > 1$ for $y = 1$!

Binary Targets / Losses for Scores

- treat y like a continuous target and use any regression loss, e.g.,

$$\ell(y, \hat{y}) := (y - \hat{y})^2$$

— but this does also penalize $\hat{y} > 1$ for $y = 1$!

- **hinge loss**:

$$\ell(y, \hat{y}) := 2[y + \hat{y} - 2y\hat{y}]_0 := 2\max(y + \hat{y} - 2y\hat{y}, 0)$$

$$= 2\begin{cases} 1 - \hat{y}, & \text{if } y = 1, \hat{y} \leq 1 \\ \hat{y}, & \text{if } y = 0, \hat{y} \geq 0 \\ 0, & \text{else} \end{cases}$$

Binary Targets / Losses for Scores

- treat y like a continuous target and use any regression loss, e.g.,

$$\ell(y, \hat{y}) := (y - \hat{y})^2$$

— but this does also penalize $\hat{y} > 1$ for $y = 1$!

- **hinge loss**:

$$\ell(y, \hat{y}) := 2[y + \hat{y} - 2y\hat{y}]_0 := 2\max(y + \hat{y} - 2y\hat{y}, 0)$$

$$= 2\begin{cases} 1 - \hat{y}, & \text{if } y = 1, \hat{y} \leq 1 \\ \hat{y}, & \text{if } y = 0, \hat{y} \geq 0 \\ 0, & \text{else} \end{cases}$$

Lemma

Hinge loss is minimized by

$$\hat{y} := \arg\max_{\hat{y}} \hat{n}_{\hat{y}}$$

Note: Usually hinge loss is used for target encoding $\{+1, -1\}$ instead of $\{0, 1\}$ and then equals $\ell(y, \hat{y}) := [1 - y\hat{y}]_0$.

Binary Targets / Losses for Scores (2/2)

- **squared hinge loss**:

$$\ell(y,\hat{y}) := (2[y+\hat{y}-2y\hat{y}]_0)^2 := (2\max(y+\hat{y}-2y\hat{y},0))^2$$
$$= 2\begin{cases}(1-\hat{y})^2, & \text{if } y=1, \hat{y}\leq 1\\ \hat{y}^2, & \text{if } y=0, \hat{y}\geq 0\\ 0, & \text{else}\end{cases}$$

Binary Targets / Losses for Scores (2/2)

- **squared hinge loss**:

$$\begin{aligned}\ell(y,\hat{y}) :=& (2[y+\hat{y}-2y\hat{y}]_0)^2 := (2\max(y+\hat{y}-2y\hat{y},0))^2\\ =& 2\begin{cases}(1-\hat{y})^2, & \text{if } y=1, \hat{y}\leq 1\\ \hat{y}^2, & \text{if } y=0, \hat{y}\geq 0\\ 0, & \text{else}\end{cases}\end{aligned}$$

Lemma

Squared hinge loss is minimized by the relative positive class frequency

$$\hat{y} := \frac{\hat{n}_1}{|\mathcal{D}^{train}|}$$

Note: Usually hinge loss is used for target encoding $\{+1,-1\}$ instead of $\{0,1\}$ and then squared hinge loss equals $\ell(y,\hat{y}) := ([1-y\hat{y}]_0)^2$.

Certainty for Nominal Targets

Predict not just the class/label $\hat{y} \in \mathcal{Y}$, but provide for each possible label $y \in \mathcal{Y}$

- a **probability** / **certainty factor** $\hat{y}(y) \in [0, 1]$ and then predict

$$\hat{y}' := \arg\max_{y \in \mathcal{Y}} \hat{y}(y)$$

- an unbounded certainty factor / **score** $\hat{y}(y) \in \mathbb{R}$ and then predict

$$\hat{y}' := \arg\max_{y \in \mathcal{Y}} \hat{y}(y)$$

Nominal Targets / Losses for Probabilities

- treat y like a continuous target and use any multivariate regression loss, e.g.,

$$\ell(y, \hat{y}) := \sum_{y' \in \mathcal{Y}} (\delta(y = y') - \hat{y}(y))^2$$

Nominal Targets / Losses for Probabilities

- treat y like a continuous target and use any multivariate regression loss, e.g.,

$$\ell(y, \hat{y}) := \sum_{y' \in \mathcal{Y}} (\delta(y = y') - \hat{y}(y))^2$$

- **multinomial negative log-likelihood**:

$$L(y, \hat{y}) := \prod_{y' \in \mathcal{Y}} \hat{y}(y')^{\delta(y'=y)}$$

$$\ell(y, \hat{y}) := -\log L(y, \hat{y}) = -\prod_{y' \in \mathcal{Y}} \delta(y' = y) \log \hat{y}(y')$$

Nominal Targets / Losses for Probabilities

- treat y like a continuous target and use any multivariate regression loss, e.g.,

$$\ell(y, \hat{y}) := \sum_{y' \in \mathcal{Y}} (\delta(y = y') - \hat{y}(y))^2$$

- **multinomial negative log-likelihood**:

$$L(y, \hat{y}) := \prod_{y' \in \mathcal{Y}} \hat{y}(y')^{\delta(y'=y)}$$

$$\ell(y, \hat{y}) := -\log L(y, \hat{y}) = -\prod_{y' \in \mathcal{Y}} \delta(y' = y) \log \hat{y}(y')$$

Lemma

Both, multivariate squared error and multinomial negative log-likelihood, are minimized by the ***relative class frequencies****:*

$$\hat{y}(y') := \frac{\hat{n}_{y'}}{|\mathcal{D}^{train}|} = \frac{1}{|\mathcal{D}^{train}|} \sum_{y \in \mathcal{D}^{train}} \delta(y = y')$$

Certainty for Set-Valued Targets

For set-valued targets, a score/certainty factor for every set $y \in \mathcal{Y} := \mathcal{P}(I)$ would have to be predicted.

But usually, one predicts just a score $\hat{y}(i)$ for every label $i \in I$.

If non-negative, such scores induce a distribution on the power set via

$$p(y) := \frac{1}{Z} \prod_{i \in y} \hat{y}(i)$$

Note: Z is the normalizing constant, $Z := \sum_{y \subseteq I} \prod_{i \in y} \hat{y}(i)$.

Set-Valued Targets / Losses

- Negative **Normalized Discounted Cumulative Gain** (neg. NDCG):

$$\begin{aligned}
\ell(y,\hat{y}) :=& 1 - \text{NDCG}(y,\hat{y}) \\
\text{NDCG}(y,\hat{y}) :=& \frac{1}{\sum_{i=1}^{|y|} \frac{1}{\log(1+i)}} \sum_{i \in y} \frac{1}{\log(1+\text{rank}(\hat{y},i))} \\
\text{with } \text{rank}(\hat{y},i) :=& |\{i' \in I \mid \hat{y}(i') \geq y(i)\}
\end{aligned}$$

Set-Valued Targets / Losses

- Negative **Normalized Discounted Cumulative Gain** (neg. NDCG):

$$\ell(y, \hat{y}) := 1 - \text{NDCG}(y, \hat{y})$$

$$\text{NDCG}(y, \hat{y}) := \frac{1}{\sum_{i=1}^{|y|} \frac{1}{\log(1+i)}} \sum_{i \in y} \frac{1}{\log(1 + \text{rank}(\hat{y}, i))}$$

$$\text{with rank}(\hat{y}, i) := |\{i' \in I \mid \hat{y}(i') \geq y(i)\}|$$

Example:

$y := \{1, 3, 6\}$,

y'	1	2	3	4	5	6
$\hat{y}(y')$	0.5	0.4	0.7	0.1	0.0	0.1

Set-Valued Targets / Losses

- Negative **Normalized Discounted Cumulative Gain** (neg. NDCG):

$$\ell(y, \hat{y}) := 1 - \text{NDCG}(y, \hat{y})$$

$$\text{NDCG}(y, \hat{y}) := \frac{1}{\sum_{i=1}^{|y|} \frac{1}{\log(1+i)}} \sum_{i \in y} \frac{1}{\log(1 + \text{rank}(\hat{y}, i))}$$

$$\text{with } \text{rank}(\hat{y}, i) := |\{i' \in I \mid \hat{y}(i') \geq y(i)\}|$$

Example:

$y := \{1, 3, 6\}$,

y'	1	2	3	4	5	6
$\hat{y}(y')$	0.5	0.4	0.7	0.1	0.0	0.1
$\text{rank}(\hat{y}, y')$	2	3	1	4	6	5

Set-Valued Targets / Losses

- Negative **Normalized Discounted Cumulative Gain** (neg. NDCG):

$$\ell(y,\hat{y}) := 1 - \text{NDCG}(y,\hat{y})$$

$$\text{NDCG}(y,\hat{y}) := \frac{1}{\sum_{i=1}^{|y|} \frac{1}{\log(1+i)}} \sum_{i \in y} \frac{1}{\log(1+\text{rank}(\hat{y},i))}$$

$$\text{with rank}(\hat{y},i) := |\{i' \in I \mid \hat{y}(i') \geq y(i)\}|$$

Example:

$y := \{1,3,6\}$,

y'	1	2	3	4	5	6
$\hat{y}(y')$	0.5	0.4	0.7	0.1	0.0	0.1
$\text{rank}(\hat{y},y')$	2	3	1	4	6	5

$$\text{NDCG}(y,\hat{y}) = \frac{1}{\frac{1}{\log 2} + \frac{1}{\log 3} + \frac{1}{\log 4}} \left(\frac{1}{\log(1+2)} + \frac{1}{\log(1+1)} + \frac{1}{\log(1+5)}\right)$$
$$= 0.947$$

Note: Here NCDG for binary relevances is given. NDCG also is defined more generally for non-binary relevances.

Set-Valued Targets / Losses

Lemma

Negative NDCG is minimized by any score $\hat{y}$ that induces a ranking by relative class frequency, esp. **relative class frequencies** *themselves:*

$$\hat{y}(y') := \frac{\hat{n}_{y'}}{|\mathcal{D}^{train}|} = \frac{1}{|\mathcal{D}^{train}|} \sum_{y \in \mathcal{D}^{train}} \delta(y = y')$$

Outline

Summary of Tasks & Error Measures

	point estimation (just the prediction)	density estimation (prediction plus variance/certainty)
univariate targets:		
continuous target (regression)	Root Mean Squared Error (RMSE) Mean Average Error (MAE) ϵ-insensitive error	Gaussian Likelihood
binary nominal target (binary classification)	Misclassification Rate Misclassification Cost	Hinge loss Squared hinge loss Binomial Likelihood
multivariate targets:		
nominal target (multiclass classification)	Misclassification Rate Misclassification Cost	Multinomial Likelihood
set-valued target (multi-label classification)	Recall, Precision, F1 Recall@10, Precision@10	Normalized Discounted Cumulative Gain (NDCG) Mean Average Precision (MAP)
ranking target (ranking)	Area under the curve (AUC)	

Conclusion

- **Prediction** is the task to learn an **unknown dependency** of a **target** from **predictors** from observed data (**training data**).
- Part of the problem setting is a **loss** that defines how bad different incorrect predictions are.
- As the dependency to learn is unknown, different **models** are assessed by their performance on an a fresh sample (**test set**).
- Different prediction problems can be described by
  1. the **target space** $\mathcal{Y}$ and
  2. the **loss** ℓ.
- The most common prediction tasks are
  1. **regression**: continuous target ($\mathcal{Y} := \mathbb{R}$),
     esp. **least squares regression** (squared loss $\ell = (y - \hat{y})^2$).
  2. **binary classification** ($\mathcal{Y} := \{0, 1\}$)),
     esp. not cost-sensitive (0/1 loss, misclassification rate $\ell = \delta(y \neq \hat{y})$).

References

Wolfgang Jank.
Business Analytics for Managers.
Springer, 2011.

Sérgio Moro, Raul Laureano, and Paulo Cortez.
Using data mining for bank direct marketing: An application of the crisp-dm methodology.
2011.