ORIE 4741: Learning with Big Messy Data

Loss functions

Professor Udell

Operations Research and Information Engineering
Cornell

November 4, 2021

Announcements 11/2/21

- hw5 will come out this Thursday or Friday
- section this week: post-hoc interpretability techniques (SHAP, LIME)

Announcements 11/4/21

- hw5 will come out today or tomorrow
- section this week: post-hoc interpretability techniques (SHAP, LIME)
- teamwork issues on the project? let's talk!
- let me see your faces!

Poll

My team is changing the direction of our project, compared to our proposal

A. yes

B. no

Poll

My team is changing the direction of our project, compared to our proposal

A. yes
B. no

My team has different team members, compared to our proposal

A. yes
B. no

Regularized empirical risk minimization

choose model by solving

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} \ell(x_i, y_i; w) + r(w)$$

with variable $w \in \mathbf{R}^d$

- parameter vector $w \in \mathbf{R}^d$
- loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \mathbf{R}^d \to \mathbf{R}$
- regularizer $r : \mathbf{R}^d \to \mathbf{R}$

Regularized empirical risk minimization

choose model by solving

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} \ell(x_i, y_i; w) + r(w)$$

with variable $w \in \mathbf{R}^d$

- parameter vector $w \in \mathbf{R}^d$
- loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \mathbf{R}^d \to \mathbf{R}$
- regularizer $r : \mathbf{R}^d \to \mathbf{R}$

why?

- want to minimize the **risk** $\mathbb{E}_{(x,y)\sim P}\ell(x, y; w)$
- approximate it by the **empirical risk** $\sum_{i=1}^{n} \ell(x, y; w)$
- add regularizer to help model generalize

Loss functions

what kind of loss functions should we use?

depends on **type** of data

- real
- boolean
- ordinal
- nominal
- ...

and on **noise** in data

- small?
- large but sparse?
- from some probabilistic model?
- ...

Outline

Loss functions for real-valued data

- quadratic
- ℓ_1
- huber
- quantile
- ...

Least squares regression

least squares (ℓ_2) regression:

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i)^2 + r(w)$$

special case: no covariates. what is

$$\operatorname*{argmin}_w \frac{1}{n}\sum_{i=1}^{n}(y_i - w)^2?$$

Least squares regression

least squares (ℓ_2) regression:

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i)^2 + r(w)$$

special case: no covariates. what is

$$\underset{w}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - w)^2?$$

A: mean(y)!

ℓ_1 regression

ℓ_1 regression:

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n}|y_i - w^T x_i| + r(w)$$

special case: no covariates. what is

$$\operatorname*{argmin}_{w} \frac{1}{n}\sum_{i=1}^{n}|y_i - w|?$$

ℓ_1 regression

ℓ_1 regression:

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} |y_i - w^T x_i| + r(w)$$

special case: no covariates. what is

$$\operatorname*{argmin}_{w} \frac{1}{n}\sum_{i=1}^{n} |y_i - w|?$$

- if pn of the y_i's are bigger than w,
- then as w increases to $w + \delta$,
- $\frac{1}{n}\sum_{i:y_i>w} |y_i - w|$ decreases by $p\delta$
- $\frac{1}{n}\sum_{i:y_i<w} |y_i - w|$ increases by $(1-p)\delta$
- if $p = \frac{1}{2}$, objective stays the same

ℓ_1 regression

ℓ_1 regression:

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} |y_i - w^T x_i| + r(w)$$

special case: no covariates. what is

$$\operatorname*{argmin}_w \frac{1}{n}\sum_{i=1}^{n} |y_i - w|?$$

- if pn of the y_i's are bigger than w,
- then as w increases to $w + \delta$,
- $\frac{1}{n}\sum_{i:y_i>w} |y_i - w|$ decreases by $p\delta$
- $\frac{1}{n}\sum_{i:y_i<w} |y_i - w|$ increases by $(1 - p)\delta$
- if $p = \frac{1}{2}$, objective stays the same

A: $w = \text{median}(y)$!

Notation

define the positive and negative parts of $x \in \mathbf{R}$

$$(x)_+ = \max(x, 0), \quad (x)_- = \max(-x, 0)$$

Quantile regression

Quantile regression: for $\alpha \in (0, 1)$,

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} \alpha(y_i - w^T x_i)_+ + (1-\alpha)(y_i - w^T x_i)_-$$

special case: no covariates. what is

$$\operatorname*{argmin}_w \frac{1}{n}\sum_{i=1}^{n} \alpha(y_i - w)_+ + (1-\alpha)(y_i - w)_-?$$

Quantile regression

Quantile regression: for $\alpha \in (0, 1)$,

$$\text{minimize} \quad \frac{1}{n} \sum_{i=1}^{n} \alpha(y_i - w^T x_i)_+ + (1 - \alpha)(y_i - w^T x_i)_-$$

special case: no covariates. what is

$$\operatorname*{argmin}_{w} \frac{1}{n} \sum_{i=1}^{n} \alpha(y_i - w)_+ + (1 - \alpha)(y_i - w)_-?$$

- if pn of the y_i's are bigger than w,
- then as w increases to $w + \delta$,
- first term decreases by $p\alpha\delta$
- second term increases by $(1 - p)(1 - \alpha)\delta$
- so if $p = 1 - \alpha$, objective stays the same

Quantile regression

Quantile regression: for $\alpha \in (0, 1)$,

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} \alpha(y_i - w^T x_i)_+ + (1-\alpha)(y_i - w^T x_i)_-$$

special case: no covariates. what is

$$\operatorname*{argmin}_w \frac{1}{n}\sum_{i=1}^{n} \alpha(y_i - w)_+ + (1-\alpha)(y_i - w)_- ?$$

- if pn of the y_i's are bigger than w,
- then as w increases to $w + \delta$,
- first term decreases by $p\alpha\delta$
- second term increases by $(1-p)(1-\alpha)\delta$
- so if $p = 1 - \alpha$, objective stays the same

A: w is the αth quantile of y!

Huber regression

Huber regression:

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} \mathbf{huber}(y_i - w^T x_i) + r(w)$$

where we define the Huber function

$$\mathbf{huber}(z) = \begin{cases} \frac{1}{2}z^2 & |z| \le 1 \\ |z| - \frac{1}{2} & |z| > 1 \end{cases}$$

Huber regression

Huber regression:

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} \mathbf{huber}(y_i - w^T x_i) + r(w)$$

where we define the Huber function

$$\mathbf{huber}(z) = \begin{cases} \frac{1}{2}z^2 & |z| \leq 1 \\ |z| - \frac{1}{2} & |z| > 1 \end{cases}$$

Huber decomposes error into a small (Gaussian) part and a large (sparse) part

$$\mathbf{huber}(x) = \inf_{s+n=x} |s| + \frac{1}{2}n^2$$

(proof: take derivative)

Robust statistics

the ℓ_1 and Huber loss functions are called **robust** loss functions

Q: when would you want to use a robust loss function?

Robust statistics

the ℓ_1 and Huber loss functions are called **robust** loss functions

Q: when would you want to use a robust loss function?
A: for **robustness** in the presence of large outliers

- large, infrequenct sensor malfunctions
- people lying on surveys
- anything that's not a sum of small iid random variables

Demo: robust regression

https://github.com/ORIE4741/demos/blob/master/robust_regression.ipynb

- least squares regression: mean error is 0
- ℓ_1 regression: median error is 0
- quantile regression: αth quantile of error is 0

Outline

Regression

Classification

The prediction space

Multiclass classification

Ordinal regression

Beyond linear models

Loss functions for classification

suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

Loss functions for classification

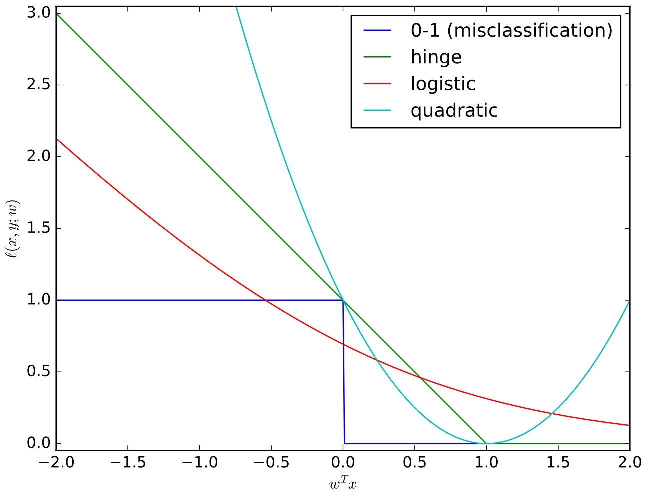
suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

- 0-1 loss $\mathbb{1}(y \neq \mathbf{sign}(w^T x))$
- quadratic loss $(y - w^T x)^2$
- hinge loss $(1 - yw^T x)_+$
- logistic loss $\log(1 + \exp\left(-w^T x\right))$
- ...

trade off dislike of false positives vs false negatives

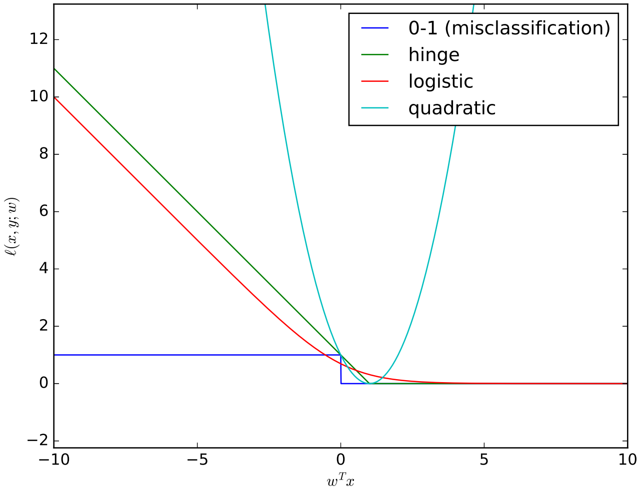
Loss functions for classification

$$y = 1$$



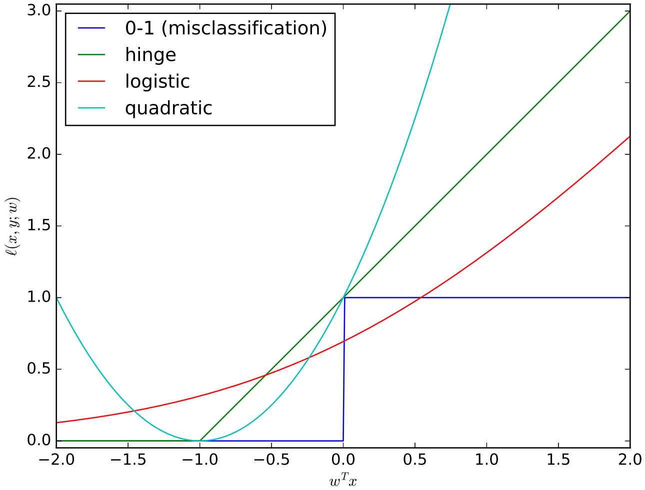
Loss functions for classification

$$y = 1$$



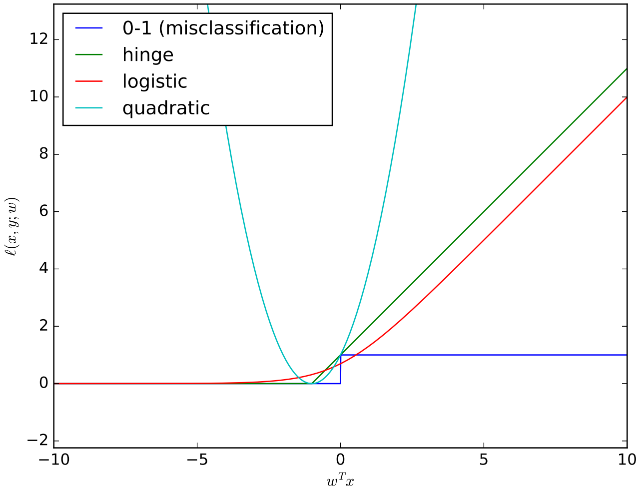
Loss functions for classification

$$y = -1$$



Loss functions for classification

$$y = -1$$



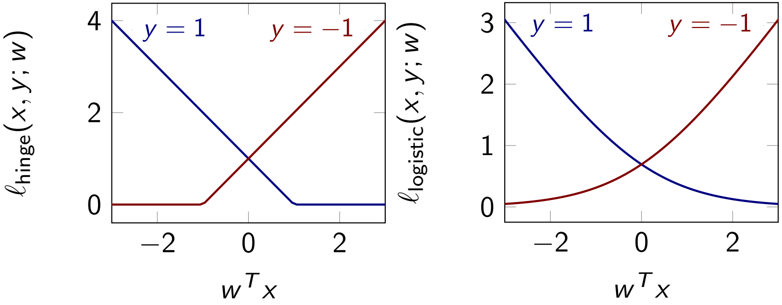
Losses for classification

- hinge loss

$$\ell_{\text{hinge}}(x, y; w) = (1 - yw^T x)_+$$

- logistic loss

$$\ell_{\text{logistic}}(x, y; w) = \log(1 + \exp\left(-yw^T x\right))$$



Logistic loss: interpretation

- logistic function maps real numbers to probabilities

$$\text{logistic}(u) = \frac{\exp(u)}{1+\exp(u)} = \frac{1}{1+\exp(-u)}$$

- suppose that given $w^T x$, y is a Bernoulli random variable

$$y = \begin{cases} 1 & \text{with prob } \text{logistic}(w^T x) \\ -1 & \text{with prob } (1 - \text{logistic}(w^T x)) = \text{logistic}(-w^T x) \end{cases}$$

  notice $\mathbb{P}(y|w, x) = \text{logistic}(yw^T x)$

- logistic loss is -log $\mathbb{P}(y|w, x)$

$$\begin{aligned} \ell_{\text{logistic}}(x, y; w) &= -\log(\text{logistic}(yw^T x)) \\ &= -\log\left(\frac{1}{1+\exp(-yw^T x)}\right) \\ &= \log\left(1+\exp\left(-yw^T x\right)\right) \end{aligned}$$

Hinge loss: interpretation

Hinge loss $\ell_{\text{hinge}}(x, y; w) = (1 - yw^T x)_+$. Solve

$$\begin{array}{ll} \text{minimize} & \|w\|^2 \\ \text{subject to} & \sum_{(x,y)\in\mathcal{D}} \ell_{\text{hinge}}(x, y; w) = 0 \end{array}$$

Hinge loss: interpretation

Hinge loss $\ell_{\text{hinge}}(x, y; w) = (1 - yw^T x)_+$. Solve

$$\begin{array}{ll} \text{minimize} & \|w\|^2 \\ \text{subject to} & \sum_{(x,y)\in\mathcal{D}} \ell_{\text{hinge}}(x, y; w) = 0 \end{array}$$

Poll: does this problem always have a solution?

A. yes

B. no

Hinge loss: interpretation

Hinge loss $\ell_{\text{hinge}}(x, y; w) = (1 - yw^T x)_+$. Solve

$$\begin{array}{ll} \text{minimize} & \|w\|^2 \\ \text{subject to} & \sum_{(x,y)\in\mathcal{D}} \ell_{\text{hinge}}(x, y; w) = 0 \end{array}$$

Poll: does this problem always have a solution?

A. yes

B. no

Poll: does this problem always have a solution, if the data is separable?

A. yes

B. no

Hinge loss: interpretation

Hinge loss $\ell_{\text{hinge}}(x, y; w) = (1 - yw^T x)_+$. Solve

$$\begin{array}{ll} \text{minimize} & \|w\|^2 \\ \text{subject to} & \sum_{(x,y)\in\mathcal{D}} \ell_{\text{hinge}}(x, y; w) = 0 \end{array}$$

- solution classifies every point correctly, with a safety margin:

$$yw^T x \geq 1, \qquad (x, y) \in \mathcal{D}.$$

Hinge loss: interpretation

Hinge loss $\ell_{\text{hinge}}(x, y; w) = (1 - yw^T x)_+$. Solve

$$\begin{array}{ll} \text{minimize} & \|w\|^2 \\ \text{subject to} & \sum_{(x,y)\in\mathcal{D}} \ell_{\text{hinge}}(x, y; w) = 0 \end{array}$$

- solution classifies every point correctly, with a safety margin:

$$yw^T x \geq 1, \qquad (x, y) \in \mathcal{D}.$$

- compare to perceptron:

Hinge loss: interpretation

Hinge loss $\ell_{\text{hinge}}(x, y; w) = (1 - yw^T x)_+$. Solve

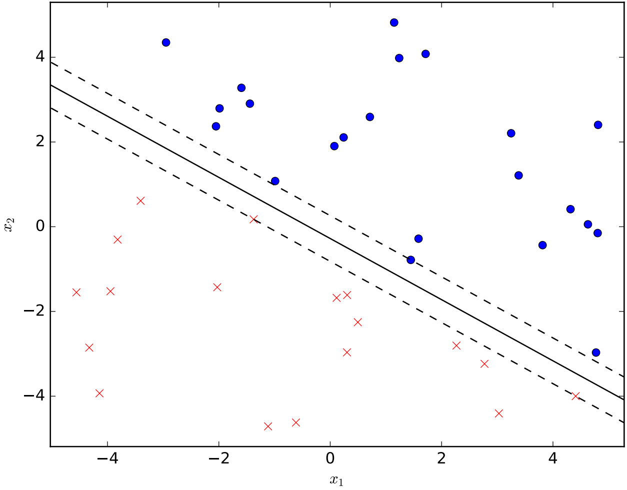
$$\begin{array}{ll} \text{minimize} & \|w\|^2 \\ \text{subject to} & \sum_{(x,y)\in\mathcal{D}} \ell_{\text{hinge}}(x, y; w) = 0 \end{array}$$

- solution classifies every point correctly, with a safety margin:

$$yw^T x \geq 1, \qquad (x, y) \in \mathcal{D}.$$

- compare to perceptron: unique solution, safety margin

Hinge loss: exact fit



solid line: $w^T x = 0$; dashed lines: $w^T x = \pm 1$

Hinge loss: interpretation

$$
\begin{aligned}
yx^T w &= \text{distance to classification boundary}, &\quad \text{if } \|w\| = 1 \\
yx^T \frac{w}{\|w\|} &= \text{distance to classification boundary}, &\quad \text{always}
\end{aligned}
$$

so if $yx^T w \geq 1$ for every $(x, y) \in \mathcal{D}$,

$$\text{distance to classification boundary} = yx^T \frac{w}{\|w\|} \geq \frac{1}{\|w\|}$$

for every $(x, y) \in \mathcal{D}$.

Support Vector Machine (SVM)

now instead solve the **support vector machine** problem (SVM)

$$\text{minimize} \quad \sum_{i=1}^{n} \ell_{\text{hinge}}(x_i, y_i; w) + \lambda \|w\|^2$$

Support Vector Machine (SVM)

now instead solve the **support vector machine** problem (SVM)

$$\text{minimize} \quad \sum_{i=1}^{n} \ell_{\text{hinge}}(x_i, y_i; w) + \lambda \|w\|^2$$

Poll: does this problem always have a solution?

A. yes

B. no

Support Vector Machine (SVM)

now instead solve the **support vector machine** problem (SVM)

$$\text{minimize} \quad \sum_{i=1}^{n} \ell_{\text{hinge}}(x_i, y_i; w) + \lambda \|w\|^2$$

Poll: does this problem always have a solution?

A. yes

B. no

- allows some mistakes
- trades off the severity of mistakes with the safety margin

Loss functions for classification

suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

Loss functions for classification

suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

A. 0-1 loss $\mathbb{1}(y \neq \textbf{sign}(w^T x))$
B. quadratic loss $(y - w^T x)^2$
C. hinge loss $(1 - yw^T x)_+$
D. logistic loss $\log(1 + \exp\left(-w^T x\right))$
E. ...

trade off dislike of false positives vs false negatives

Loss functions for classification

suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

A. 0-1 loss $\mathbb{1}(y \neq \textbf{sign}(w^T x))$
B. quadratic loss $(y - w^T x)^2$
C. hinge loss $(1 - yw^T x)_+$
D. logistic loss $\log(1 + \exp(-w^T x))$
E. ...

trade off dislike of false positives vs false negatives

properties: (select any loss that is)

- continuous?

Loss functions for classification

suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

A. 0-1 loss $\mathbb{1}(y \neq \textbf{sign}(w^T x))$
B. quadratic loss $(y - w^T x)^2$
C. hinge loss $(1 - yw^T x)_+$
D. logistic loss $\log(1 + \exp\left(-w^T x\right))$
E. ...

trade off dislike of false positives vs false negatives

properties: (select any loss that is)

- continuous? quadratic, hinge, logistic
- differentiable?

Loss functions for classification

suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

A. 0-1 loss $\mathbb{1}(y \neq \textbf{sign}(w^T x))$
B. quadratic loss $(y - w^T x)^2$
C. hinge loss $(1 - yw^T x)_+$
D. logistic loss $\log(1 + \exp\left(-w^T x\right))$
E. ...

trade off dislike of false positives vs false negatives

properties: (select any loss that is)

- continuous? quadratic, hinge, logistic
- differentiable? quadratic, logistic
- insensitive to outliers?

Loss functions for classification

suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

A. 0-1 loss $\mathbb{1}(y \neq \textbf{sign}(w^T x))$
B. quadratic loss $(y - w^T x)^2$
C. hinge loss $(1 - yw^T x)_+$
D. logistic loss $\log(1 + \exp\left(-w^T x\right))$
E. ...

trade off dislike of false positives vs false negatives

properties: (select any loss that is)

- continuous? quadratic, hinge, logistic
- differentiable? quadratic, logistic
- insensitive to outliers? 0-1
- sensitive to outliers?

Loss functions for classification

suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

A. 0-1 loss $\mathbb{1}(y \neq \textbf{sign}(w^T x))$
B. quadratic loss $(y - w^T x)^2$
C. hinge loss $(1 - yw^T x)_+$
D. logistic loss $\log(1 + \exp\left(-w^T x\right))$
E. ...

trade off dislike of false positives vs false negatives

properties: (select any loss that is)

- continuous? quadratic, hinge, logistic
- differentiable? quadratic, logistic
- insensitive to outliers? 0-1
- sensitive to outliers? quadratic
- quadratic?

Loss functions for classification

suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

A. 0-1 loss $\mathbb{1}(y \neq \mathbf{sign}(w^T x))$
B. quadratic loss $(y - w^T x)^2$
C. hinge loss $(1 - yw^T x)_+$
D. logistic loss $\log(1 + \exp\left(-w^T x\right))$
E. ...

trade off dislike of false positives vs false negatives

properties: (select any loss that is)

- continuous? quadratic, hinge, logistic
- differentiable? quadratic, logistic
- insensitive to outliers? 0-1
- sensitive to outliers? quadratic
- quadratic? quadratic
- probabilistic interpretation?

Loss functions for classification

suppose $\mathcal{Y} = \{-1, 1\}$. let $\ell(x, y; w) =$

A. 0-1 loss $\mathbb{1}(y \neq \textbf{sign}(w^T x))$
B. quadratic loss $(y - w^T x)^2$
C. hinge loss $(1 - yw^T x)_+$
D. logistic loss $\log(1 + \exp\left(-w^T x\right))$
E. ...

trade off dislike of false positives vs false negatives

properties: (select any loss that is)

- continuous? quadratic, hinge, logistic
- differentiable? quadratic, logistic
- insensitive to outliers? 0-1
- sensitive to outliers? quadratic
- quadratic? quadratic
- probabilistic interpretation? logistic

Outline

Ordinal regression and multiclass classification for trees

predicting different kinds of data is easy for trees:

- pick an error (impurity) metric
- choose split to greedily minimize error metric
- predict majority class (classification) or median (regression)

Ordinal regression and multiclass classification for trees

predicting different kinds of data is easy for trees:

- pick an error (impurity) metric
- choose split to greedily minimize error metric
- predict majority class (classification) or median (regression)

predicting different kinds of data is harder for linear models:

- model produces continuous value(s)
- to predict, we must map continuous output to correct kind of predictions (boolean, ordinal, nominal, . . .)

Recap linear models

- input space $\mathbf{R}^d$
- output space $\mathcal{Y}$
  - regression: $\mathcal{Y} = \mathbf{R}$
  - classification: $\mathcal{Y} = \{-1, 1\}$
- parameter space $\mathbf{R}^d$
- hypothesis class $h \in \mathcal{H}$

$$\mathcal{H} = \{h : \mathbf{R}^d \times \mathbf{R}^d \to \mathbf{R}\}$$

  e.g., $\mathcal{H} = \{h : h(x; w) = w^T x\}$
- rewrite the objective using this notation

$$\text{minimize} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i; w)) + r(w)$$

  with variable $w \in \mathbf{R}^d$

The prediction space

- input space $\mathcal{X}$
- output space $\mathcal{Y}$
- parameter space $\mathcal{W}$
- prediction space $\mathcal{Z}$
- hypothesis class $h \in \mathcal{H}$

$$\mathcal{H} = \{h : \mathcal{X} \times \mathcal{W} \to \mathcal{Z}\}$$

- rewrite the objective using this notation

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i; w)) + r(w)$$

  with variable $w \in \mathcal{W}$
- loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbf{R}$ maps between prediction space and output space

How to predict?

given

- a loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbf{R}$
- a hypothesis class $h : \mathcal{X} \times \mathcal{W}$, and
- model parameters $w \in \mathcal{W}$ fit to data

Q: how to predict $\hat{y}$ for a new sample x?

How to predict?

given

- a loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbf{R}$
- a hypothesis class $h : \mathcal{X} \times \mathcal{W}$, and
- model parameters $w \in \mathcal{W}$ fit to data

Q: how to predict $\hat{y}$ for a new sample x?
A: predict $\hat{y}$ by solving

$$\hat{y} = \operatorname*{argmin}_{y \in \mathcal{Y}} \ell(y, h(x; w))$$

How to predict?

given

- a loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbf{R}$
- a hypothesis class $h : \mathcal{X} \times \mathcal{W}$, and
- model parameters $w \in \mathcal{W}$ fit to data

Q: how to predict $\hat{y}$ for a new sample x?
A: predict $\hat{y}$ by solving

$$\hat{y} = \operatorname*{argmin}_{y \in \mathcal{Y}} \ell(y, h(x; w))$$

MLE interpretation: if $z = w^T x$, $\ell(y, z) = -\log P(y \mid z)$, then $\hat{y}$ is **most probable** $y \in \mathcal{Y}$ given $z = w^T x$.

Prediction: examples

given

- a loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbf{R}$
- a hypothesis class $h : \mathcal{X} \times \mathcal{W}$, and
- model parameters $w \in \mathcal{W}$ fit to data

predict $\hat{y}$ by solving

$$\hat{y} = \operatorname*{argmin}_{y \in \mathcal{Y}} \ell(y, h(x; w))$$

for quadratic loss, $\mathcal{Y} = \mathbf{R}$, $w^T x = 5.2$, $\hat{y} =$

A. 5.2
B. 1
C. -5.2
D. -1
E. 0

Prediction: examples

given

- a loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbf{R}$
- a hypothesis class $h : \mathcal{X} \times \mathcal{W}$, and
- model parameters $w \in \mathcal{W}$ fit to data

predict $\hat{y}$ by solving

$$\hat{y} = \operatorname*{argmin}_{y \in \mathcal{Y}} \ell(y, h(x; w))$$

for quadratic loss, $\mathcal{Y} = \{-1, 1\}$, $w^T x = 5.2$, $\hat{y} =$

A. 5.2
B. 1
C. -5.2
D. -1
E. 0

Prediction: examples

given

- a loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbf{R}$
- a hypothesis class $h : \mathcal{X} \times \mathcal{W}$, and
- model parameters $w \in \mathcal{W}$ fit to data

predict $\hat{y}$ by solving

$$\hat{y} = \operatorname*{argmin}_{y \in \mathcal{Y}} \ell(y, h(x; w))$$

for hinge loss, $\mathcal{Y} = \{-1, 1\}$, $w^T x = 5.2$, $\hat{y} =$

A. 5.2
B. 1
C. -5.2
D. -1
E. 0

Prediction: examples

given

- a loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbf{R}$
- a hypothesis class $h : \mathcal{X} \times \mathcal{W}$, and
- model parameters $w \in \mathcal{W}$ fit to data

predict $\hat{y}$ by solving

$$\hat{y} = \operatorname*{argmin}_{y \in \mathcal{Y}} \ell(y, h(x; w))$$

for huber loss, $\mathcal{Y} = \mathbf{R}$, $w^T x = 5.2$, $\hat{y} =$

A. 5.2
B. 1
C. -5.2
D. -1
E. 0

Prediction: examples

given

- a loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbf{R}$
- a hypothesis class $h : \mathcal{X} \times \mathcal{W}$, and
- model parameters $w \in \mathcal{W}$ fit to data

predict $\hat{y}$ by solving

$$\hat{y} = \operatorname*{argmin}_{y \in \mathcal{Y}} \ell(y, h(x; w))$$

for logistic loss, $\mathcal{Y} = \{-1, 1\}$, $w^T x = 5.2$, $\hat{y} =$

A. 5.2
B. 1
C. -5.2
D. -1
E. 0

Outline

Multiclass classification

how to predict **nominal** values?

Multiclass classification

how to predict **nominal** values?

- **idea 1: classification**
  1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
  2. predict entries of $\psi(y)$
  3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Multiclass classification

how to predict **nominal** values?

- **idea 1: classification**
  1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
  2. predict entries of $\psi(y)$
  3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$
- **idea 2: learning probabilities**
  1. learn the probability $\mathbb{P}(y = y' \mid x)$ for every $y' \in \mathcal{Y}$
  2. predict $y = \mathrm{argmax}_{y' \in \mathcal{Y}} \mathbb{P}(y = y' \mid x)$
  3. $z = h(x; w)$ will parametrize probability distribution

Multiclass classification: examples

examples:

- classifying which breed of dog is present in an image
- classifying the type of heart disease given a electrocardiogram (EKG)
- predicting if a water well is ok, needs repair, or is defunct
- more examples from projects?

Multiclass classification via binary classification

idea 1: classification

1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Q: how to pick $\psi(y)$? (suppose $\mathcal{Y} = \{1, \ldots, k\}$)

Multiclass classification via binary classification

idea 1: classification

1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Q: how to pick $\psi(y)$? (suppose $\mathcal{Y} = \{1, \ldots, k\}$)

Multiclass classification via binary classification

idea 1: classification

1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Q: how to pick $\psi(y)$? (suppose $\mathcal{Y} = \{1, \ldots, k\}$)

- one-hot encoding:

$$\begin{aligned} \psi(y) &= (-1, \ldots, \overbrace{1}^{y\text{th entry}}, \ldots, -1) \\ &= 2(\mathbf{1}(y = 1), \ldots, \mathbf{1}(y = k)) - 1 \in \{-1, 1\}^k \end{aligned}$$

(resulting scheme is called **one-vs-all** classification)

Multiclass classification via binary classification

idea 1: classification

1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Q: how to pick $\psi(y)$? (suppose $\mathcal{Y} = \{1, \ldots, k\}$)

- one-hot encoding:

$$\begin{aligned} \psi(y) &= (-1, \ldots, \overbrace{1}^{y\text{th entry}}, \ldots, -1) \\ &= 2(\mathbf{1}(y=1), \ldots, \mathbf{1}(y=k)) - 1 \in \{-1, 1\}^k \end{aligned}$$

  (resulting scheme is called **one-vs-all** classification)
- binary codes:

Multiclass classification via binary classification

idea 1: classification

1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Q: how to pick $\psi(y)$? (suppose $\mathcal{Y} = \{1, \ldots, k\}$)

- one-hot encoding:

$$\begin{aligned}\psi(y) &= (-1, \ldots, \overbrace{1}^{y\text{th entry}}, \ldots, -1) \\ &= 2(\mathbf{1}(y=1), \ldots, \mathbf{1}(y=k)) - 1 \in \{-1, 1\}^k\end{aligned}$$

  (resulting scheme is called **one-vs-all** classification)
- binary codes:
  - define binary expansion of y, $\mathrm{bin}(y) \in \{-1, 1\}^{\log(k)}$
  - let $\psi(y) = 2\,\mathrm{bin}(y) - 1 \in \{-1, 1\}^{\log(k)}$

Multiclass classification via binary classification

idea 1: classification

1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Q: how to pick $\psi(y)$? (suppose $\mathcal{Y} = \{1, \ldots, k\}$)

- one-hot encoding:

$$\begin{aligned}\psi(y) &= (-1, \ldots, \overbrace{1}^{y\text{th entry}}, \ldots, -1) \\ &= 2(\mathbf{1}(y=1), \ldots, \mathbf{1}(y=k)) - 1 \in \{-1, 1\}^k\end{aligned}$$

  (resulting scheme is called **one-vs-all** classification)
- binary codes:
  - define binary expansion of y, $\mathrm{bin}(y) \in \{-1, 1\}^{\log(k)}$
  - let $\psi(y) = 2\,\mathrm{bin}(y) - 1 \in \{-1, 1\}^{\log(k)}$
- error-correcting codes

Multiclass classification via binary classification

idea 1: classification

1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Q: how to pick $\psi(y)$? (suppose $\mathcal{Y} = \{1, \ldots, k\}$)

- one-hot encoding:

$$\begin{aligned}\psi(y) &= (-1, \ldots, \overbrace{1}^{y\text{th entry}}, \ldots, -1) \\ &= 2(\mathbf{1}(y = 1), \ldots, \mathbf{1}(y = k)) - 1 \in \{-1, 1\}^k\end{aligned}$$

  (resulting scheme is called **one-vs-all** classification)
- binary codes:
  - define binary expansion of y, $\text{bin}(y) \in \{-1, 1\}^{\log(k)}$
  - let $\psi(y) = 2\,\text{bin}(y) - 1 \in \{-1, 1\}^{\log(k)}$
- error-correcting codes

these vary in the **dimension** of $\psi(y)$ = dimension of z

Multiclass classification via binary classification

idea 1: classification

1. encode $y \in \mathcal{Y}$ as a vector $\psi(y) \in \{-1, 1\}^k$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Q: how to predict entries of $\psi(y) \in \{-1, 1\}^k$?

Multiclass classification via binary classification

idea 1: classification

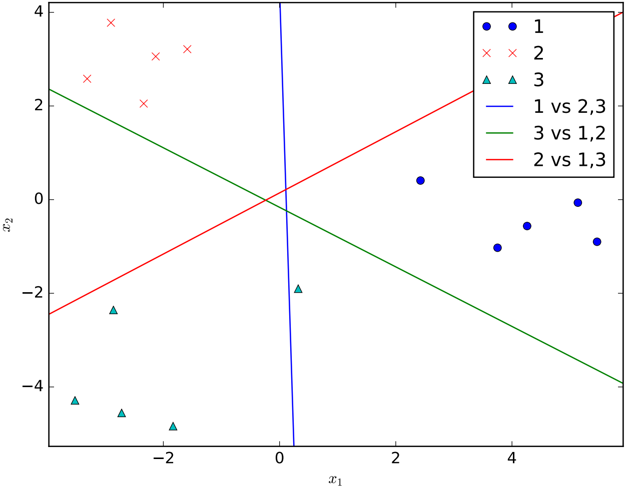
1. encode $y \in \mathcal{Y}$ as a vector $\psi(y) \in \{-1, 1\}^k$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Q: how to predict entries of $\psi(y) \in \{-1, 1\}^k$?

- reduce to a bunch of binary problems!
- let $W \in \mathbf{R}^{k \times d}$, so $z = Wx \in \mathbf{R}^k$
- pick your favorite loss function ℓ^{bin} for binary classification
- fit parameter W by minimizing loss function

$$\ell^{\text{nom}}(y, z) = \sum_{i=1}^{k} \ell^{\text{bin}}(\psi(y)_i, z_i)$$

One-vs-All classification



Multiclass classification via learning probabilities

(for concreteness, suppose $\mathcal{Y} = \{1, \ldots, k\}$)

idea 2: learning probabilities

1. learn the probability $\mathbb{P}(y = y' \mid x)$ for every $y' \in \mathcal{Y}$
2. predict $y = \mathrm{argmax}_{y' \in \mathcal{Y}} \mathbb{P}(y = y' \mid x)$
3. $z = h(x; w) \in \mathbf{R}^k$ will parametrize probability distribution

Q: how to predict probabilities?

Multiclass classification via learning probabilities

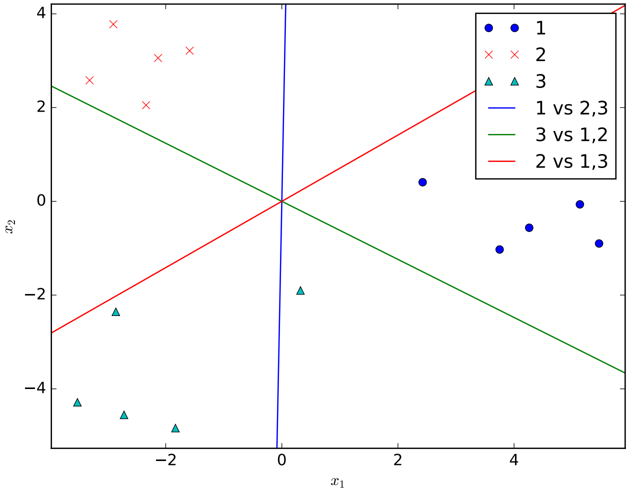
- let $W \in \mathbf{R}^{k \times d}$, so $Wx \in \mathbf{R}^k$
- **multinomial logit** takes a hint from logistic: let $z = h(x; W) = Wx$, and suppose

$$\mathbb{P}(y = i \mid z) = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)}$$

  (ensures probabilities are positive and sum to 1)
- fit by minimizing negative log likelihood

$$\begin{aligned} \ell(y, z) &= -\log(\mathbb{P}(y \mid z)) \\ &= -\log\left(\frac{\exp(z_y)}{\sum_{j=1}^{k} \exp(z_j)}\right) \end{aligned}$$

Multinomial classification



Outline

Ordinal regression

how to predict **ordinal** values?

Ordinal regression

how to predict **ordinal** values?

- **idea 0: regression**
  1. encode $y \in \mathcal{Y}$ in $\mathbf{R}$

Ordinal regression

how to predict **ordinal** values?

- **idea 0: regression**
  1. encode $y \in \mathcal{Y}$ in $\mathbf{R}$
- **idea 1: classification**
  1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
  2. predict entries of $\psi(y)$
  3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

Ordinal regression

how to predict **ordinal** values?

- **idea 0: regression**
  1. encode $y \in \mathcal{Y}$ in $\mathbf{R}$
- **idea 1: classification**
  1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
  2. predict entries of $\psi(y)$
  3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$
- **idea 2: learning probabilities**
  1. learn the probability $\mathbb{P}(y = y' \mid x)$ for every $y' \in \mathcal{Y}$
  2. predict $y = \mathrm{argmax}_{y' \in \mathcal{Y}} \mathbb{P}(y = y' \mid x)$
  3. $z = h(x; w)$ will parametrize probability distribution

Ordinal regression

(for concreteness, suppose $\mathcal{Y} = \{1, \ldots, k\}$)

idea 0: regression

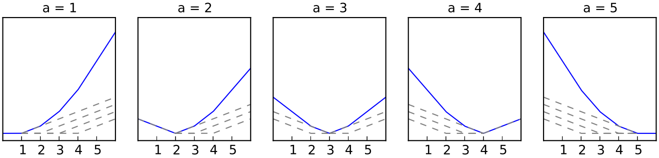
1. encode $y \in \mathcal{Y}$ in $\mathbf{R}$
2. predict with $\mathcal{Z} = \mathbf{R}$

- quadratic loss

$$\ell(y, z) = (y - z)^2$$

- ordinal hinge loss

$$\ell(y, z) = \sum_{y'=1}^{y-1} (1 - z + y')_+ + \sum_{y'=y+1}^{k} (1 + z - y')_+$$



Ordinal regression via predicting a vector

idea 1: classification (suppose $\mathcal{Y} = \{1, \dots, k\}$)

1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

- how to encode y as a vector?

Ordinal regression via predicting a vector

idea 1: classification (suppose $\mathcal{Y} = \{1, \ldots, k\}$)

1. encode $y \in \mathcal{Y}$ as a vector $\psi(y)$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

- how to encode y as a vector? how about

$$\psi(y) = (1, \ldots, 1, \overbrace{-1}^{y\text{th entry}}, \ldots, -1) \in \{-1, 1\}^{k-1}$$

  (resulting scheme is called **bigger-vs-smaller** classification)
- let $W \in \mathbf{R}^{k-1 \times d}$, so $z = Wx \in \mathbf{R}^{k-1}$
- pick your favorite loss function ℓ^{bin} for binary classification
- fit model W by minimizing loss function

$$\ell^{\text{ord}}(y; z) = \sum_{i=1}^{k-1} \ell^{\text{bin}}(\psi(y)_i; z_i)$$

Ordinal regression via predicting a vector

- set $\psi(y) = (1, \ldots, 1, \overbrace{-1}^{y\text{th entry}}, \ldots, -1) \in \{-1, 1\}^{k-1}$
- let $W \in \mathbf{R}^{k-1 \times d}$, so $z = Wx \in \mathbf{R}^{k-1}$
- fit parameter W by minimizing loss function

$$\ell^{\text{ord}}(y; z) = \sum_{i=1}^{k-1} \ell^{\text{bin}}(\psi(y)_i, z_i)$$

- ith column of W defines a line separating levels $y \leq i$ from levels $y > i$

Q: How to predict $\hat{y}$ given x and W?

Ordinal regression via predicting a vector

- set $\psi(y) = (1, \ldots, 1, \overbrace{-1}^{y\text{th entry}}, \ldots, -1) \in \{-1, 1\}^{k-1}$
- let $W \in \mathbf{R}^{k-1 \times d}$, so $z = Wx \in \mathbf{R}^{k-1}$
- fit parameter W by minimizing loss function

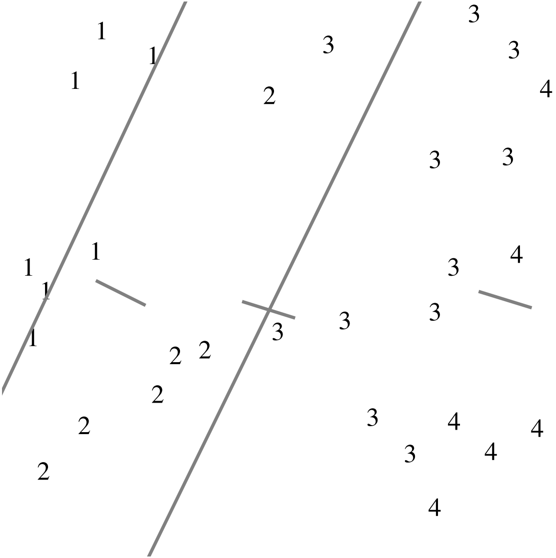
$$\ell^{\text{ord}}(y; z) = \sum_{i=1}^{k-1} \ell^{\text{bin}}(\psi(y)_i, z_i)$$

- ith column of W defines a line separating levels $y \leq i$ from levels $y > i$

Q: How to predict $\hat{y}$ given x and W?
A: Compute $z = Wx$, and predict

$$\hat{y} = \operatorname*{argmin}_{y \in \mathcal{Y}} \ell^{\text{ord}}(y; z)$$

Ordinal regression



Outline

Coding and decoding

we now have four different spaces

- input space $\mathcal{X}$
- output space $\mathcal{Y}$
- parameter space $\mathcal{W}$
- prediction space $\mathcal{Z}$

a **model** is given by a choice of

- loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbf{R}$,
- regularizer $r : \mathcal{W} \to \mathbf{R}$, and
- hypothesis class $h : \mathcal{X} \times \mathcal{W} \to \mathcal{Z}$

we **fit** the model by solving

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i; w)) + r(w)$$

to find $w \in \mathcal{W}$

given a parameter $w \in \mathcal{W}$ and a new input $x \in \mathcal{X}$, we **predict** $y \in \mathcal{Y}$ by solving

$$y = \operatorname*{argmin}_{y \in \mathcal{Y}} \ell(y, h(x_i; w))$$

What models fit in this framework?

- linear models
- linear models with feature transformations
- decision trees
- neural networks
- generalized additive models
- unsupervised learning (!)
- ...

Resources

- quantile regression https://www.cscu.cornell.edu/news/statnews/stnews70.pdf