ORIE 4741: Learning with Big Messy Data

Loss functions

Professor Udell
Operations Research and Information Engineering
Cornell

November 7, 2017
Outline

Maximum likelihood estimation

Regression

Classification

The prediction space

Multiclass classification

Ordinal regression

Beyond linear models
Probabilistic setup

▶ suppose you know a function $p : \mathbb{R} \to [0, 1]$ so that
\[
\mathbb{P}(y_i = y \mid x_i, w) = p(y; x_i, w)
\]
▶ for example, if $y_i = w^T x_i + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, then
\[
\mathbb{P}(y_i = y \mid x_i, w) \sim \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(y_i - w^T x_i)^2}{2\sigma^2} \right)
\]
▶ likelihood of data given parameter $w$ is
\[
L(\mathcal{D}; w) = \prod_{i=1}^{n} \mathbb{P}(y_i = y \mid x_i, w)
\]
▶ for example, for linear model with Gaussian error,
\[
L(\mathcal{D}; w) \sim \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(y_i - w^T x_i)^2}{2\sigma^2} \right)
\]
Maximum Likelihood Estimation (MLE)

**MLE:** choose \( w \) to maximize \( L(\mathcal{D}; w) \)

- **likelihood**
  \[
  L(\mathcal{D}; w) = \prod_{i=1}^{n} p(y_i; x_i, w)
  \]

- **negative log likelihood**
  \[
  \ell(\mathcal{D}; w) = -\log L(\mathcal{D}; w)
  \]

- **maximize** \( L(\mathcal{D}; w) \) \( \iff \) **minimize** \( \ell(\mathcal{D}; w) \)
Example: Maximum Likelihood Estimation (MLE)

- for linear model with Gaussian error,

\[
\ell(D; w) \sim - \log \left( \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(y_i - w^T x_i)^2}{2\sigma^2} \right) \right)
\]

\[
= \sum_{i=1}^{n} - \log \left( \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(y_i - w^T x_i)^2}{2\sigma^2} \right) \right)
\]

\[
= \sum_{i=1}^{n} \left( \frac{1}{2} \log(2\pi\sigma^2) - \log \left( \exp \left( -\frac{(y_i - w^T x_i)^2}{2\sigma^2} \right) \right) \right)
\]

\[
= \frac{n}{2} \log(2\pi\sigma^2) + \sum_{i=1}^{n} \frac{1}{2\sigma^2} (y_i - w^T x_i)^2
\]

so maximize \( L(D; w) \iff \text{minimize} \sum_{i=1}^{n} (y_i - w^T x_i)^2 \)
what if I have beliefs about what $w$ should be before I begin?

- $w$ should be small
- $w$ should be sparse
- $w$ should be nonnegative

**idea:** impose **prior** on $w$ to specify

$$\mathbb{P}(w)$$

before seeing any data
Maximum-a-posteriori estimation

after I see data, compute posterior probability

\[ P(D; w) = P(D \mid w) P(w) \]

**maximum a posteriori (MAP estimation):** choose \( w \) to maximize posterior probability
Maximum-a-posteriori estimation

after I see data, compute posterior probability

\[ P(D; w) = P(D \mid w) P(w) \]

maximum a posteriori (MAP estimation): choose \( w \) to maximize posterior probability

n.b. this is not what a true Bayesian would do
(see, e.g., Bishop, Pattern Recognition and Machine Learning)
Ridge regression: interpretation as MAP estimator

- prior probability of model $w \sim \mathcal{N}(0, I_d)$
- noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, $i = 1, \ldots, n$
- response $y_i = w^T x_i + \epsilon_i$, $i = 1, \ldots, n$

$$
\mathbb{P}(D; w) = \mathbb{P}(D \mid w) \mathbb{P}(w)
\approx \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right) \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{w_i^2}{2}\right)
= (2\pi\sigma^2)^{-\frac{n}{2}} \prod_{i=1}^{n} \left(\exp\left(-\frac{(y_i - w^T x_i)^2}{2\sigma^2}\right)\right) (2\pi)^{-\frac{d}{2}} \prod_{i=1}^{d} \left(\exp\left(-\frac{w_i^2}{2}\right)\right)
\ell(D; w) = -\log(\mathbb{P}(D; w))
= \frac{n}{2} \log(2\pi\sigma^2) + \frac{d}{2} \log(2\pi) + \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \frac{1}{2} \sum_{i=1}^{d} w_i^2
$$
Ridge regression: interpretation as MAP estimator

- prior probability of model $w \sim \mathcal{N}(0, I_d)$
- noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, $i = 1, \ldots, n$
- response $y_i = w^T x_i + \epsilon_i$, $i = 1, \ldots, n$

\[
P(D; w) = P(D | w) P(w)
\sim \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(y_i - w^T x_i)^2}{2\sigma^2} \right) \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{w_i^2}{2} \right)
= (2\pi\sigma^2)^{-\frac{n}{2}} \prod_{i=1}^{n} \left( \exp \left( -\frac{(y_i - w^T x_i)^2}{2\sigma^2} \right) \right) (2\pi)^{-\frac{d}{2}} \prod_{i=1}^{d} \left( \exp \left( -\frac{w_i^2}{2} \right) \right)
\]

\[
\ell(D; w) = -\log (P(D; w))
= \frac{n}{2} \log(2\pi\sigma^2) + \frac{d}{2} \log(2\pi) + \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \frac{1}{2} \sum_{i=1}^{d} w_i^2
\]

... aha! and we have ridge regression with $\lambda = \sigma^2$
Recap: regularized empirical risk minimization

choose model by solving

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

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

- parameter vector $w \in \mathbb{R}^d$
- loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^d \rightarrow \mathbb{R}$
- regularizer $r : \mathbb{R}^d \rightarrow \mathbb{R}$
Recap: regularized empirical risk minimization

choose model by solving

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

with variable \( w \in \mathbb{R}^d \)

- parameter vector \( w \in \mathbb{R}^d \)
- loss function \( \ell : \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^d \rightarrow \mathbb{R} \)
- regularizer \( r : \mathbb{R}^d \rightarrow \mathbb{R} \)

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

Maximum likelihood estimation

Regression

Classification

The prediction space

Multiclass classification

Ordinal regression

Beyond linear models
Loss functions for real-valued data

- quadratic
- $\ell_1$
- huber
- quantile
- ...

...
Least squares regression finds the mean

least squares ($\ell_2$) 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

$$\arg\min_w \frac{1}{n} \sum_{i=1}^{n} (y_i - w)^2?$$
Least squares regression finds the mean

least squares ($\ell_2$) 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

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

A: mean($y$)!
ℓ₁ regression finds the median

ℓ₁ regression:

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

special case: no covariates. what is

\[
\arg\min_w \frac{1}{n} \sum_{i=1}^{n} |y_i - w|
\]
\( \ell_1 \) regression finds the median

\( \ell_1 \) regression:

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

special case: no covariates. what is

\[
\arg\min_w \frac{1}{n} \sum_{i=1}^{n} |y_i - w|?
\]

- 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
\( \ell_1 \) regression finds the median

\( \ell_1 \) regression:

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

special case: no covariates. what is

\[
\arg\min_w \frac{1}{n} \sum_{i=1}^{n} |y_i - w|
\]

- 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:** \( w = \text{median}(y) \)!
Huber regression

Huber regression:

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

where we define the Huber function

\[
\text{huber}(z) = \begin{cases} 
\frac{1}{2}z^2 & |z| \leq 1 \\
|z| - \frac{1}{2} & |z| > 1 
\end{cases}
\]
Huber regression

Huber regression:

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

where we define the Huber function

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

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

\[
\text{huber}(x) = \inf_{s+n=x} |s| + \frac{1}{2} n^2
\]

(proof: take derivative)
Robust statistics

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

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

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

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

$$(x)_+ = \max(x, 0), \quad (x)_- = \max(-x, 0)$$
Quantile regression finds the right quantile

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

\[
\arg \min_w \frac{1}{n} \sum_{i=1}^{n} \alpha (y_i - w)_+ + (1 - \alpha) (y_i - w)_-
\]
Quantile regression finds the right quantile

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

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

special case: no covariates. what is $\argmin_w \frac{1}{n} \sum_{i=1}^{n} \alpha (y_i - w)_+ + (1 - \alpha) (y_i - w)_-$?

- 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 finds the right quantile

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

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

- 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 $\alpha$th quantile of $y$!
Demo: robust regression
Outline

Maximum likelihood estimation

Regression

Classification

The prediction space

Multiclass classification

Ordinal regression

Beyond linear models
Loss functions for classification

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

- 0-1 loss $\mathbb{1}(y \neq \text{sign}(w^Tx))$
- quadratic loss $(y - w^Tx)^2$
- hinge loss $(1 - yw^Tx)_+$
- logistic loss $\log(1 + \exp(-w^Tx))$
- ... 

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**
  \[
  \ell_{\text{logistic}}(x, y; w) = \log(1 + \exp(-yw^T x))
  \]
Logistic loss: interpretation

- logistic function maps real numbers to probabilities
  \[
  \text{logistic}(u) = \frac{\exp(u)}{1 + \exp(u)} = \frac{1}{1 + \exp(-u)}
  \]

- given \( w^T x \), \( y \) is a Bernoulli random variable
  \[
  y = \begin{cases} 
  1 & \text{with prob } \text{logistic}(w^T x) \\
  -1 & \text{with prob } (1 - \text{logistic}(w^T x)) = \text{logistic}(-w^T x) 
  \end{cases}
  \]

- logistic loss is \(-\log\) likelihood of \( y \) given \( w^T x \)
  \[
  \ell_{\text{logistic}}(x, y; w) = -\log(\text{logistic}(yw^T x))
  \]
  \[
  = -\log\left( \frac{1}{1 + \exp(-yw^T x)} \right)
  = \log\left( 1 + \exp\left( -yw^T x \right) \right)
  \]
suppose we solve

\[
\begin{align*}
\text{minimize} & \quad \|w\|^2 \\
\text{subject to} & \quad \sum_{i=1}^{n} \ell_{\text{hinge}}(x_i, y_i; w) = 0
\end{align*}
\]

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

\[\text{solution classifies every point correctly, with a safety margin: for every } i = 1, \ldots, n,\]
\[1 \leq y_i w^T x_i\]

\[\text{compare to perceptron}\]
\[\text{as } \|w\|^2 \text{ gets smaller, the margin gets bigger}\]
Hinge loss: interpretation
now instead solve the support vector machine problem (SVM)

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

- 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 $\ell(x, y; w) =$

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

trade off dislike of false positives vs false negatives
Loss functions for classification

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

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

trade off dislike of false positives vs false negatives properties:

- continuous?
Loss functions for classification

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

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

trade off dislike of false positives vs false negatives properties:

- continuous? quadratic, hinge, logistic
- differentiable?
Loss functions for classification

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

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

trade off dislike of false positives vs false negatives properties:

- continuous? quadratic, hinge, logistic
- differentiable? quadratic, logistic
- insensitive to outliers?
Loss functions for classification

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

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

trade off dislike of false positives vs false negatives properties:

- 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 $\ell(x, y; w) =$

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

trade off dislike of false positives vs false negatives properties:

- 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 $\ell(x, y; w) =$

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

trade off dislike of false positives vs false negatives properties:

- 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 $\ell(x, y; w) =$

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

trade off dislike of false positives vs false negatives properties:

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

Maximum likelihood estimation

Regression

Classification

The prediction space

Multiclass classification

Ordinal regression

Beyond linear models
Recap linear models

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

$$\mathcal{H} = \{ h : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R} \}$$

$$= \{ h(x; w) = w^T x \}$$

- rewrite the objective using this notation

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

with variable $w \in \mathbb{R}^d$
The prediction space

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

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

- rewrite the objective using this notation

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

with variable $w \in \mathcal{W}$

- loss function $\ell : \mathcal{Y} \times \mathcal{Z} \rightarrow \mathbf{R}$ maps between prediction space and output space
How to predict?

given

- a loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbb{R}$
- 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 $\ell : \mathcal{Y} \times \mathcal{Z} \rightarrow \mathbb{R}$
- 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

$$
\hat{y} = \arg\min_{y \in \mathcal{Y}} \ell(y, h(x; w))
$$

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

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

predict $\hat{y}$ by solving

$$\hat{y} = \arg\min_{y \in \mathcal{Y}} \ell(y, h(x; w))$$

- for quadratic loss,
Prediction: examples

given

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

predict \( \hat{y} \) by solving

\[
\hat{y} = \arg\min_{y \in \mathcal{Y}} \ell(y, h(x; w))
\]

- for quadratic loss, \( \mathcal{Y} = \mathcal{Z} \), and \( \hat{y} = w^T x \)
- for \( \ell_1 \), Huber loss, or quantile loss,
Prediction: examples

given

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

predict $\hat{y}$ by solving

$$\hat{y} = \arg\min_{y \in \mathcal{Y}} \ell(y, h(x; w))$$

▶ for quadratic loss, $\mathcal{Y} = \mathcal{Z}$, and $\hat{y} = w^T x$
▶ for $\ell_1$, Huber loss, or quantile loss, $\mathcal{Y} = \mathcal{Z}$, and $\hat{y} = w^T x$
▶ for hinge loss $\ell(y, h(x; w)) = (1 - yw^T x)_+$,
Prediction: examples

given

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

predict $\hat{y}$ by solving

$$\hat{y} = \arg\min_{y \in \mathcal{Y}} \ell(y, h(x; w))$$

- for quadratic loss, $\mathcal{Y} = \mathcal{Z}$, and $\hat{y} = w^T x$
- for $\ell_1$, Huber loss, or quantile loss, $\mathcal{Y} = \mathcal{Z}$, and $\hat{y} = w^T x$
- for hinge loss $\ell(y, h(x; w)) = (1 - yw^T x)_+$, $\hat{y} = \text{sign}(w^T x)$
- for logistic loss $\ell(y, h(x; w)) = \log(1 + \exp(-yw^T x))$, 
Prediction: examples

given

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

predict \( \hat{y} \) by solving

\[
\hat{y} = \arg\min_{y \in \mathcal{Y}} \ell(y, h(x; w))
\]

- for quadratic loss, \( \mathcal{Y} = \mathcal{Z} \), and \( \hat{y} = w^T x \)
- for \( \ell_1 \), Huber loss, or quantile loss, \( \mathcal{Y} = \mathcal{Z} \), and \( \hat{y} = w^T x \)
- for hinge loss \( \ell(y, h(x; w)) = (1 - yw^T x)_+ \), \( \hat{y} = \text{sign}(w^T x) \)
- for logistic loss \( \ell(y, h(x; w)) = \log(1 + \exp(-yw^T x)) \), \( \hat{y} = \text{sign}(w^T x) \)
Outline

Maximum likelihood estimation

Regression

Classification

The prediction space

**Multiclass classification**

Ordinal regression

Beyond linear models
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 $y = \arg\max_{y' \in \mathcal{Y}} \mathbb{P}(y = y' \mid x)$
  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 $\psi(y) \in \{-1, 1\}^k$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

**Q:** how to 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 $\psi(y) \in \{-1, 1\}^k$
2. predict entries of $\psi(y)$
3. each entry of $z = h(x; w)$ will predict corresponding entry of $\psi(y)$

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

- one-hot encoding: if $y = i$,

$$
\psi(y) = (-1, \ldots, \underbrace{1}_i, \ldots, -1) \in \{-1, 1\}^k
$$

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

idea 1: classification

1. encode $y \in Y$ 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 pick $\psi(y)$? (suppose $Y = \{1, \ldots, k\}$)

- one-hot encoding: if $y = i$,
  
  $$\psi(y) = (-1, \ldots, \underbrace{1}_{i\text{th entry}}, \ldots, -1) \in \{-1, 1\}^k$$

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

- binary codes:
  
  - define binary expansion of $y$, $\text{bin}(y) \in \{0, 1\}^{\log(k)}$
  - let $\psi(y) = 2 \text{bin}(y) - 1 \in \{-1, 1\}^{\log(k)}$
Multiclass classification via binary classification

idea 1: classification

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

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

- one-hot encoding: if \( y = i \),
  
  \[
  \psi(y) = (-1, \ldots, 1, \ldots, -1) \in \{-1, 1\}^k
  \]

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

- binary codes:
  
  - define binary expansion of \( y \), \( \text{bin}(y) \in \{0, 1\}^{\log(k)} \)
  - let \( \psi(y) = 2 \text{bin}(y) - 1 \in \{-1, 1\}^{\log(k)} \)

- error-correcting codes
Multiclass classification via binary classification

**idea 1: classification**

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

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

- one-hot encoding: if \( y = i \),
  \[
  \psi(y) = (-1, \ldots, 1_{ith\ \text{entry}}, \ldots, -1) \in \{-1, 1\}^k
  \]
  (resulting scheme is called **one-vs-all** classification)

- binary codes:
  - define binary expansion of \( y \), \( \text{bin}(y) \in \{0, 1\}^{\log(k)} \)
  - let \( \psi(y) = 2 \cdot \text{bin}(y) - 1 \in \{-1, 1\}^{\log(k)} \)

- error-correcting codes

these vary in the **dimension** of \( \psi(y) = \text{dimension of } z \)
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 $\psi(y) \in \{-1, 1\}^k$?
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 $W \in \mathbb{R}^{k \times d}$, so $z = Wx \in \mathbb{R}^k$
- pick your favorite loss function $\ell^{\text{bin}}$ 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

The diagram illustrates the concept of One-vs-All classification in a two-dimensional space. The axes represent the features $x_1$ and $x_2$, and the points are classified into three categories represented by different symbols and colors:

- Circles (●) represent category 1.
- Xs (×) represent category 2.
- Triangles (▲) represent category 3.

The lines indicate the decision boundaries for the following pair-wise comparisons:

- Red line: $1$ vs $2,3$.
- Green line: $3$ vs $1,2$.
- Blue line: $2$ vs $1,3$.

These lines are used to separate the points into their respective categories based on their coordinates in the feature space.
Multiclass classification via learning probabilities

(for concreteness, suppose $\mathcal{Y} = \{1, \ldots, k\}$) idea 2: learning probabilities

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

Q: how to predict probabilities?
Multiclass classification via learning probabilities

- let $W \in \mathbb{R}^{k \times d}$, so $Wx \in \mathbb{R}^k$

- **multinomial logit** takes a hint from logistic: let $z = h(x; W) = Wx$, and suppose

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

(ensures probabilities are positive and sum to 1)

- fit by minimizing negative log likelihood

$$\ell(y, z) = -\log(P(y \mid z))$$

$$= -\log\left(\frac{\exp(z_y)}{\sum_{j=1}^k \exp(z_j)}\right)$$
Multinomial classification

The diagram illustrates a two-dimensional space with axes labeled $x_1$ and $x_2$. Points are classified into three categories: 1, 2, and 3, represented by different markers. Lines indicate the decision boundaries for the following comparisons:

- 1 vs 2,3
- 3 vs 1,2
- 2 vs 1,3
Outline

Maximum likelihood estimation

Regression

Classification

The prediction space

Multiclass classification

Ordinal regression

Beyond linear models
Ordinal regression

how to predict ordinal values?
Ordinal regression

how to predict **ordinal** values?

- **idea 0: regression**
  1. encode $y \in Y$ in $\mathbb{R}$
how to predict **ordinal** values?

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

- **idea 1: classification**
  1. encode $y \in Y$ 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 \( \mathbb{R} \)

▶ 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 \( y = \text{argmax}_{y' \in \mathcal{Y}} \mathbb{P}(y = y' \mid x) \)
   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 $\mathbb{R}$
2. predict with $Z = \mathbb{R}$

- 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**

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 $\psi(y)$

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

▶ how to encode $y$ as a vector?
Ordinal regression via predicting a vector

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 \( \psi(y) \)

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

- how to encode \( y \) as a vector? how about

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

- let \( W \in \mathbb{R}^{k-1 \times d} \), so \( z = Wx \in \mathbb{R}^{k-1} \)
- pick your favorite loss function \( \ell_{\text{bin}} \) 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 $\psi(y) = (1, \ldots, 1, \underbrace{-1}, \ldots, -1) \in \{-1, 1\}^{k-1}$
- let $W \in \mathbb{R}^{k-1 \times d}$, so $z = Wx \in \mathbb{R}^{k-1}$
- fit parameter $W$ by minimizing loss function

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

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

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

- $i$th 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

$$\hat{y} = \arg\min_{y \in \mathcal{Y}} \ell^{\text{ord}}(y; z)$$
Ordinal regression
Regularization for ordinal regression

- need to ensure that
  \[ P(y > 1 \mid z) \geq P(y > 2 \mid z) \geq \ldots \geq P(y > k - 1 \mid z) \]

- since \( P(y > i \mid z) \sim \exp((i)z_i) \), need to ensure that
  \[ z_1 \geq z_2 \geq \ldots \geq z_{k-1} \]

- can do this by insisting that
  \[
  W = \begin{bmatrix}
  w^T & b_1 \\
  w^T & b_2 \\
  \vdots & \vdots \\
  w^T & b_{k-1}
  \end{bmatrix}
  \]
  and \( b_1 \geq b_2 \geq \ldots \geq b_{k-1} \)

- then \( z = Wx \) satisfies \( z_1 \geq z_2 \geq \ldots \geq z_{k-1} \)

- this is a kind of **regularization** on \( W \)!
Outline

Maximum likelihood estimation

Regression

Classification

The prediction space

Multiclass classification

Ordinal regression

Beyond linear models
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 $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbb{R}$,
- regularizer $r : \mathcal{W} \to \mathbb{R}$, 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

$$y = \arg\min_{y \in \mathcal{Y}} \ell(y, h(x_i; w))$$
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](https://www.cscu.cornell.edu/news/statnews/stnews70.pdf)