6.036/6.862: Introduction to Machine Learning

Lecture: starts Tuesdays 9:35am (Boston time zone)

Course website: introml.odl.mit.edu

Who's talking? Prof. Tamara Broderick

Questions? discourse.odl.mit.edu ("Lecture 3" category)

Materials: Will all be available at course website

Last Time(s)

I. Linear classifiers
II. Perceptron algorithm
III. Linear separability
IV. Perceptron theorem

Today's Plan

I. A more-complete ML analysis
II. Choosing good features
III. Evaluation

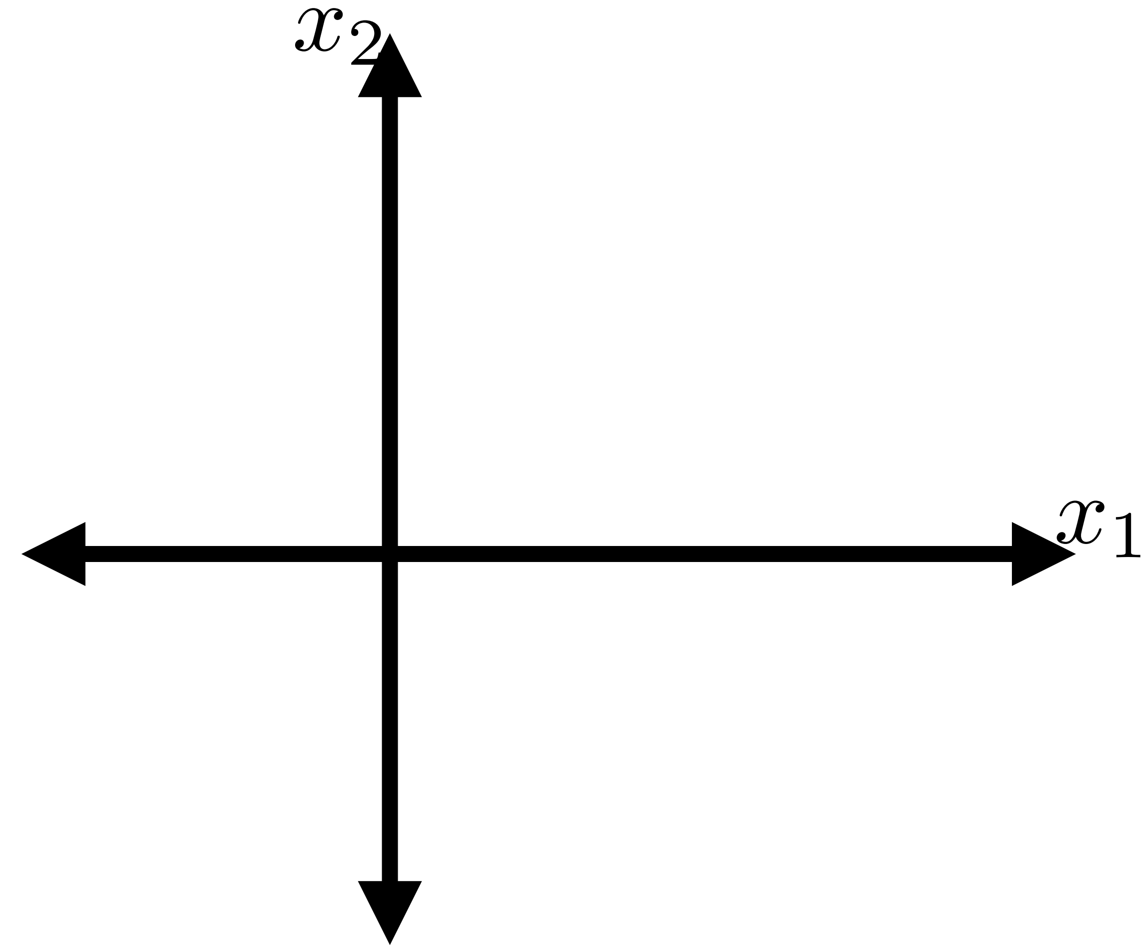
Recall

Recall

- Linear classifier h

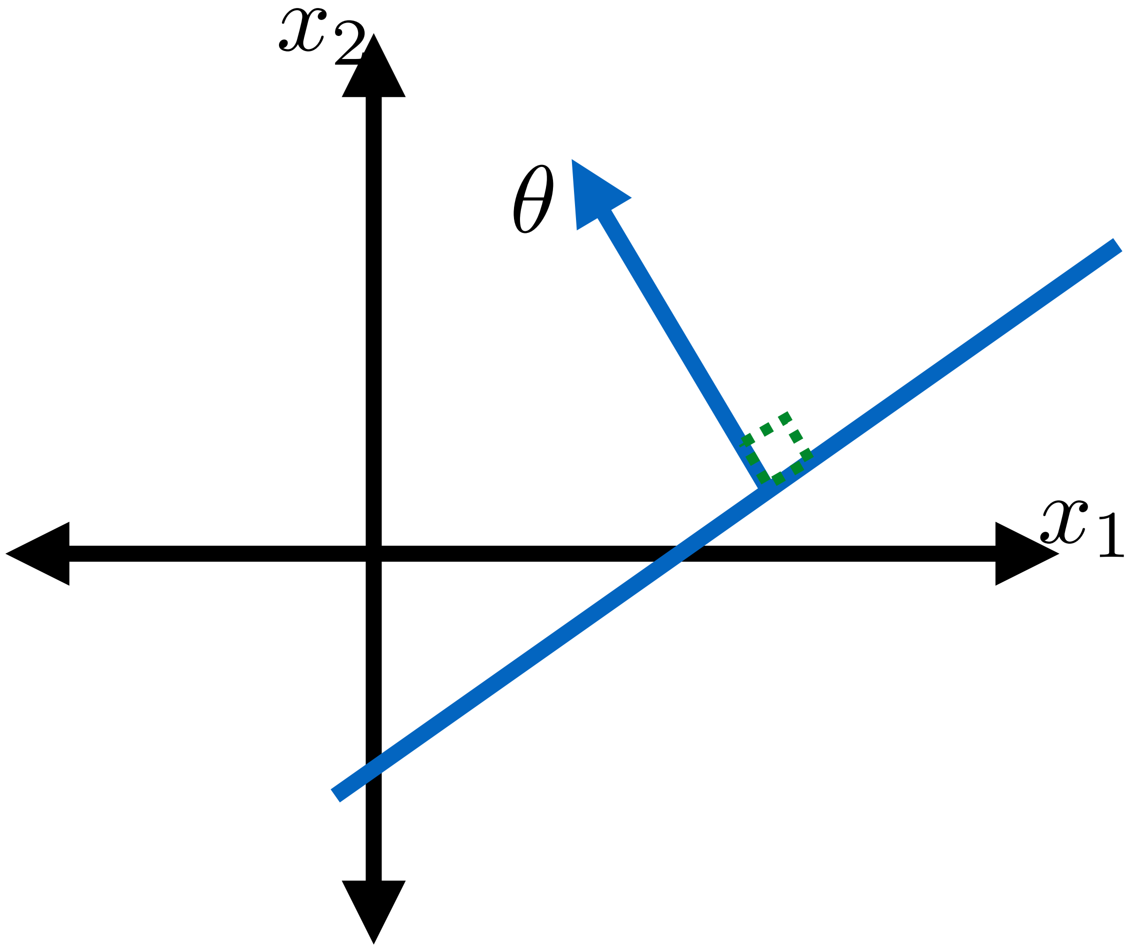
Recall

- Linear classifier h



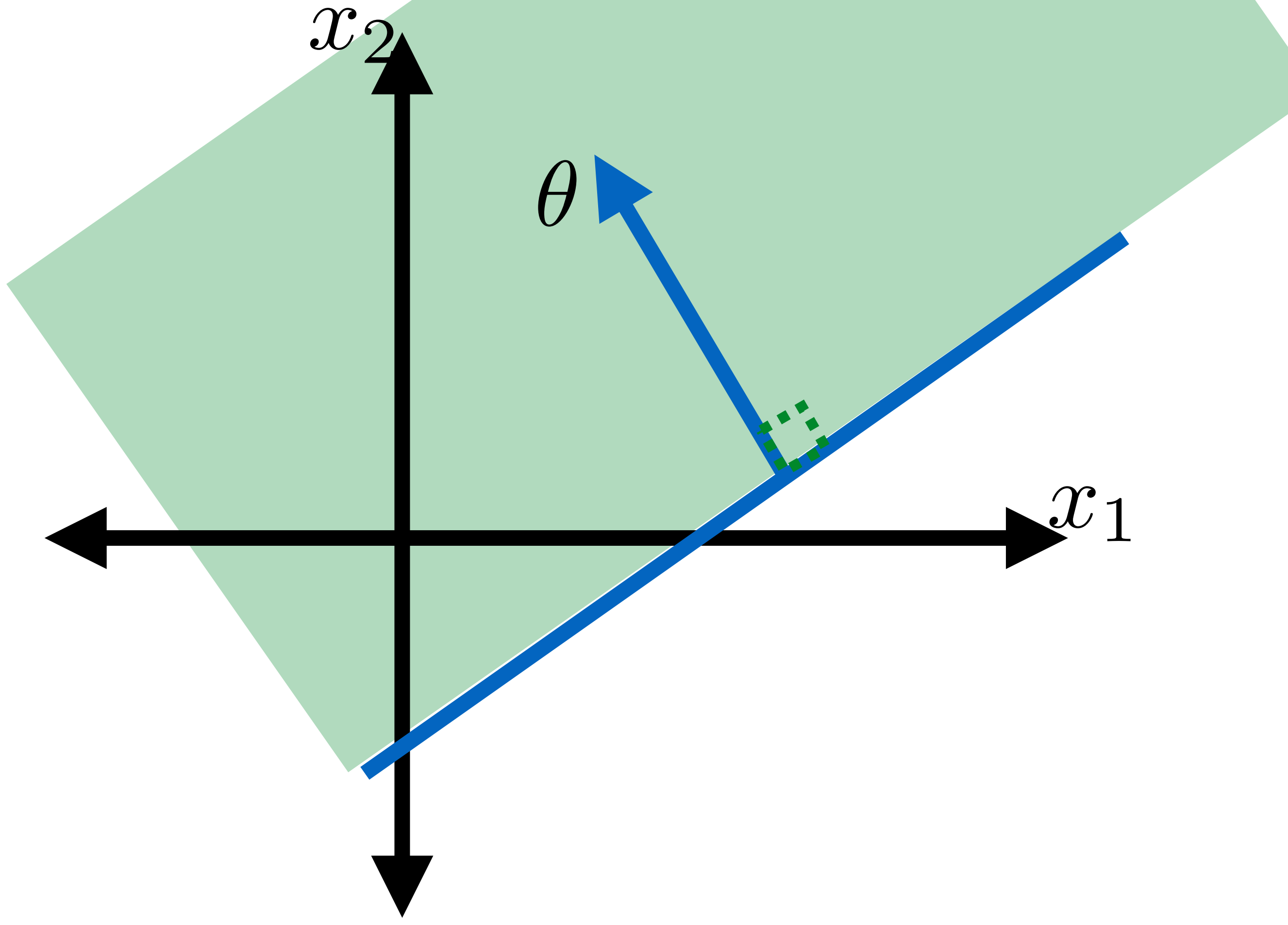
Recall

- Linear classifier h



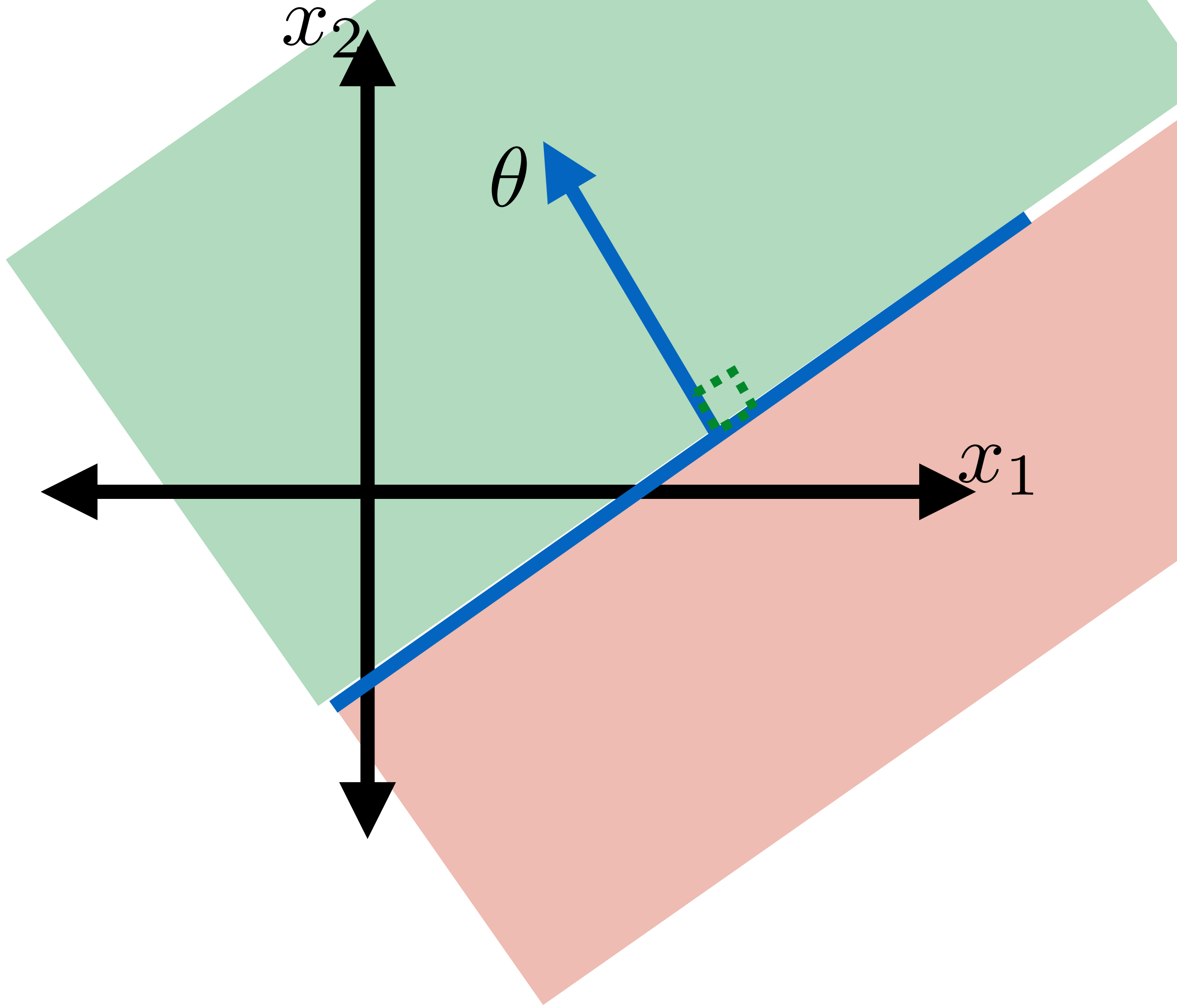
Recall

- Linear classifier h



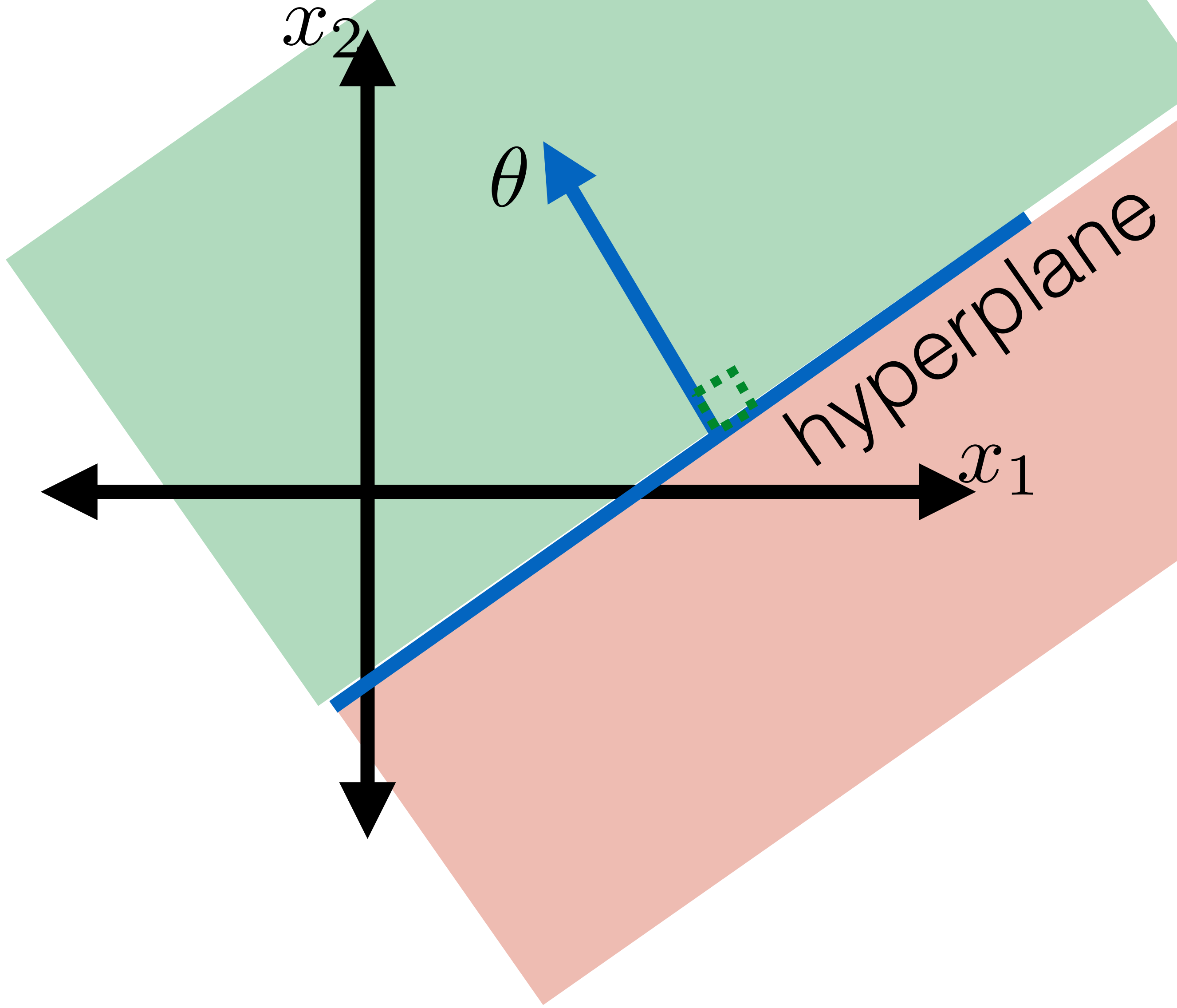
Recall

- Linear classifier h



Recall

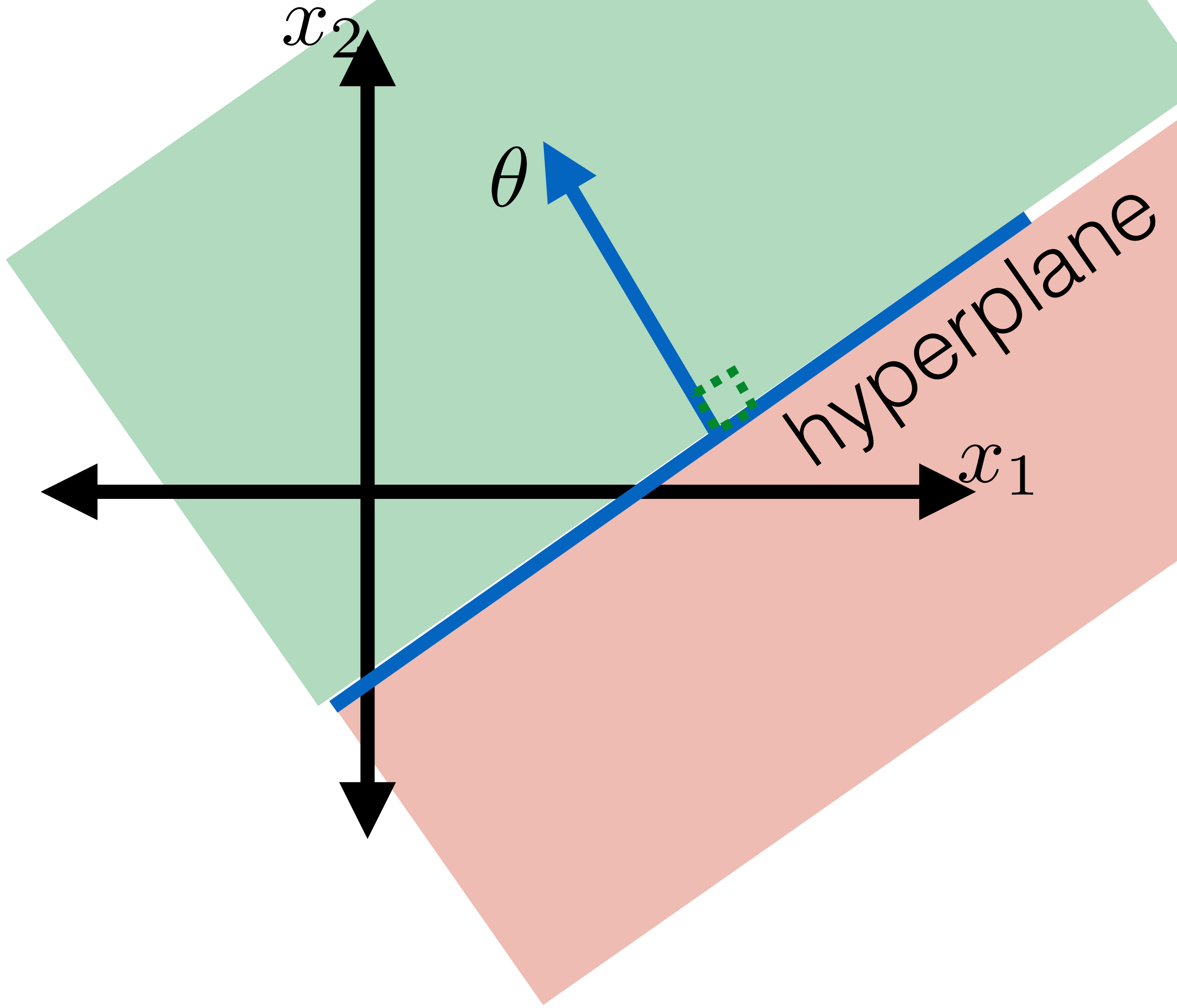
- Linear classifier h



Recall

- Linear classifier h
- 0-1 Loss

$$L(g, a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$



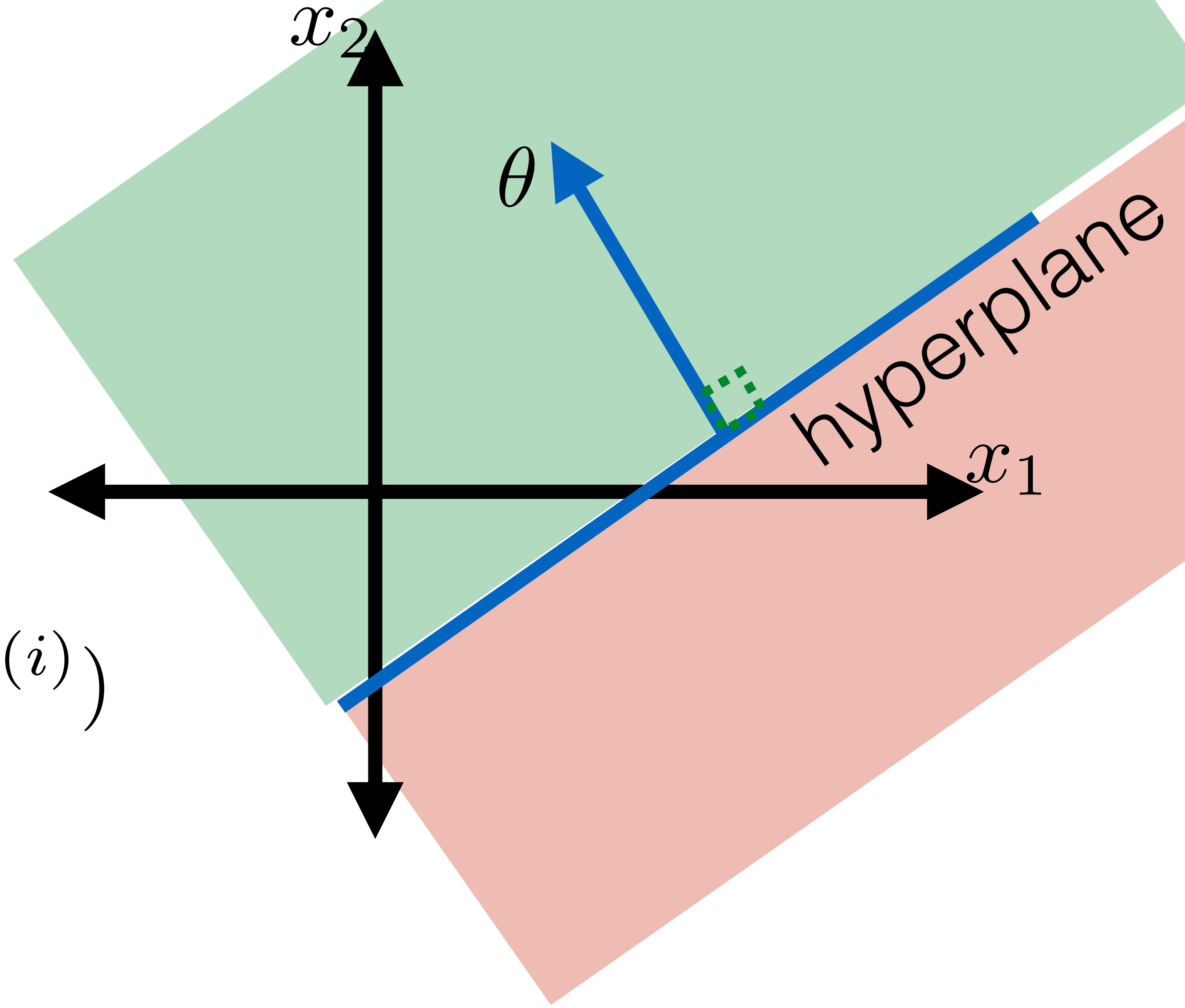
Recall

- Linear classifier h
- 0-1 Loss

$$L(g,a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$

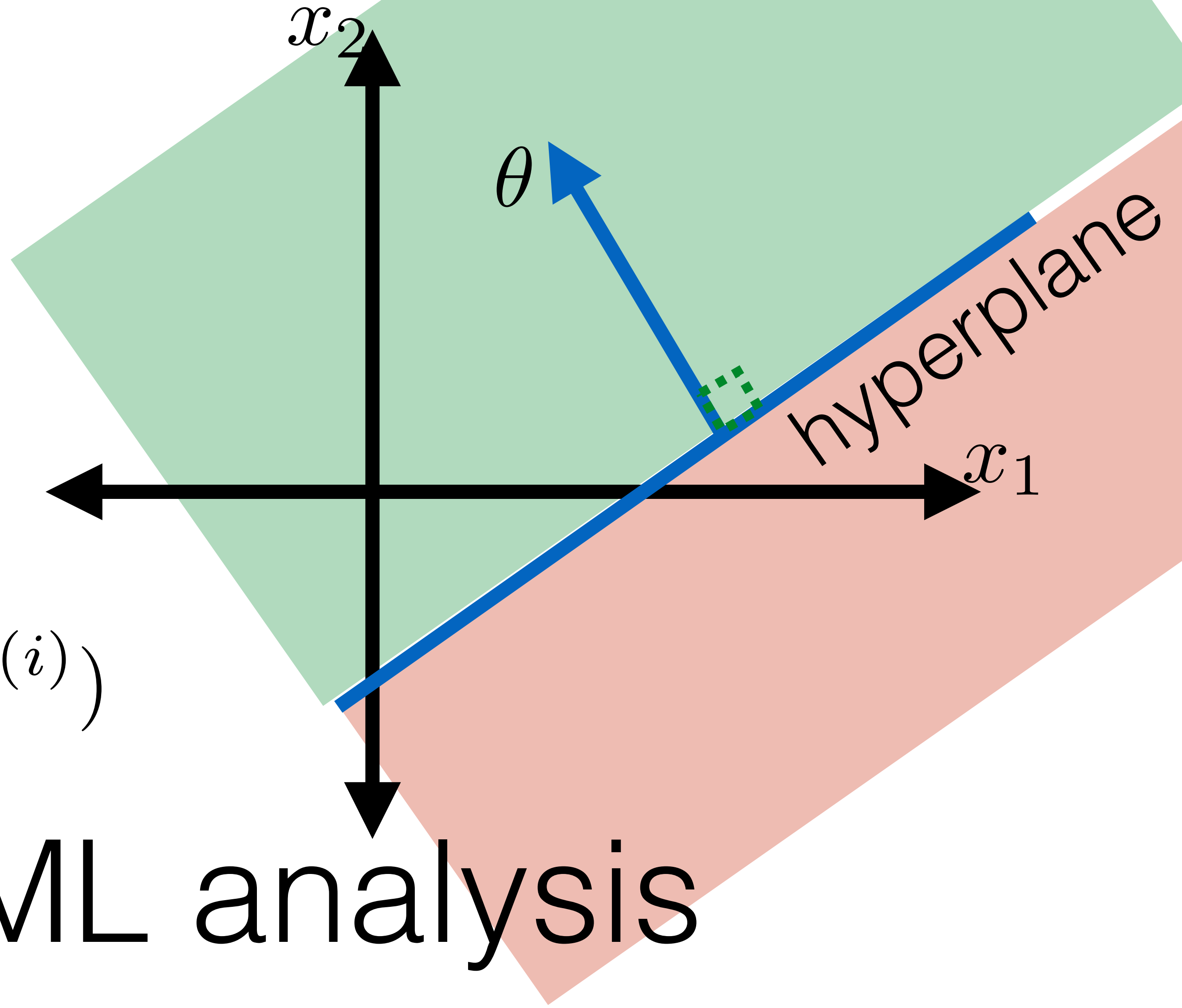
- Training error

$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$$



Recall

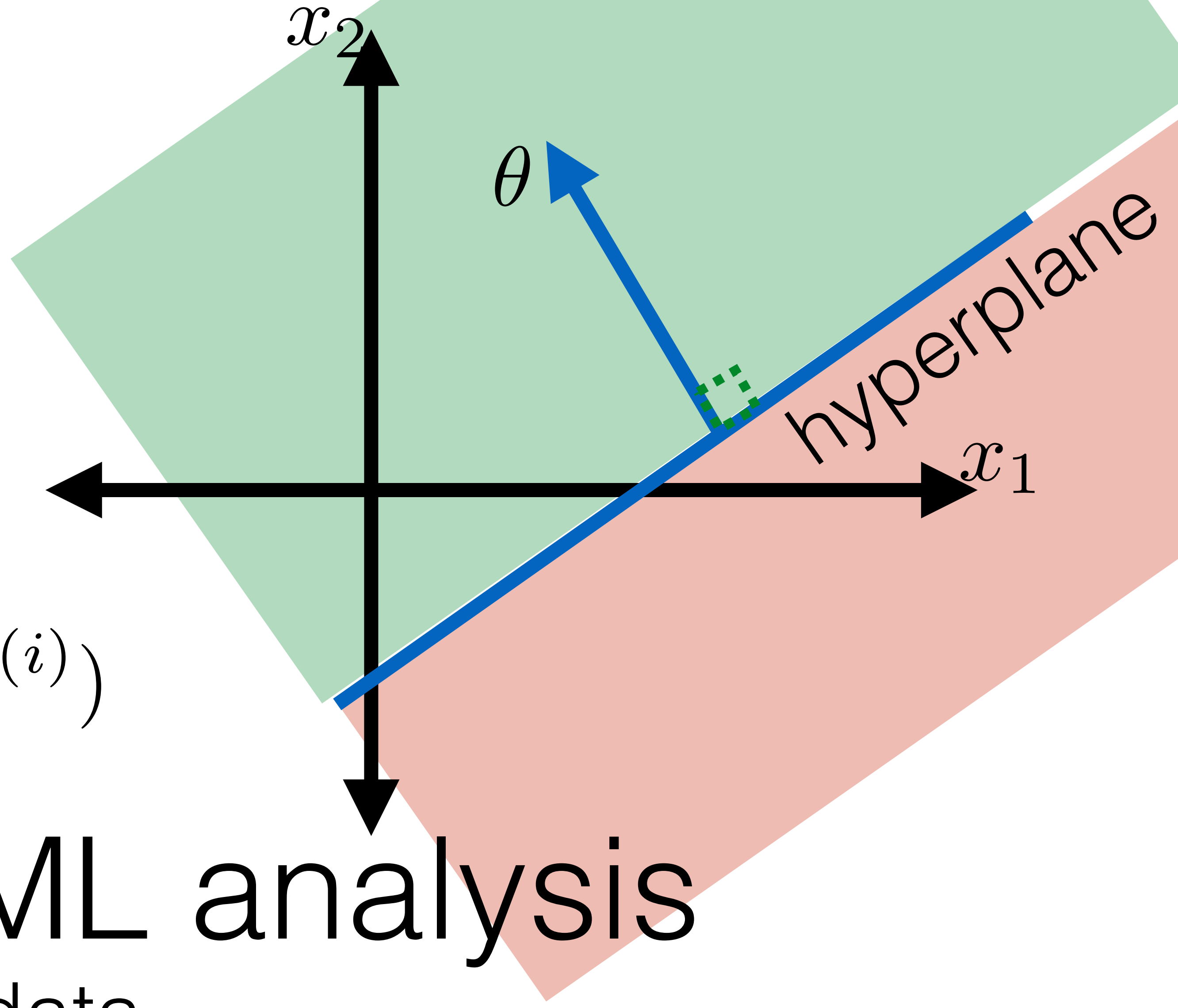
- Linear classifier h
- 0-1 Loss
$$L(g, a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$
- Training error
$$\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$$



A more-complete ML analysis

Recall

- Linear classifier h
- 0-1 Loss
$$L(g,a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$
- Training error
$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$$

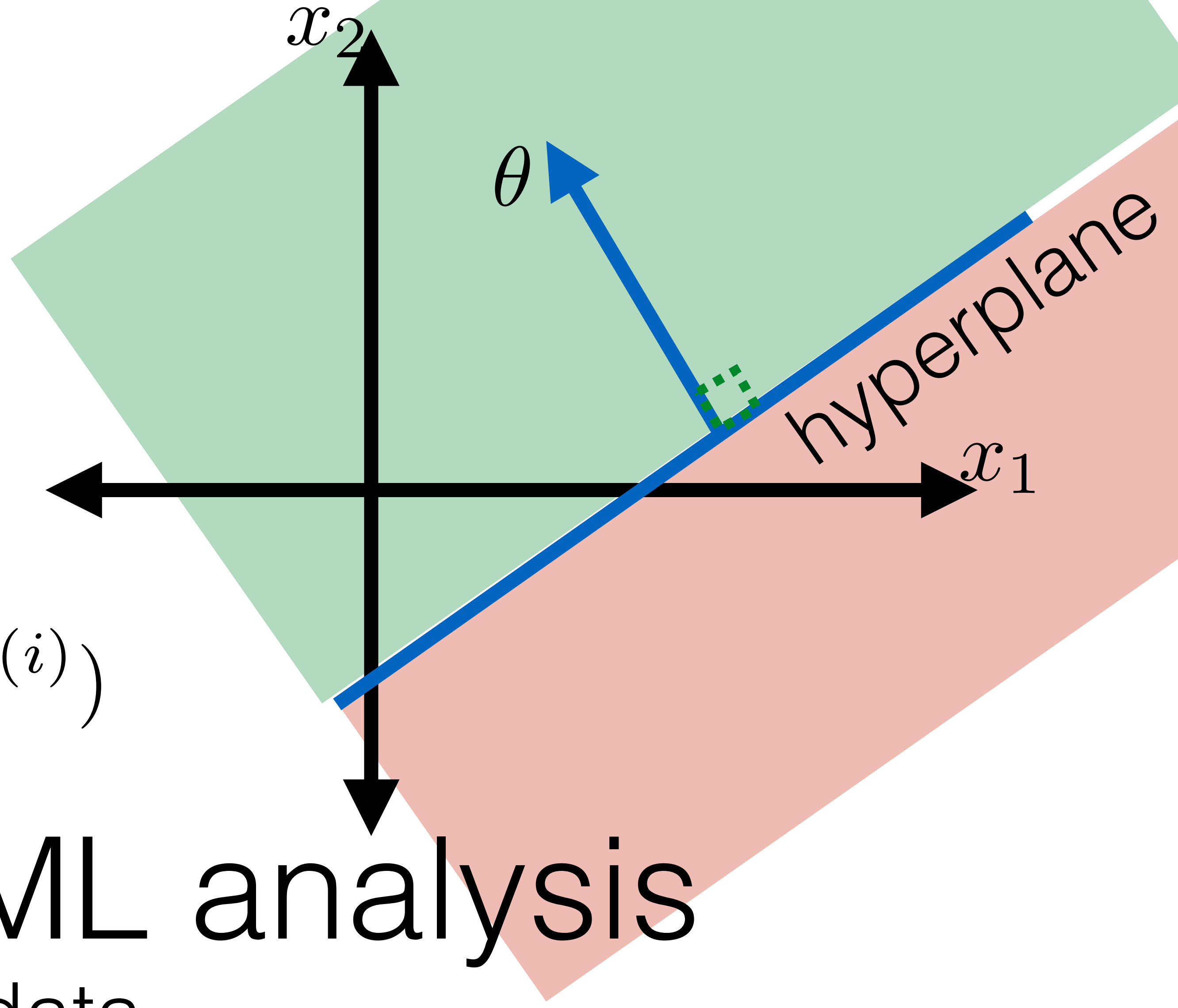


A more-complete ML analysis

1. Establish a goal & find data

Recall

- Linear classifier h
- 0-1 Loss
$$L(g, a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$
- Training error
$$\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$$

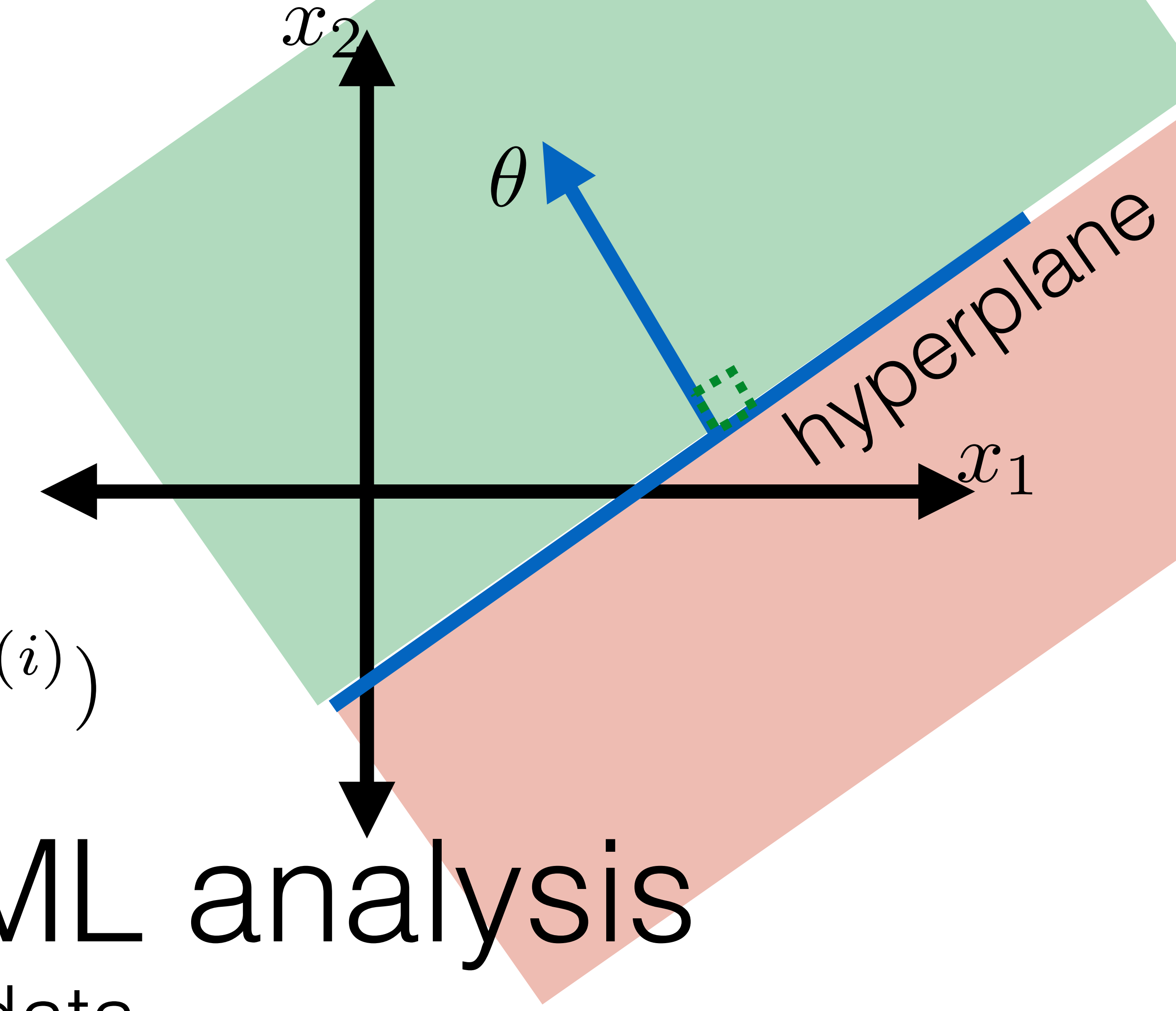


A more-complete ML analysis

1. Establish a goal & find data
    - Example goal: diagnose whether people have heart disease based on their available information

Recall

- Linear classifier h
- 0-1 Loss
$$L(g,a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$
- Training error
$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$$

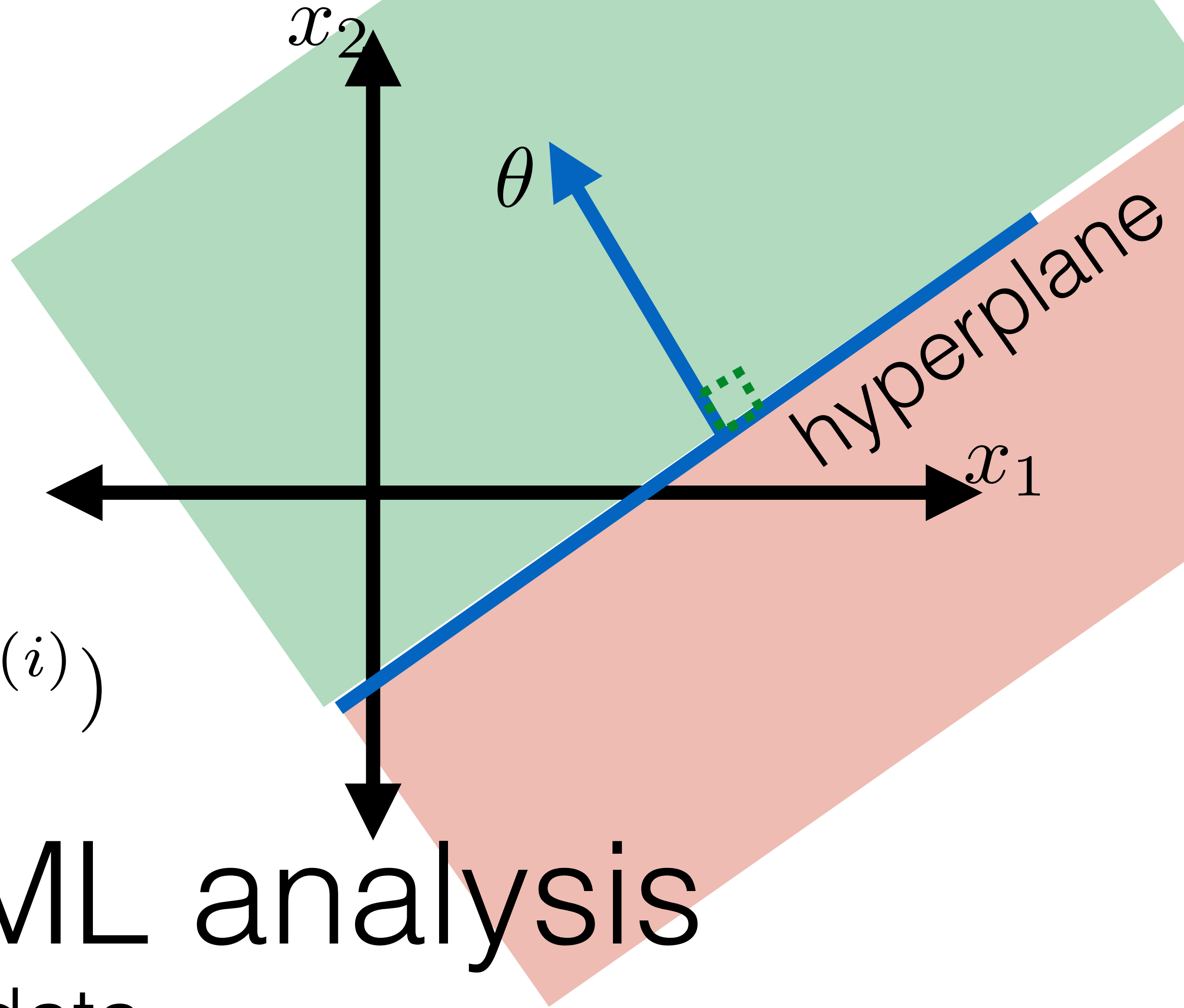


A more-complete ML analysis

1. Establish a goal & find data
    - Example goal: diagnose whether people have heart disease based on their available information
2. Encode data in useful form for the ML algorithm

Recall

- Linear classifier h
- 0-1 Loss
$$L(g,a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$
- Training error
$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$$

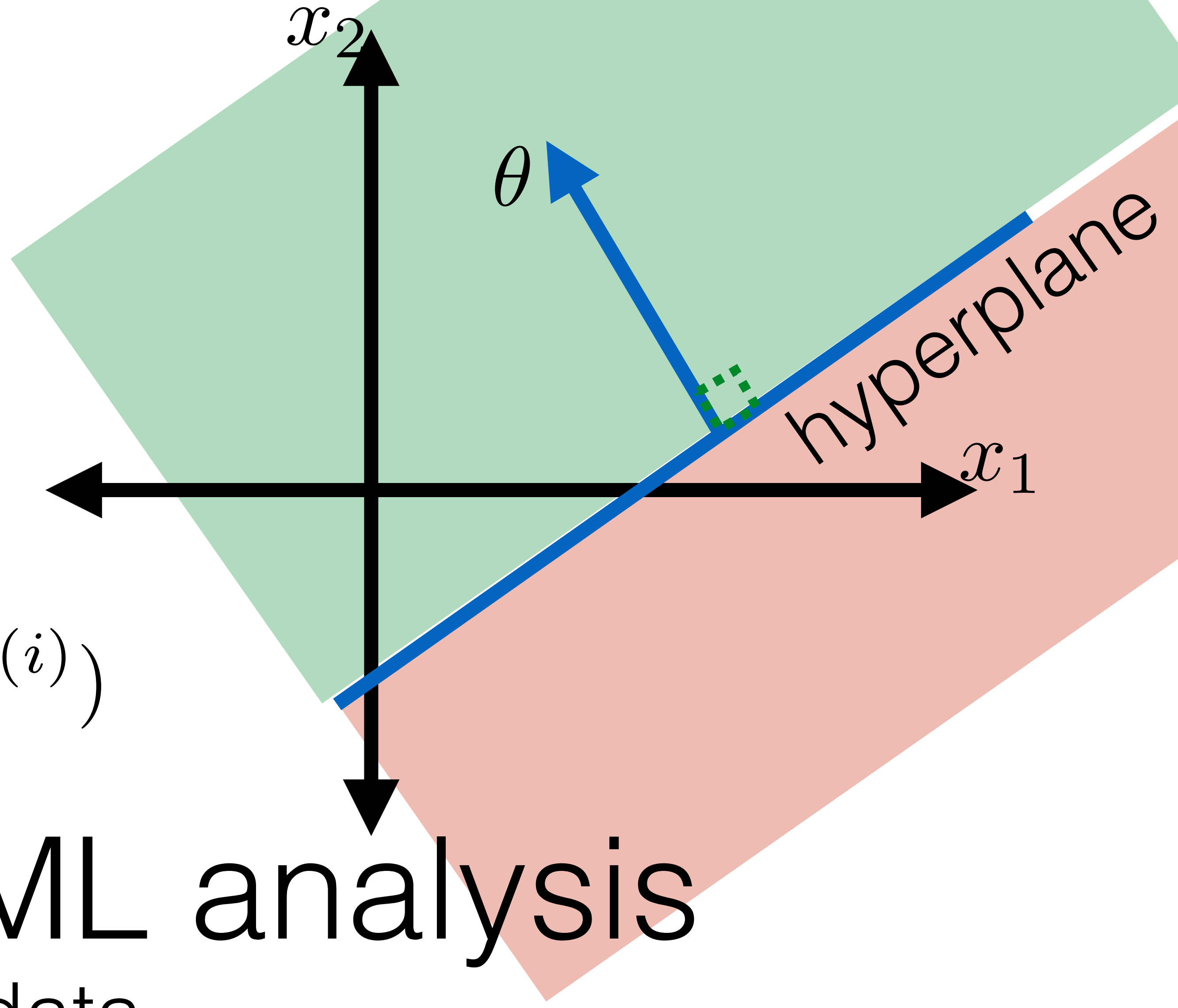


A more-complete ML analysis

1. Establish a goal & find data
   - Example goal: diagnose whether people have heart disease based on their available information
2. Encode data in useful form for the ML algorithm
3. Run the ML algorithm & return a classifier

Recall

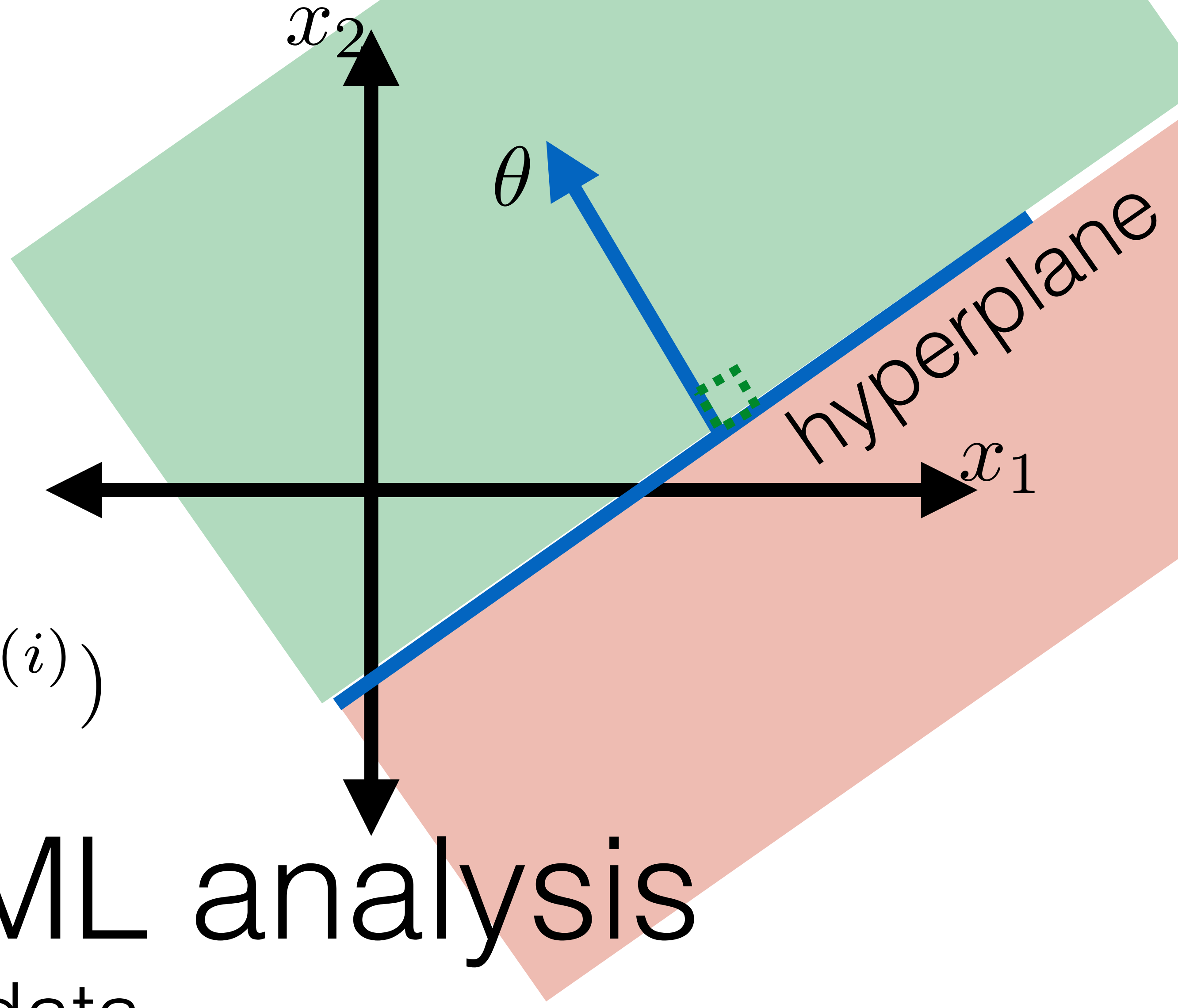
- Linear classifier h
- 0-1 Loss
$$L(g,a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$
- Training error
$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$$



A more-complete ML analysis

1. Establish a goal & find data
   - Example goal: diagnose whether people have heart disease based on their available information
2. Encode data in useful form for the ML algorithm
3. Run the ML algorithm & return a classifier
   - Example algorithms:

Recall



- Linear classifier h
- 0-1 Loss

$$L(g, a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$

- Training error

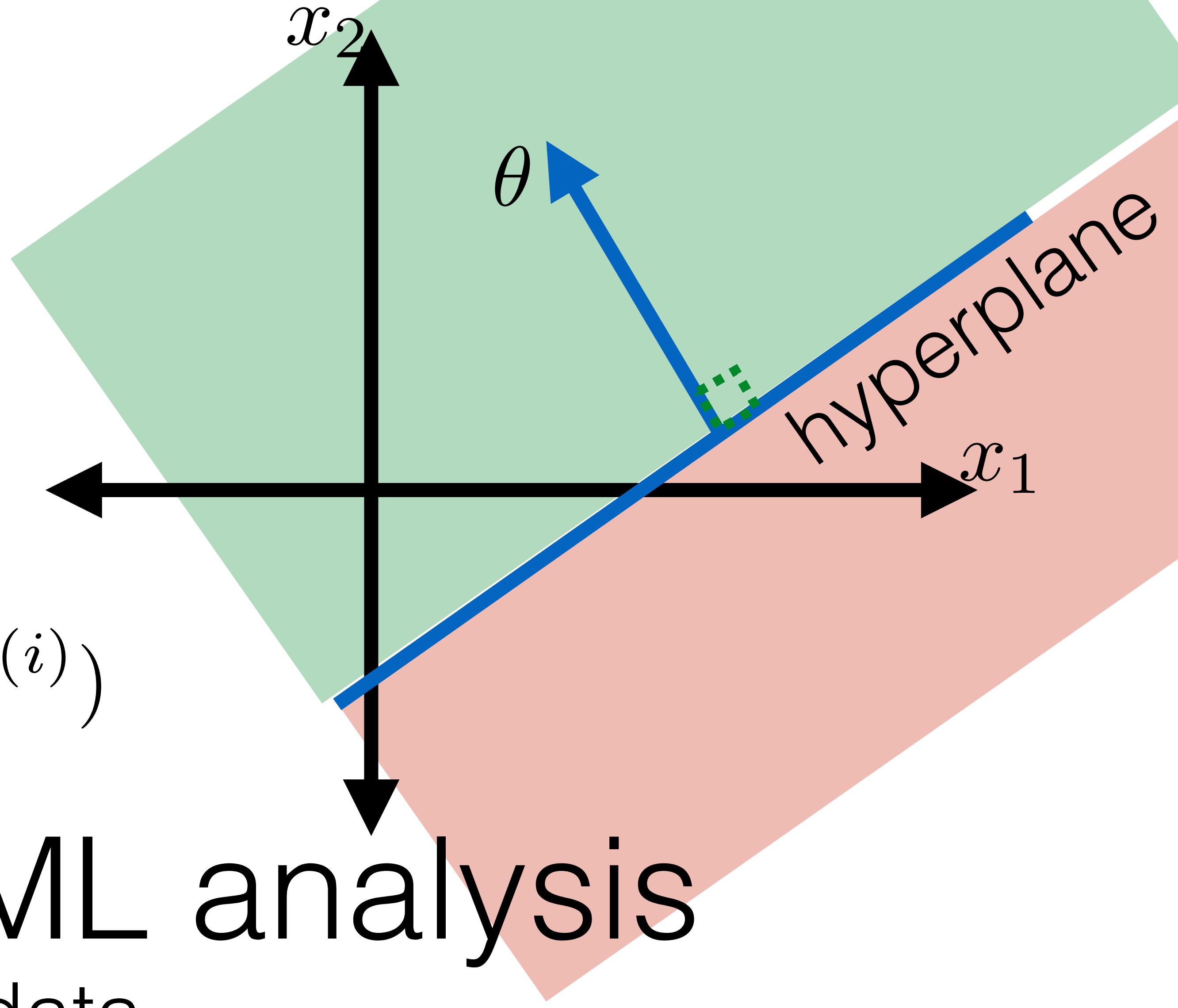
$$\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$$

A more-complete ML analysis

1. Establish a goal & find data
   - Example goal: diagnose whether people have heart disease based on their available information
2. Encode data in useful form for the ML algorithm
3. Run the ML algorithm & return a classifier
   - Example algorithms: (A) choose best classifier from a finite list; (B) perceptron; (C) averaged perceptron

Recall

- Linear classifier h
- 0-1 Loss
$$L(g, a) = \begin{cases} 0 \text{ if } g = a \\ 1 \text{ else} \end{cases}$$
- Training error
$$\mathcal{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x^{(i)}), y^{(i)})$$



A more-complete ML analysis

1. Establish a goal & find data
   - Example goal: diagnose whether people have heart disease based on their available information
2. Encode data in useful form for the ML algorithm
3. Run the ML algorithm & return a classifier
   - Example algorithms: (A) choose best classifier from a finite list; (B) perceptron; (C) averaged perceptron
4. Interpretation & evaluation

A machine learning (ML) analysis

- First, need goal & data.

A machine learning (ML) analysis

- First, need goal & data. E.g. diagnose whether people have heart disease based on their available information

A machine learning (ML) analysis

- First, need goal & data. E.g. diagnose whether people have heart disease based on their available information

	has heart disease?	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	no	55	no	nurse	pain	40s	133000
2	no	71	no	admin	beta blockers, pain	20s	34000
3	yes	89	yes	nurse	beta blockers	50s	40000
4	no	67	no	doctor	none	50s	120000

A machine learning (ML) analysis

- First, need goal & data. E.g. diagnose whether people have heart disease based on their available information
- Next, put data in useful form for learning algorithm

	has heart disease?	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	no	55	no	nurse	pain	40s	133000
2	no	71	no	admin	beta blockers, pain	20s	34000
3	yes	89	yes	nurse	beta blockers	50s	40000
4	no	67	no	doctor	none	50s	120000

A machine learning (ML) analysis

- First, need goal & data. E.g. diagnose whether people have heart disease based on their available information
- Next, put data in useful form for learning algorithm

	has heart disease?	**resting heart rate (bpm)**	**pain?**	**job**	**medicines**	**age**	**family income (USD)**
1	no	55	no	nurse	pain	40s	133000
2	no	71	no	admin	beta blockers, pain	20s	34000
3	yes	89	yes	nurse	beta blockers	50s	40000
4	no	67	no	doctor	none	50s	120000

A machine learning (ML) analysis

- First, need goal & data. E.g. diagnose whether people have heart disease based on their available information
- Next, put data in useful form for learning algorithm

	has heart disease?	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	no	55	no	nurse	pain	40s	133000
2	no	71	no	admin	beta blockers, pain	20s	34000
3	yes	89	yes	nurse	beta blockers	50s	40000
4	no	67	no	doctor	none	50s	120000

A machine learning (ML) analysis

- First, need goal & data. E.g. diagnose whether people have heart disease based on their available information
- Next, put data in useful form for learning algorithm

	has heart disease?	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	no	55	no	nurse	pain	40s	133000
2	no	71	no	admin	beta blockers, pain	20s	34000
3	yes	89	yes	nurse	beta blockers	50s	40000
4	no	67	no	doctor	none	50s	120000

	has heart disease?
1	no
2	no
3	yes
4	no

Encode data in usable form

	has heart disease?
1	no
2	no
3	yes
4	no

Encode data in usable form

- Identify the labels and encode as real numbers

	has heart disease?
1	no
2	no
3	yes
4	no

Encode data in usable form

- Identify the labels and encode as real numbers

	has heart disease?
1	no
2	no
3	yes
4	no

$$\{\text{'yes'}, \text{'no'}\} \leftrightarrow \{+1, -1\}$$

Encode data in usable form

- Identify the labels and encode as real numbers

	has heart disease?
1	no
2	no
3	yes
4	no

$$\{\text{'yes'}, \text{'no'}\} \leftrightarrow \{+1, -1\}$$

Encode data in usable form

- Identify the labels and encode as real numbers

$$\{\text{'yes'}, \text{'no'}\} \leftrightarrow \{+1, -1\}$$

	has heart disease?
1	no
2	no
3	yes
4	no

1	-1
2	-1
3	+1
4	-1

Encode data in usable form

- Identify the labels and encode as real numbers

	has heart disease?
1	no
2	no
3	yes
4	no

$$\{\text{‘yes’}, \text{‘no’}\} \leftrightarrow \{+1, -1\}$$

1	-1 $= y^{(1)}$
2	-1
3	+1
4	-1

Encode data in usable form

- Identify the labels and encode as real numbers

	has heart disease?
1	no
2	no
3	yes
4	no

$$\{\text{‘yes’}, \text{‘no’}\} \leftrightarrow \{+1, -1\}$$

1	-1 $= y^{(1)}$
2	-1
3	+1
4	-1

- Depending on your algorithm, might instead use $\{0, 1\}$

Encode data in usable form

- Identify the labels and encode as real numbers

	has heart disease?
1	no
2	no
3	yes
4	no

$$\{\text{`yes'}, \text{`no'}\} \leftrightarrow \{+1, -1\}$$

1	-1
2	-1
3	+1
4	-1

$= y^{(1)}$

- Depending on your algorithm, might instead use $\{0, 1\}$
- Save mapping to recover predictions of new points

Encode data in usable form

Encode data in usable form

- Identify the features and encode as real numbers

Encode data in usable form

- Identify the features and encode as real numbers
- Feature: any function of the data (except labels)

Encode data in usable form

- Identify the features and encode as real numbers
- Feature: any function of the data (except labels)
- Today, old features: x; new features: $\phi(x)$

Encode data in usable form

- Identify the features and encode as real numbers
- Feature: any function of the data (except labels)
- Today, old features: x; new features: $\phi(x)$

	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	55	no	nurse	pain	40s	133000
2	71	no	admin	beta blockers, pain	20s	34000
3	89	yes	nurse	beta blockers	50s	40000
4	67	no	doctor	none	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers
- Feature: any function of the data (except labels)
- Today, old features: x; new features: $\phi(x)$

	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	55	no	nurse	pain	40s	133000
2	71	no	admin	beta blockers, pain	20s	34000
3	89	yes	nurse	beta blockers	50s	40000
4	67	no	doctor	none	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers
- Feature: any function of the data (except labels)
- Today, old features: x; new features: $\phi(x)$

	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
$(x^{(1)})^\top$ 1	55	no	nurse	pain	40s	133000
2	71	no	admin	beta blockers, pain	20s	34000
3	89	yes	nurse	beta blockers	50s	40000
4	67	no	doctor	none	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers
- Feature: any function of the data (except labels)
- Today, old features: x; new features: $\phi(x)$

	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	55	no	nurse	pain	40s	133000
2	71	no	admin	beta blockers, pain	20s	34000
3	89	yes	nurse	beta blockers	50s	40000
4	67	no	doctor	none	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers
- Feature: any function of the data (except labels)
- Today, old features: x; new features: $\phi(x)$

	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	55	no	nurse	pain	40s	133000
2	71	no	admin	beta blockers, pain	20s	34000
3	89	yes	nurse	beta blockers	50s	40000
4	67	no	doctor	none	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers
- Feature: any function of the data (except labels)
- Today, old features: x; new features: $\phi(x)$

	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	55	no	nurse	pain	40s	133000
2	71	no	admin	beta blockers, pain	20s	34000
3	89	yes	nurse	beta blockers	50s	40000
4	67	no	doctor	none	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers
- Feature: any function of the data (except labels)
- Today, old features: x; new features: $\phi(x)$

	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	55	0	nurse	pain	40s	133000
2	71	0	admin	beta blockers, pain	20s	34000
3	89	1	nurse	beta blockers	50s	40000
4	67	0	doctor	none	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers
- Feature: any function of the data (except labels)
- Today, old features: x; new features: $\phi(x)$

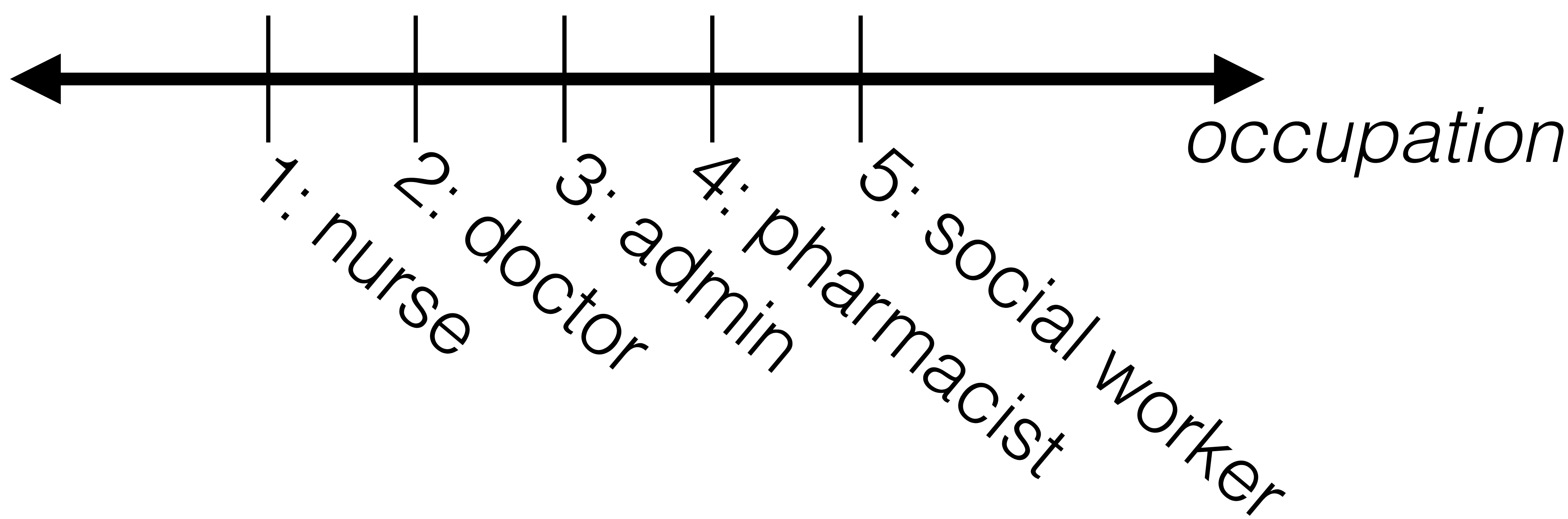
	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	55	0	nurse	pain	40s	133000
2	71	0	admin	beta blockers, pain	20s	34000
3	89	1	nurse	beta blockers	50s	40000
4	67	0	doctor	none	50s	120000

Encode categorical data

- Idea: turn each category into a unique natural number

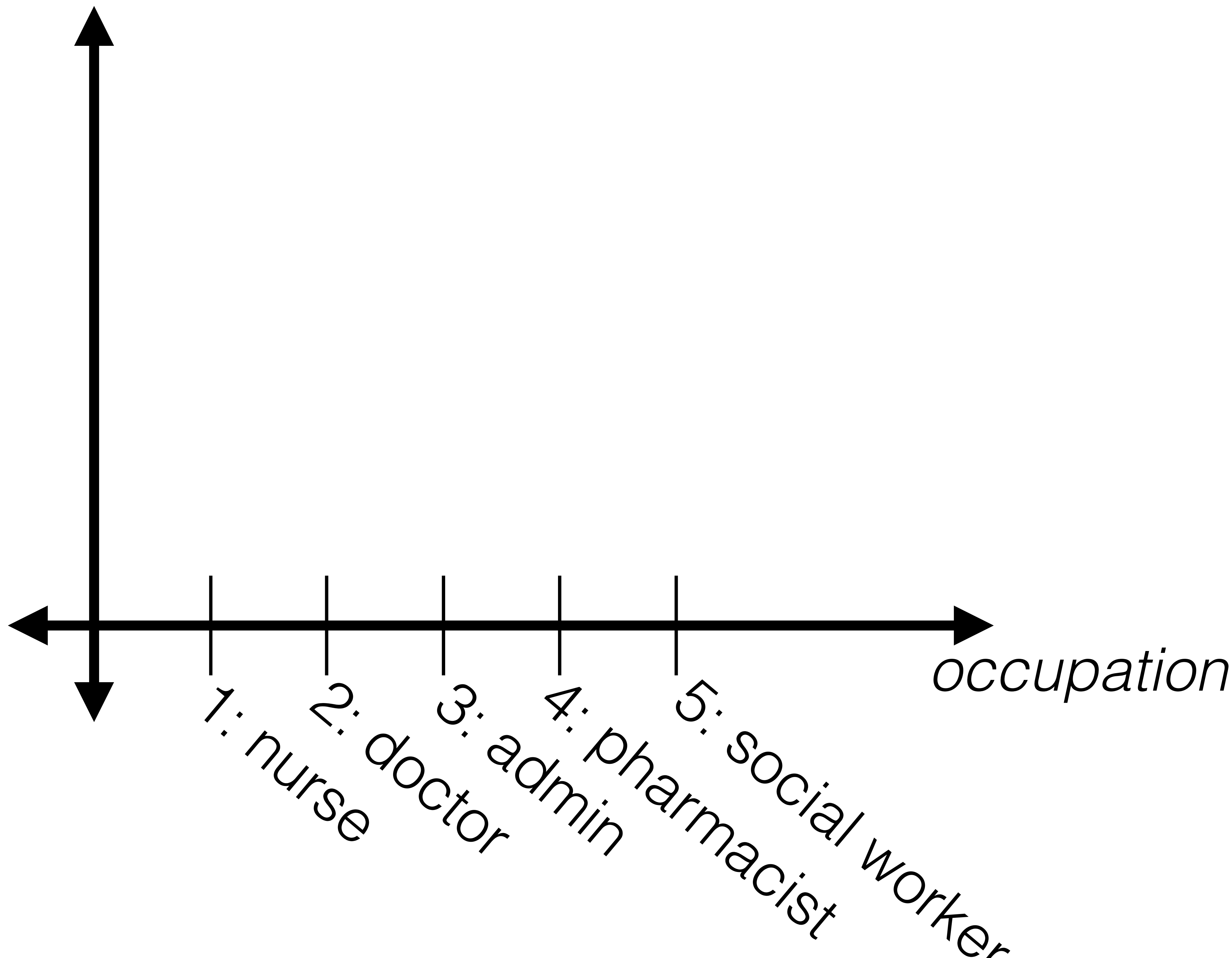
Encode categorical data

- Idea: turn each category into a unique natural number



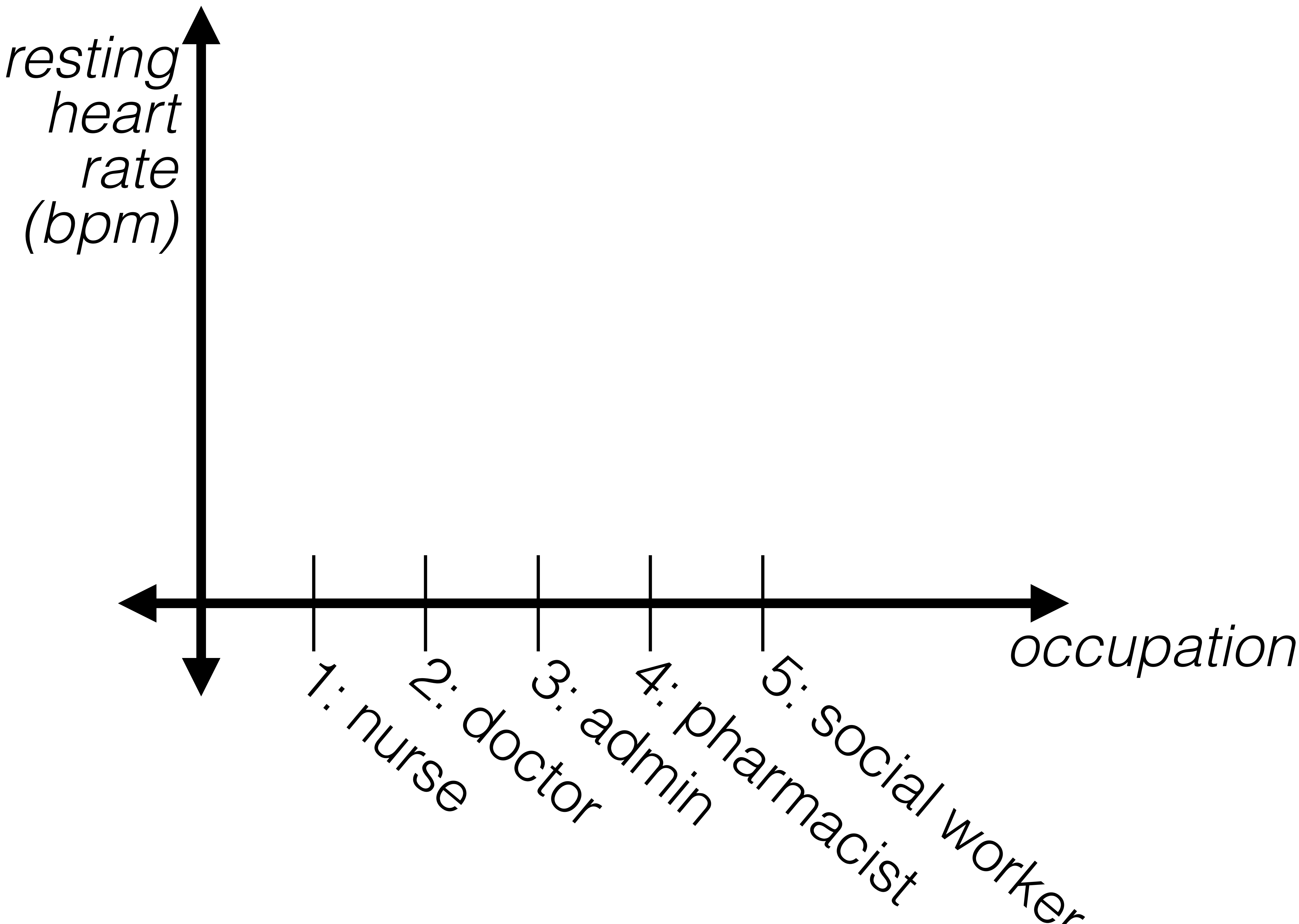
Encode categorical data

- Idea: turn each category into a unique natural number



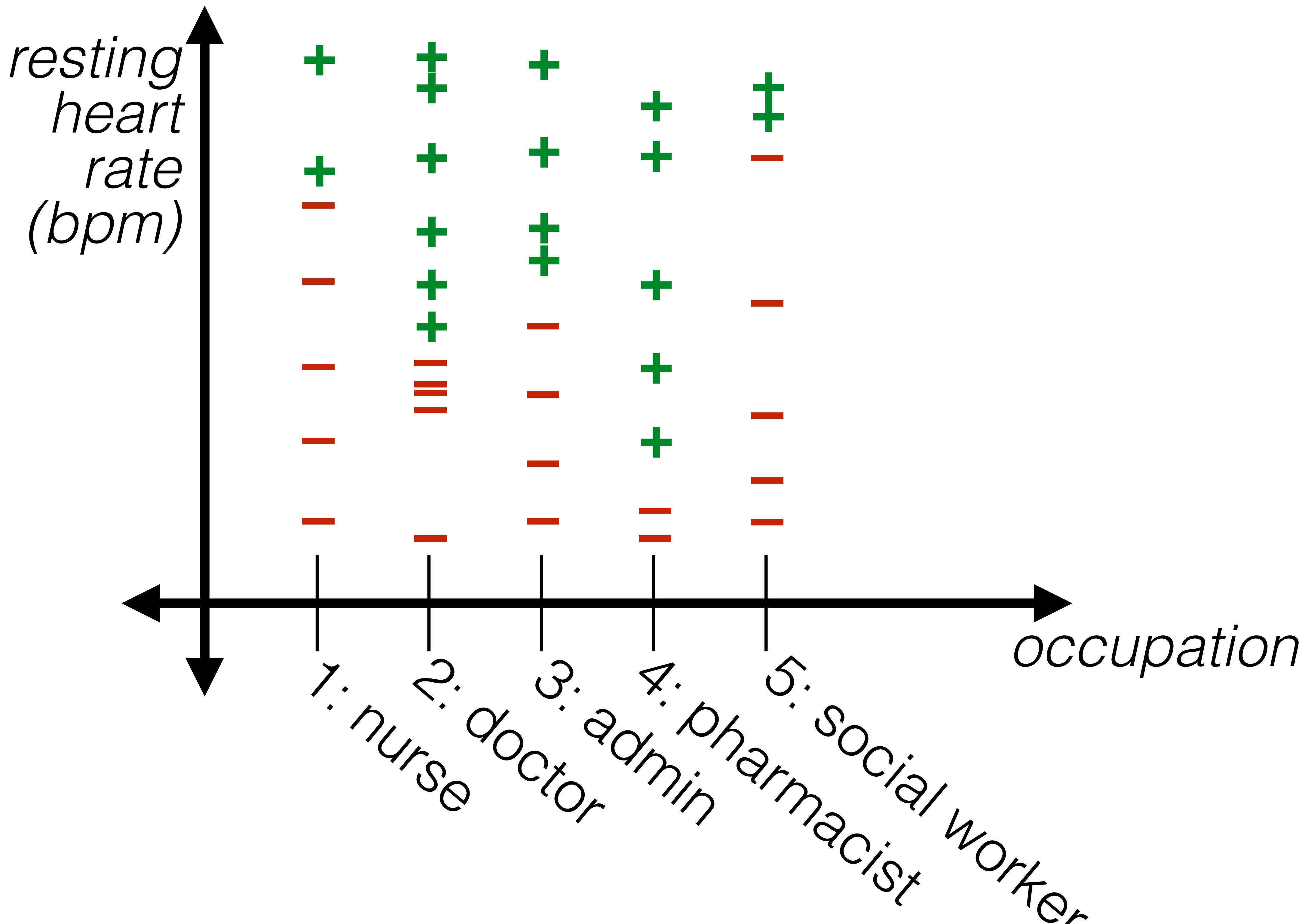
Encode categorical data

- Idea: turn each category into a unique natural number



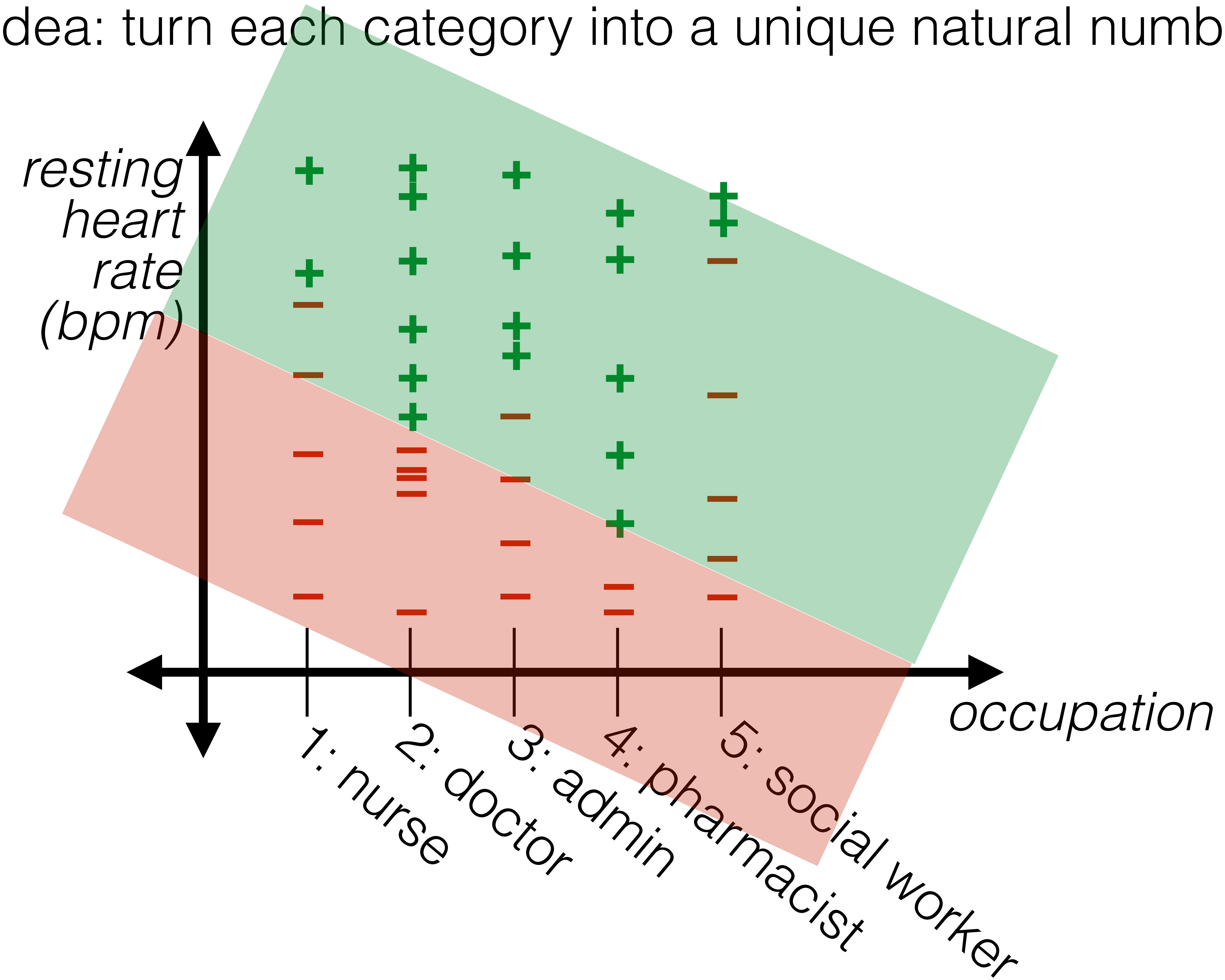
Encode categorical data

- Idea: turn each category into a unique natural number



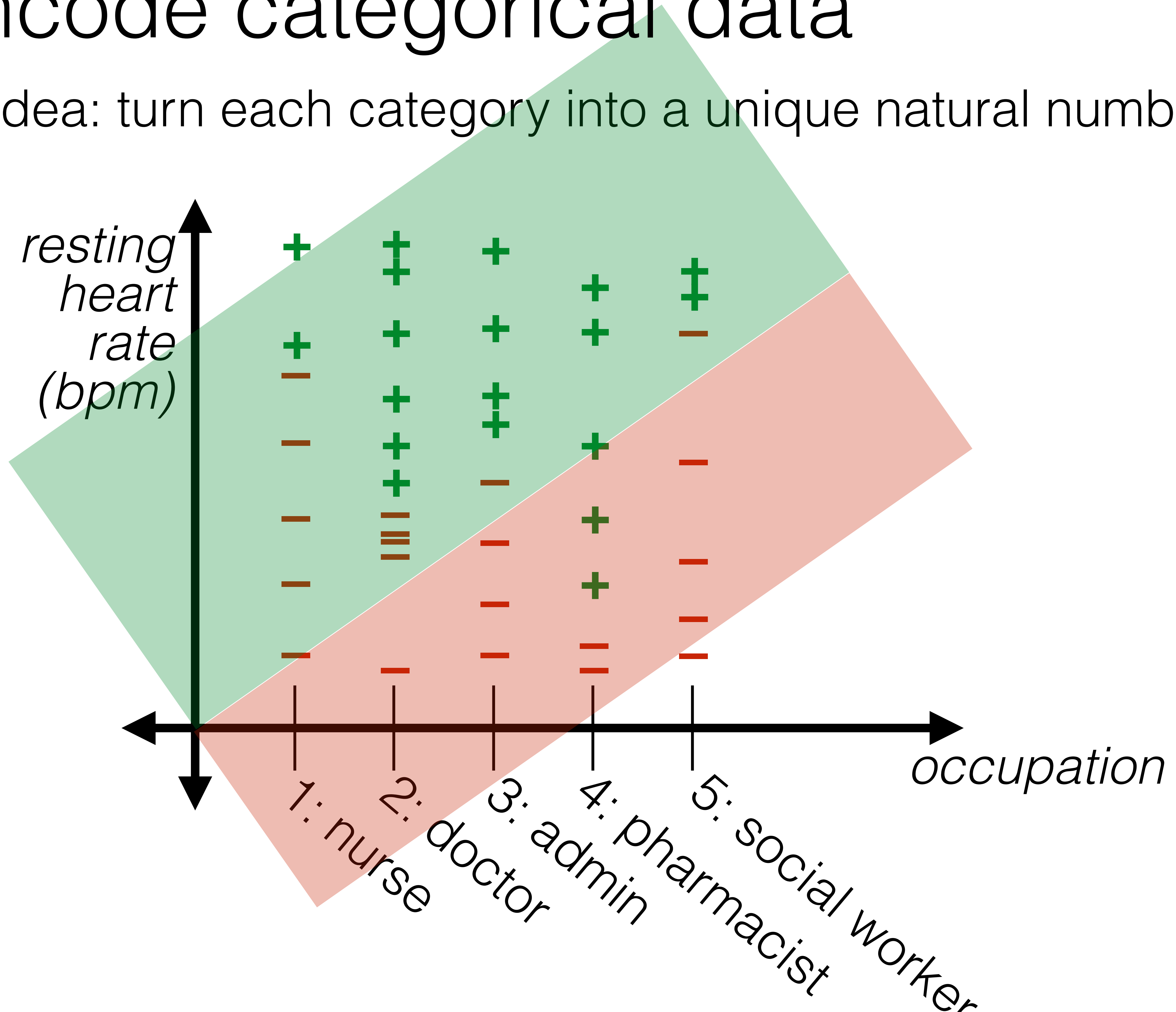
Encode categorical data

- Idea: turn each category into a unique natural number



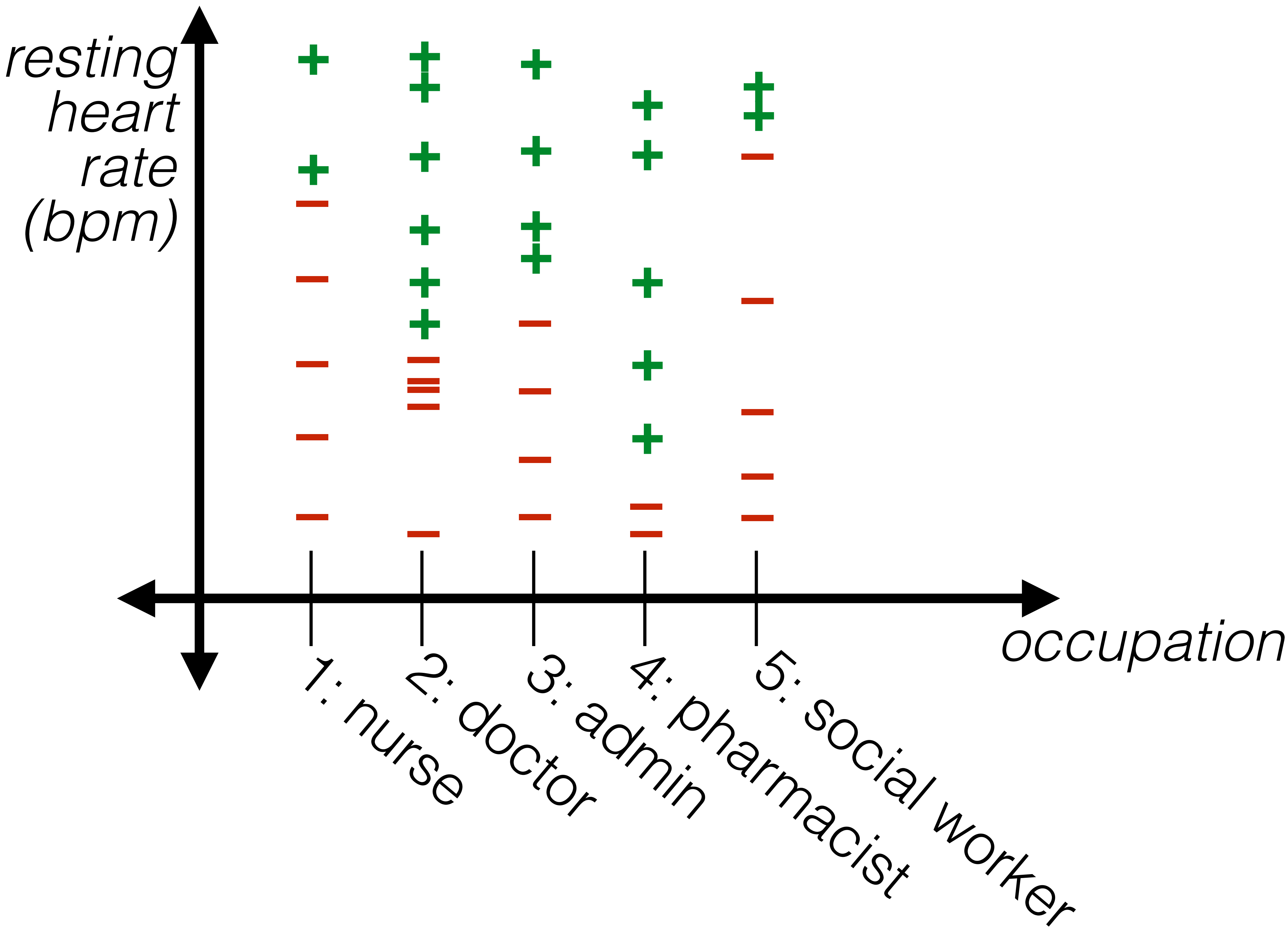
Encode categorical data

- Idea: turn each category into a unique natural number



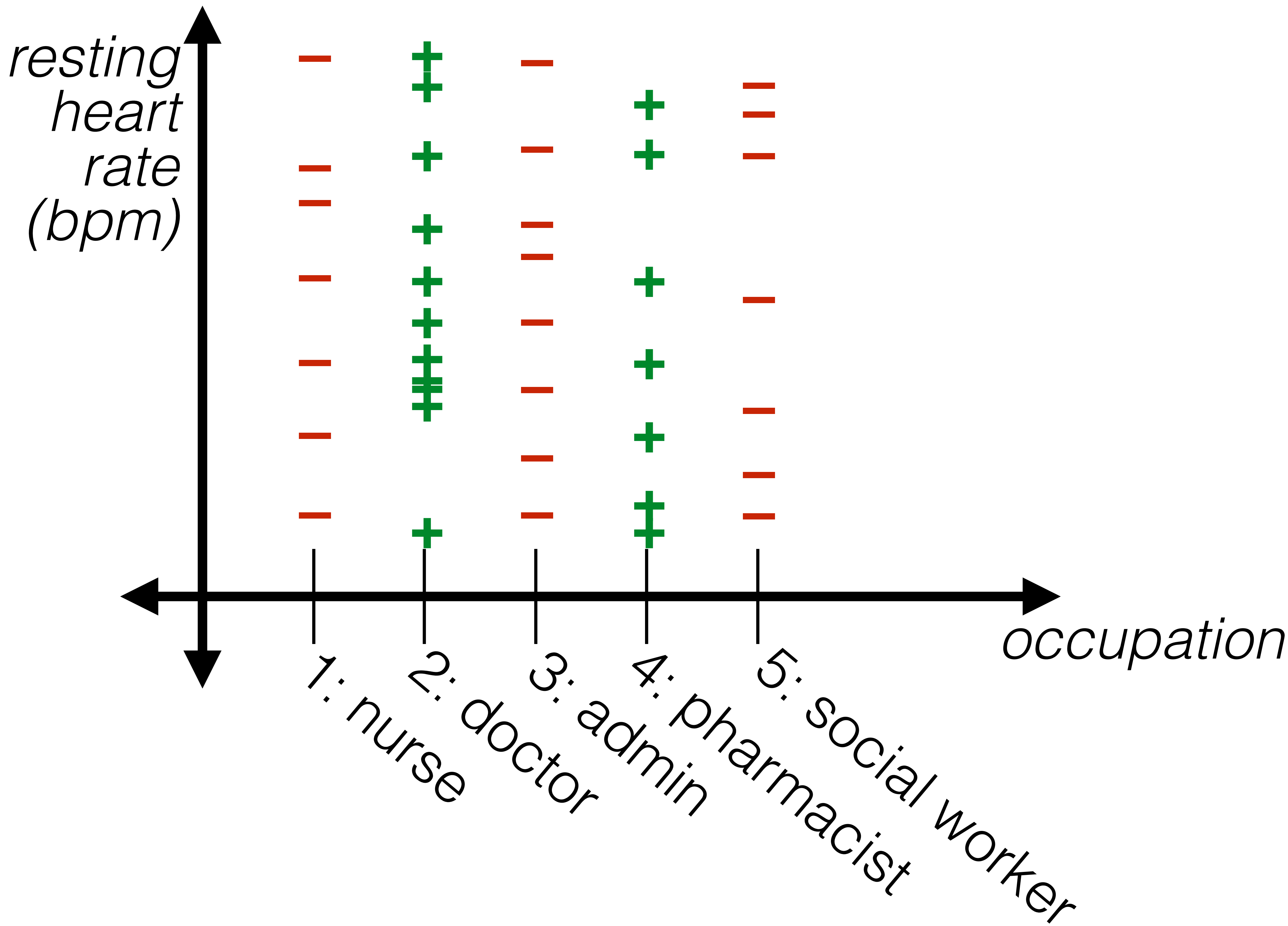
Encode categorical data

- Idea: turn each category into a unique natural number



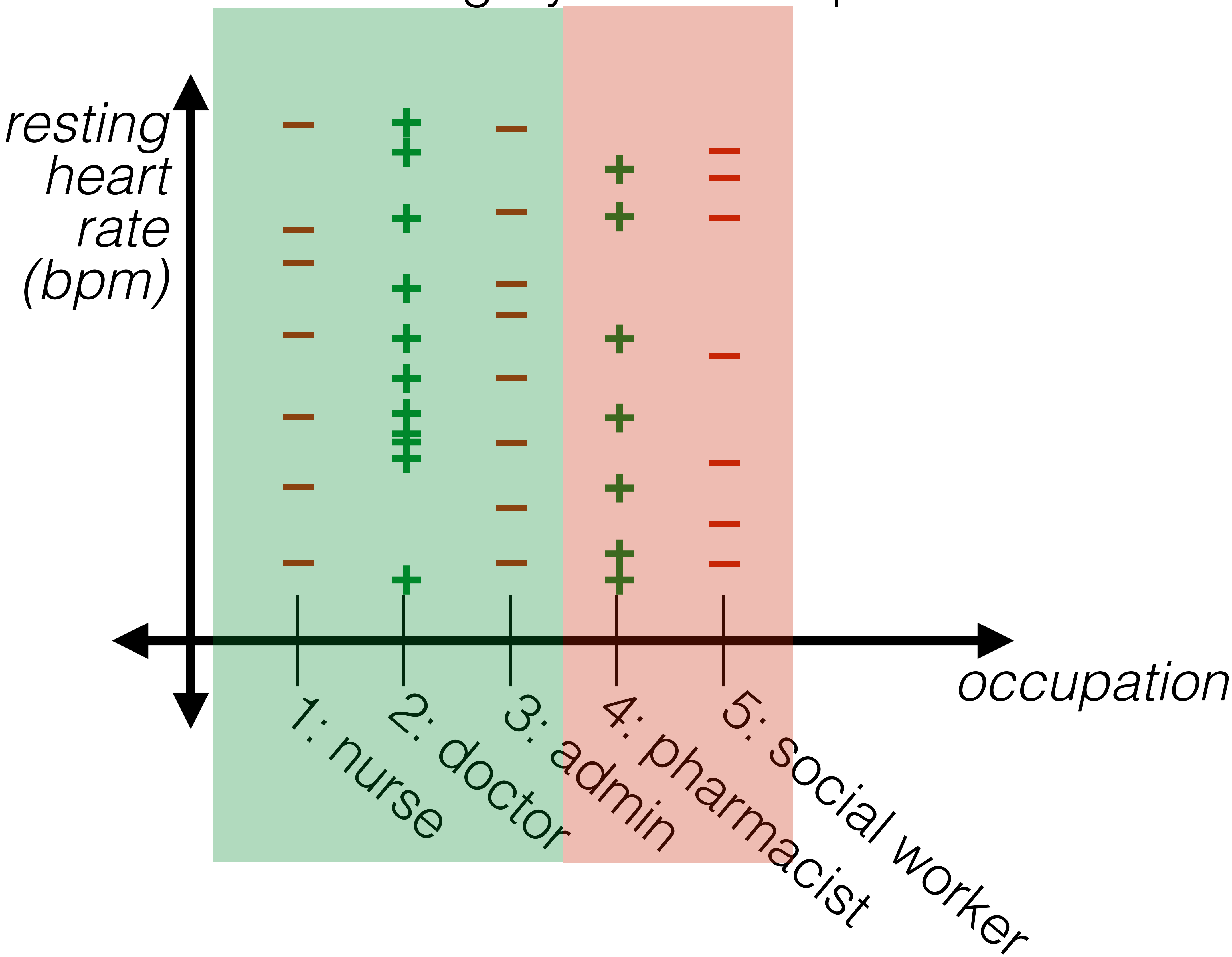
Encode categorical data

- Idea: turn each category into a unique natural number



Encode categorical data

- Idea: turn each category into a unique natural number



Encode categorical data

Encode categorical data

- Idea: turn each category into a unique binary number

Encode categorical data

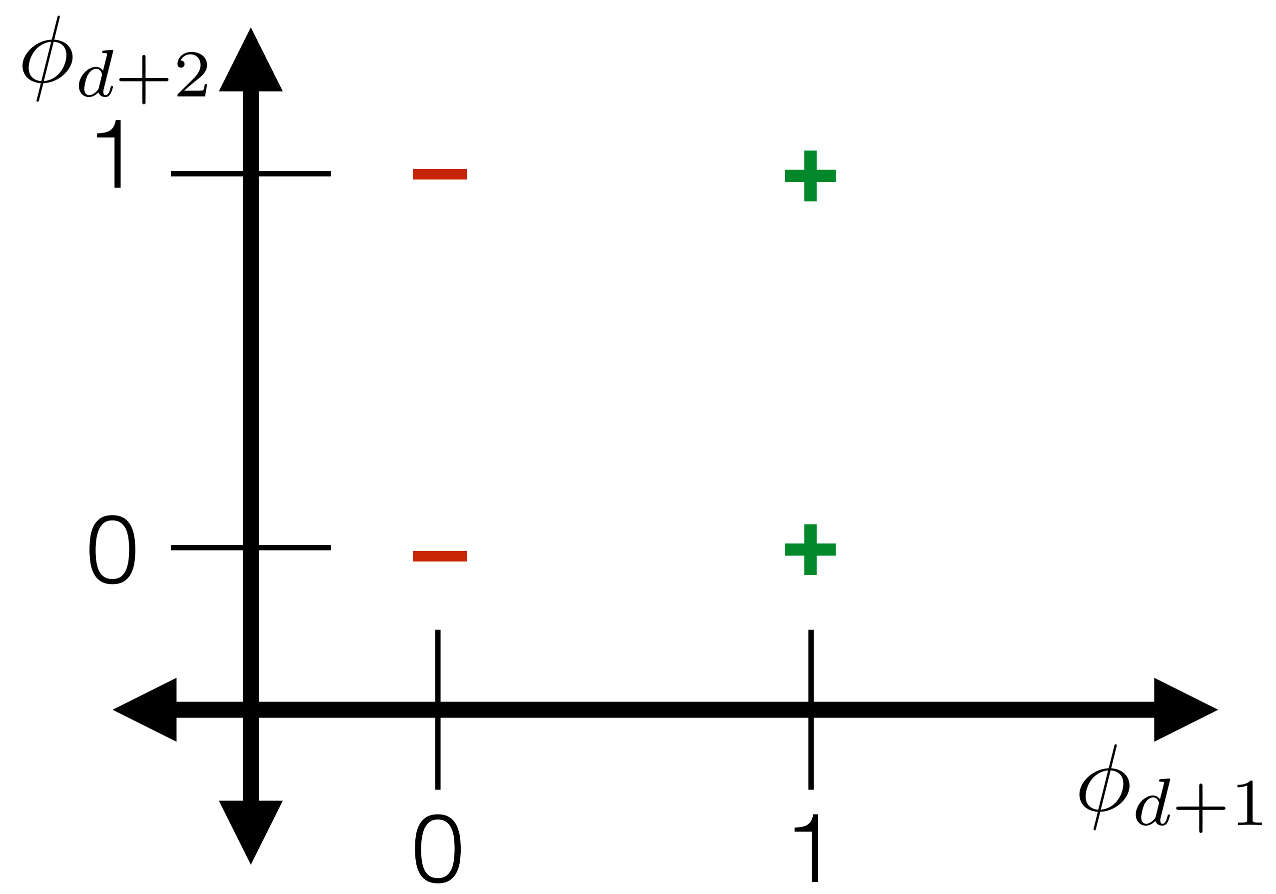
- Idea: turn each category into a unique binary number

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}
nurse	0	0	0
admin	0	0	1
pharmacist	0	1	0
doctor	0	1	1
social worker	1	0	0

Encode categorical data

- Idea: turn each category into a unique binary number

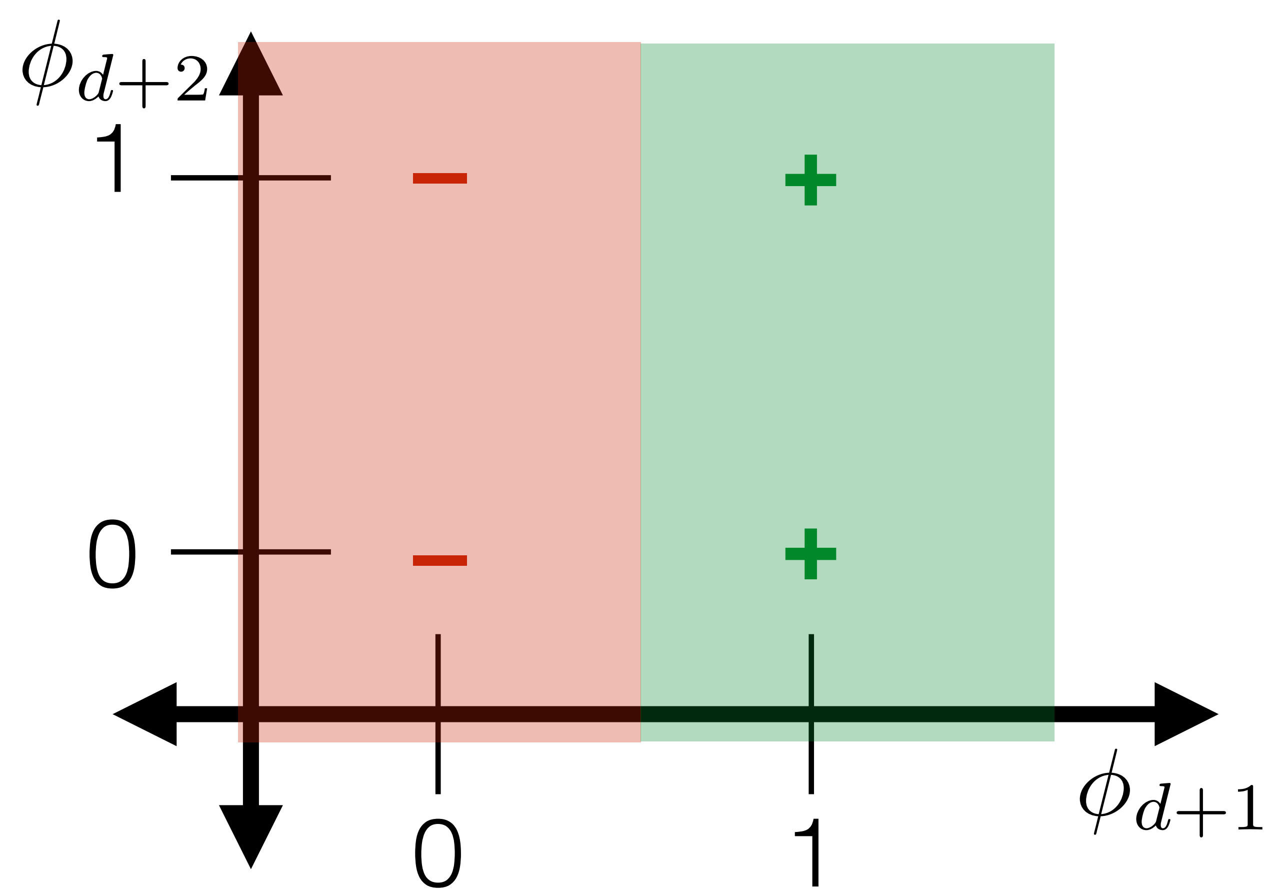
	ϕ_d	ϕ_{d+1}	ϕ_{d+2}
nurse	0	0	0
admin	0	0	1
pharmacist	0	1	0
doctor	0	1	1
social worker	1	0	0



Encode categorical data

- Idea: turn each category into a unique binary number

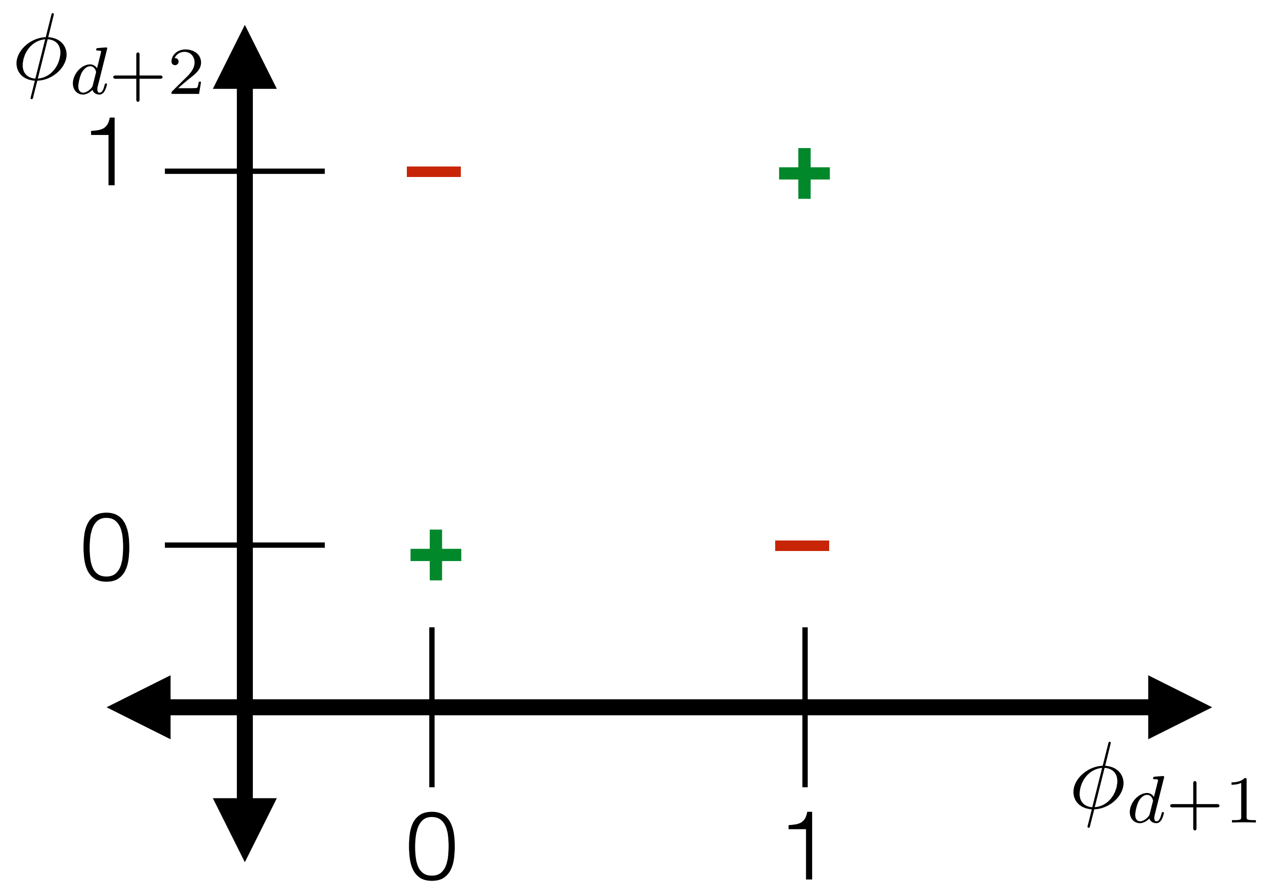
	ϕ_d	ϕ_{d+1}	ϕ_{d+2}
nurse	0	0	0
admin	0	0	1
pharmacist	0	1	0
doctor	0	1	1
social worker	1	0	0



Encode categorical data

- Idea: turn each category into a unique binary number

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}
nurse	0	0	0
admin	0	0	1
pharmacist	0	1	0
doctor	0	1	1
social worker	1	0	0



Encode categorical data

Encode categorical data

- Idea: turn each category into own unique 0-1 feature

Encode categorical data

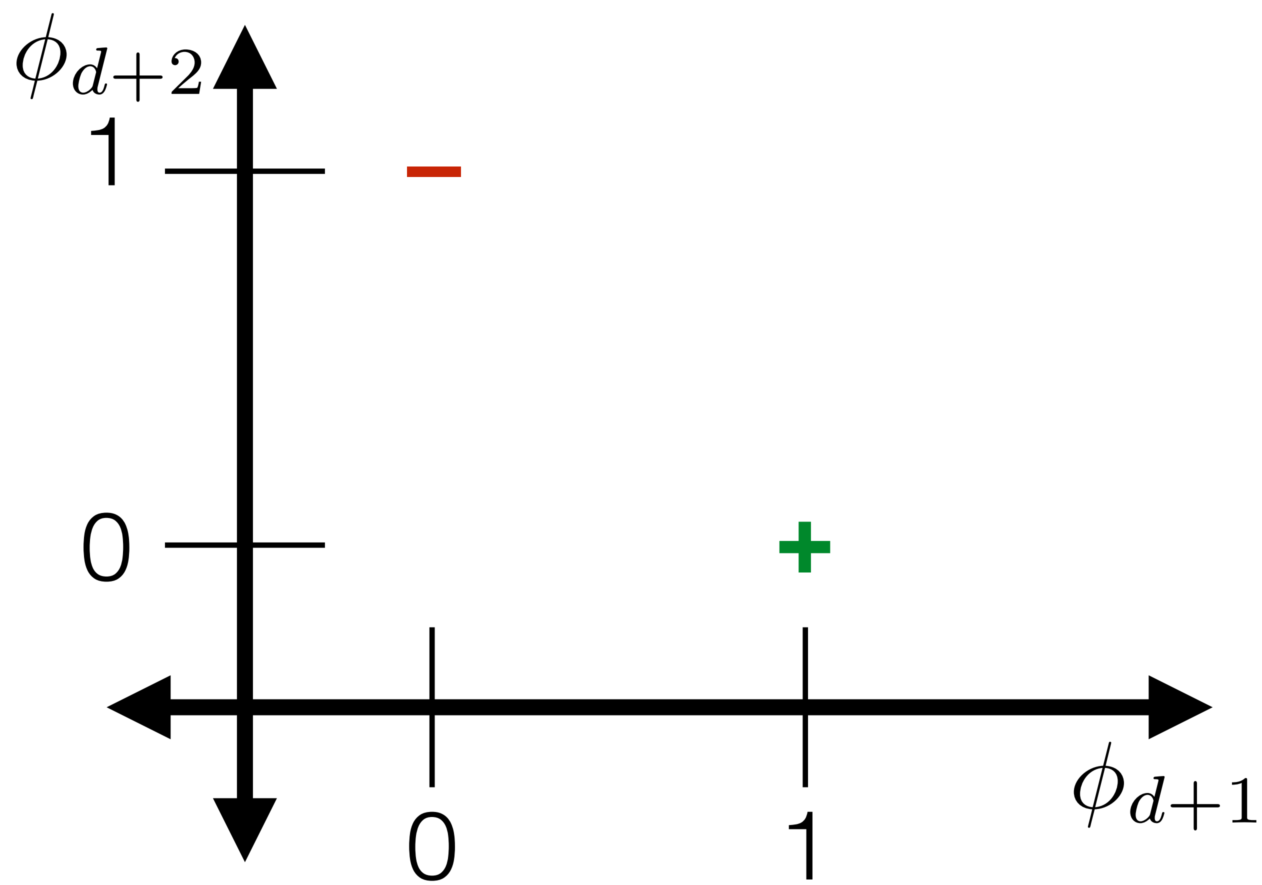
- Idea: turn each category into own unique 0-1 feature

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}	ϕ_{d+3}	ϕ_{d+4}
nurse	1	0	0	0	0
admin	0	1	0	0	0
pharmacist	0	0	1	0	0
doctor	0	0	0	1	0
social worker	0	0	0	0	1

Encode categorical data

- Idea: turn each category into own unique 0-1 feature

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}	ϕ_{d+3}	ϕ_{d+4}
nurse	1	0	0	0	0
admin	0	1	0	0	0
pharmacist	0	0	1	0	0
doctor	0	0	0	1	0
social worker	0	0	0	0	1

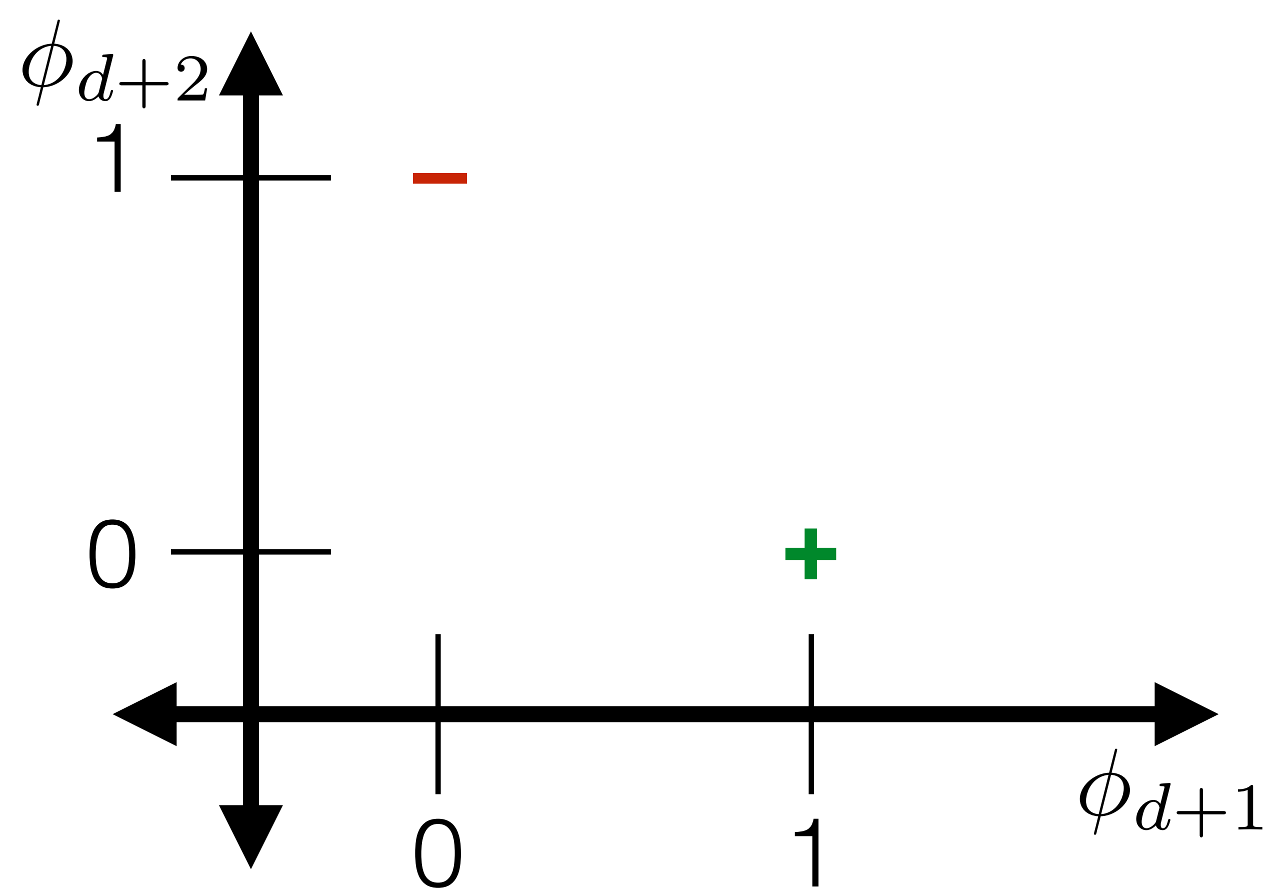


Encode categorical data

- Idea: turn each category into own unique 0-1 feature

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}	ϕ_{d+3}	ϕ_{d+4}
nurse	1	0	0	0	0
admin	0	1	0	0	0
pharmacist	0	0	1	0	0
doctor	0	0	0	1	0
social worker	0	0	0	0	1

- "one-hot encoding"



Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	job	medicines	age	family income (USD)
1	55	0	nurse	pain	40s	133000
2	71	0	admin	beta blockers, pain	20s	34000
3	89	1	nurse	beta blockers	50s	40000
4	67	0	doctor	none	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	medicines	age	family income (USD)
1	55	0	1,0,0,0,0	pain	40s	133000
2	71	0	0,1,0,0,0	beta blockers, pain	20s	34000
3	89	1	1,0,0,0,0	beta blockers	50s	40000
4	67	0	0,0,0,1,0	none	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	medicines	age	family income (USD)
1	55	0	1,0,0,0,0	pain	40s	133000
2	71	0	0,1,0,0,0	beta blockers, pain	20s	34000
3	89	1	1,0,0,0,0	beta blockers	50s	40000
4	67	0	0,0,0,1,0	none	50s	120000

Encode categorical data

pain
pain & beta blockers
beta blockers
no medications

Encode categorical data

- Should we use one-hot encoding?

pain
pain & beta blockers
beta blockers
no medications

Encode categorical data

- Should we use one-hot encoding?

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}	ϕ_{d+3}
pain	1	0	0	0
pain & beta blockers	0	1	0	0
beta blockers	0	0	1	0
no medications	0	0	0	1

Encode categorical data

- Should we use one-hot encoding?

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}	ϕ_{d+3}
pain	1	0	0	0
pain & beta blockers	0	1	0	0
beta blockers	0	0	1	0
no medications	0	0	0	1

- Idea: factored encoding

Encode categorical data

- Should we use one-hot encoding?

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}	ϕ_{d+3}
pain	1	0	0	0
pain & beta blockers	0	1	0	0
beta blockers	0	0	1	0
no medications	0	0	0	1

- Idea: factored encoding

	ϕ_d	ϕ_{d+1}
pain	1	0
pain & beta blockers	1	1
beta blockers	0	1
no medications	0	0

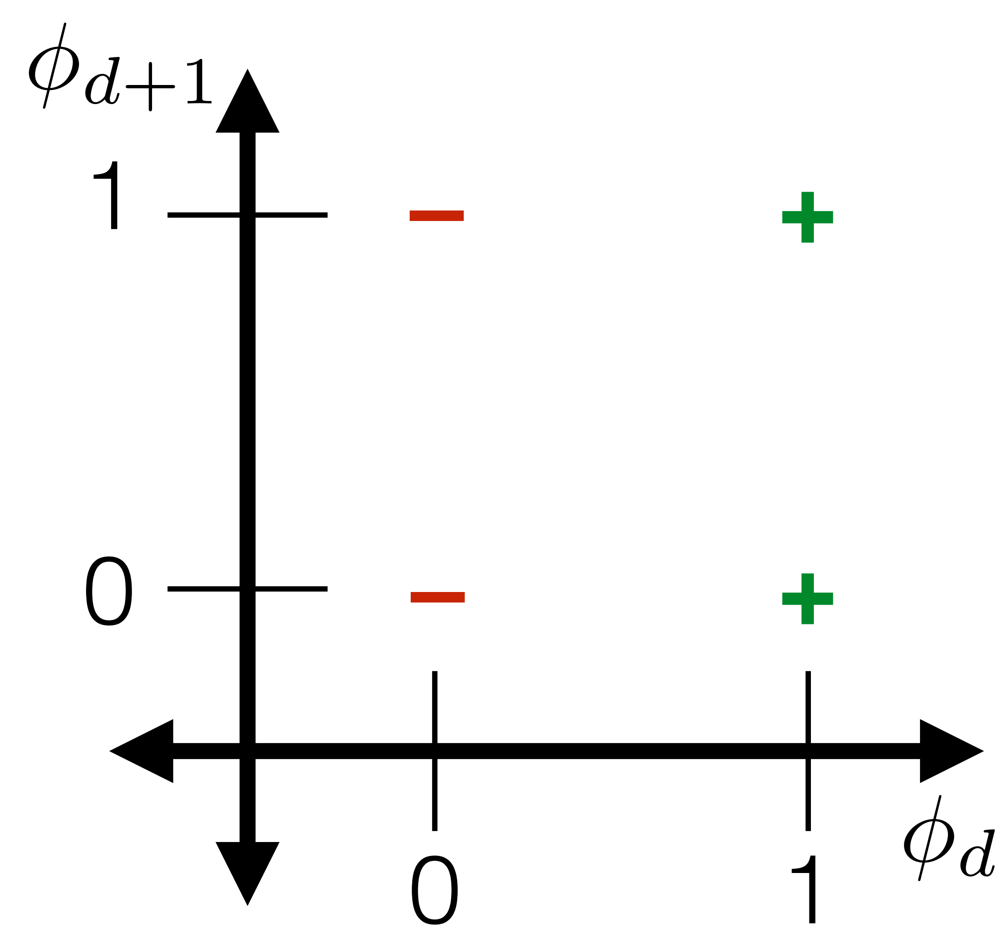
Encode categorical data

- Should we use one-hot encoding?

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}	ϕ_{d+3}
pain	1	0	0	0
pain & beta blockers	0	1	0	0
beta blockers	0	0	1	0
no medications	0	0	0	1

- Idea: factored encoding

	ϕ_d	ϕ_{d+1}
pain	1	0
pain & beta blockers	1	1
beta blockers	0	1
no medications	0	0



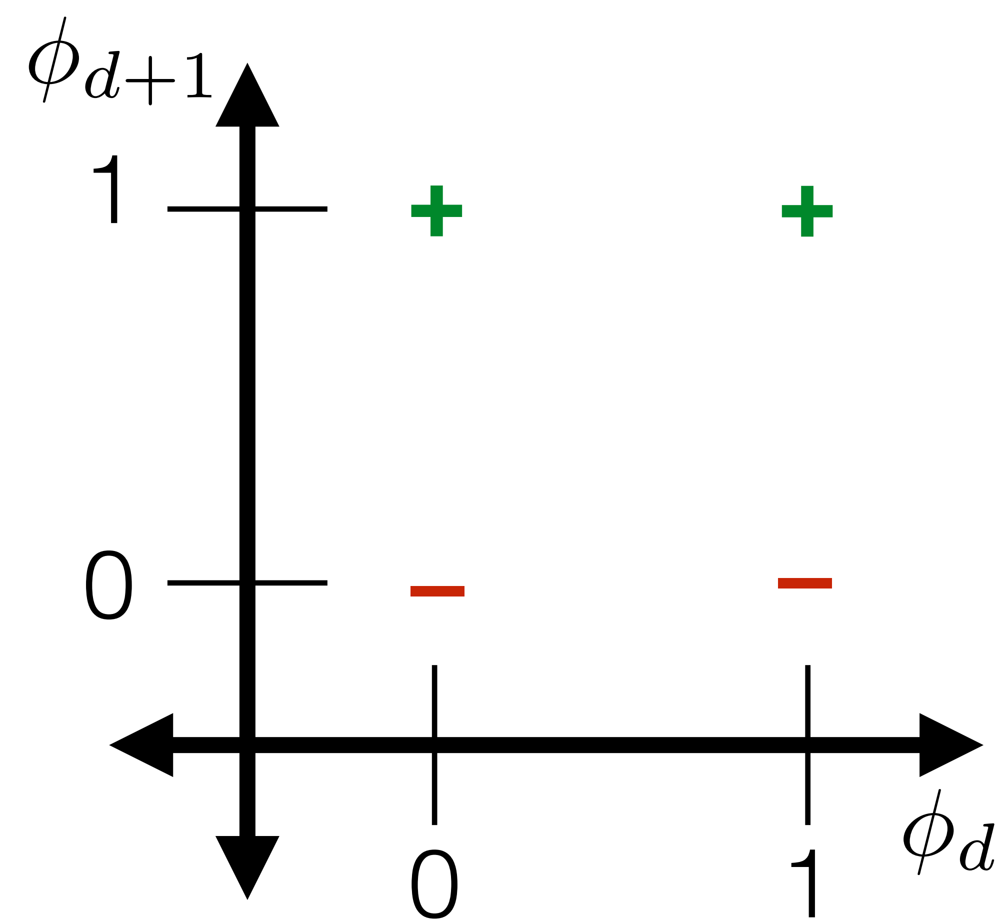
Encode categorical data

- Should we use one-hot encoding?

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}	ϕ_{d+3}
pain	1	0	0	0
pain & beta blockers	0	1	0	0
beta blockers	0	0	1	0
no medications	0	0	0	1

- Idea: factored encoding

	ϕ_d	ϕ_{d+1}
pain	1	0
pain & beta blockers	1	1
beta blockers	0	1
no medications	0	0



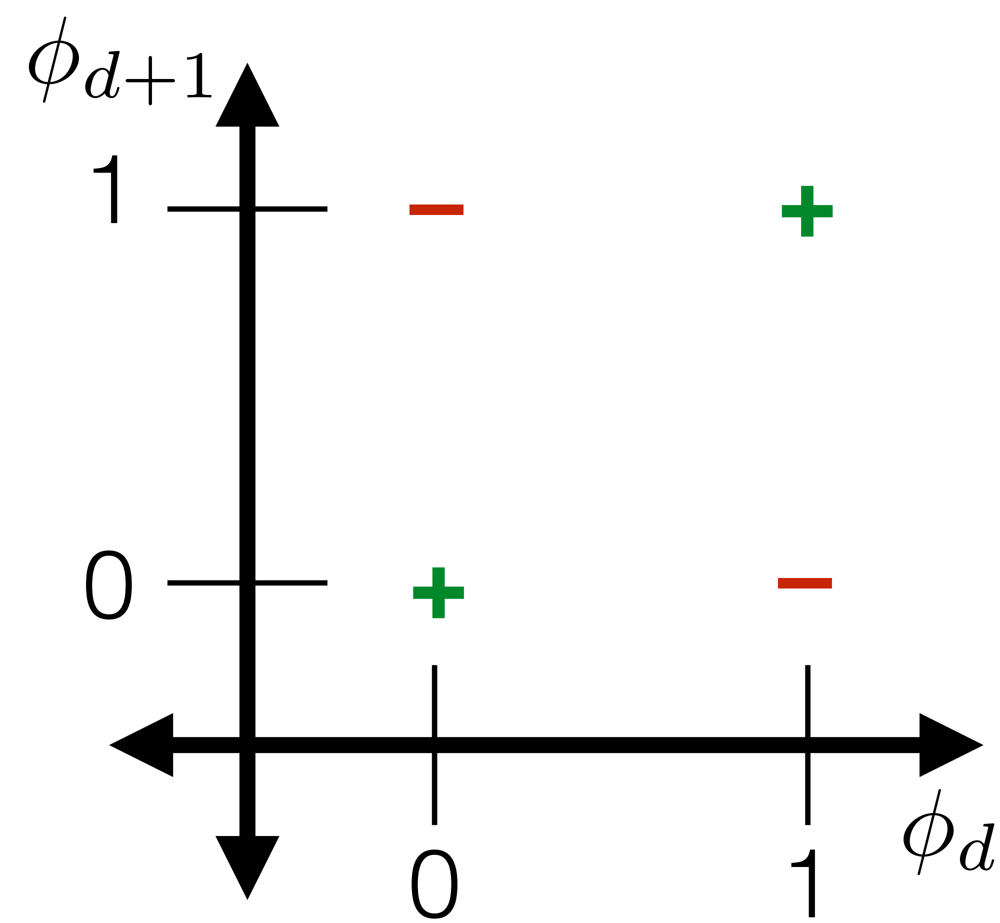
Encode categorical data

- Should we use one-hot encoding?

	ϕ_d	ϕ_{d+1}	ϕ_{d+2}	ϕ_{d+3}
pain	1	0	0	0
pain & beta blockers	0	1	0	0
beta blockers	0	0	1	0
no medications	0	0	0	1

- Idea: factored encoding

	ϕ_d	ϕ_{d+1}
pain	1	0
pain & beta blockers	1	1
beta blockers	0	1
no medications	0	0



Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	medicines	age	family income (USD)
1	55	0	1,0,0,0,0	pain	40s	133000
2	71	0	0,1,0,0,0	beta blockers, pain	20s	34000
3	89	1	1,0,0,0,0	beta blockers	50s	40000
4	67	0	0,0,0,1,0	none	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	age	family income (USD)
1	55	0	1,0,0,0,0	1,0	40s	133000
2	71	0	0,1,0,0,0	1,1	20s	34000
3	89	1	1,0,0,0,0	0,1	50s	40000
4	67	0	0,0,0,1,0	0,0	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	age	family income (USD)
1	55	0	1,0,0,0,0	1,0	40s	133000
2	71	0	0,1,0,0,0	1,1	20s	34000
3	89	1	1,0,0,0,0	0,1	50s	40000
4	67	0	0,0,0,1,0	0,0	50s	120000

Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	age	family income (USD)
1	55	0	1,0,0,0,0	1,0	45	133000
2	71	0	0,1,0,0,0	1,1	25	34000
3	89	1	1,0,0,0,0	0,1	55	40000
4	67	0	0,0,0,1,0	0,0	55	120000

Using a representative # for a range

age
45
25
55
55

Using a representative # for a range

- Potential pitfall: level of detail might be treated as meaningful (by you or others using the data)

age
45
25
55
55

Using a representative # for a range

- Potential pitfall: level of detail might be treated as meaningful (by you or others using the data)

age
45
25
55
55

[https://fusion.tv/story/287592/internet-mapping-glitch-kansas-farm/?utm_medium=social]

Using a representative # for a range

- Potential pitfall: level of detail might be treated as meaningful (by you or others using the data)
- A way to diagnose many problems: plot your data!

age
45
25
55
55

[https://fusion.tv/story/287592/internet-mapping-glitch-kansas-farm/?utm_medium=social]

Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	age	family income (USD)
1	55	0	1,0,0,0,0	1,0	45	133000
2	71	0	0,1,0,0,0	1,1	25	34000
3	89	1	1,0,0,0,0	0,1	55	40000
4	67	0	0,0,0,1,0	0,0	55	120000

Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	decade	family income (USD)
1	55	0	1,0,0,0,0	1,0	4	133000
2	71	0	0,1,0,0,0	1,1	2	34000
3	89	1	1,0,0,0,0	0,1	5	40000
4	67	0	0,0,0,1,0	0,0	5	120000

Encode ordinal data

Encode ordinal data

- Numerical data: order on data values, and differences in value are meaningful

Encode ordinal data

- Numerical data: order on data values, and differences in value are meaningful
- Categorical data: no order on data values

Encode ordinal data

- Numerical data: order on data values, and differences in value are meaningful
- Categorical data: no order on data values
- Ordinal data: order on data values, but differences not meaningful

Encode ordinal data

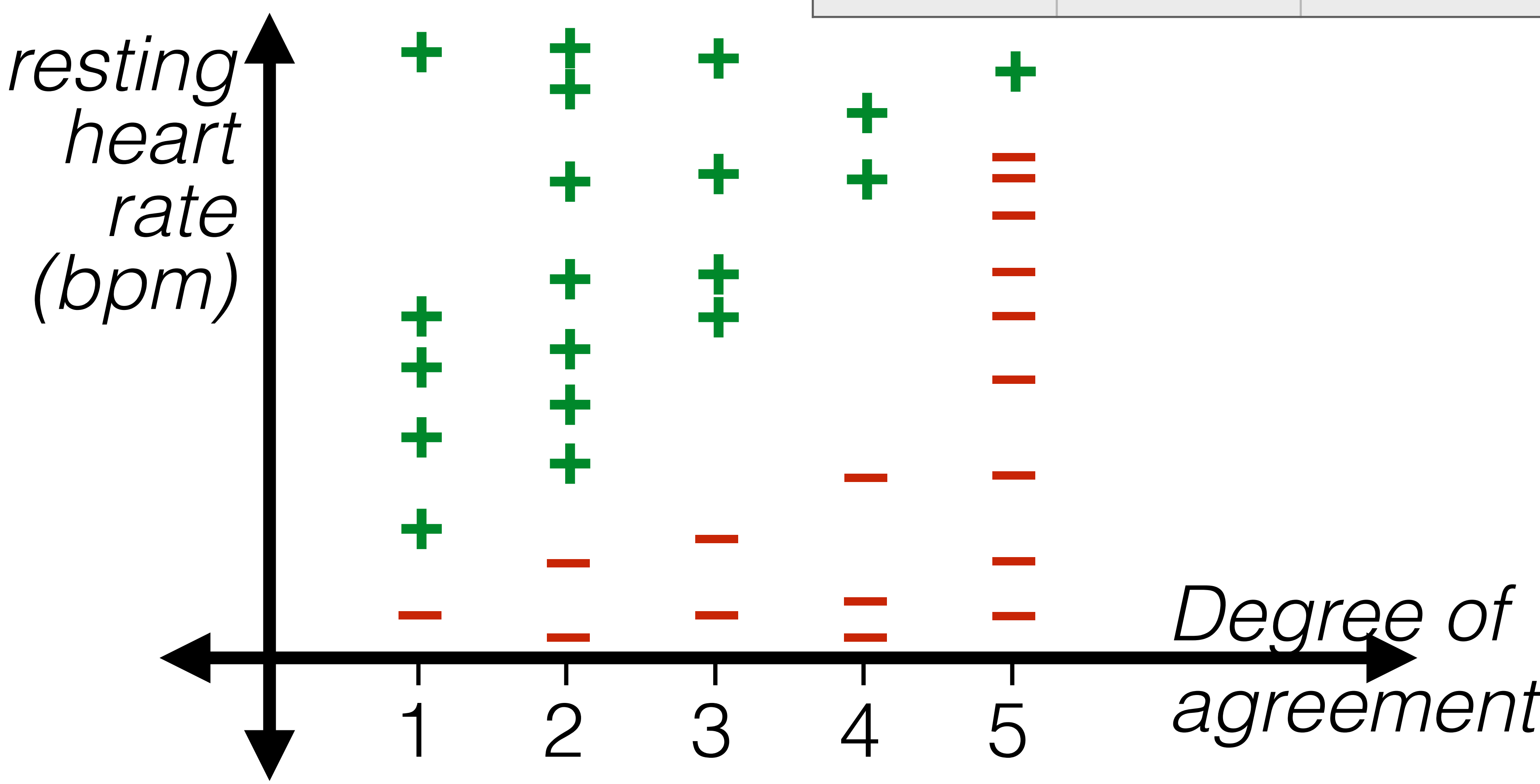
- Numerical data: order on data values, and differences in value are meaningful
- Categorical data: no order on data values
- Ordinal data: order on data values, but differences not meaningful
  - E.g. Likert scale:

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	2	3	4	5

Encode ordinal data

- Numerical data: order on data values, and differences in value are meaningful
- Categorical data: no order on data values
- Ordinal data: order on data values, but differences not meaningful
  - E.g. Likert scale:

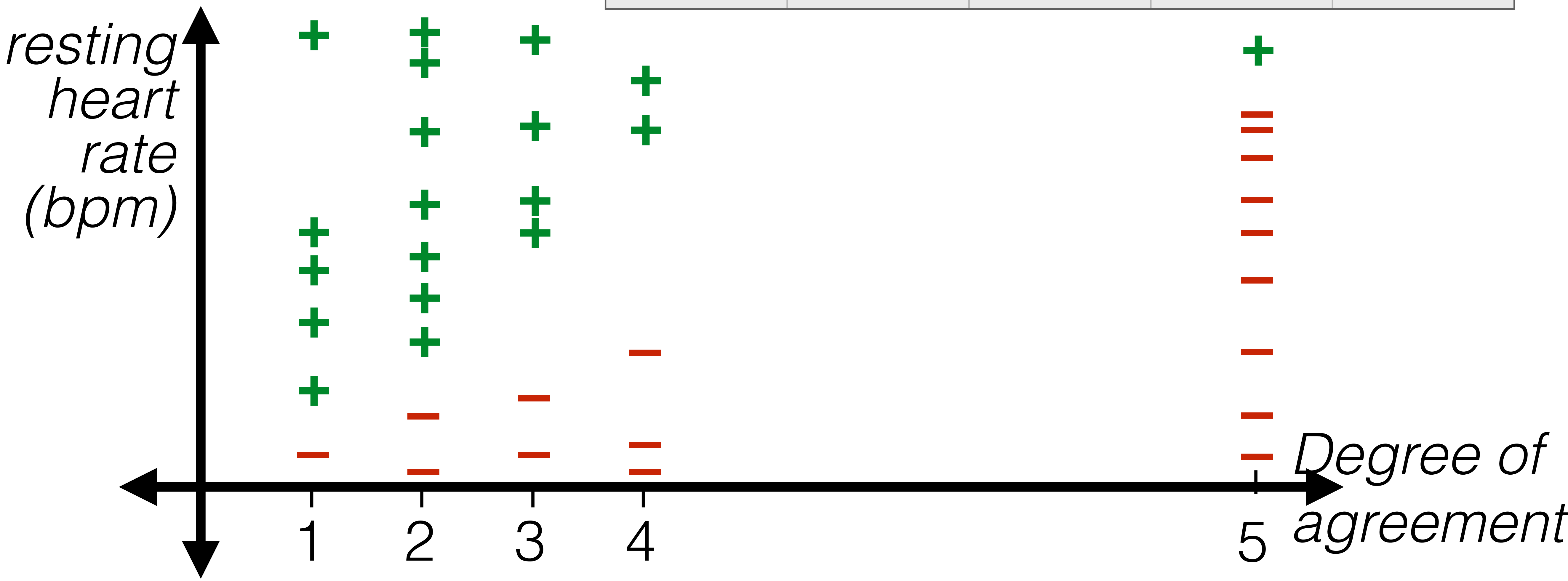
Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	2	3	4	5



Encode ordinal data

- Numerical data: order on data values, and differences in value are meaningful
- Categorical data: no order on data values
- Ordinal data: order on data values, but differences not meaningful
    - E.g. Likert scale:

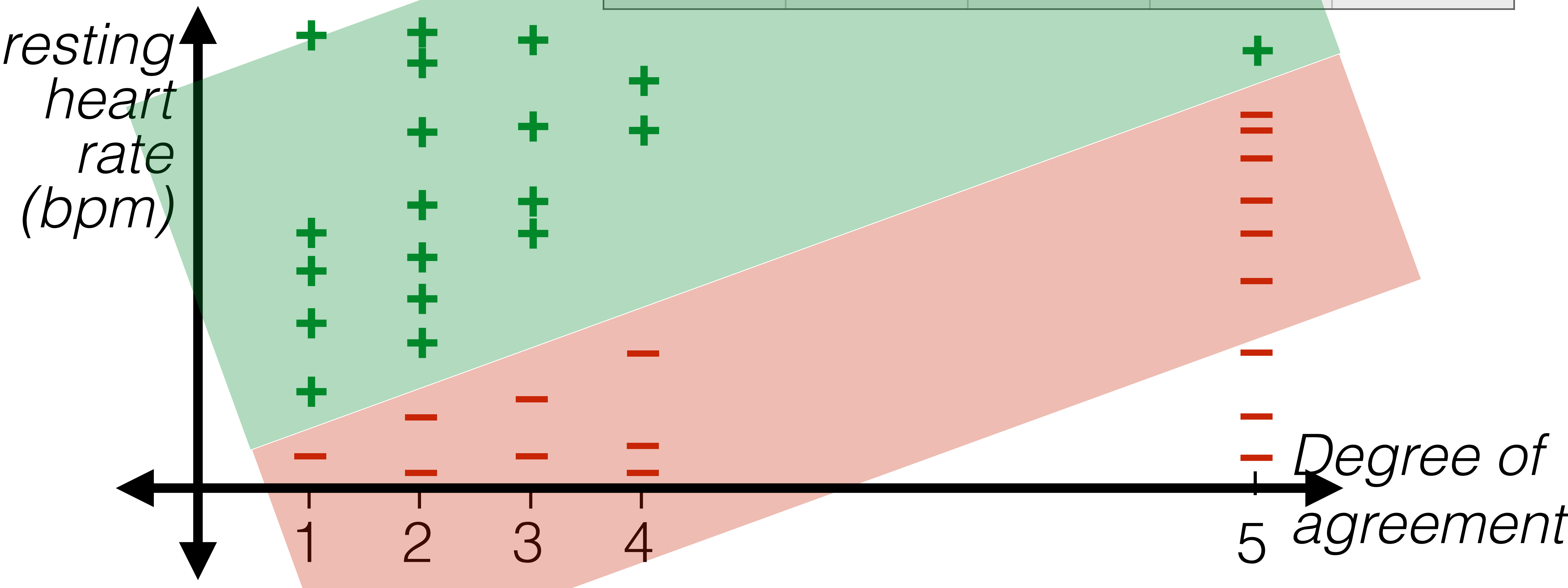
Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	2	3	4	5



Encode ordinal data

- Numerical data: order on data values, and differences in value are meaningful
- Categorical data: no order on data values
- Ordinal data: order on data values, but differences not meaningful
  - E.g. Likert scale:

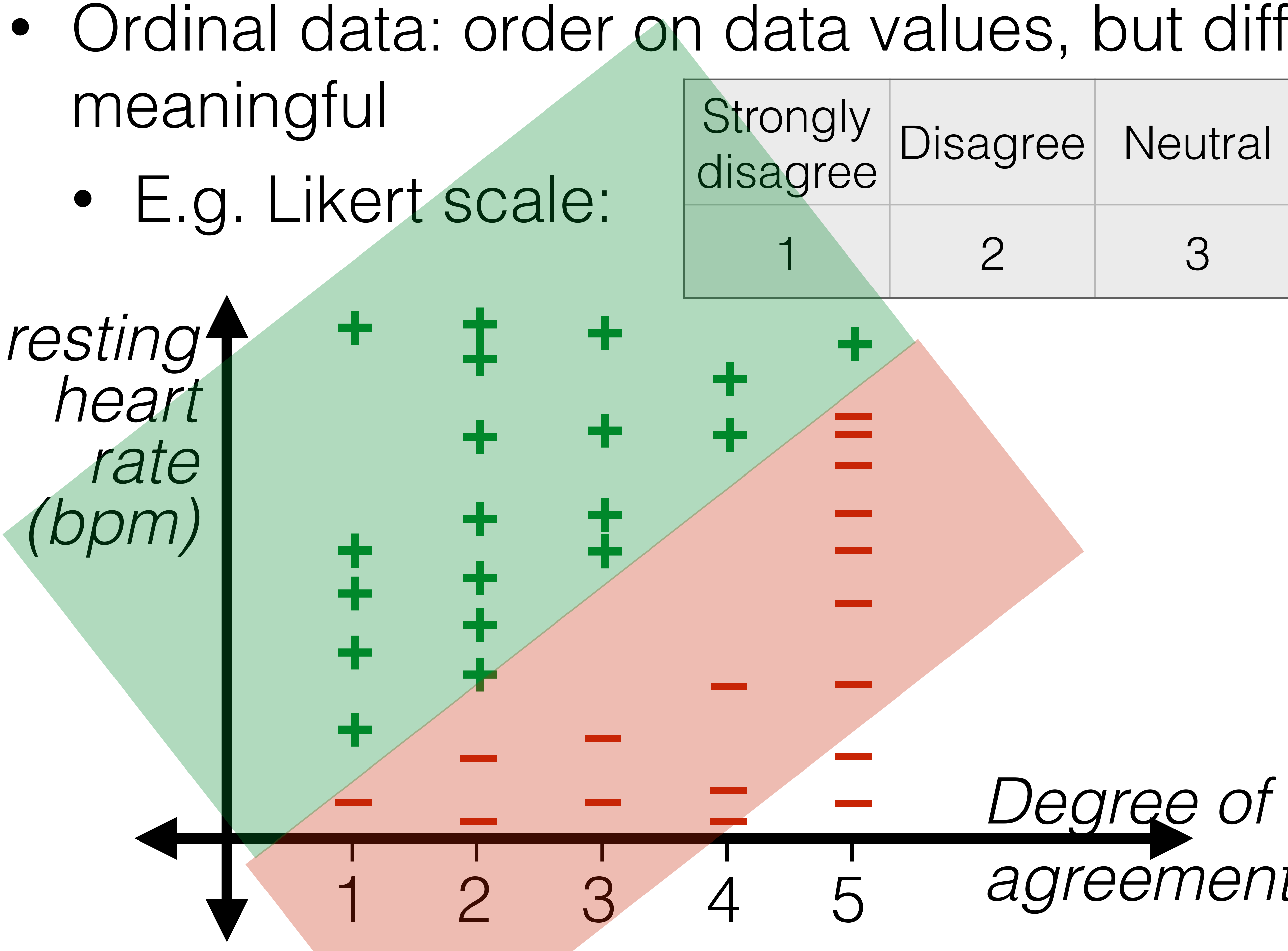
Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	2	3	4	5



Encode ordinal data

- Numerical data: order on data values, and differences in value are meaningful
- Categorical data: no order on data values
- Ordinal data: order on data values, but differences not meaningful
  - E.g. Likert scale:

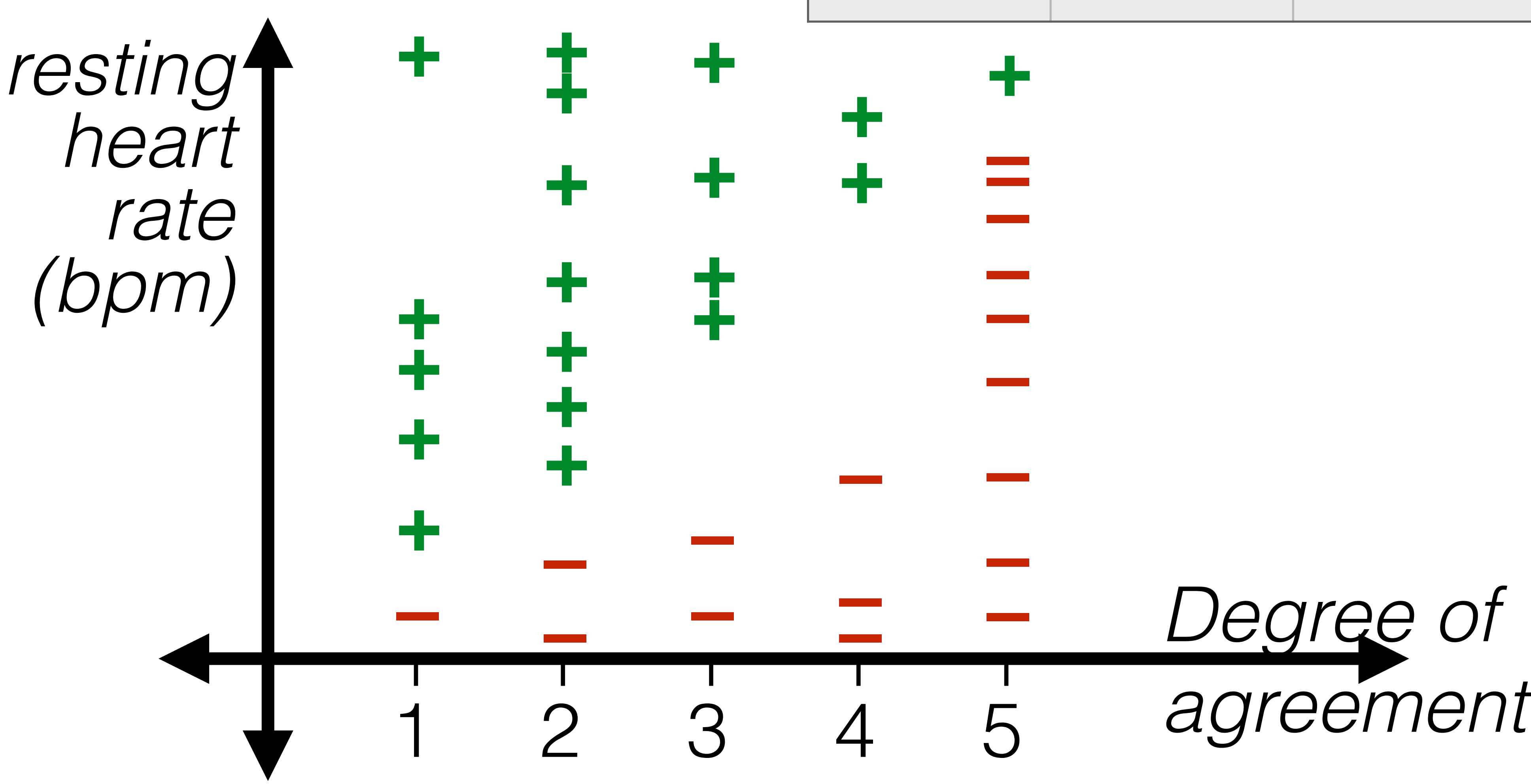
Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	2	3	4	5



Encode ordinal data

- Numerical data: order on data values, and differences in value are meaningful
- Categorical data: no order on data values
- Ordinal data: order on data values, but differences not meaningful
    - E.g. Likert scale:

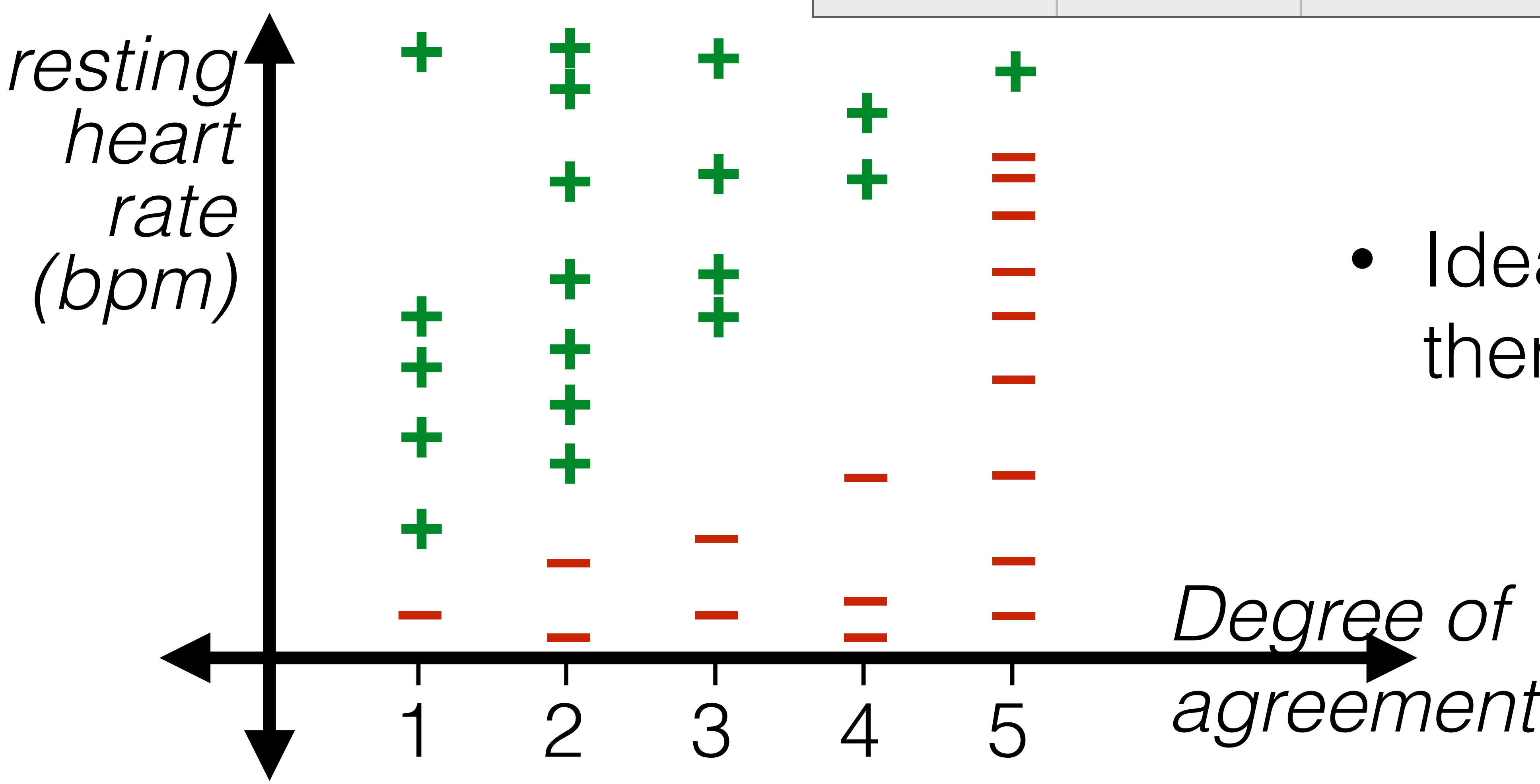
Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	2	3	4	5



Encode ordinal data

- Numerical data: order on data values, and differences in value are meaningful
- Categorical data: no order on data values
- Ordinal data: order on data values, but differences not meaningful
    - E.g. Likert scale:

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	2	3	4	5

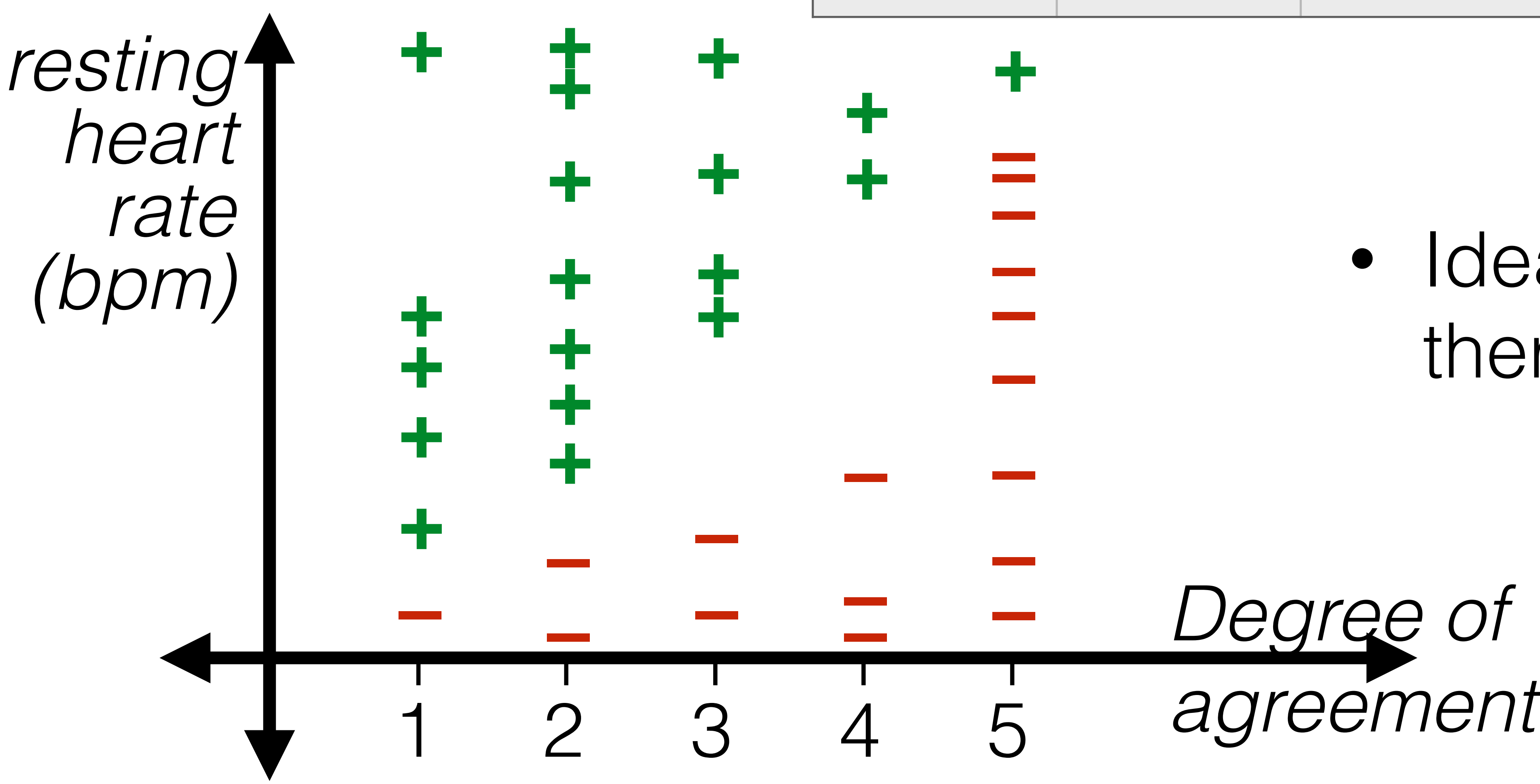


- Idea: Unary/ thermometer code

Encode ordinal data

- Numerical data: order on data values, and differences in value are meaningful
- Categorical data: no order on data values
- Ordinal data: order on data values, but differences not meaningful
  - E.g. Likert scale:

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1,0,0,0,0	1,1,0,0,0	1,1,1,0,0	1,1,1,1,0	1,1,1,1,1



- Idea: Unary/ thermometer code

Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	decade	family income (USD)
1	55	0	1,0,0,0,0	1,0	4	133000
2	71	0	0,1,0,0,0	1,1	2	34000
3	89	1	1,0,0,0,0	0,1	5	40000
4	67	0	0,0,0,1,0	0,0	5	120000

Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	decade	family income (USD)
1	55	0	1,0,0,0,0	1,0	4	133000
2	71	0	0,1,0,0,0	1,1	2	34000
3	89	1	1,0,0,0,0	0,1	5	40000
4	67	0	0,0,0,1,0	0,0	5	120000

Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	decade	family income (USD)
1	55	0	1,0,0,0,0	1,0	4	133000
2	71	0	0,1,0,0,0	1,1	2	34000
3	89	1	1,0,0,0,0	0,1	5	40000
4	67	0	0,0,0,1,0	0,0	5	120000

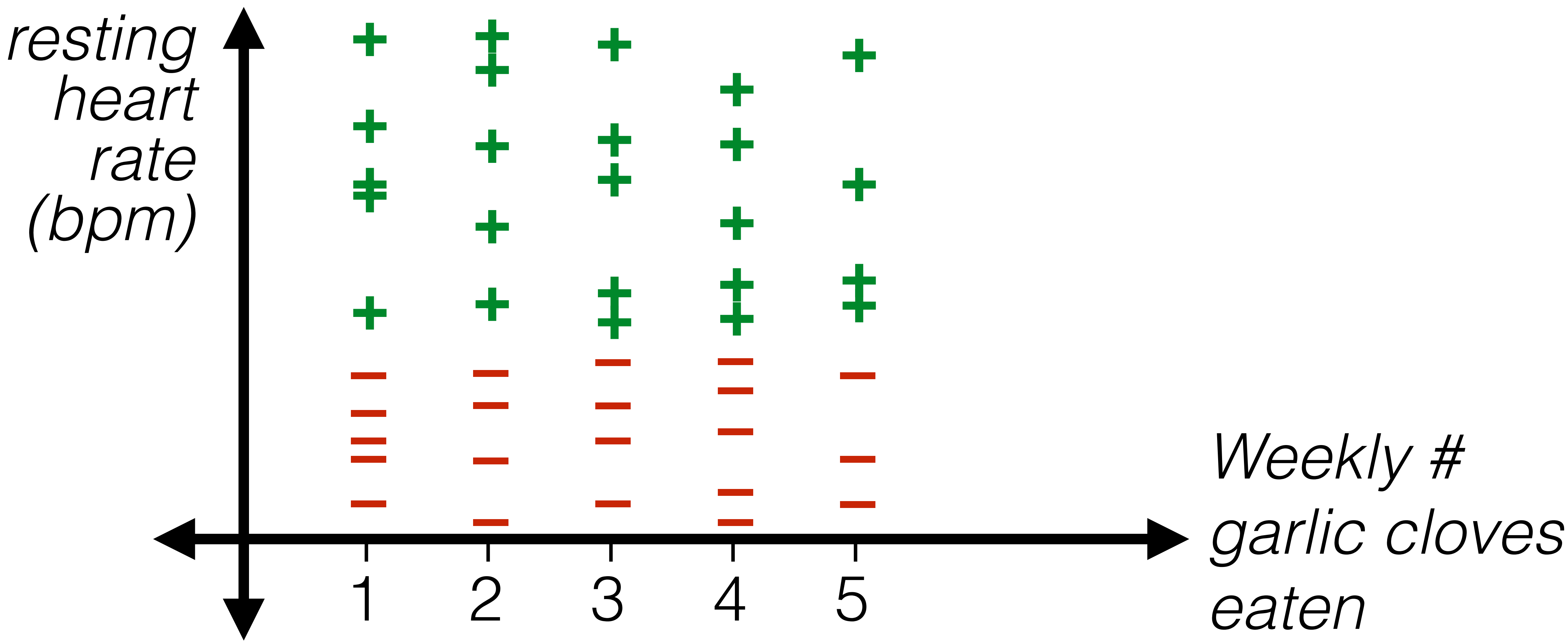
Encode numerical data

Encode numerical data

- A closer look at the output of a linear classifier

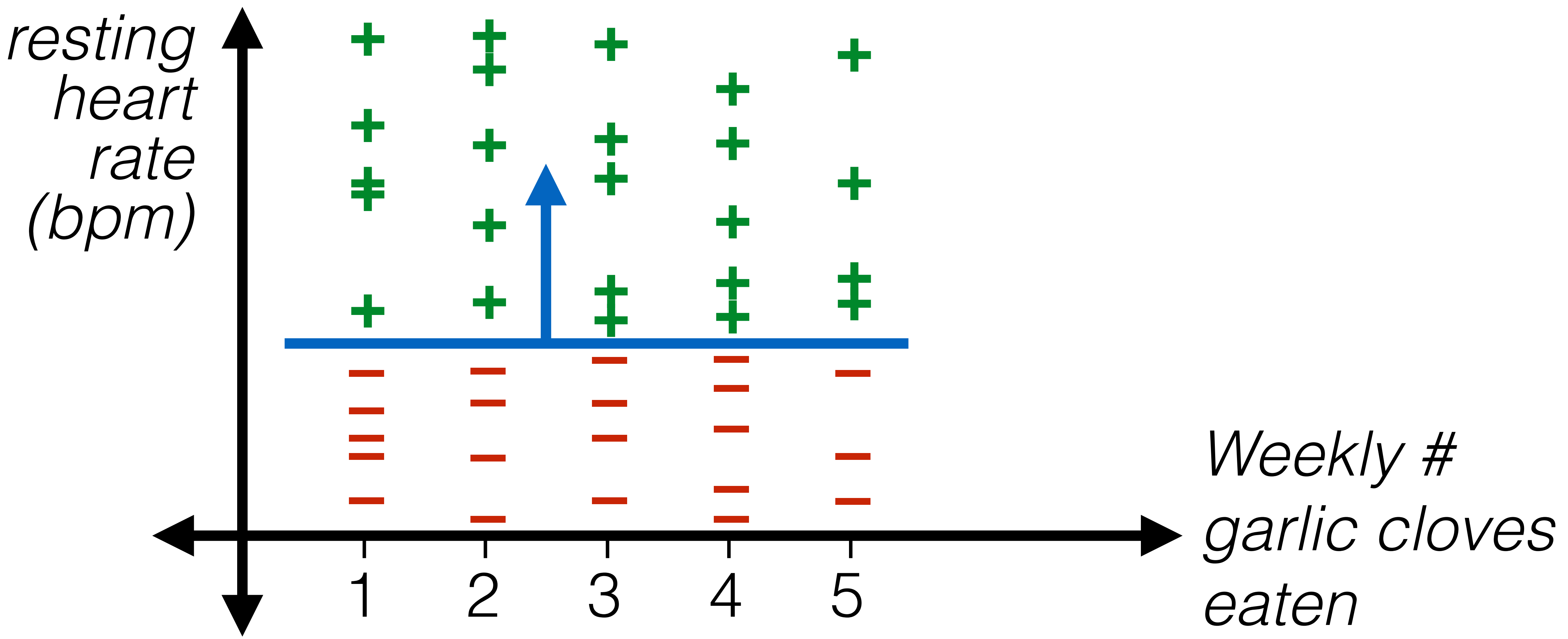
Encode numerical data

- A closer look at the output of a linear classifier



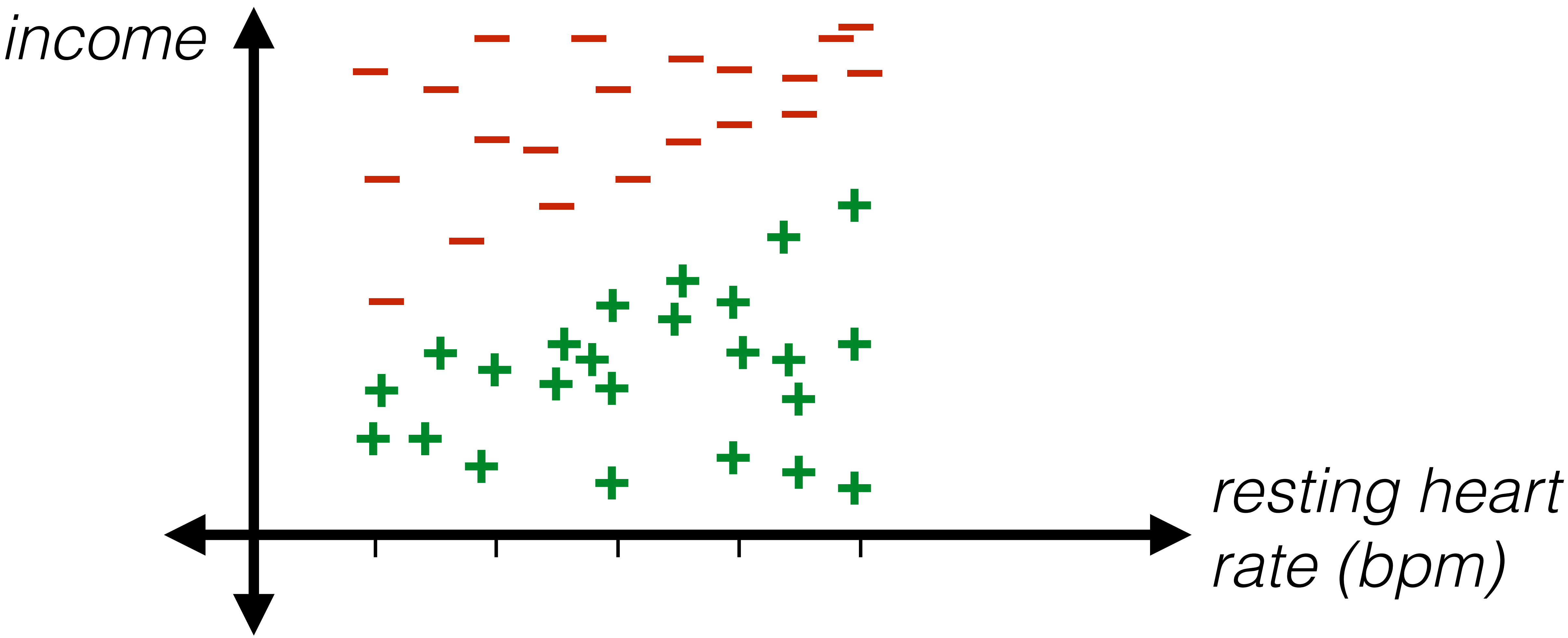
Encode numerical data

- A closer look at the output of a linear classifier



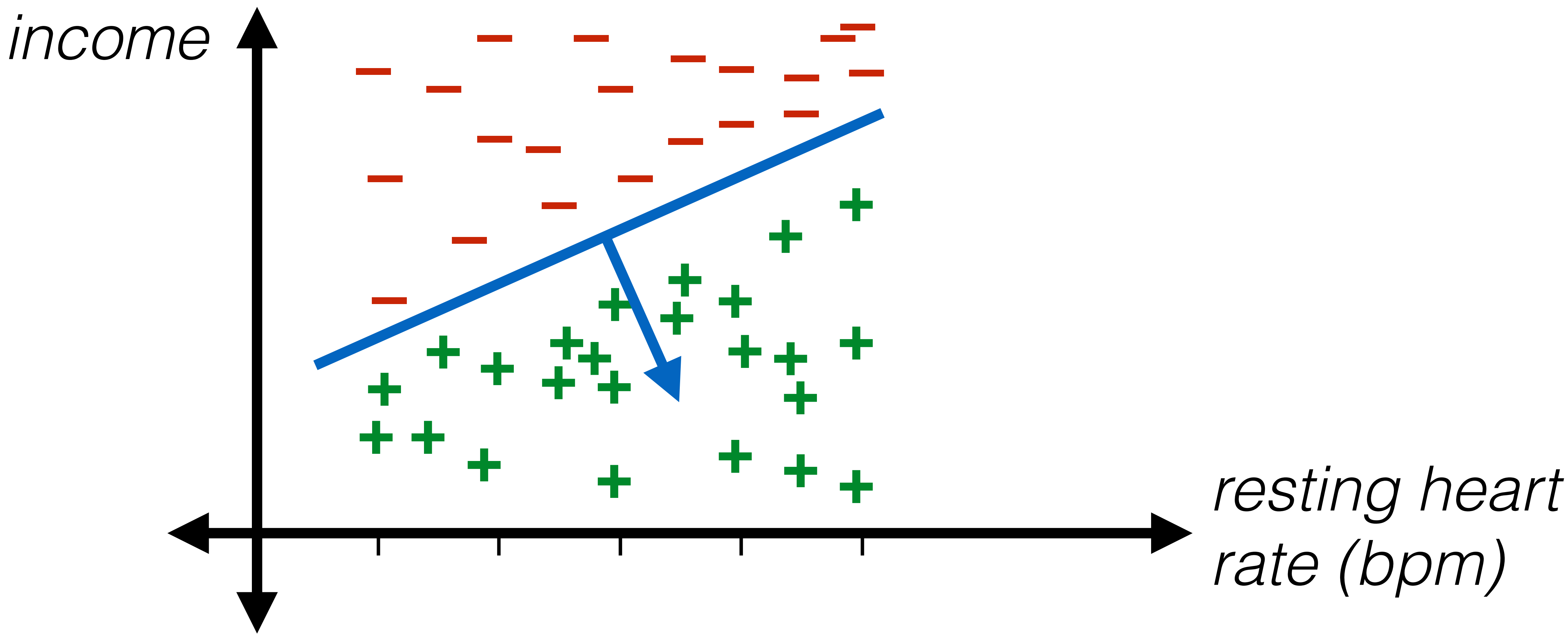
Encode numerical data

- A closer look at the output of a linear classifier



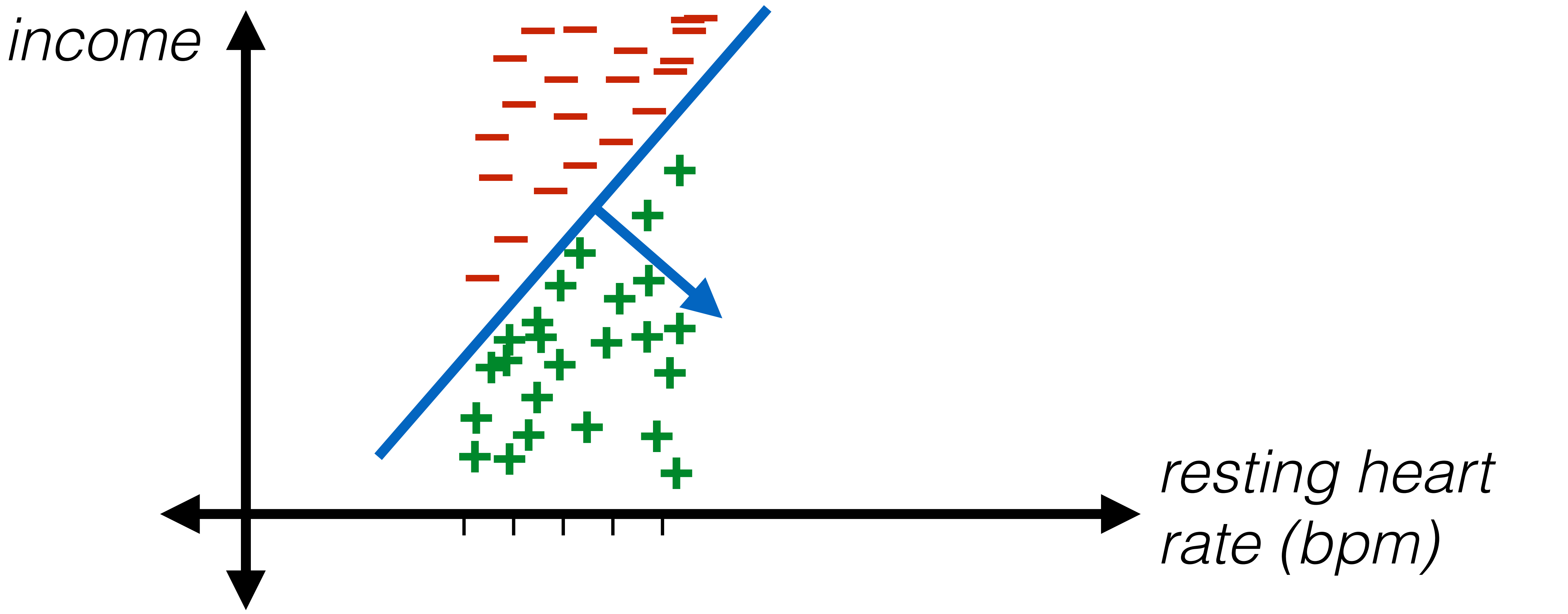
Encode numerical data

- A closer look at the output of a linear classifier



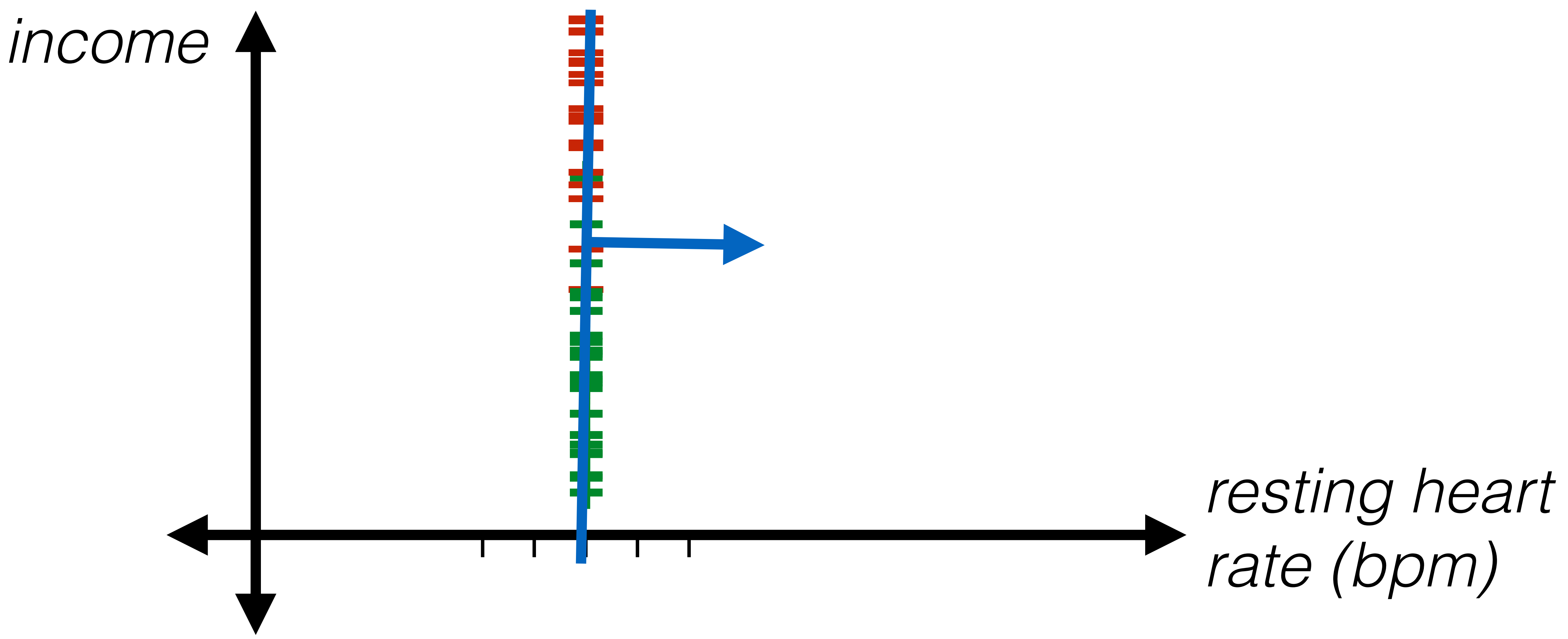
Encode numerical data

- A closer look at the output of a linear classifier



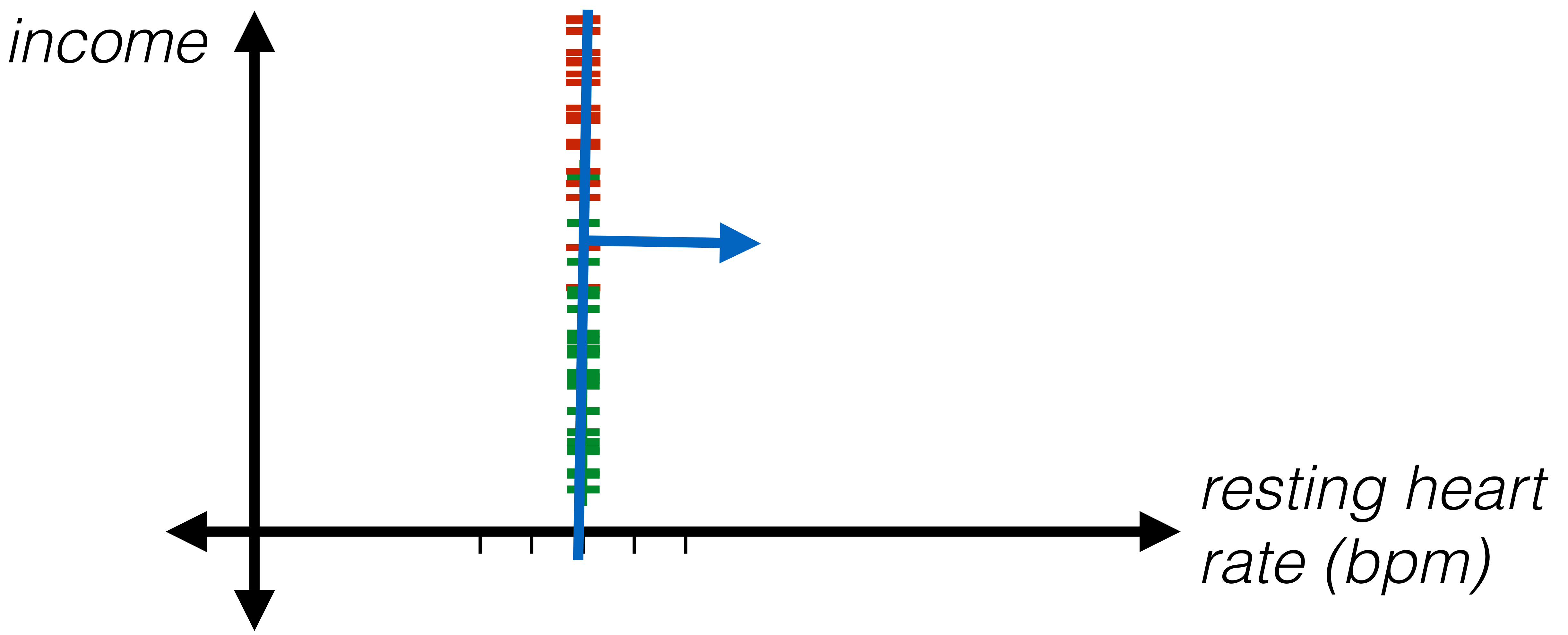
Encode numerical data

- A closer look at the output of a linear classifier



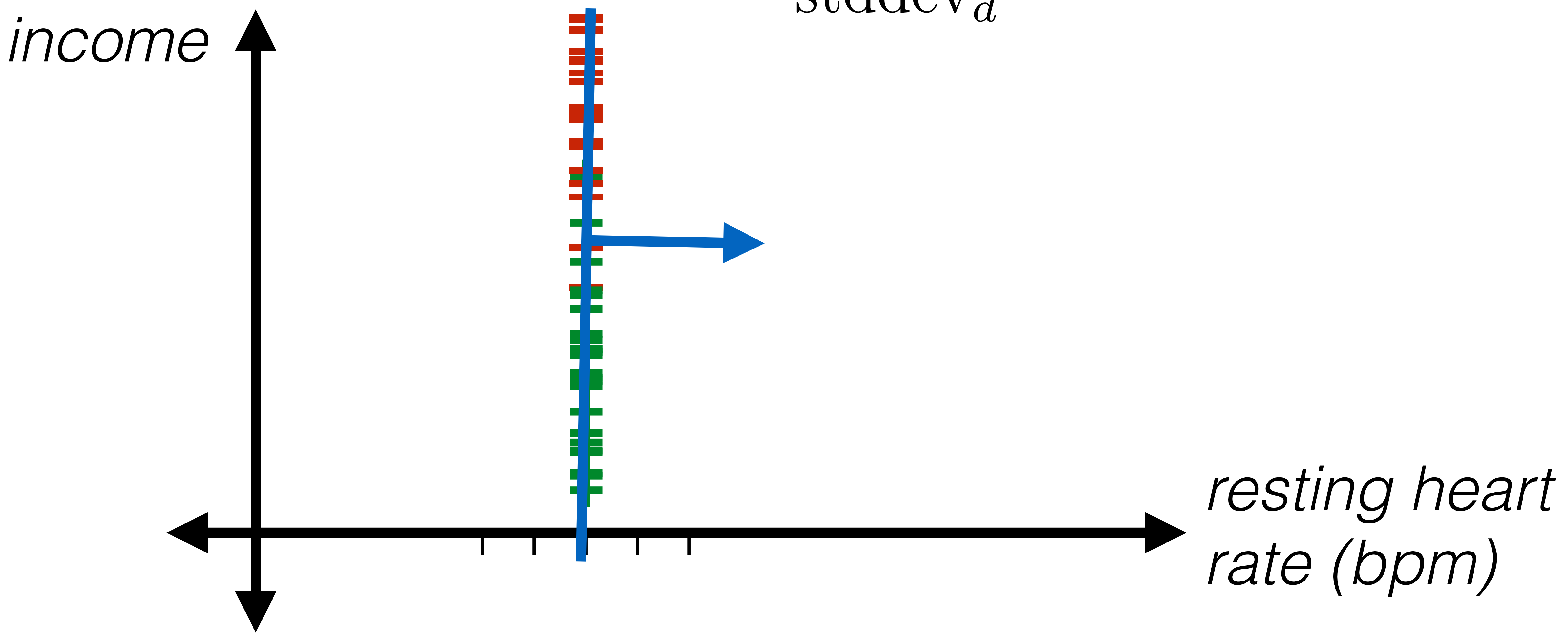
Encode numerical data

- A closer look at the output of a linear classifier
- Idea: standardize numerical data



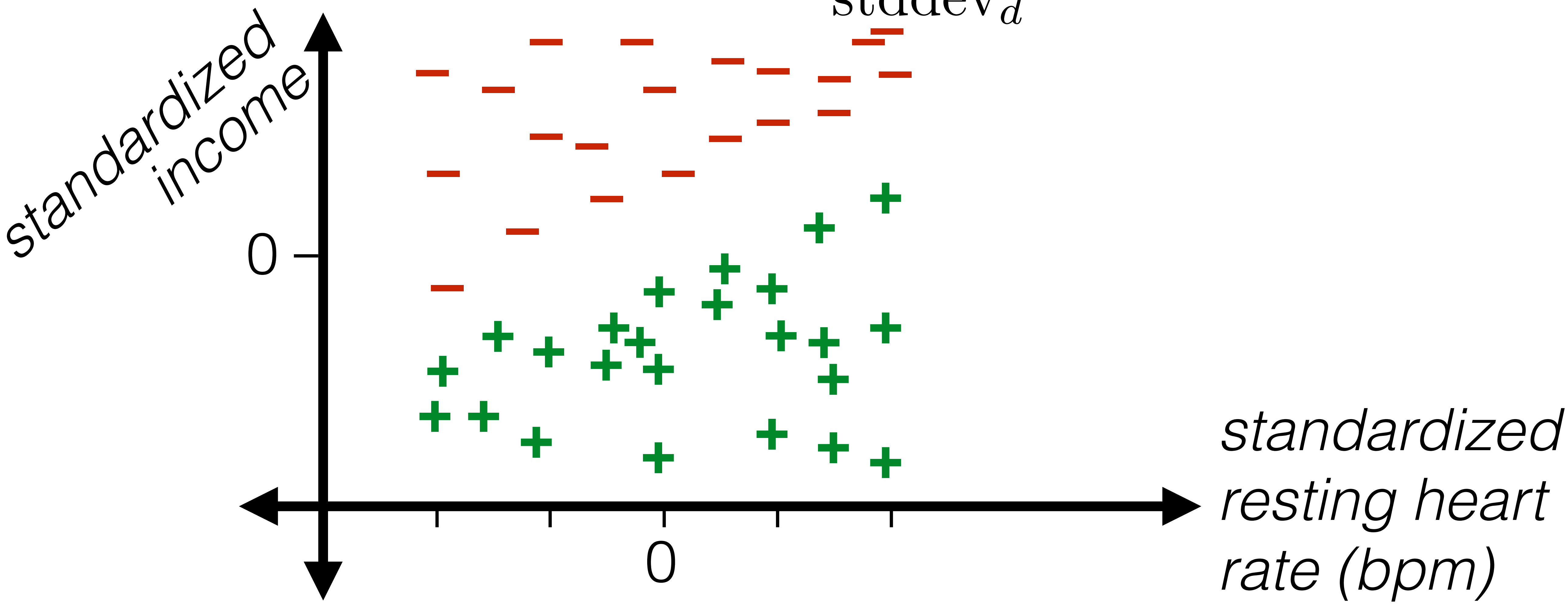
Encode numerical data

- A closer look at the output of a linear classifier
- Idea: standardize numerical data
    - For *d*th feature: $\phi_d^{(k)} = \dfrac{x_d^{(k)} - \text{mean}_d}{\text{stddev}_d}$



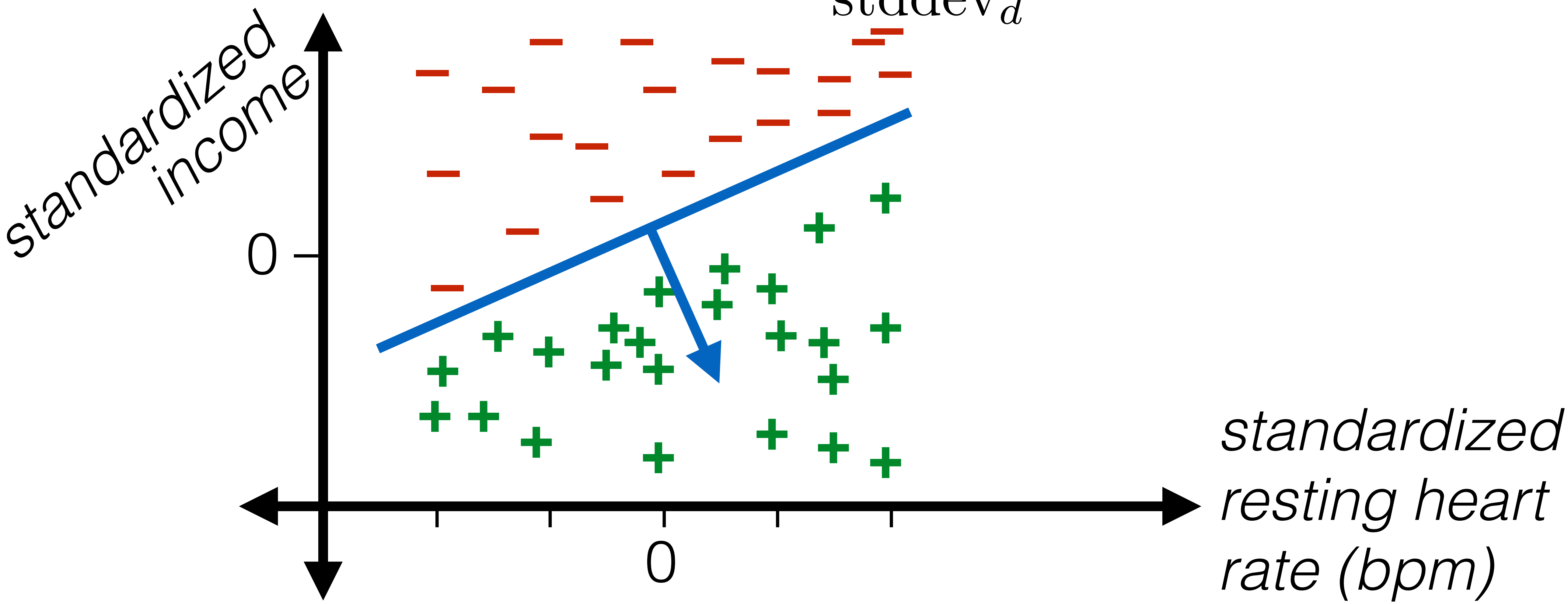
Encode numerical data

- A closer look at the output of a linear classifier
- Idea: standardize numerical data
  - For *d*th feature: $\phi_d^{(k)} = \dfrac{x_d^{(k)} - \mathrm{mean}_d}{\mathrm{stddev}_d}$



Encode numerical data

- A closer look at the output of a linear classifier
- Idea: standardize numerical data
  - For *d*th feature: $\phi_d^{(k)} = \frac{x_d^{(k)} - \text{mean}_d}{\text{stddev}_d}$



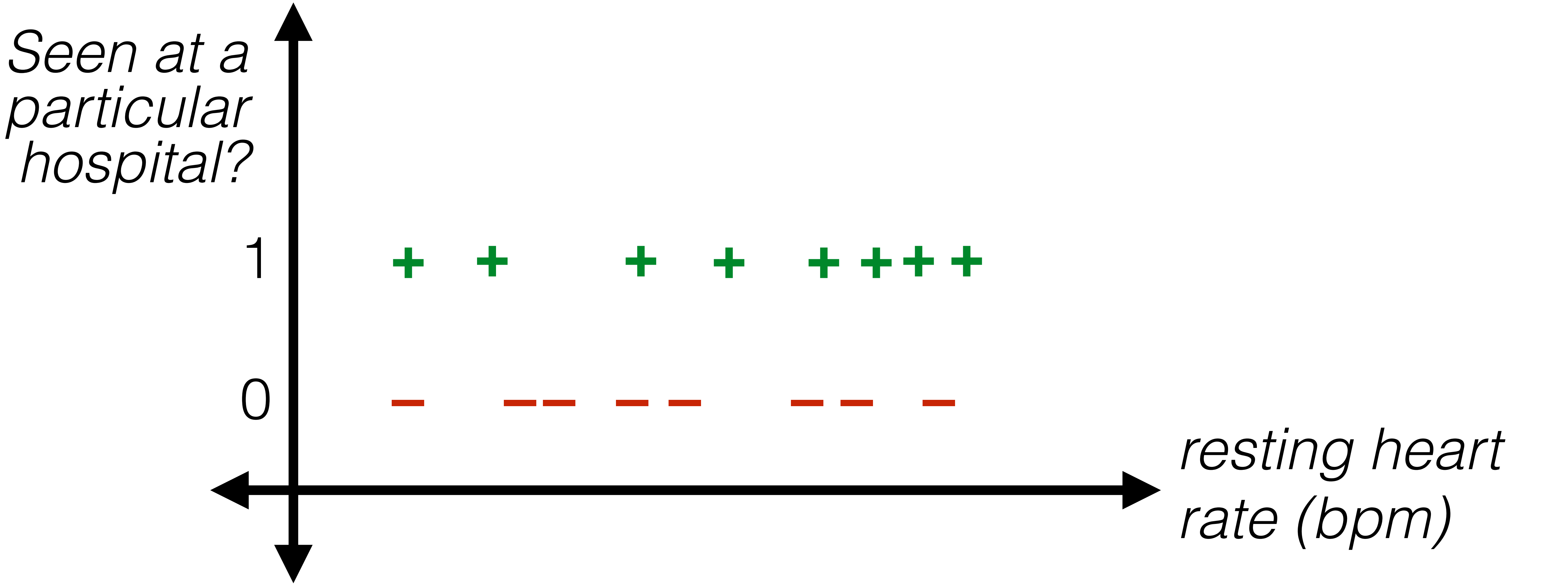
More benefits of plotting your data

More benefits of plotting your data

- And talking to experts

More benefits of plotting your data

- And talking to experts



Encode data in usable form

- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	decade	family income (USD)
1	55	0	1,0,0,0,0	1,0	4	133000
2	71	0	0,1,0,0,0	1,1	2	34000
3	89	1	1,0,0,0,0	0,1	5	40000
4	67	0	0,0,0,1,0	0,0	5	120000

Encode data in usable form

- Identify the features and encode as real numbers
- Standardize numerical features

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	decade	family income (USD)
1	55	0	1,0,0,0,0	1,0	4	133000
2	71	0	0,1,0,0,0	1,1	2	34000
3	89	1	1,0,0,0,0	0,1	5	40000
4	67	0	0,0,0,1,0	0,0	5	120000

Encode data in usable form

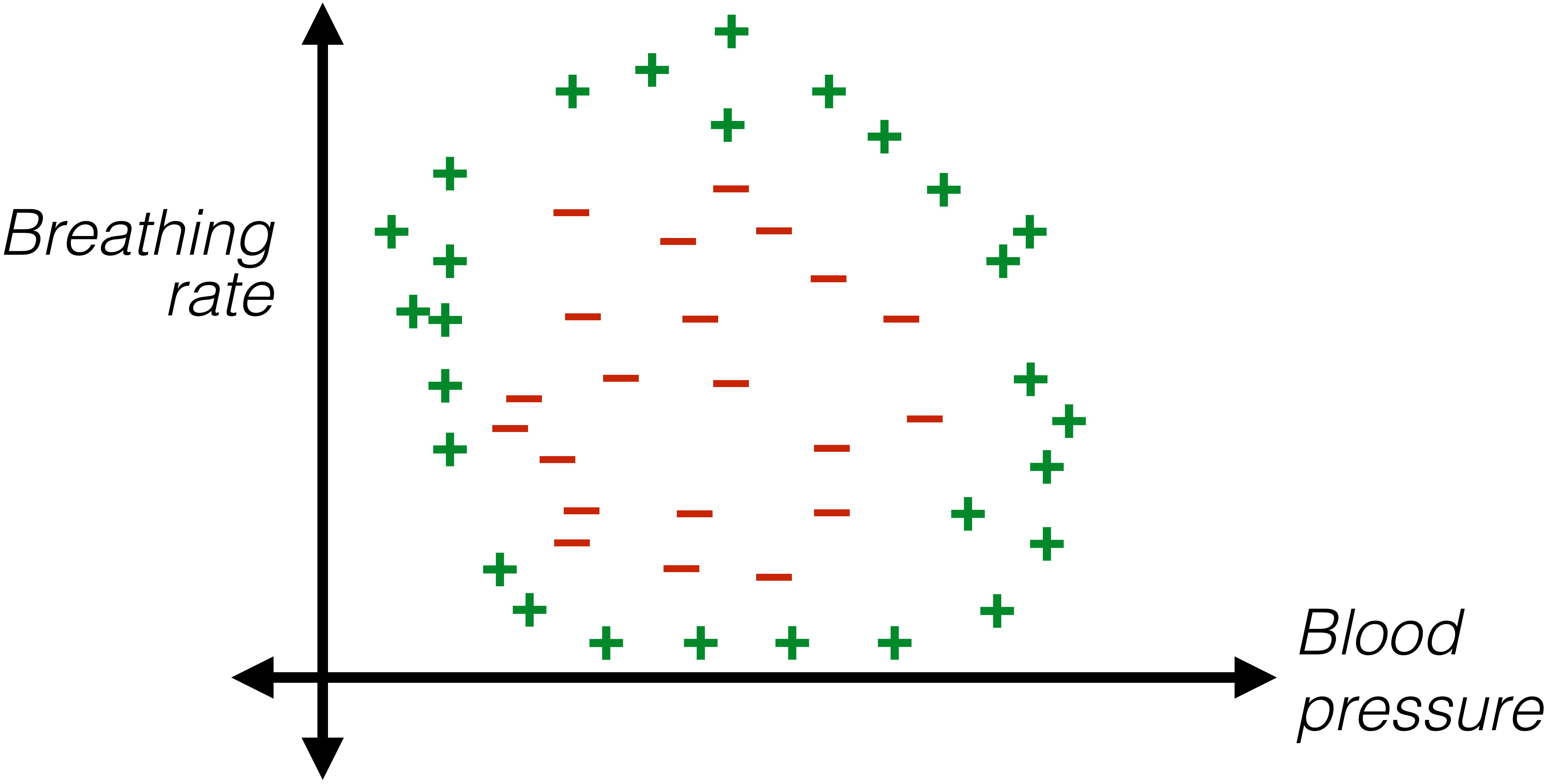
- Identify the features and encode as real numbers
- Standardize numerical features

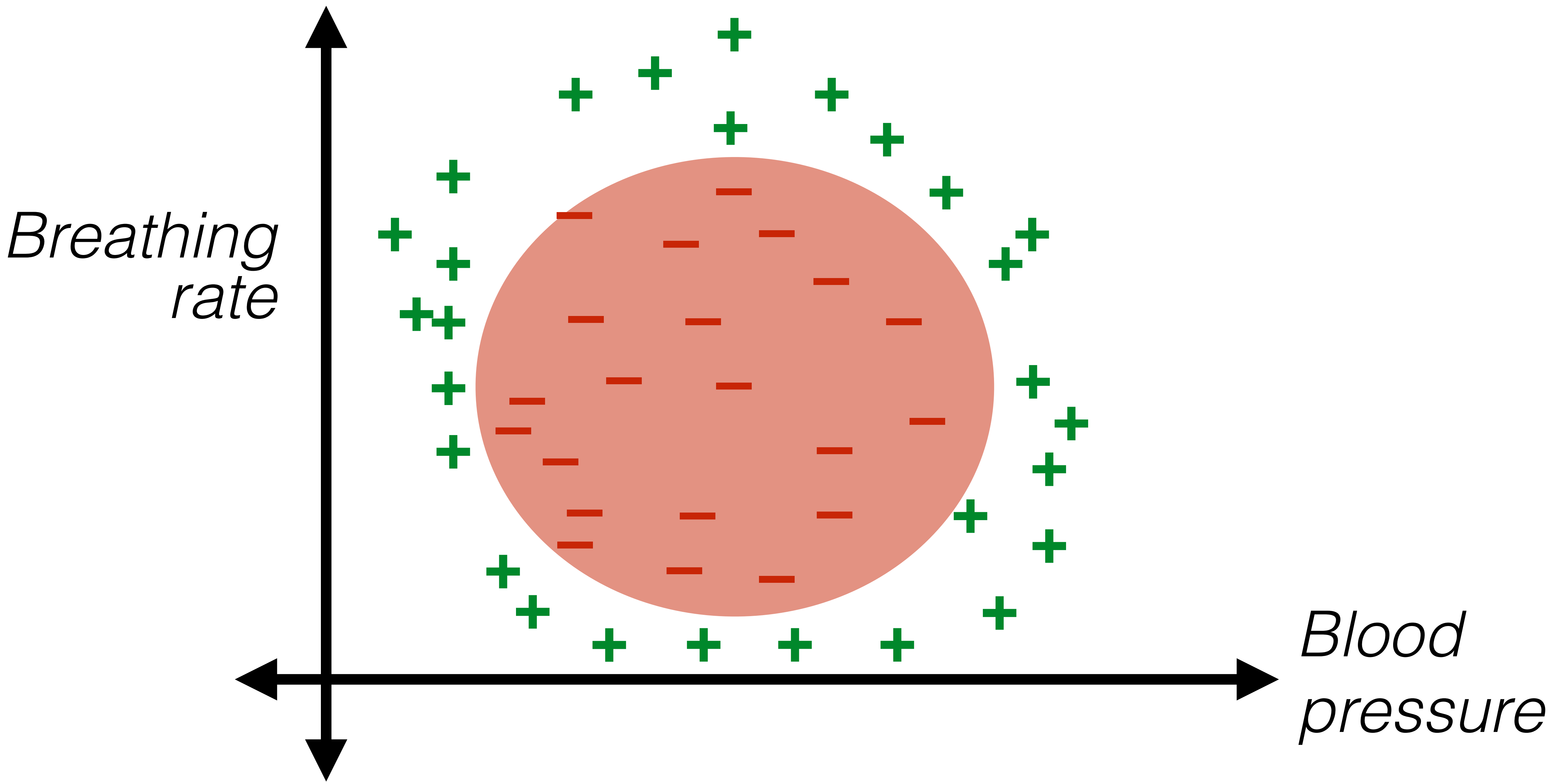
	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	decade	family income (USD)
1	55	0	1,0,0,0,0	1,0	4	133000
2	71	0	0,1,0,0,0	1,1	2	34000
3	89	1	1,0,0,0,0	0,1	5	40000
4	67	0	0,0,0,1,0	0,0	5	120000

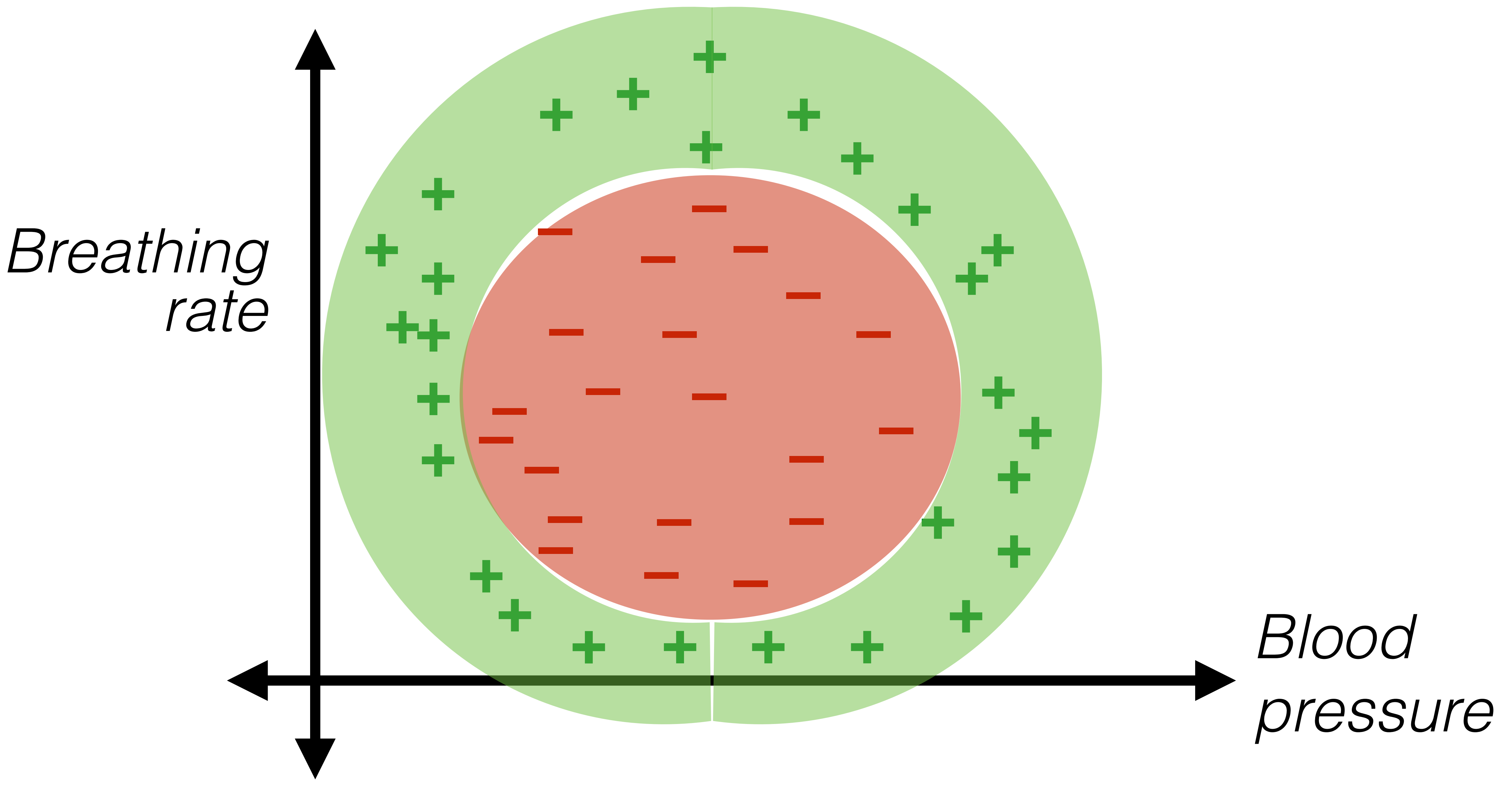
Encode data in usable form

- Identify the features and encode as real numbers
- Standardize numerical features

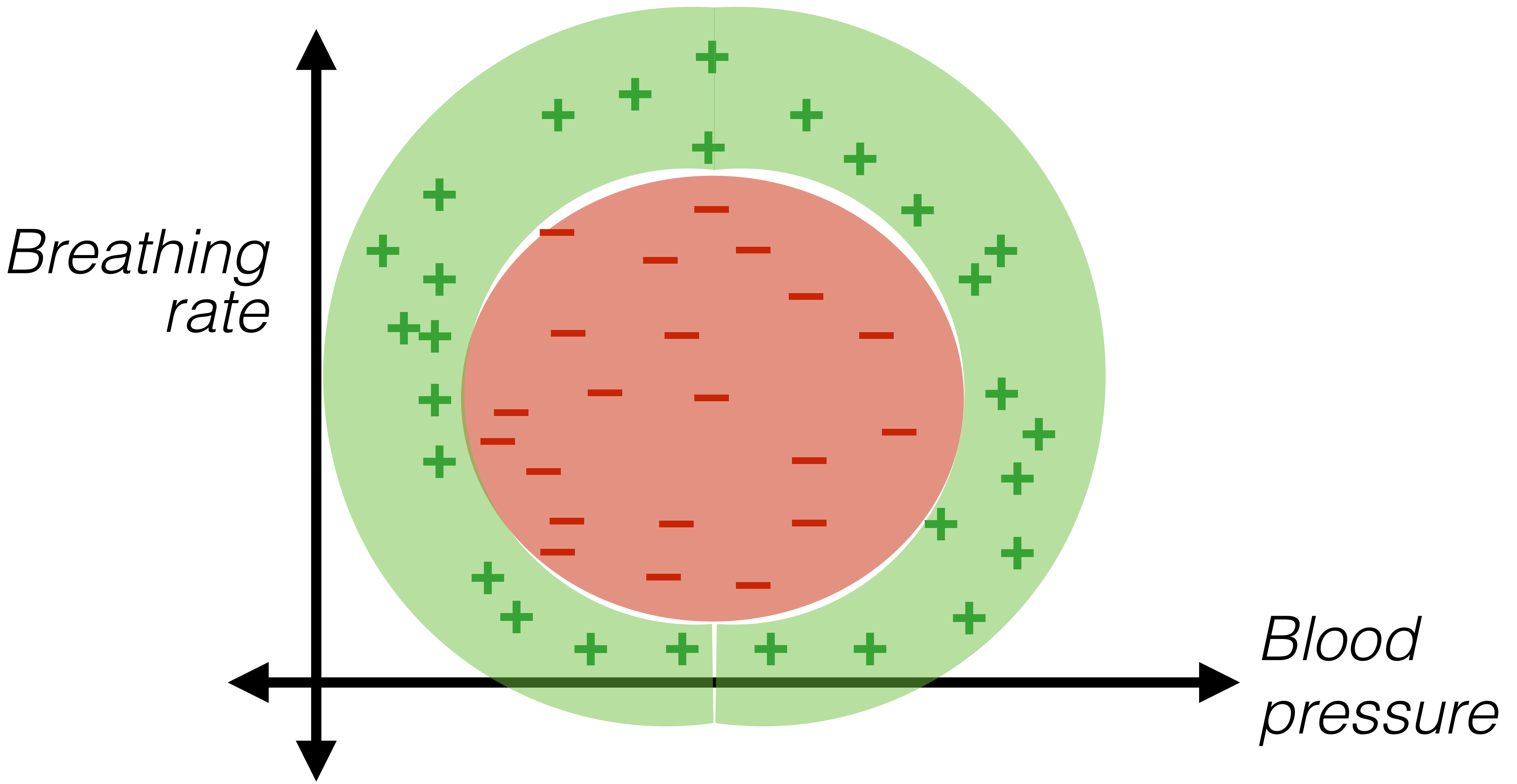
	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	decade	family income (USD)
1	-1.5	0	1,0,0,0,0	1,0	1	2.075
2	0.1	0	0,1,0,0,0	1,1	-1	-0.4
3	1.9	1	1,0,0,0,0	0,1	2	-0.25
4	-0.3	0	0,0,0,1,0	0,0	2	1.75


Breathing rate
Blood pressure


Breathing rate
Blood pressure

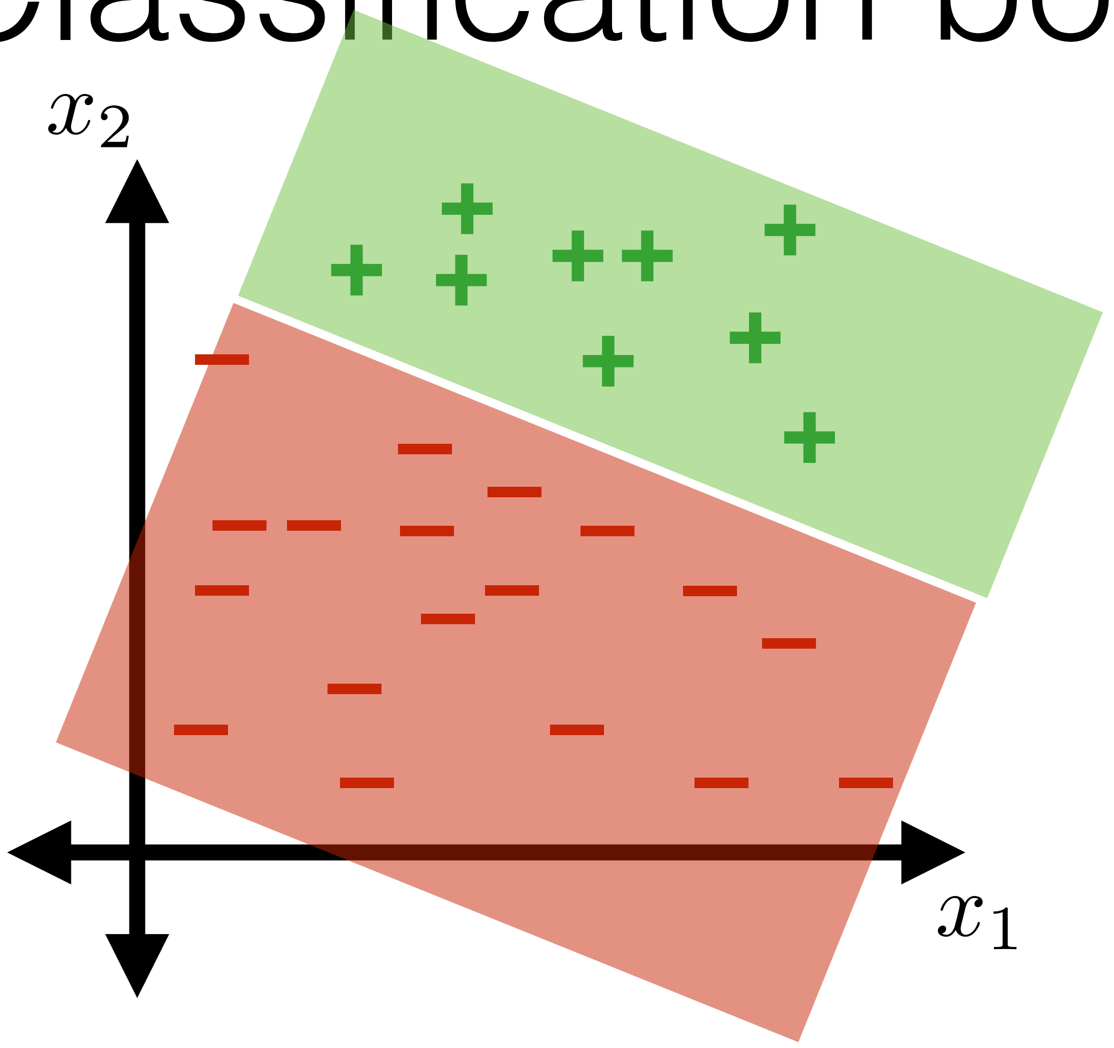

Breathing rate
Blood pressure

Nonlinear boundaries

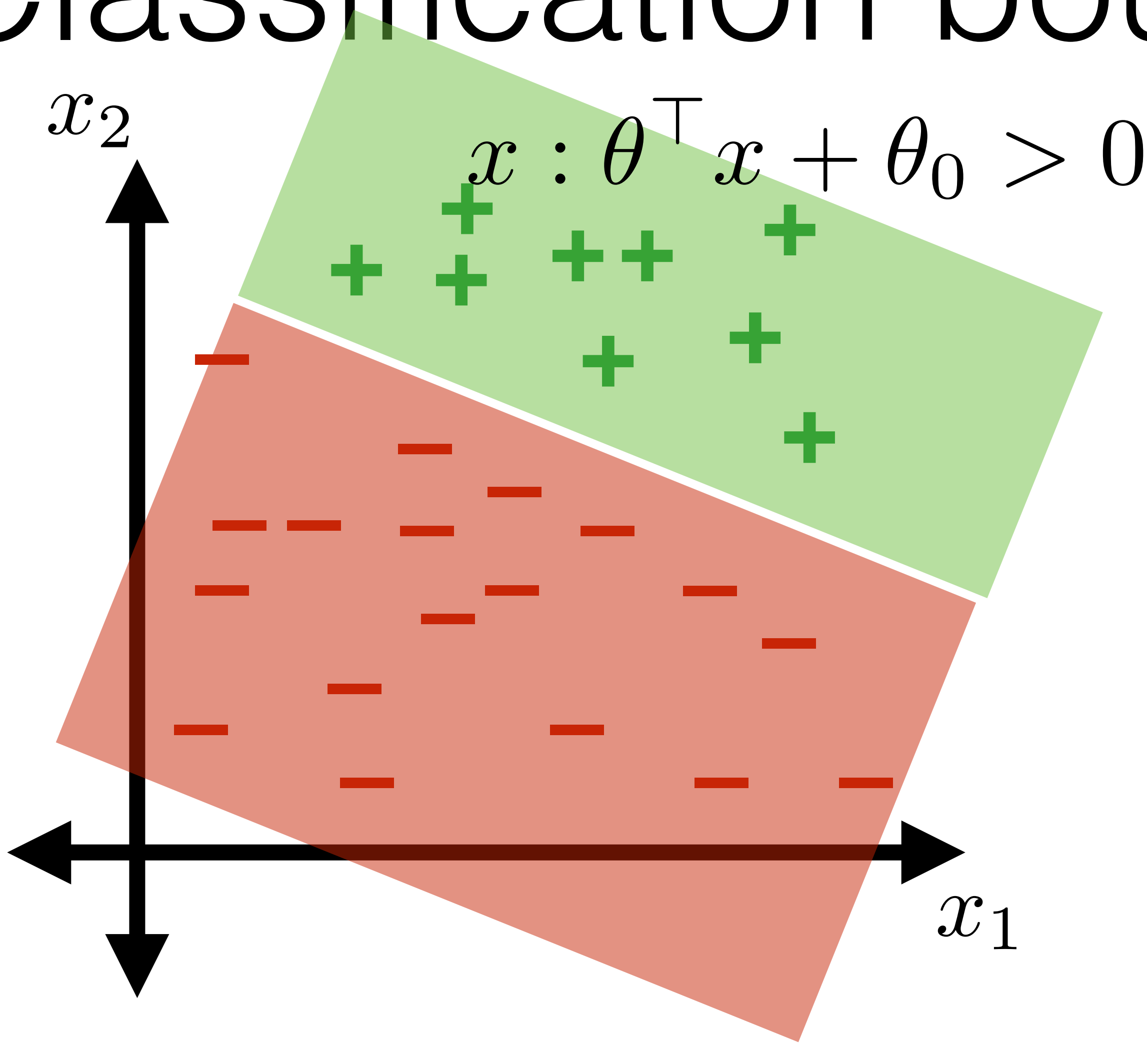


Classification boundaries

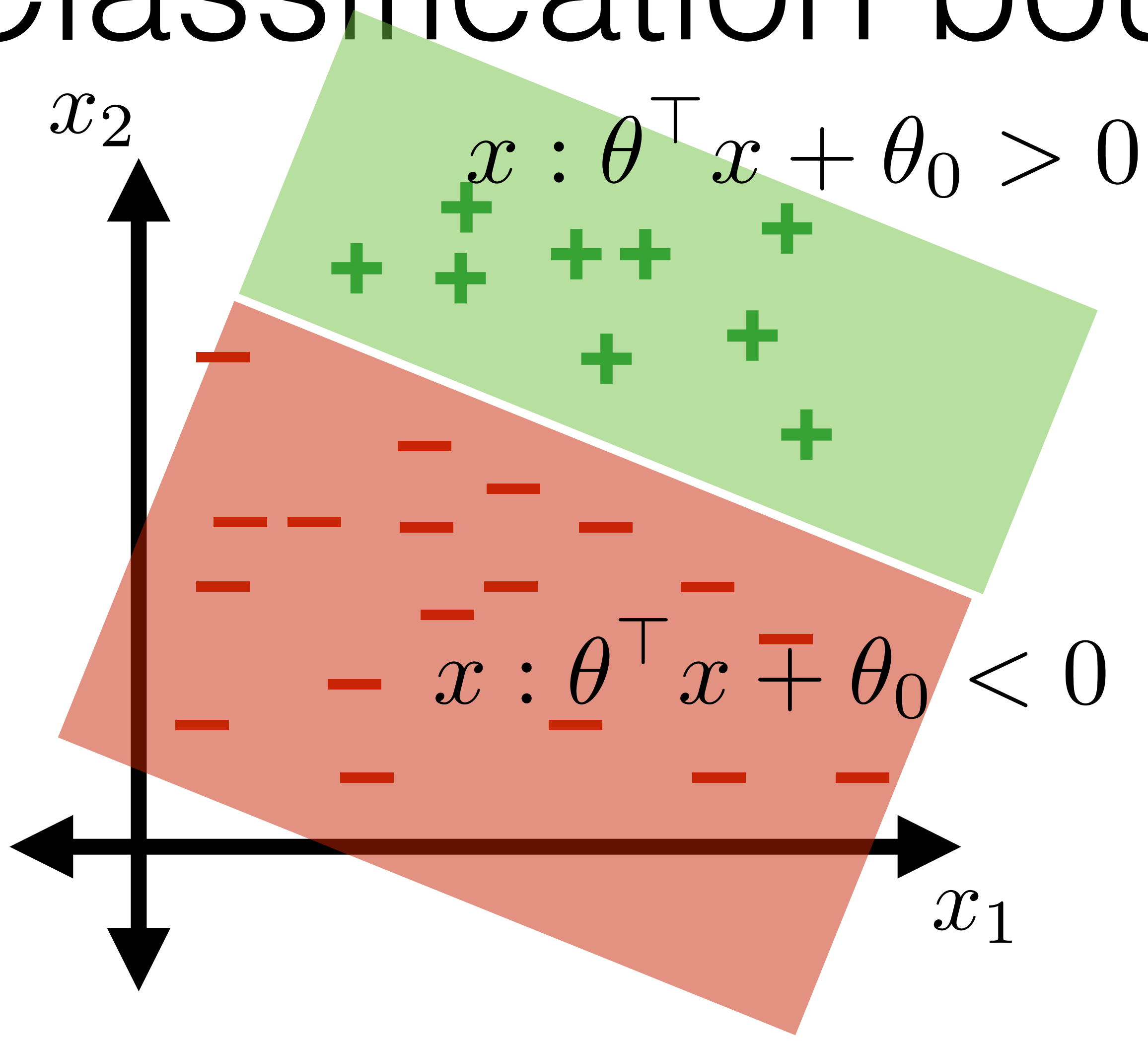
Classification boundaries



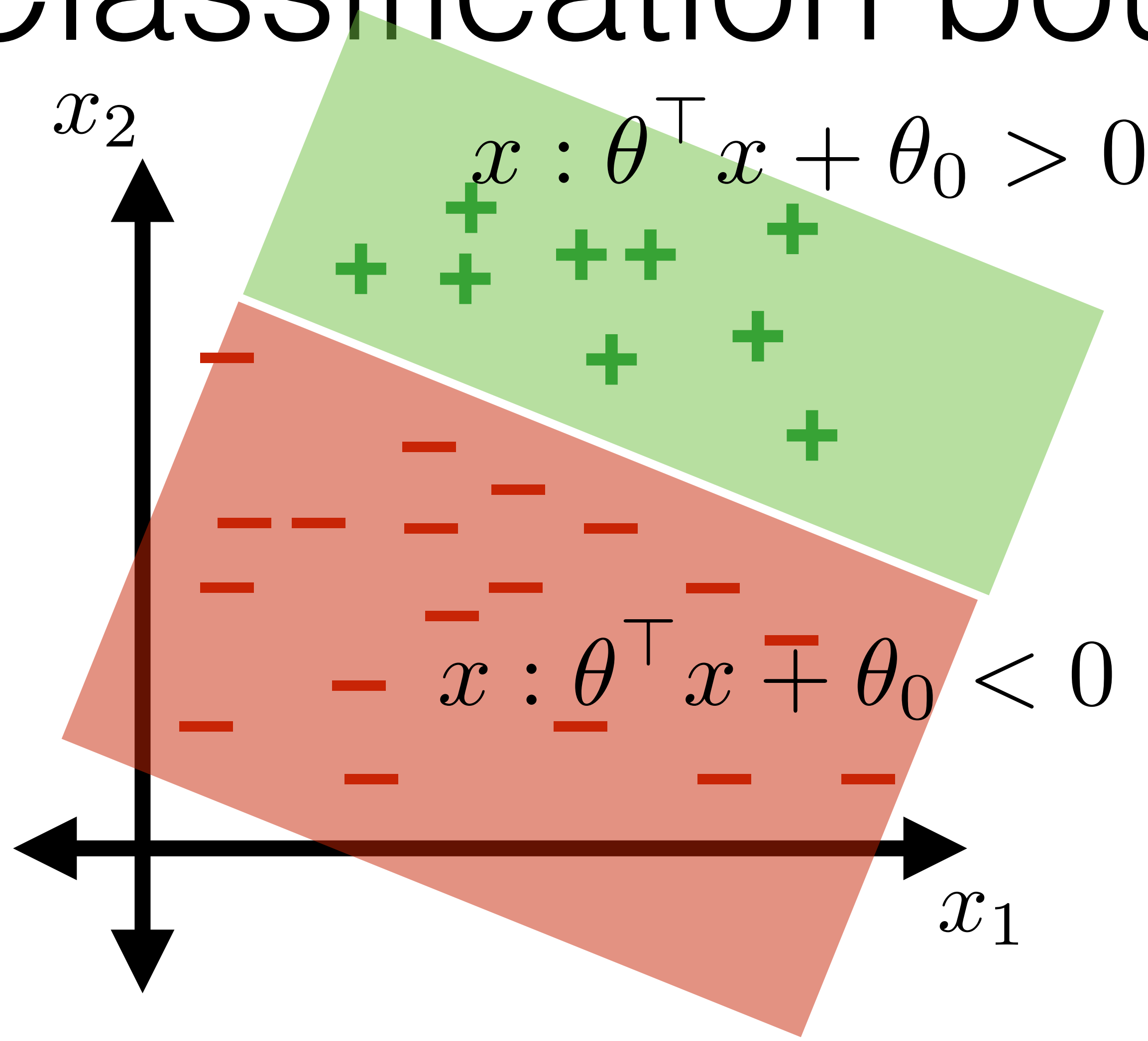
Classification boundaries

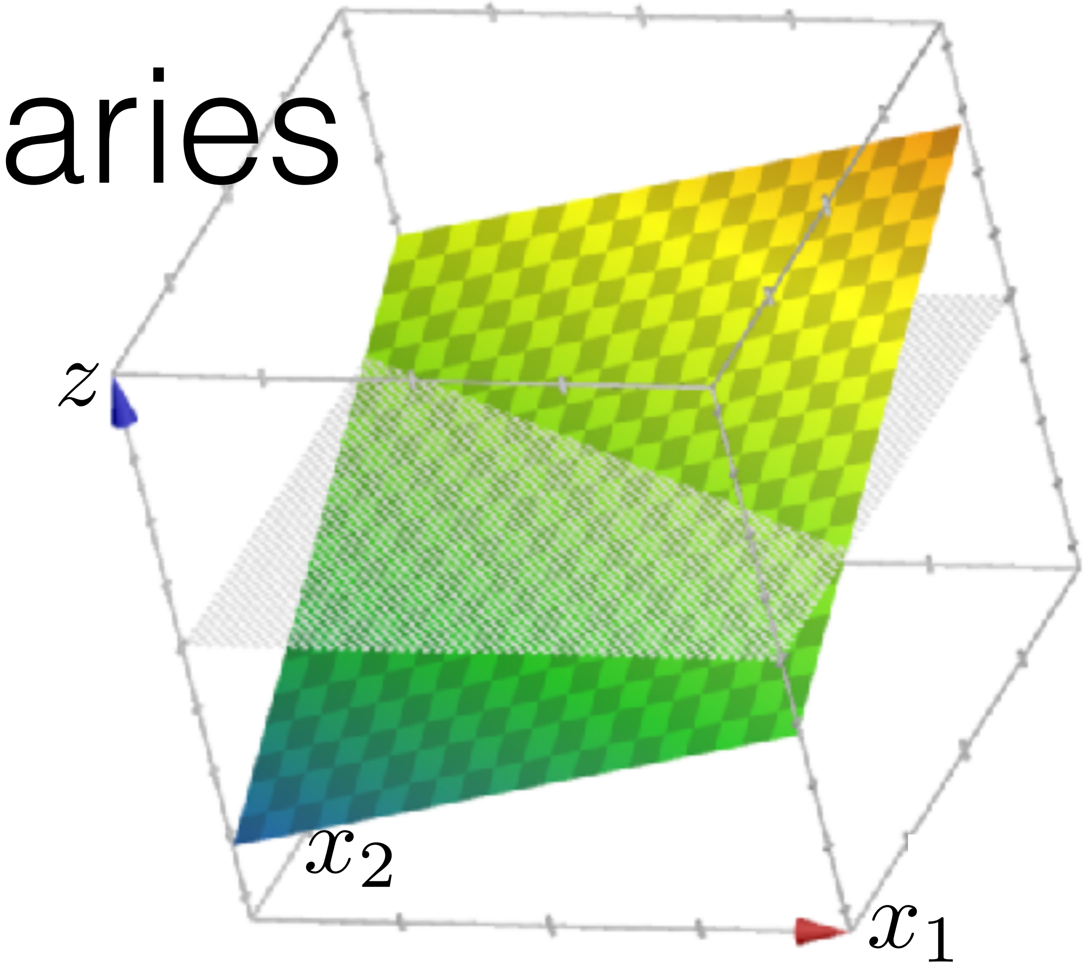


Classification boundaries

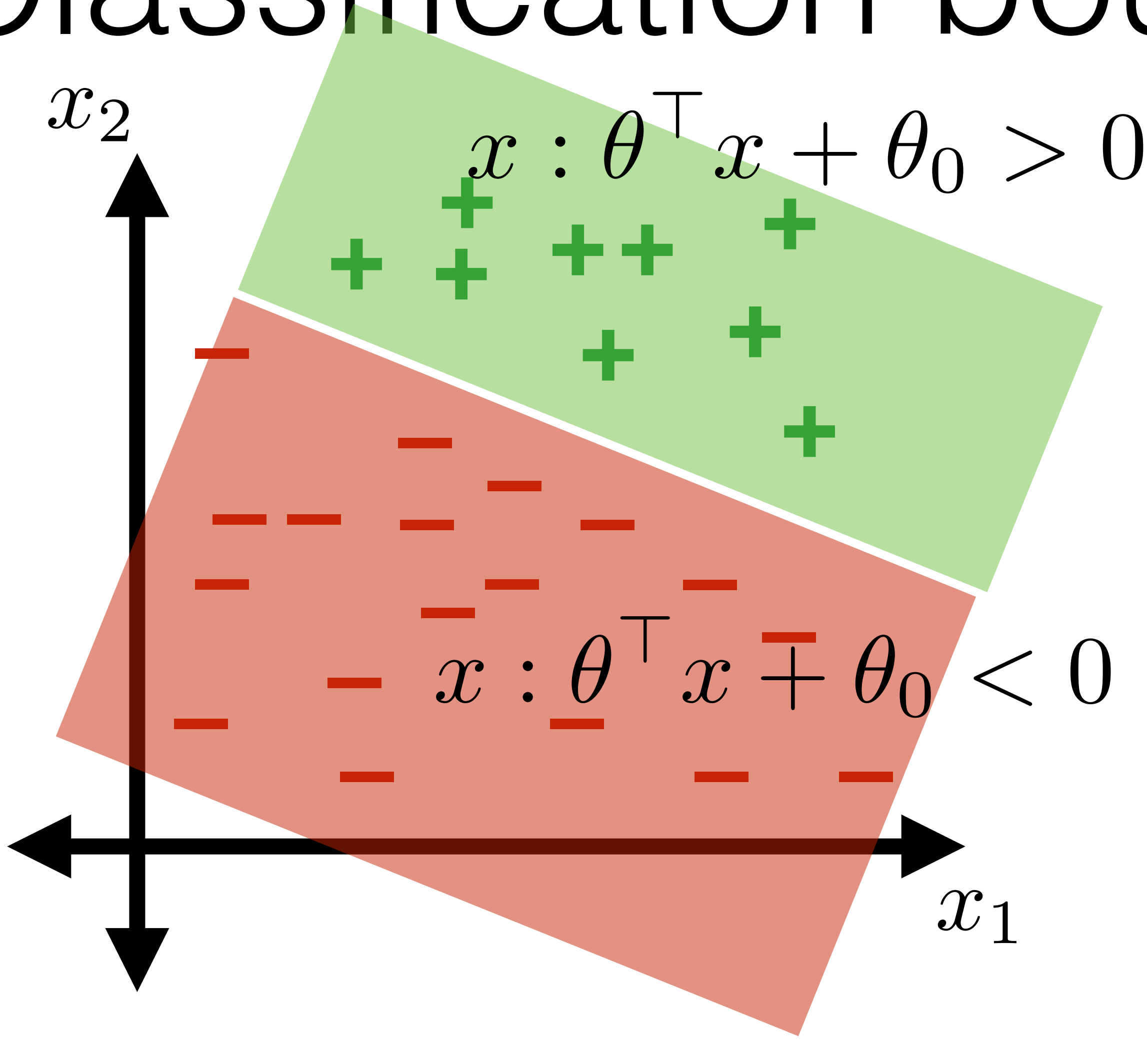


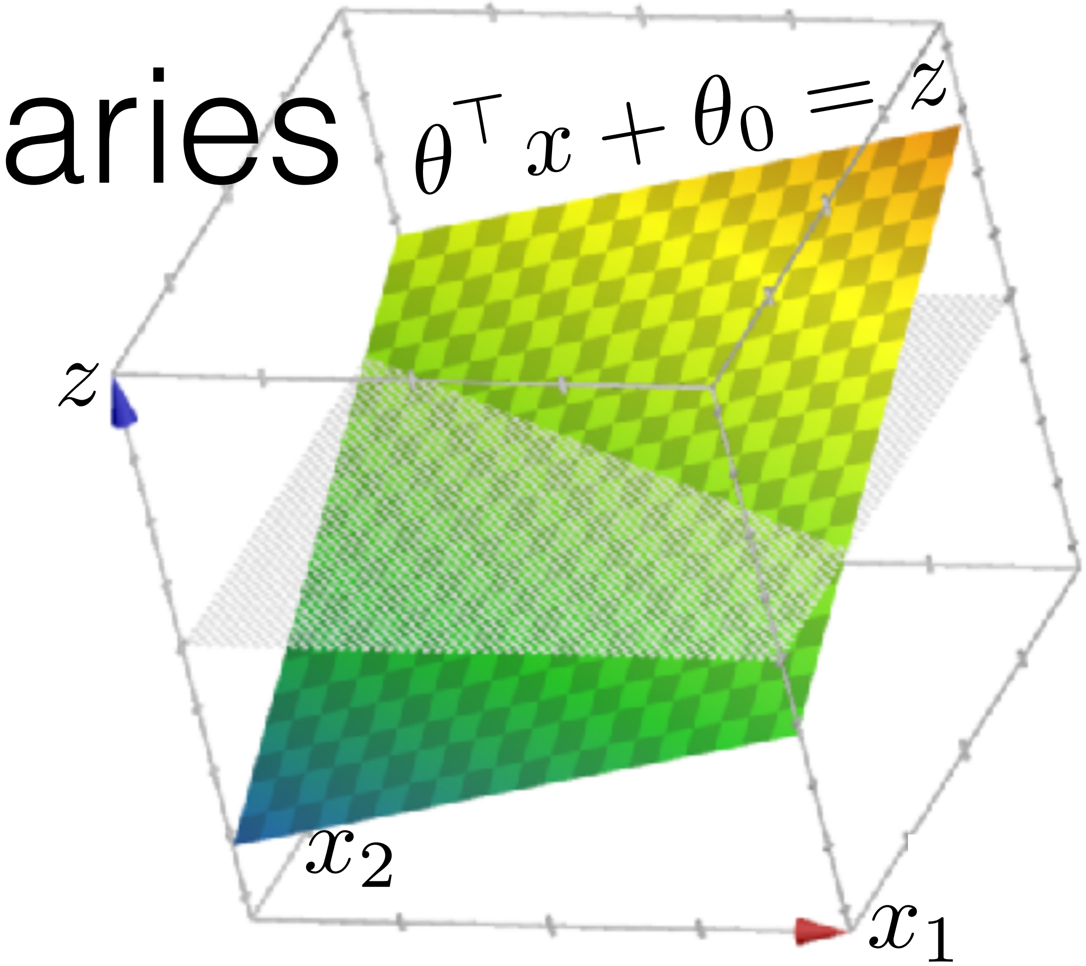
Classification boundaries



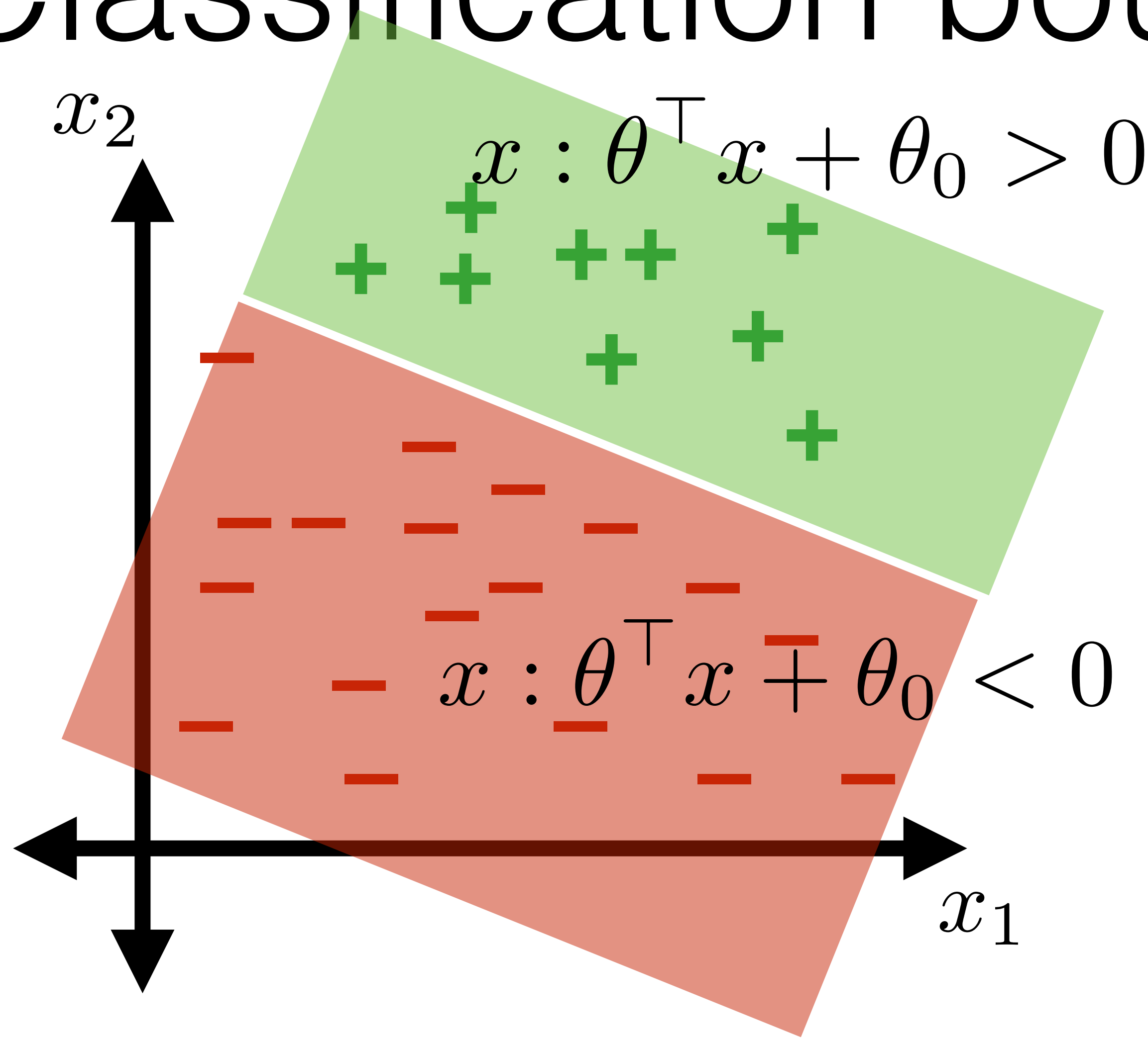


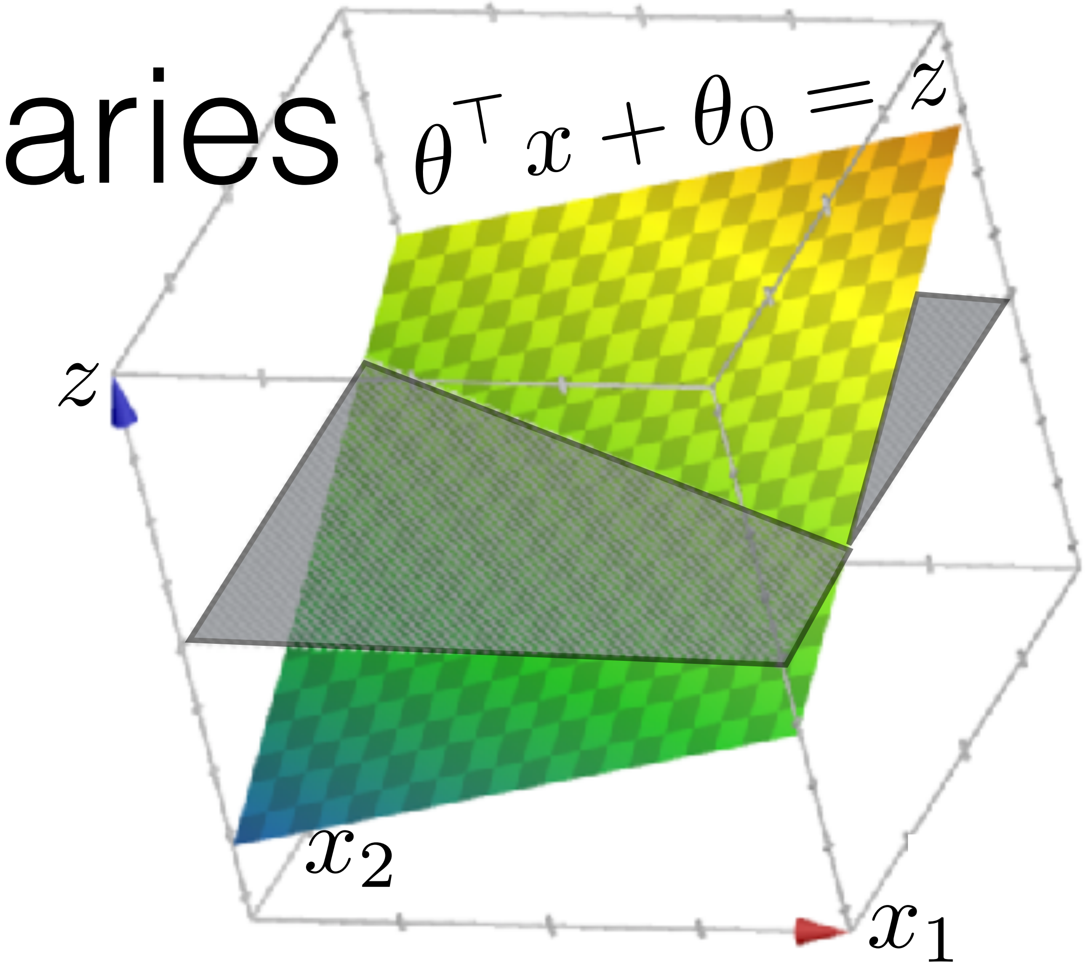
Classification boundaries



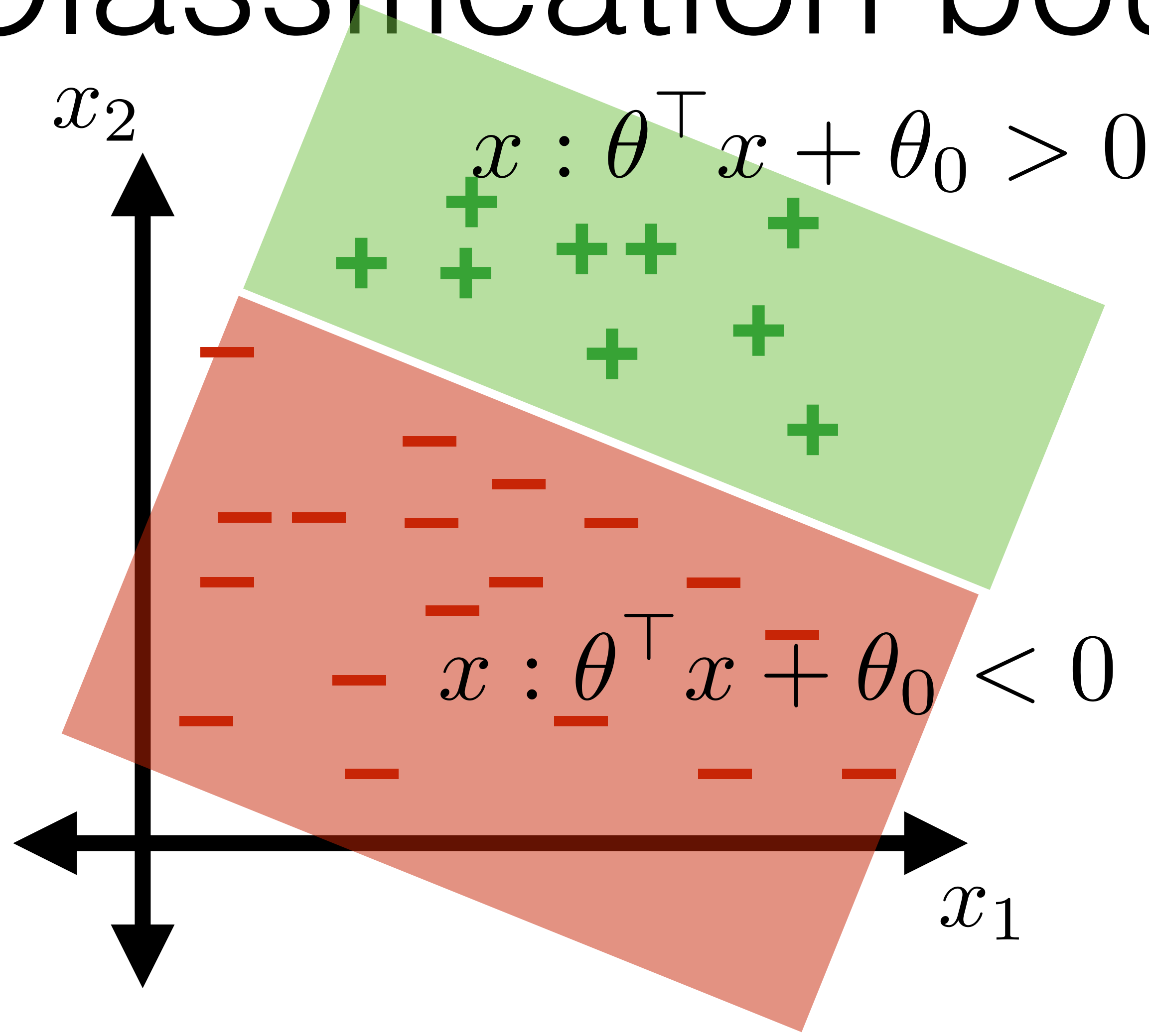


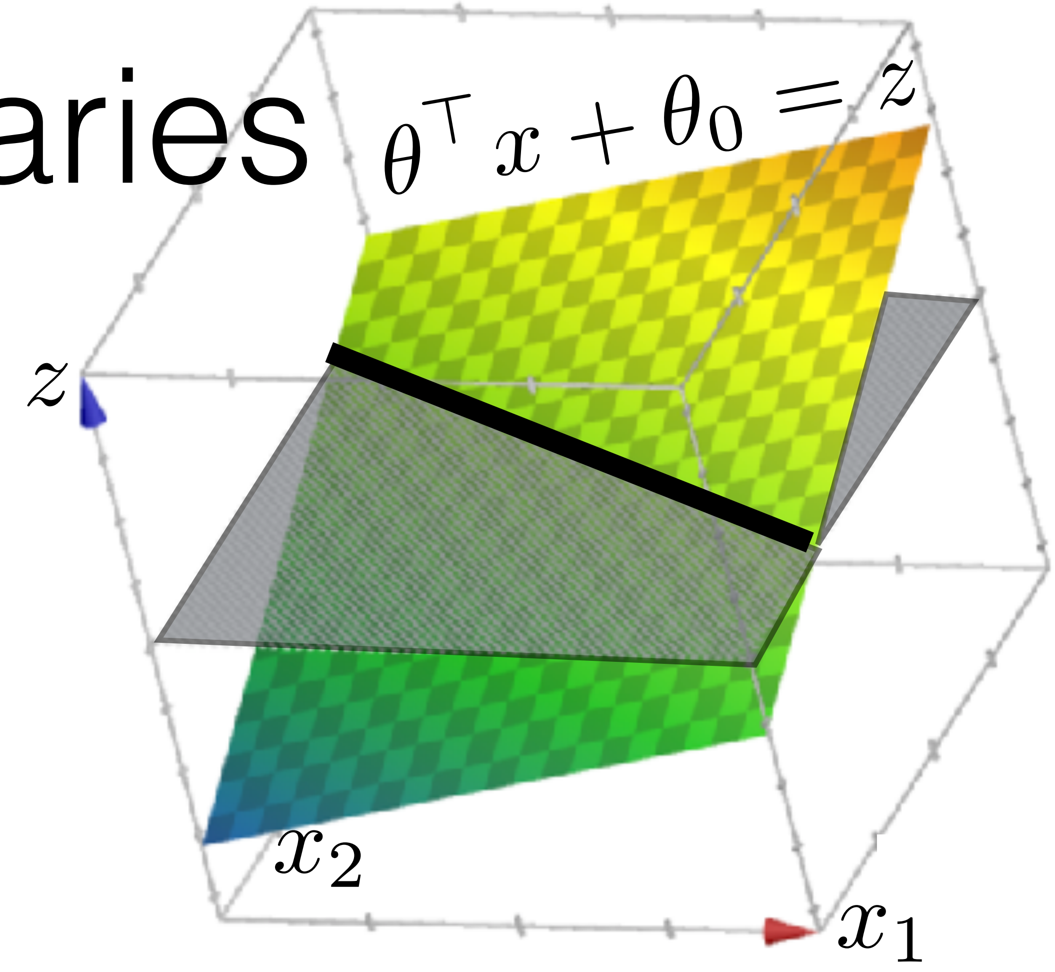
Classification boundaries



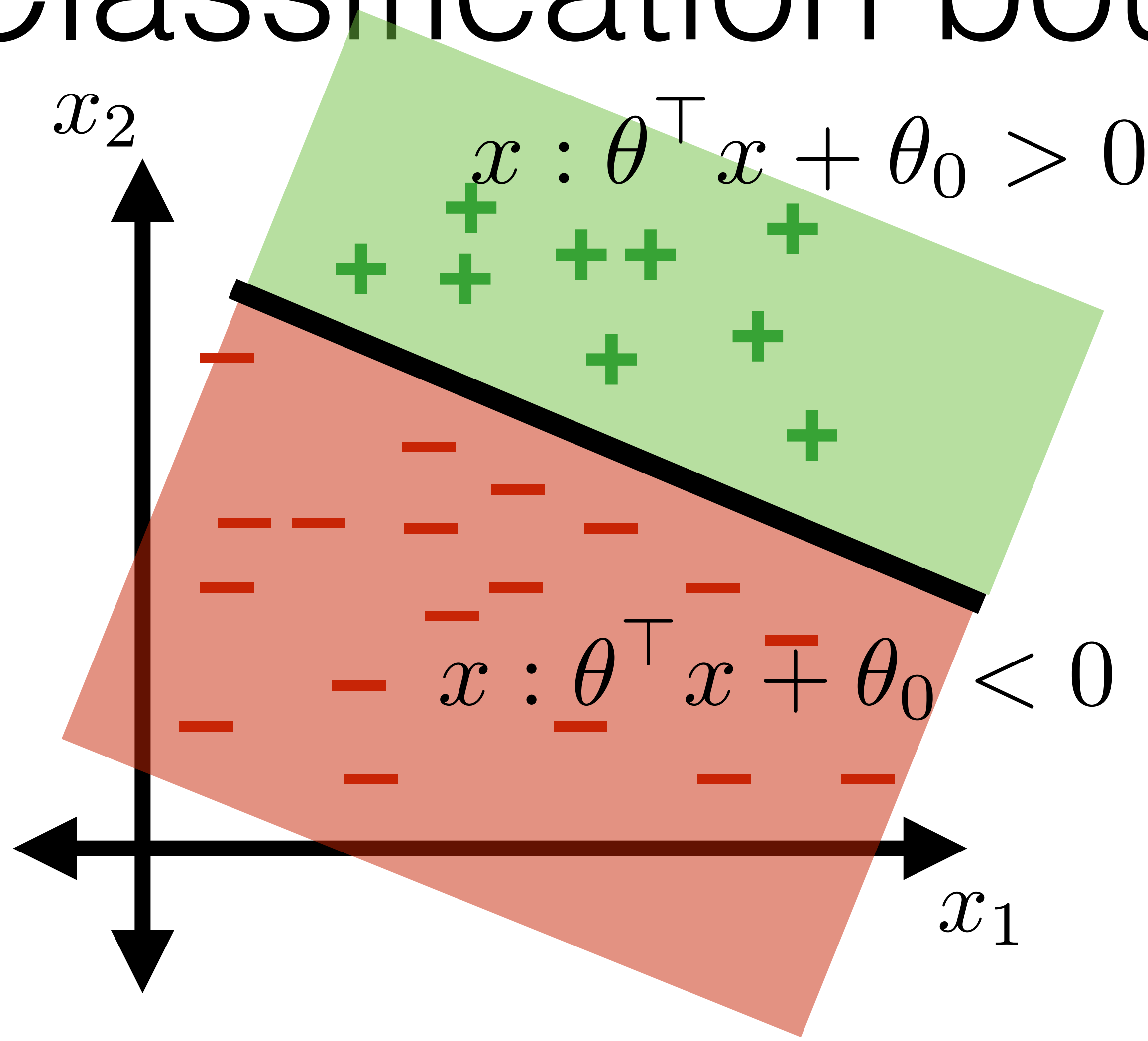


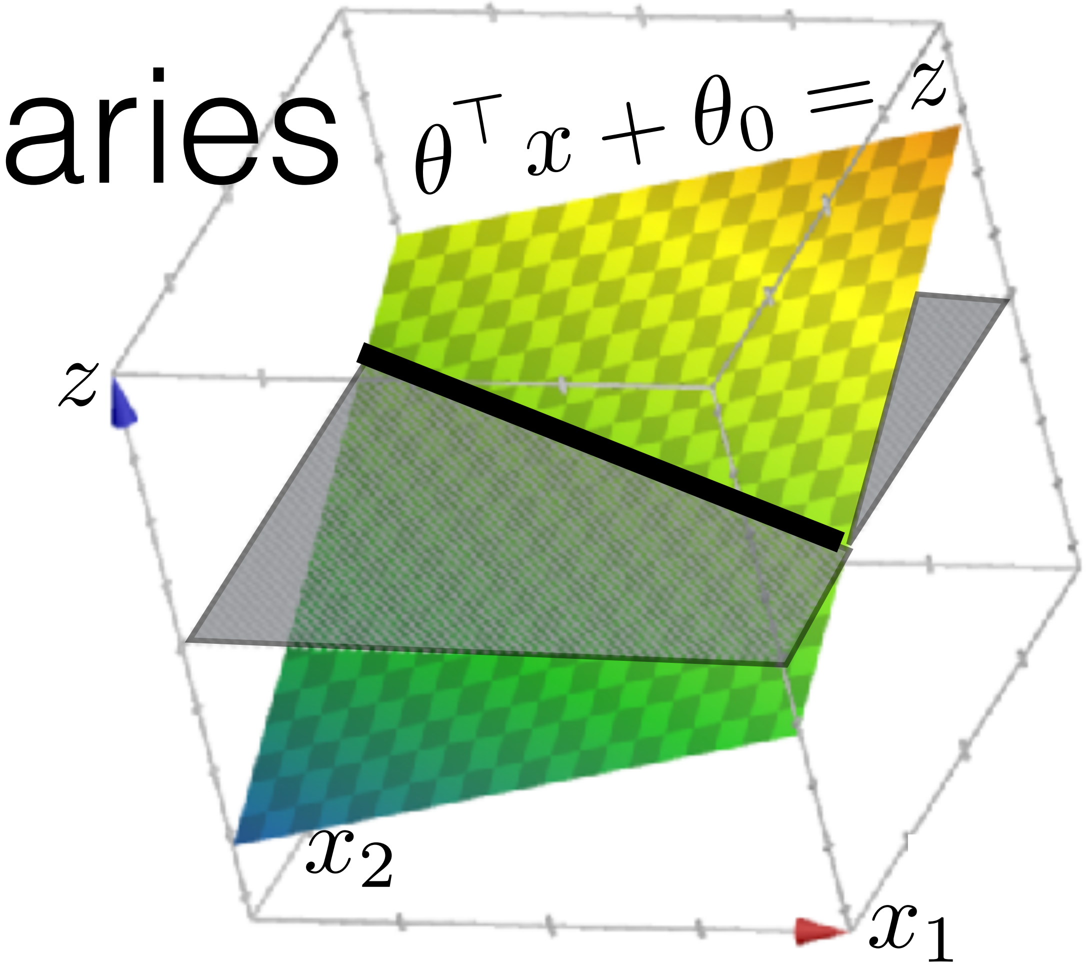
Classification boundaries



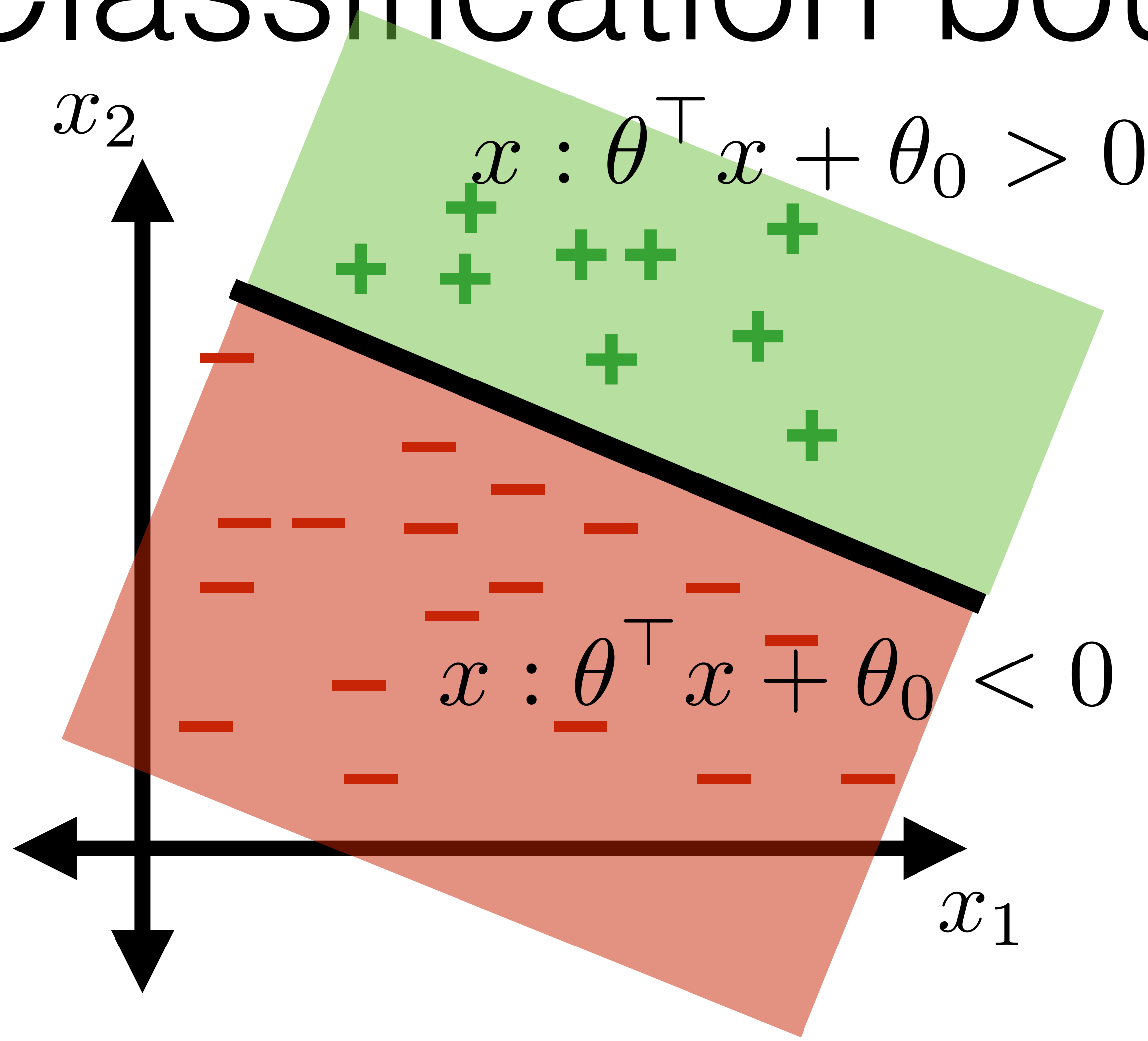


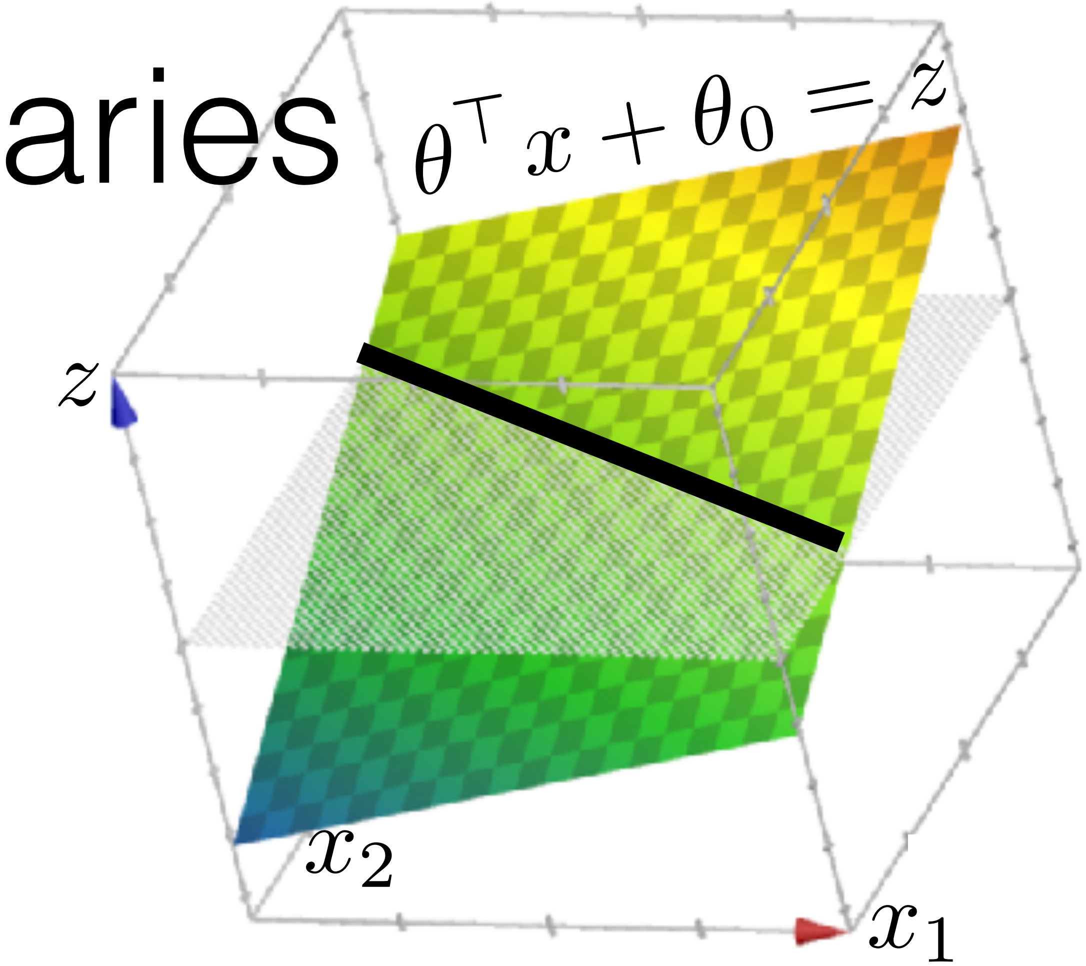
Classification boundaries



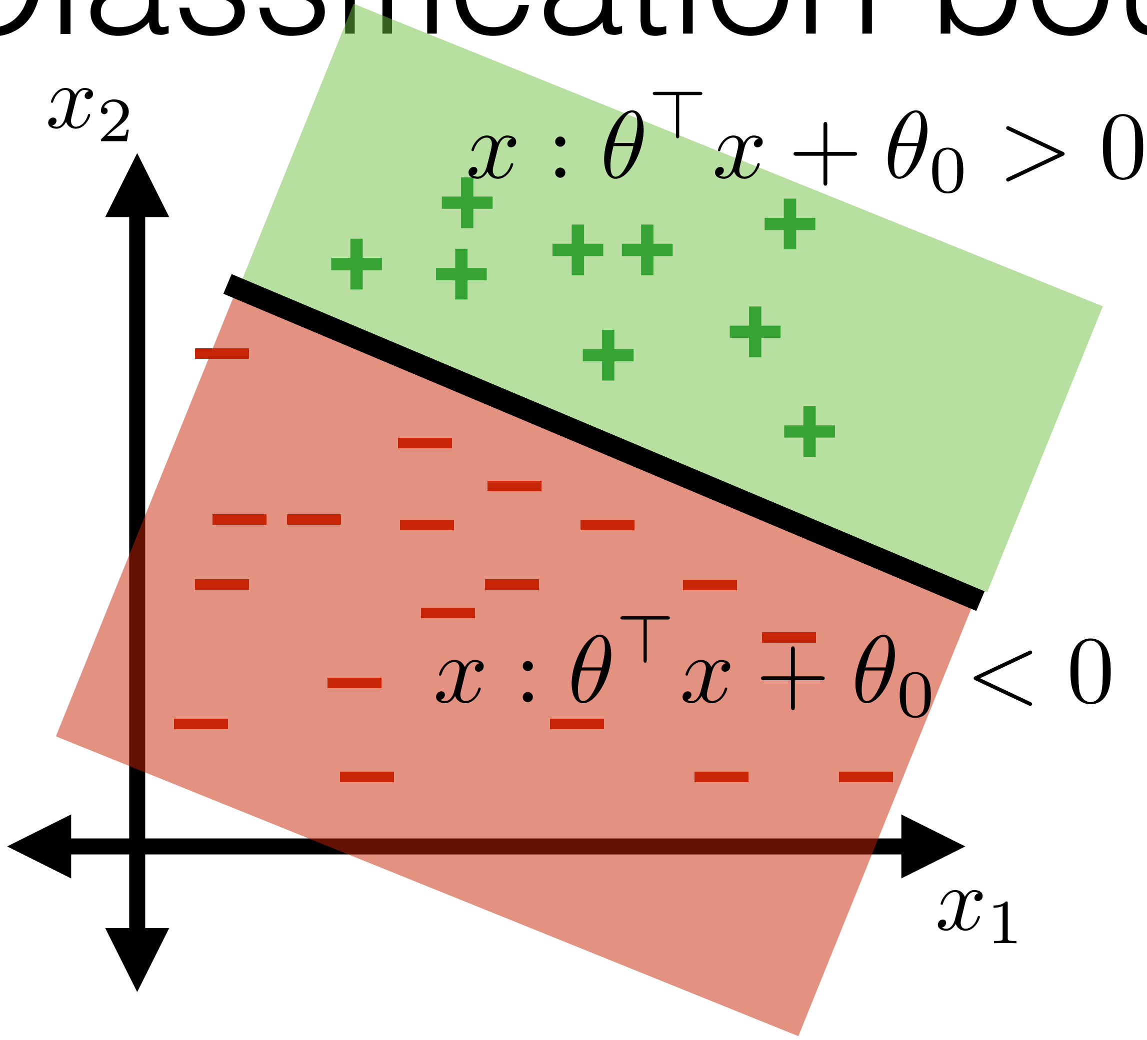


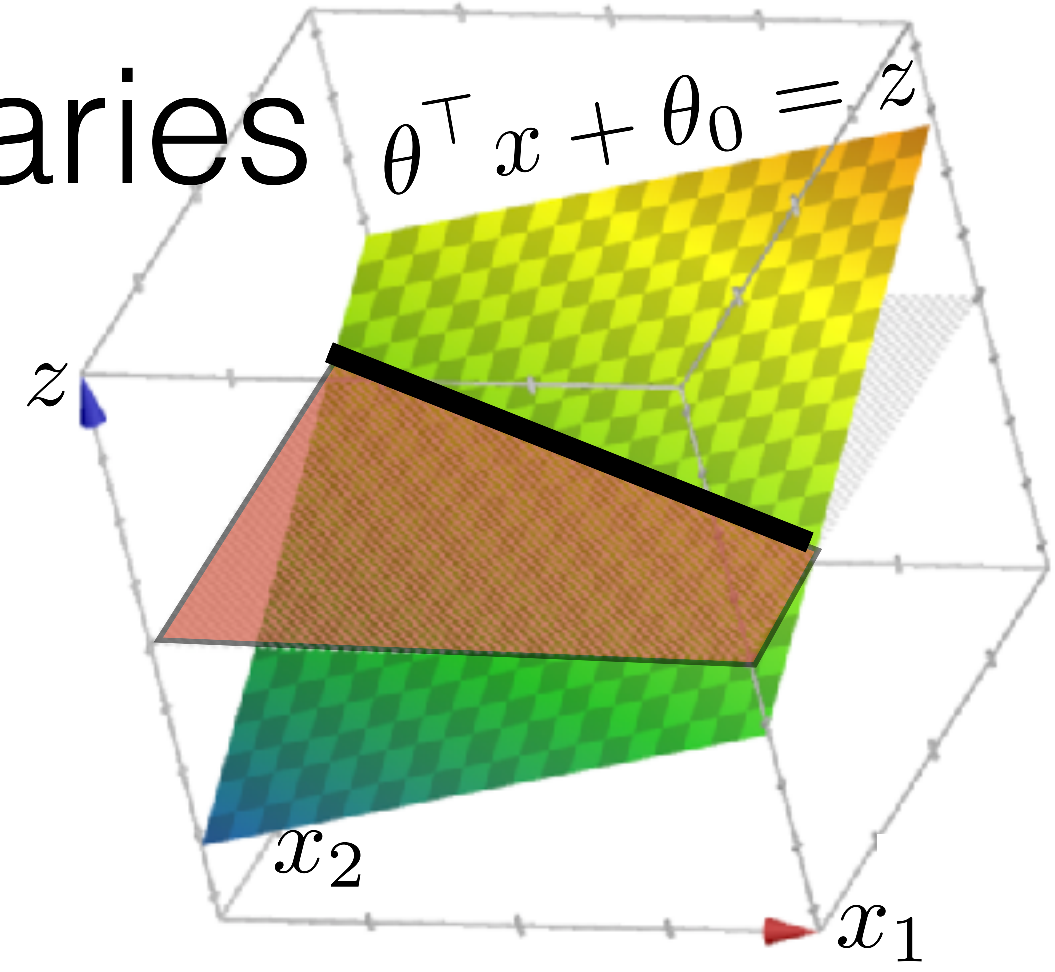
Classification boundaries



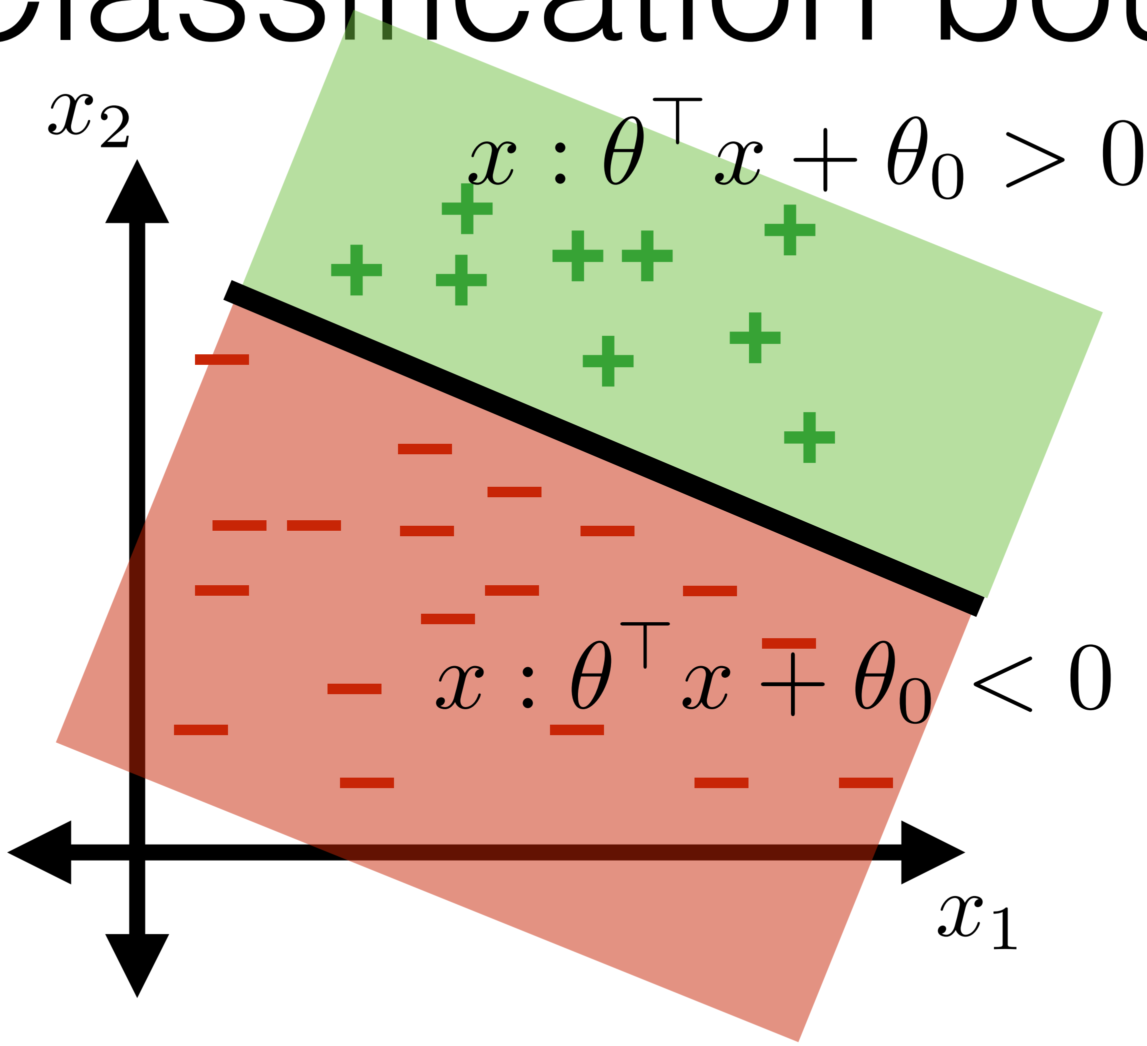


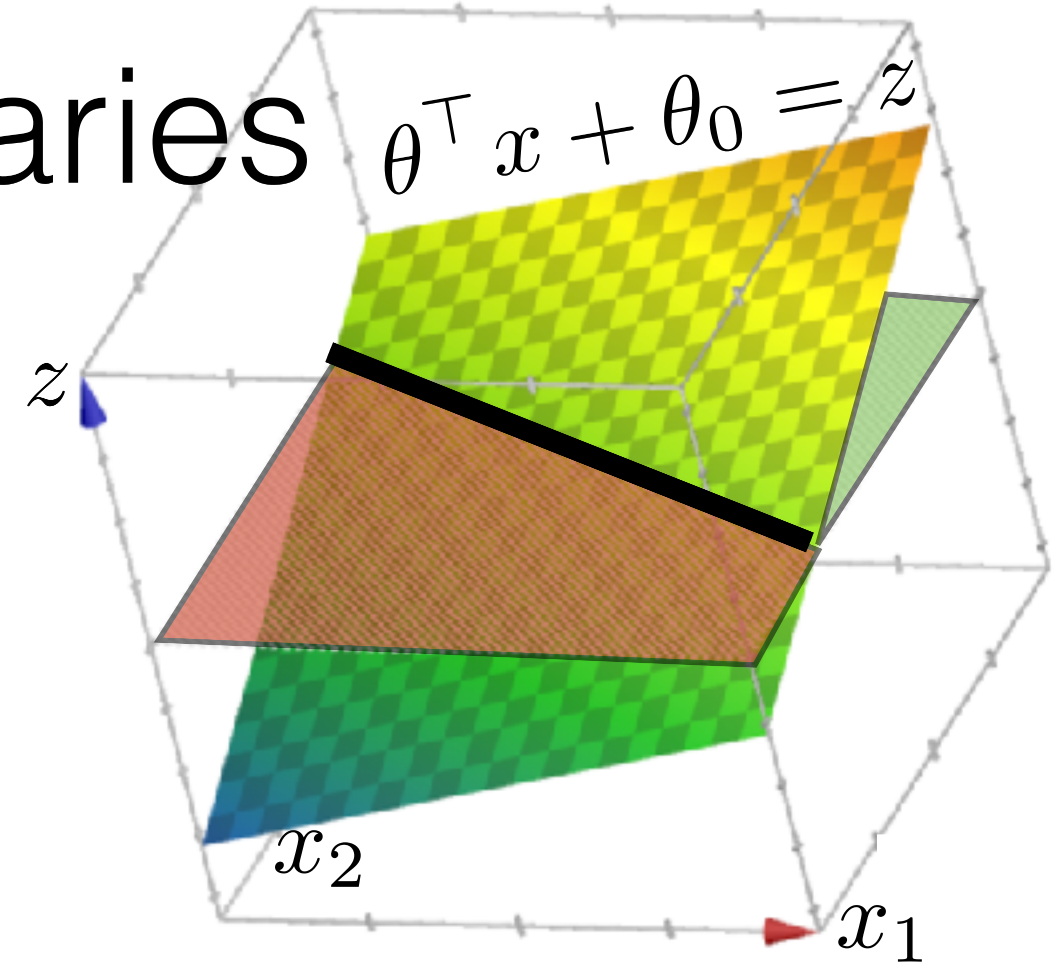
Classification boundaries



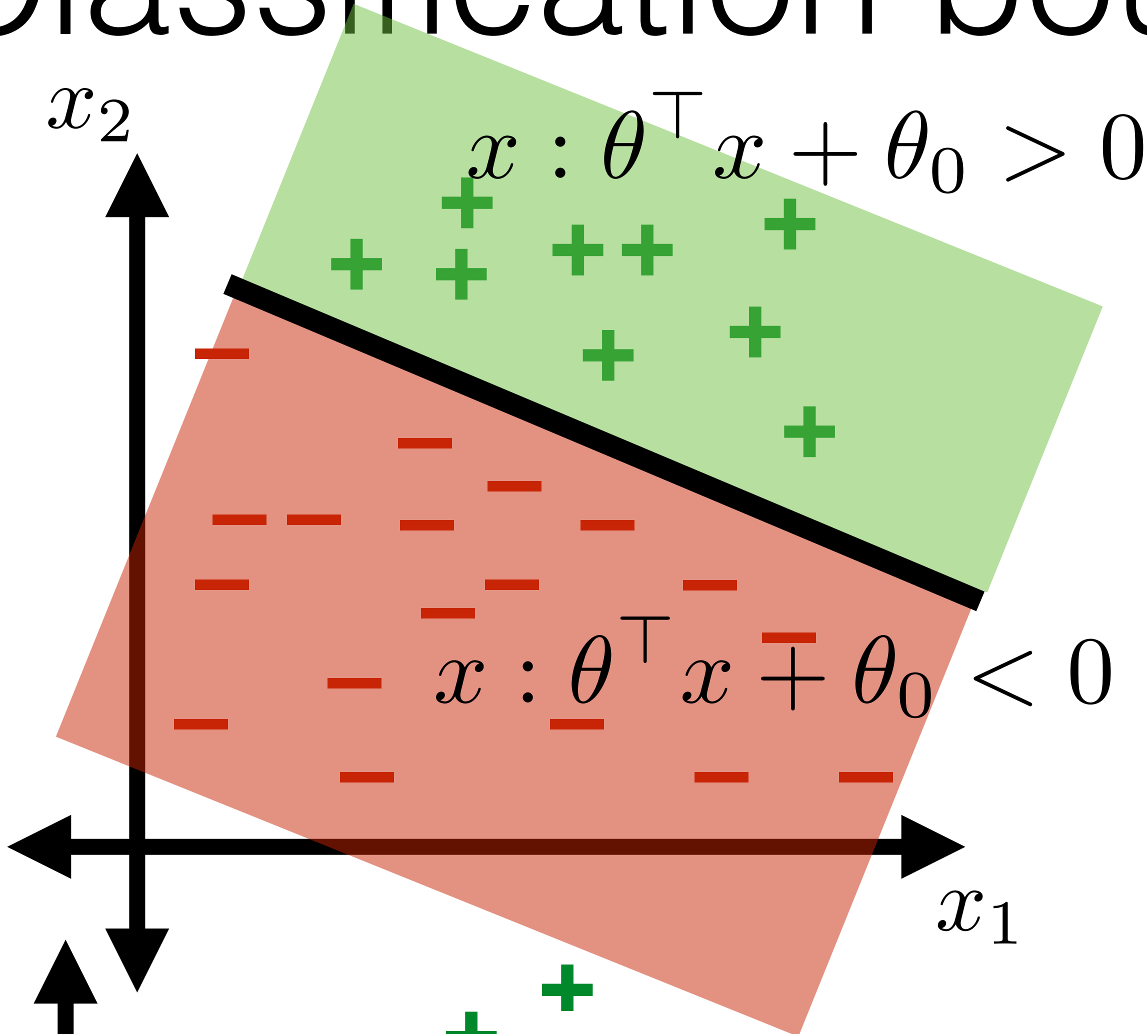


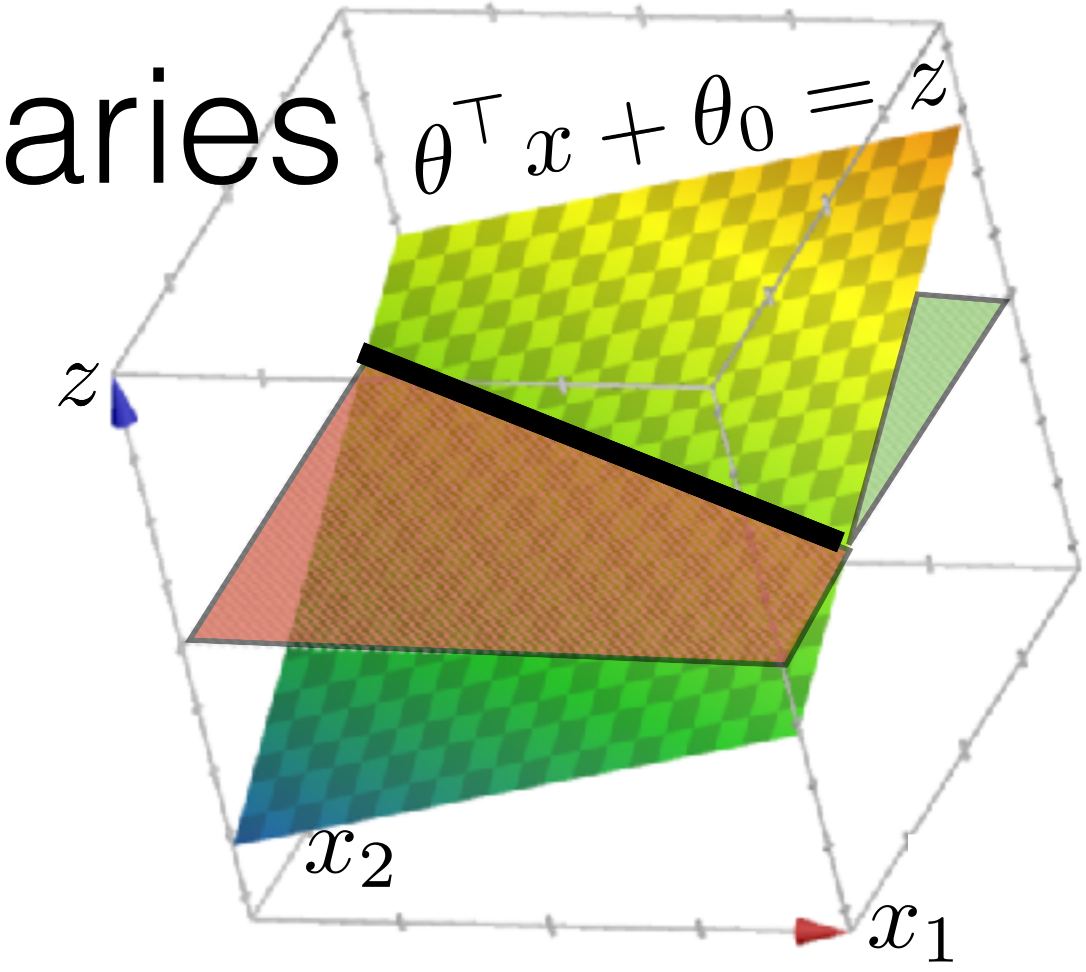
Classification boundaries

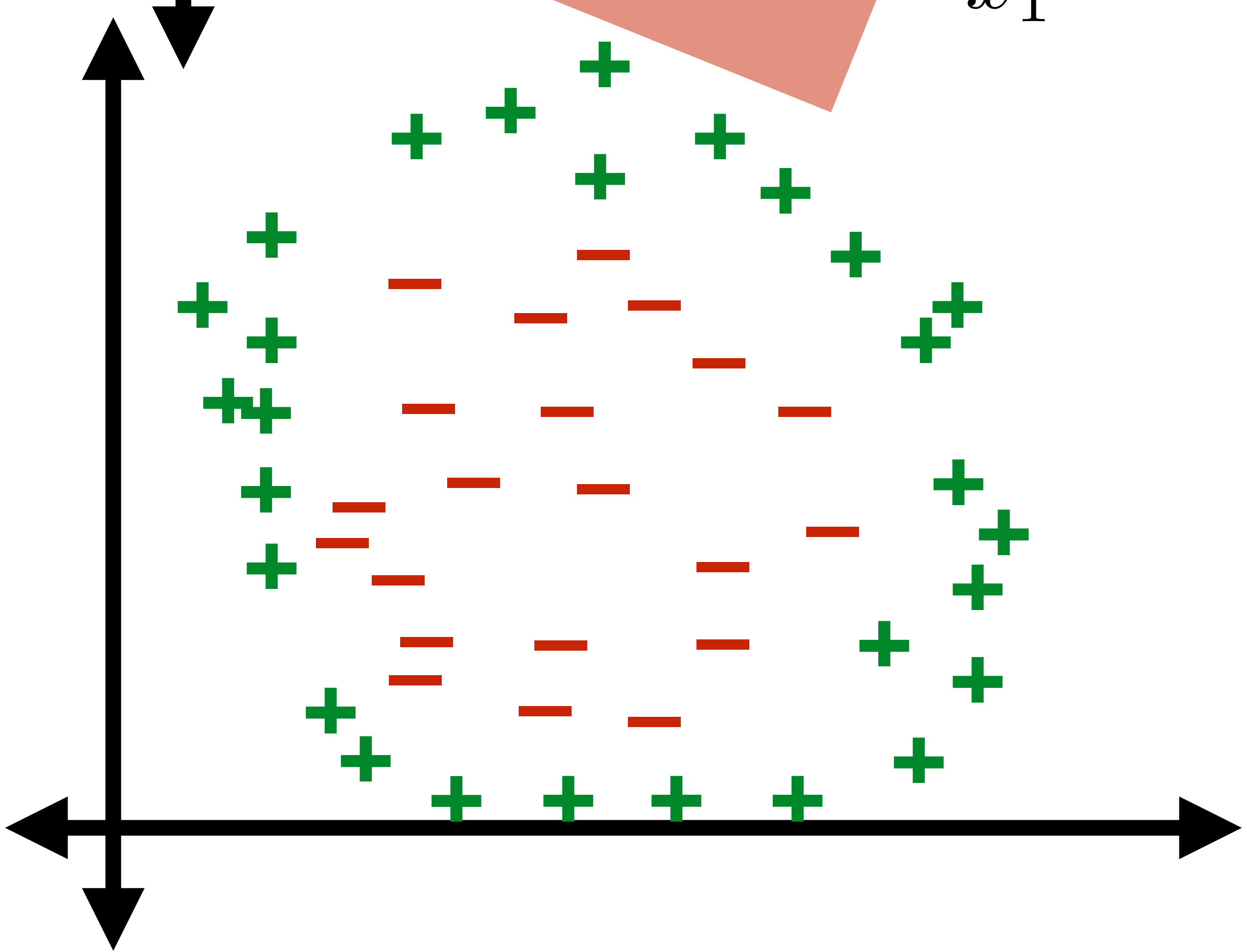




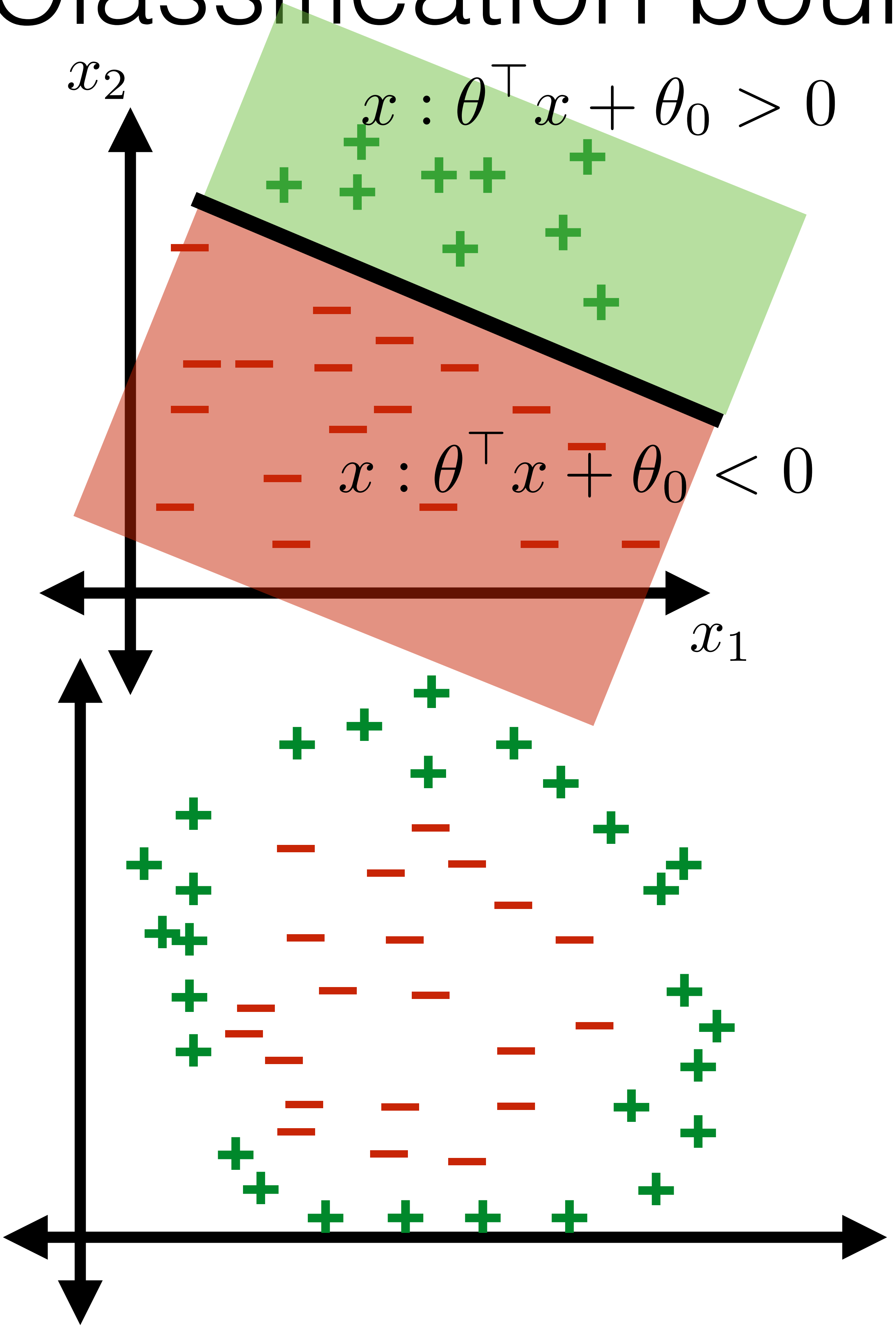
Classification boundaries

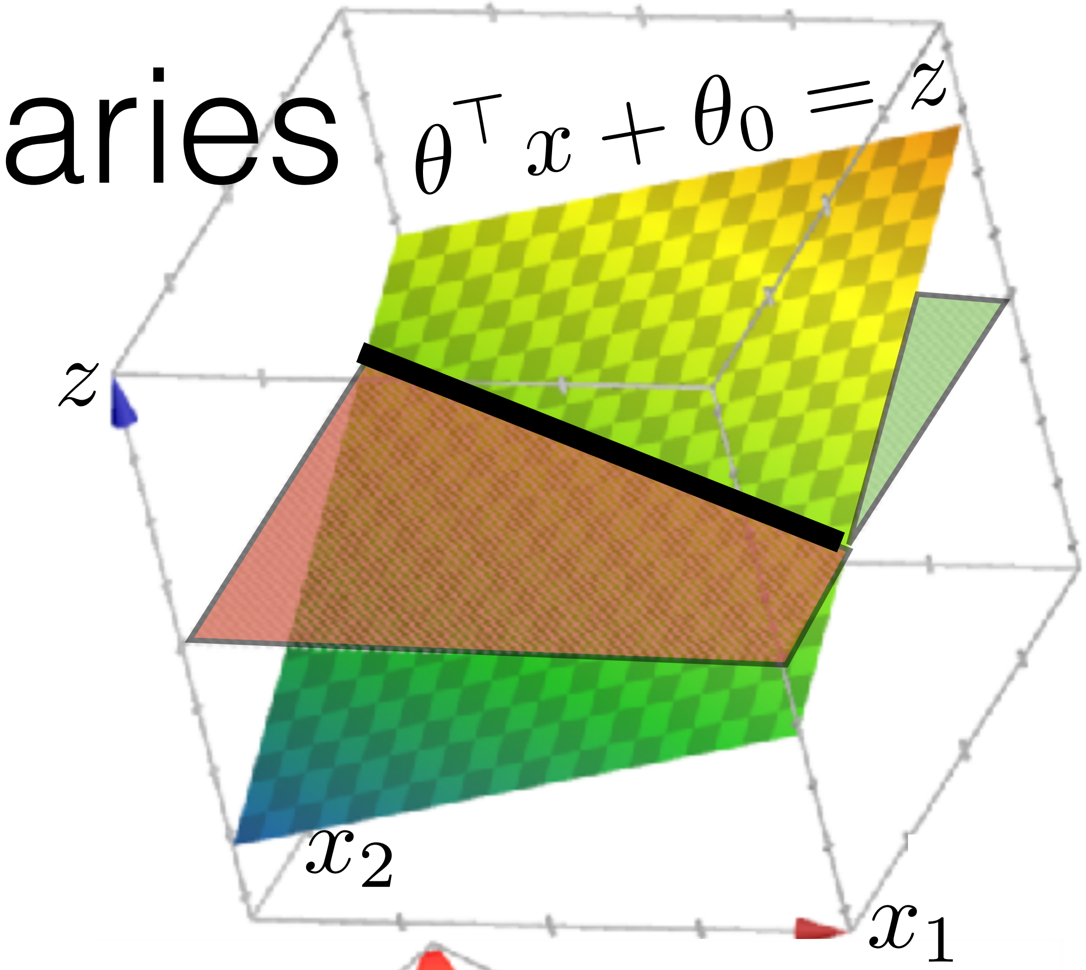


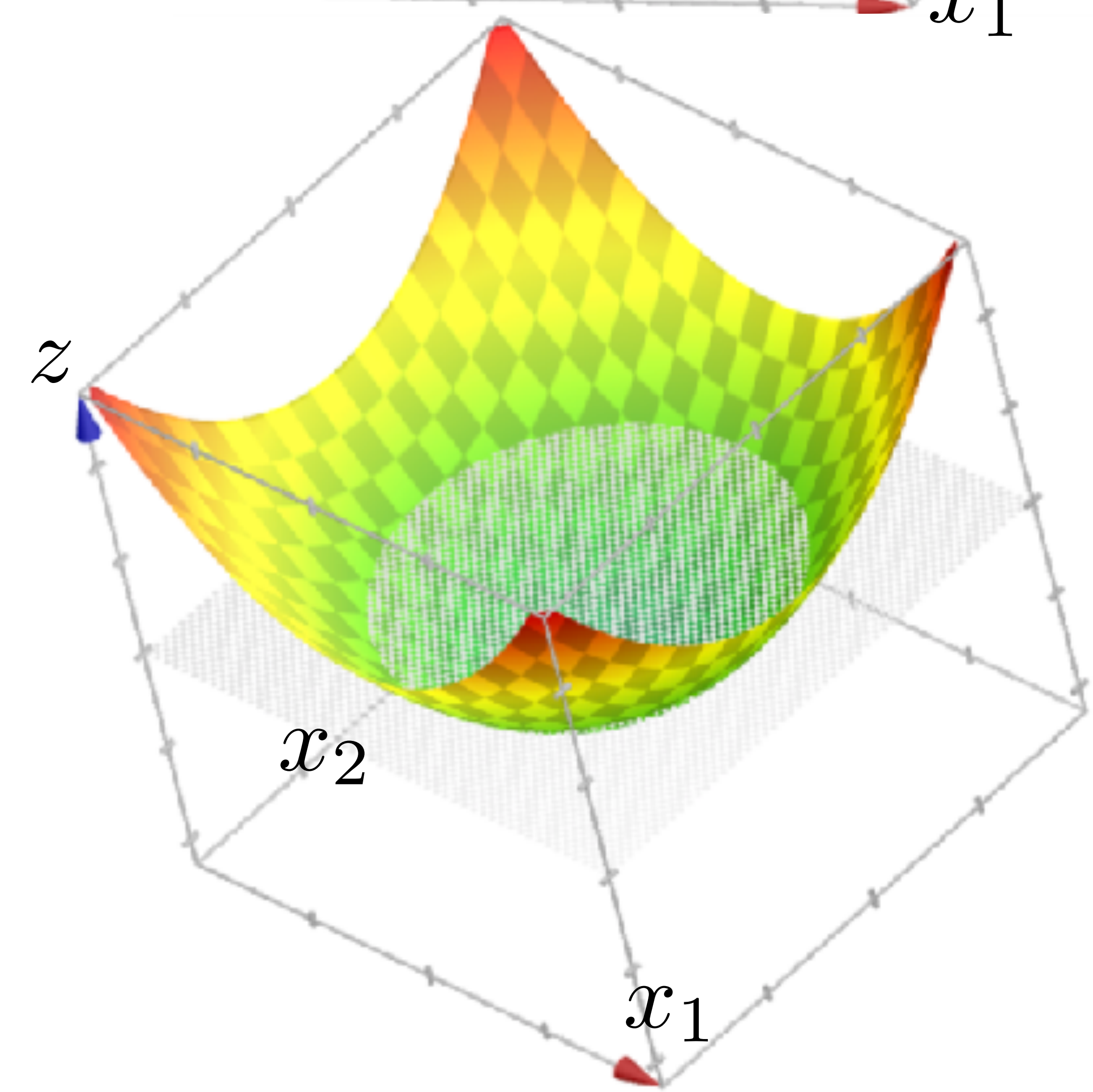




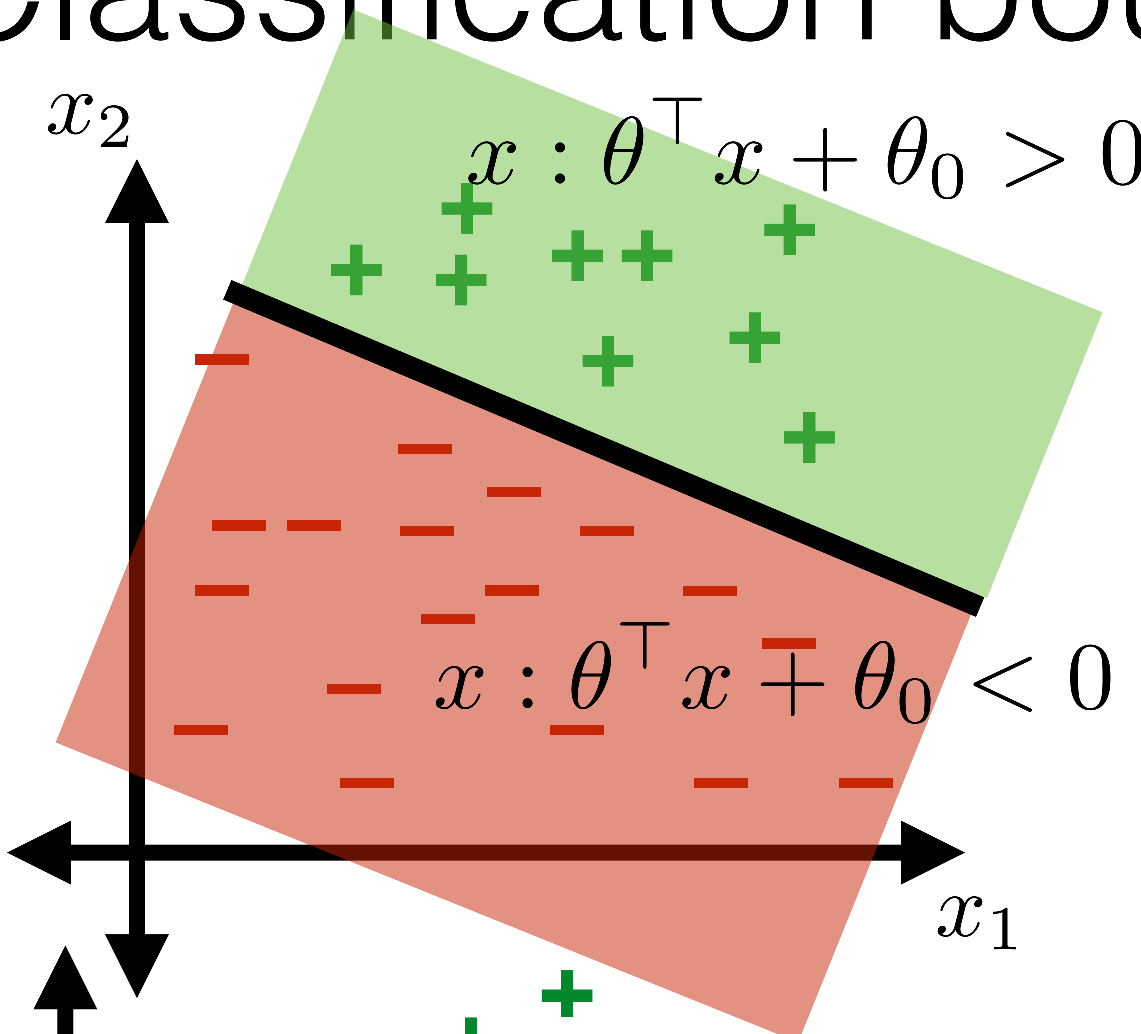
Classification boundaries

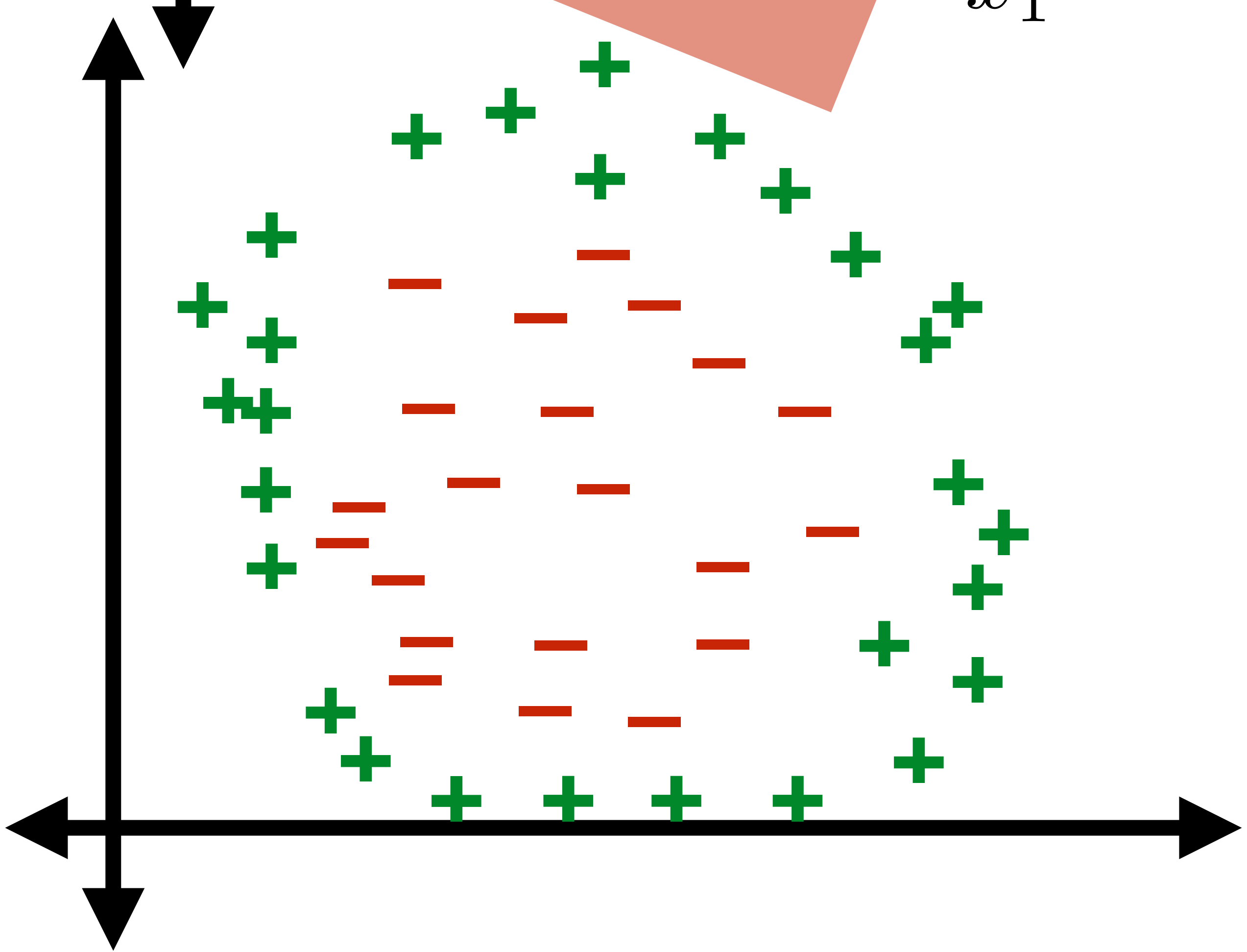


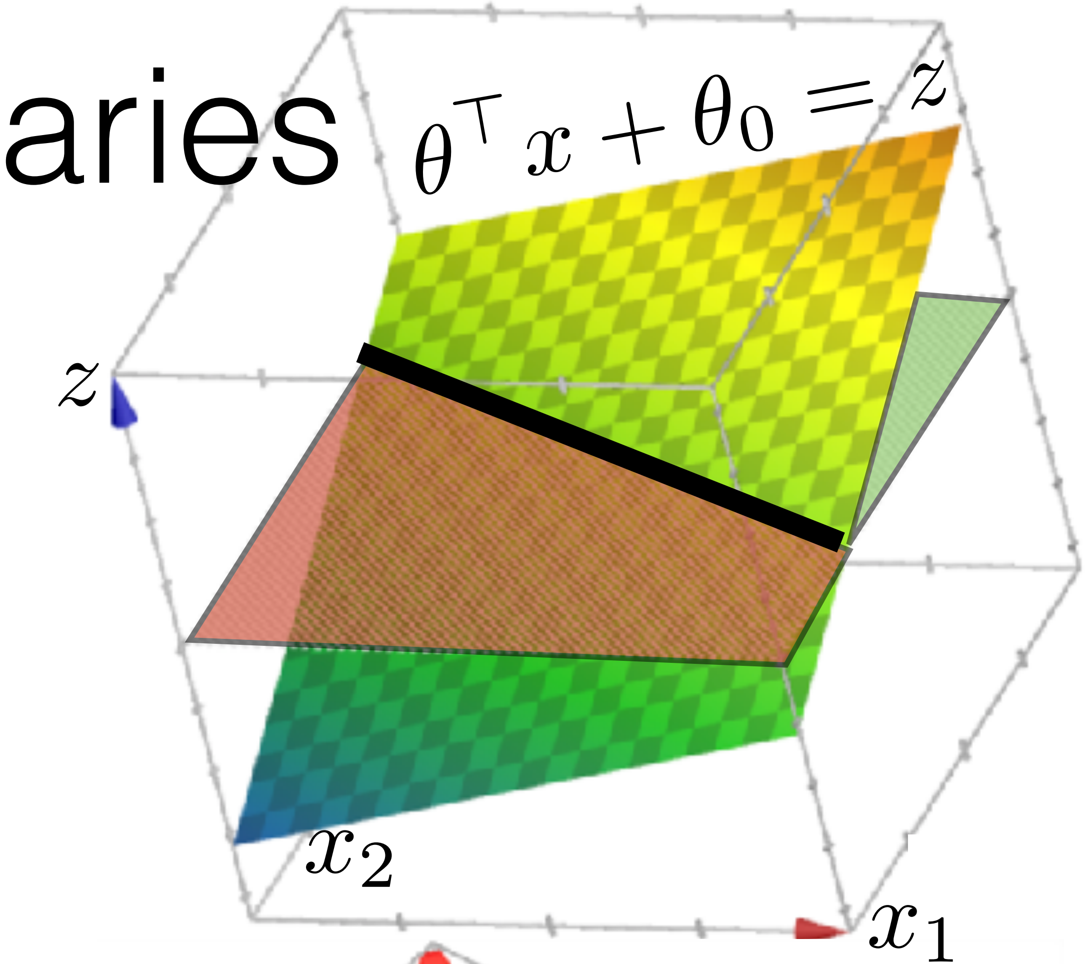


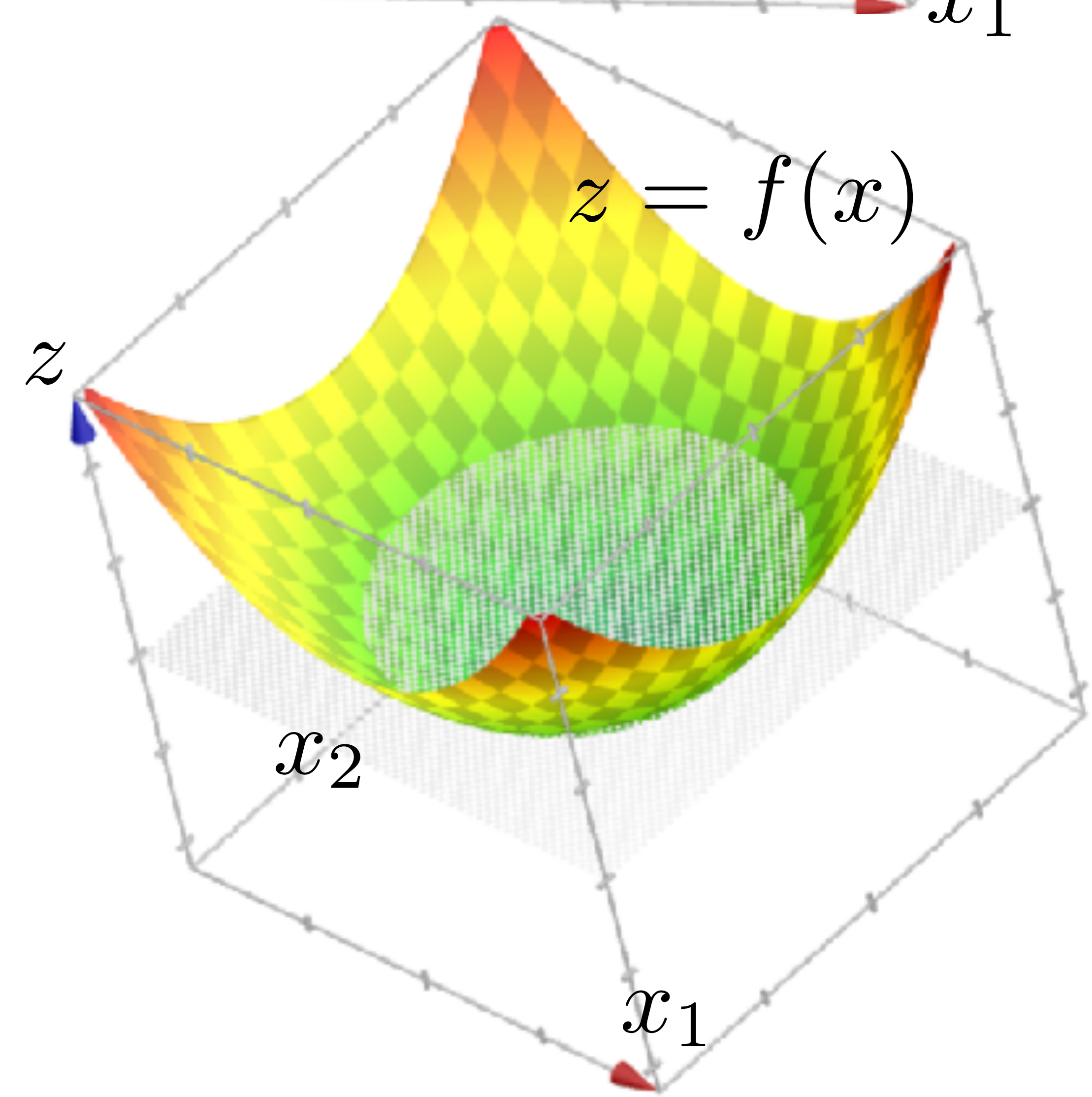


Classification boundaries

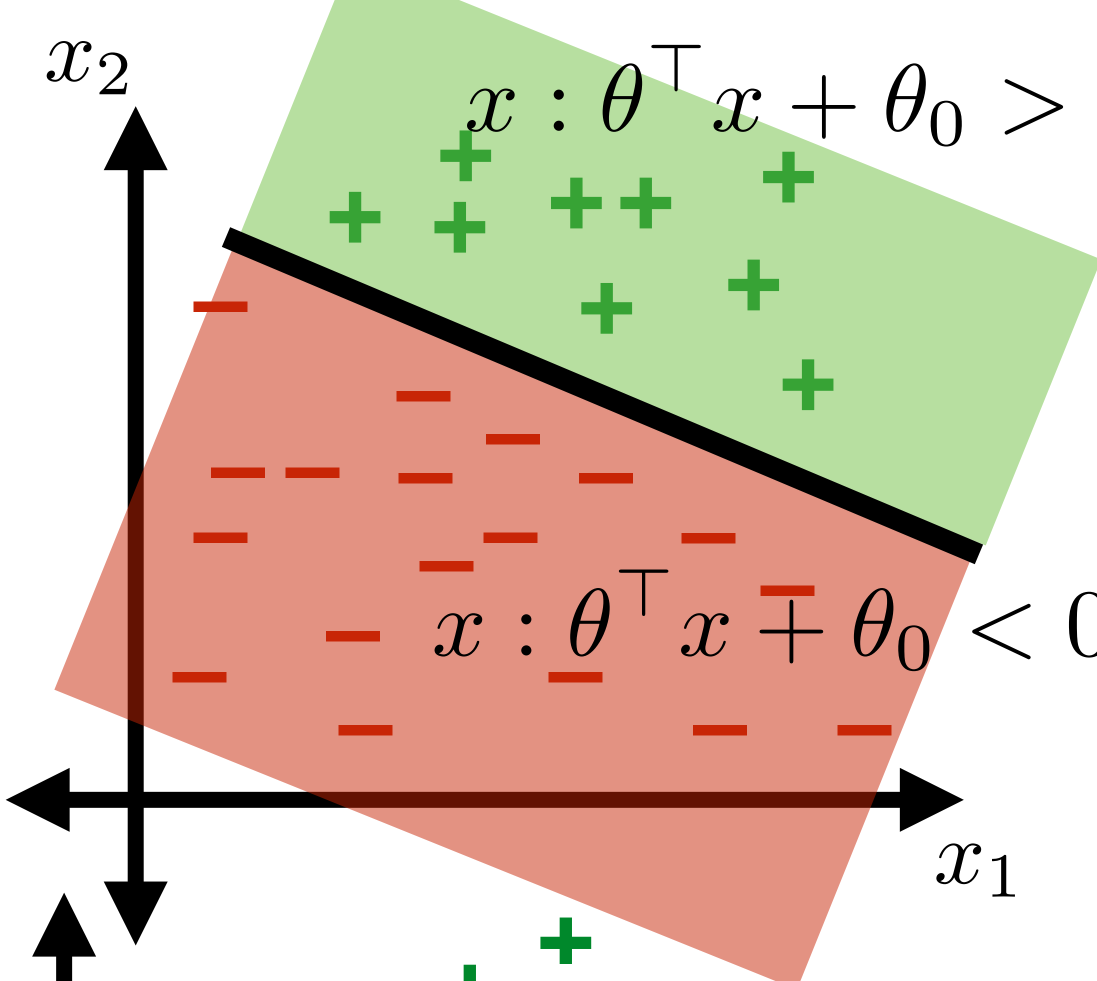


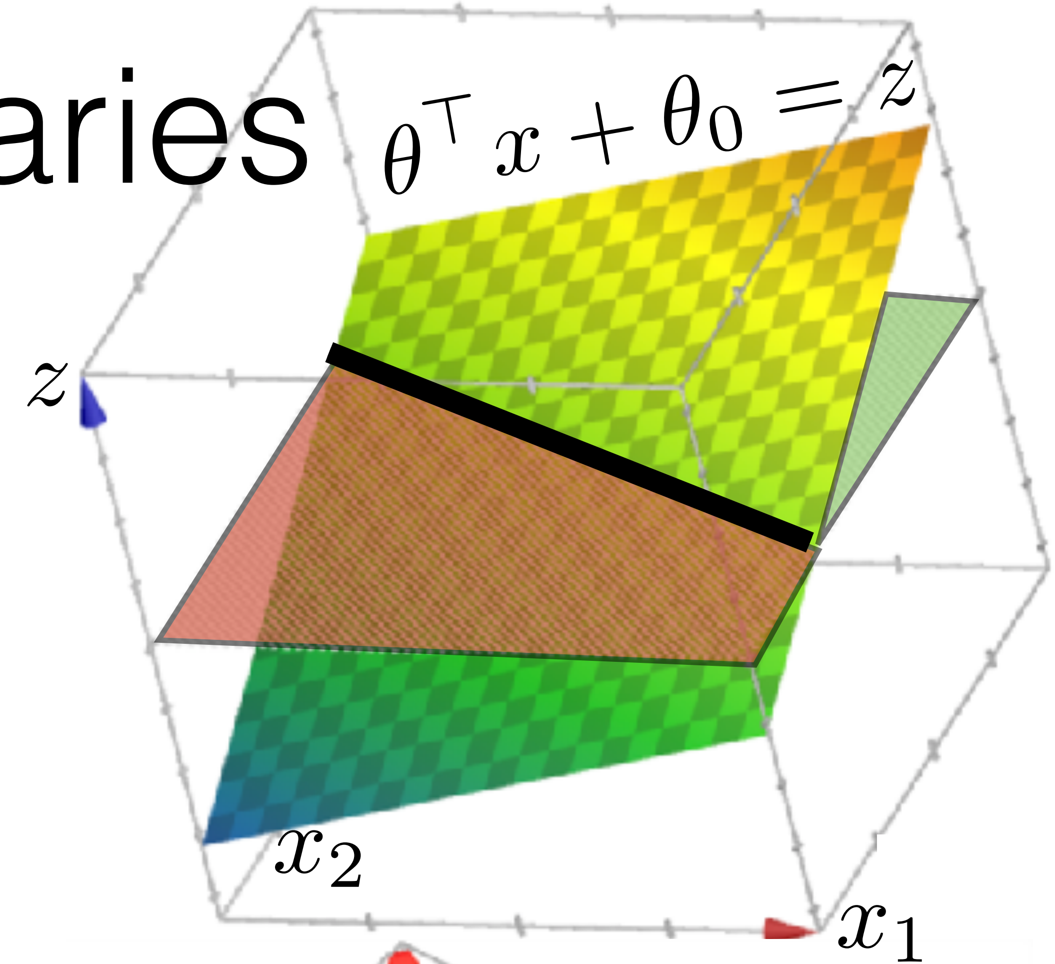


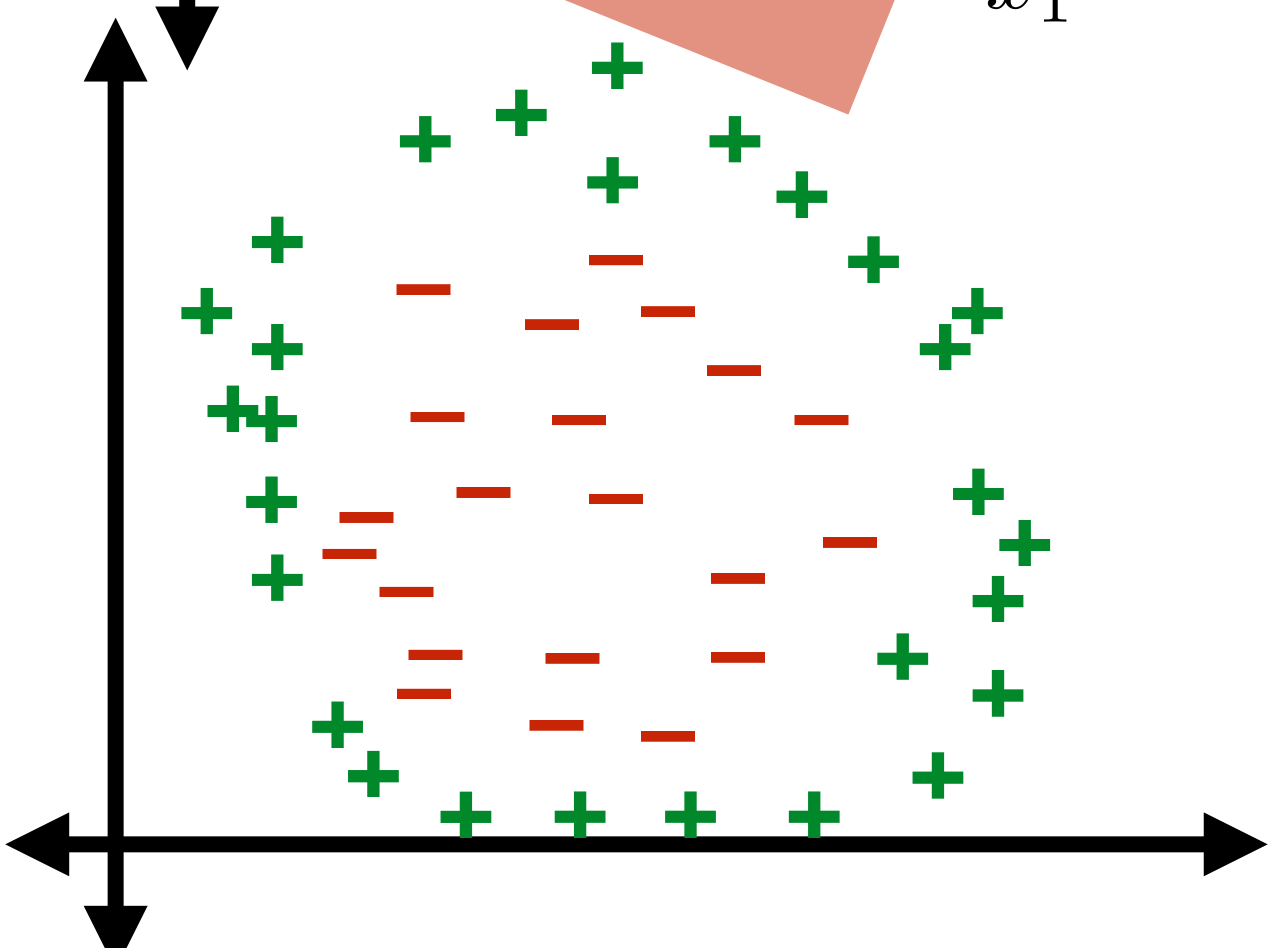


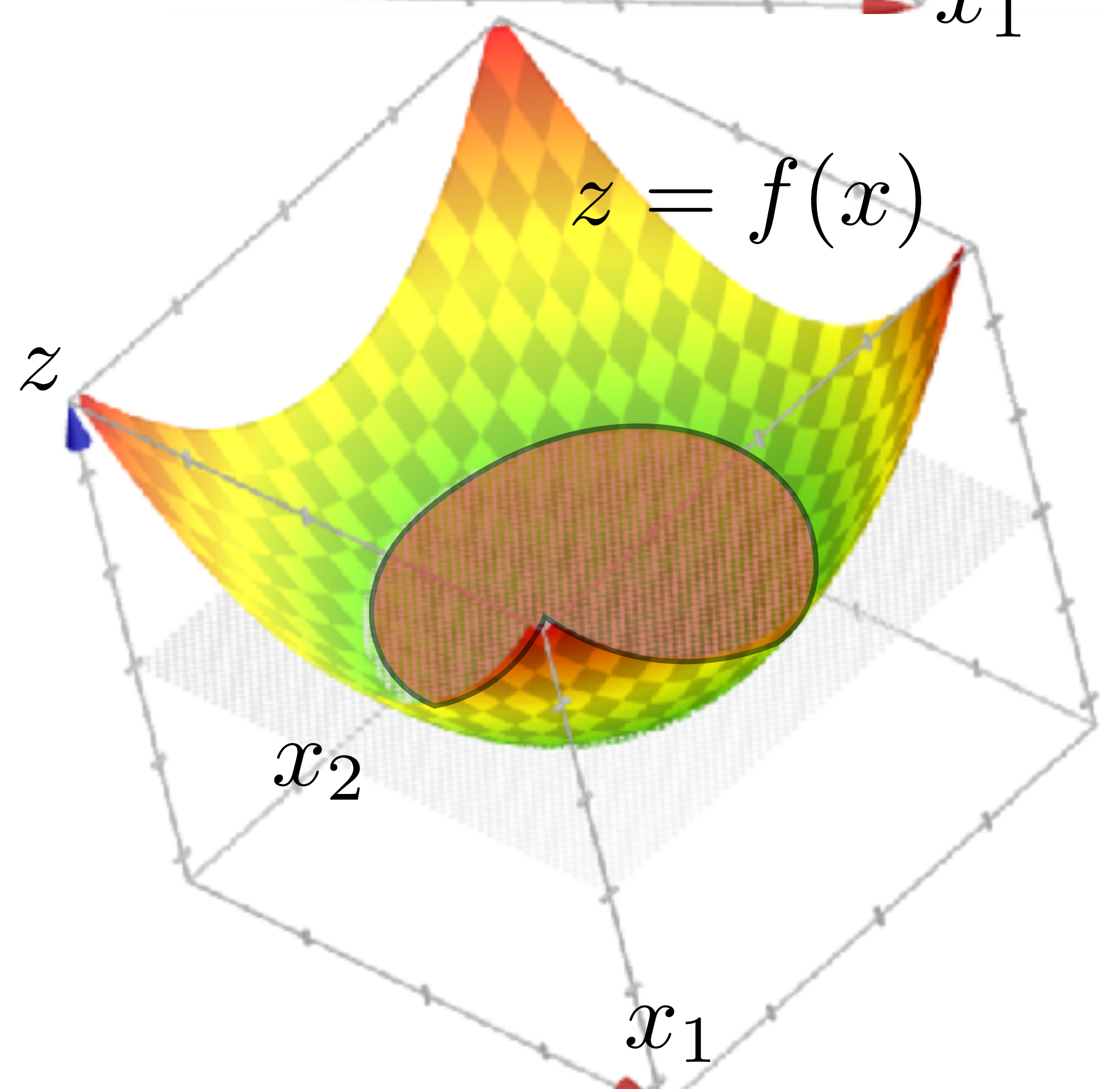


Classification boundaries

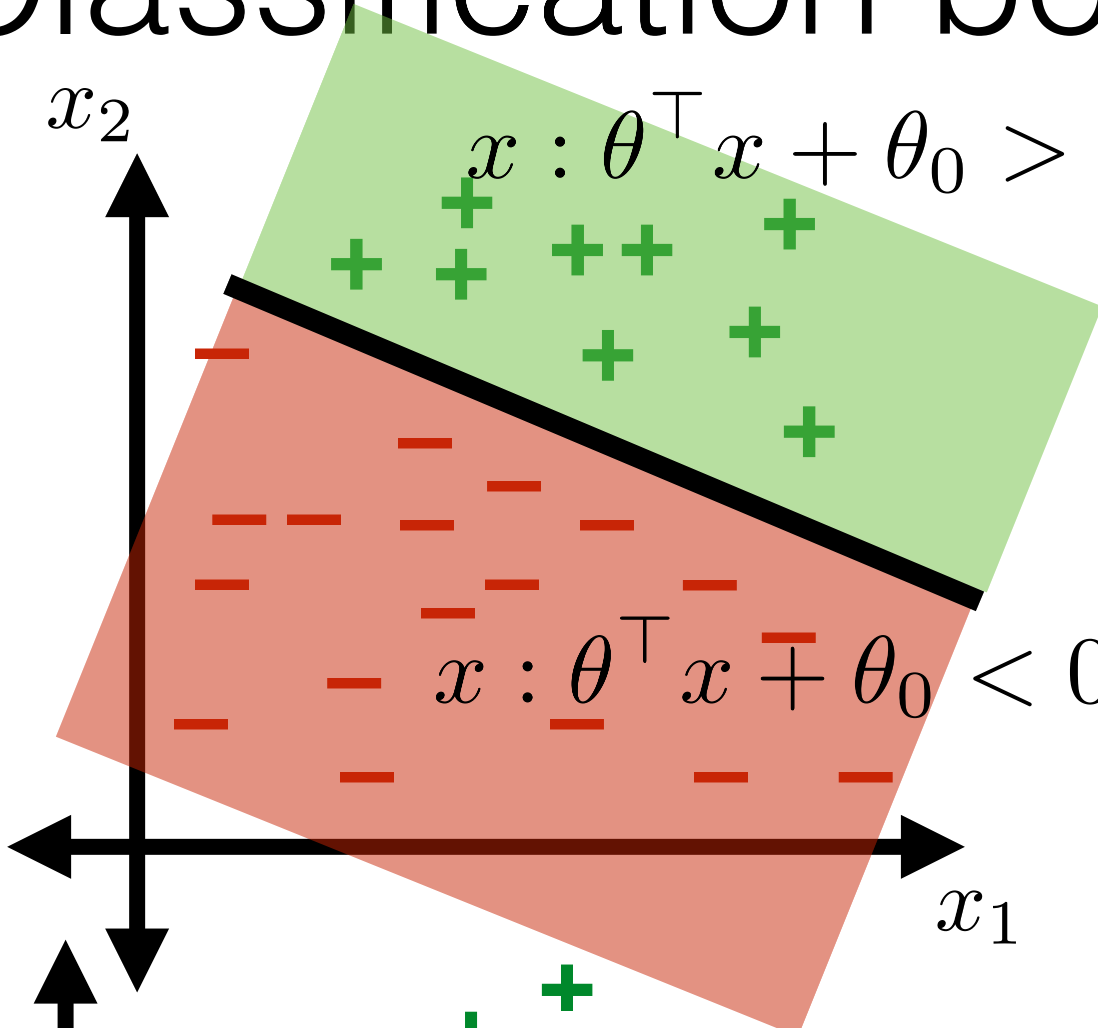


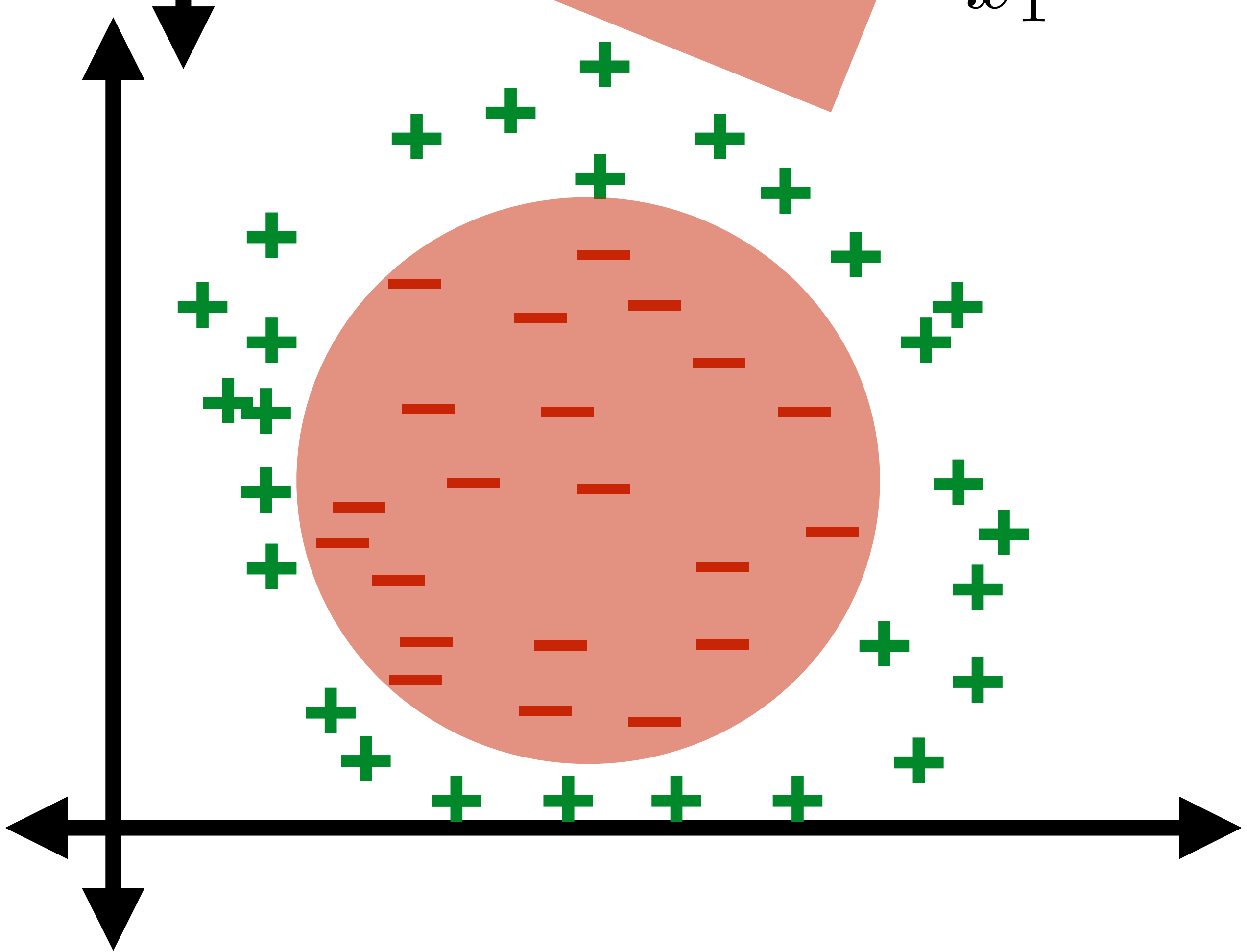


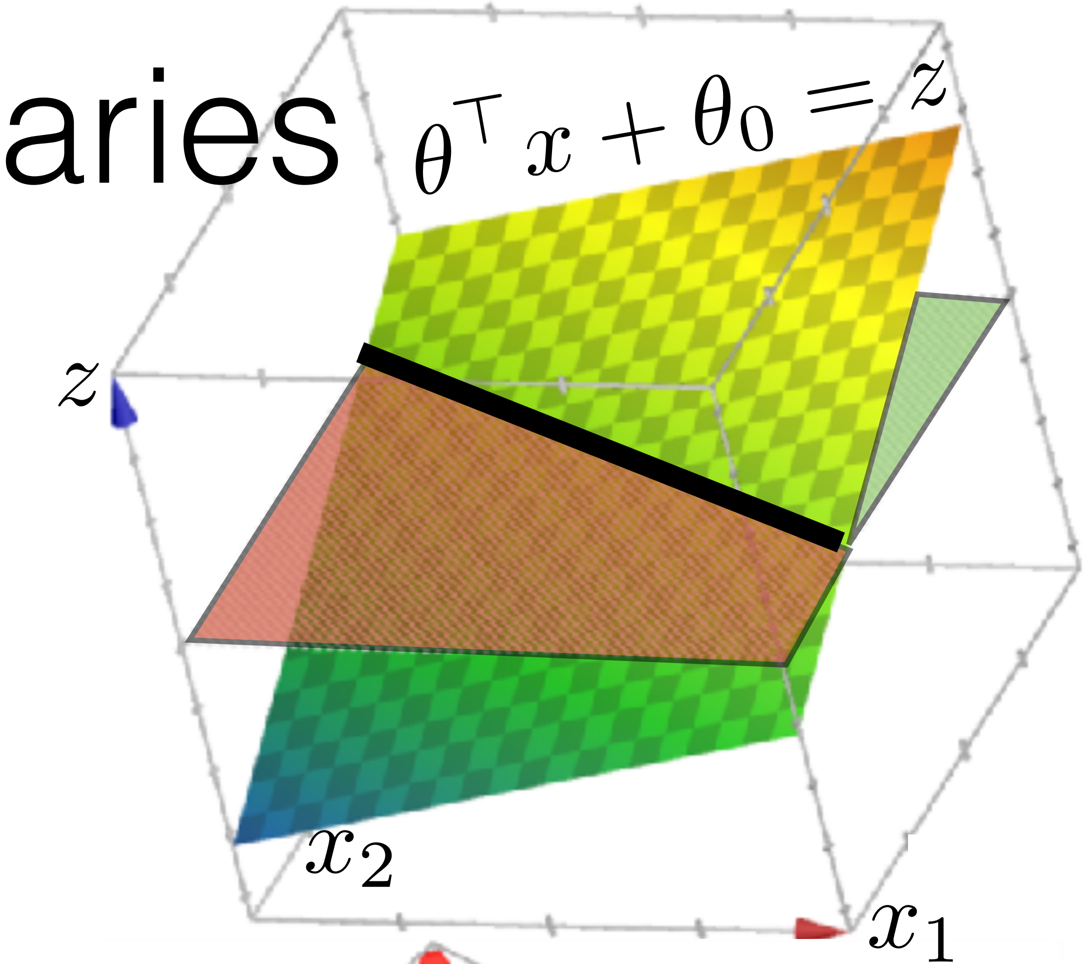


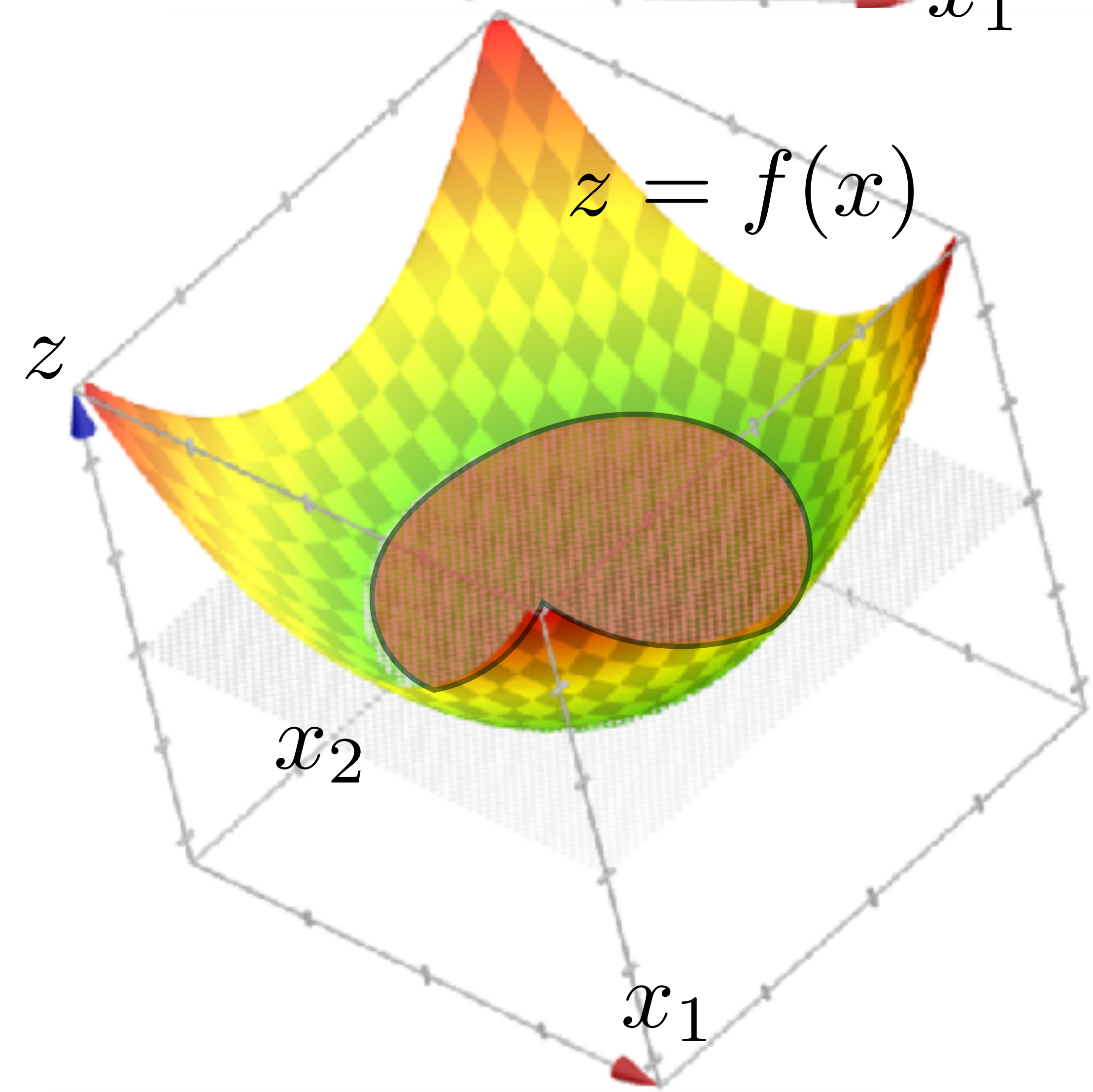


Classification boundaries

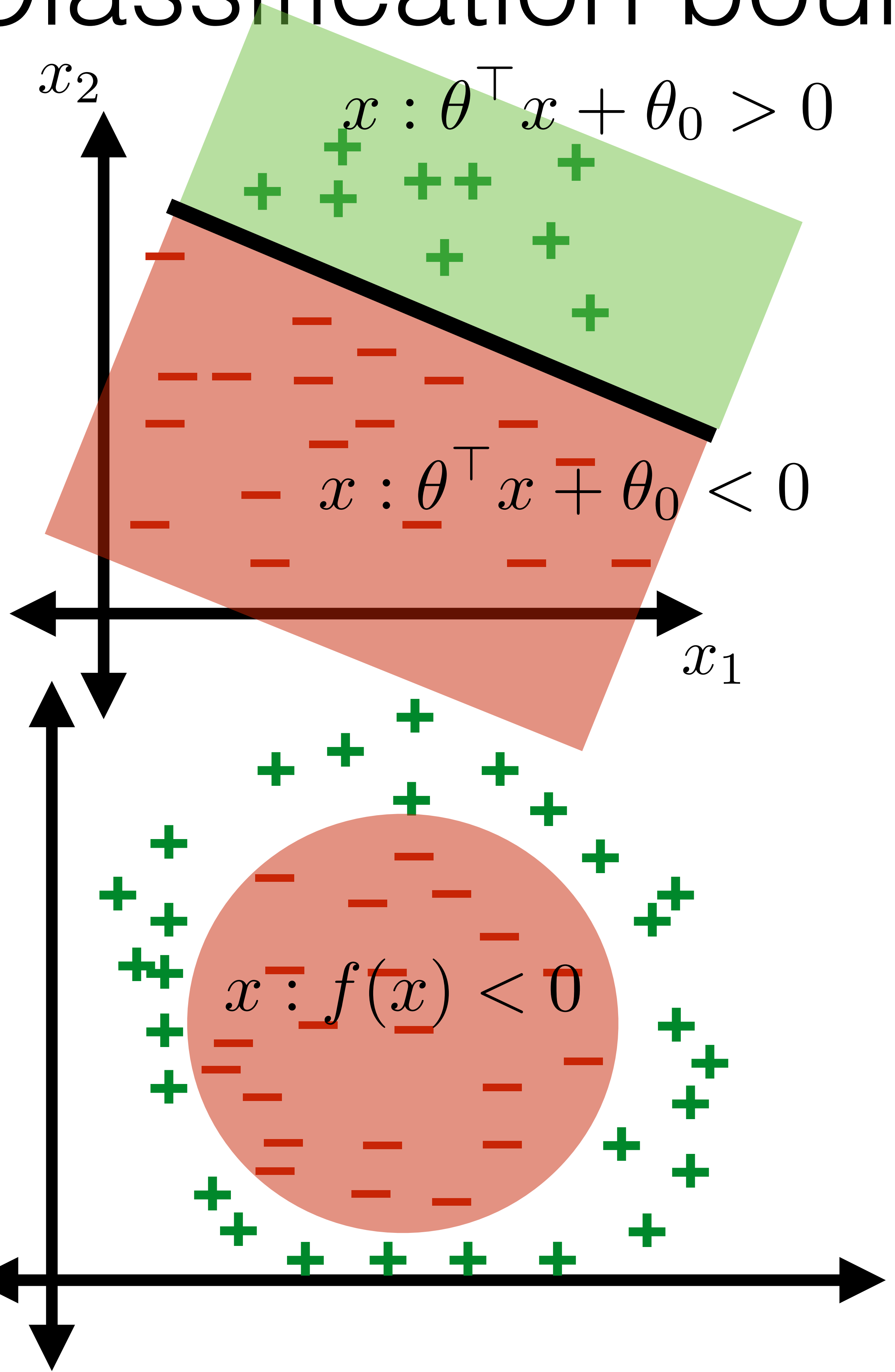


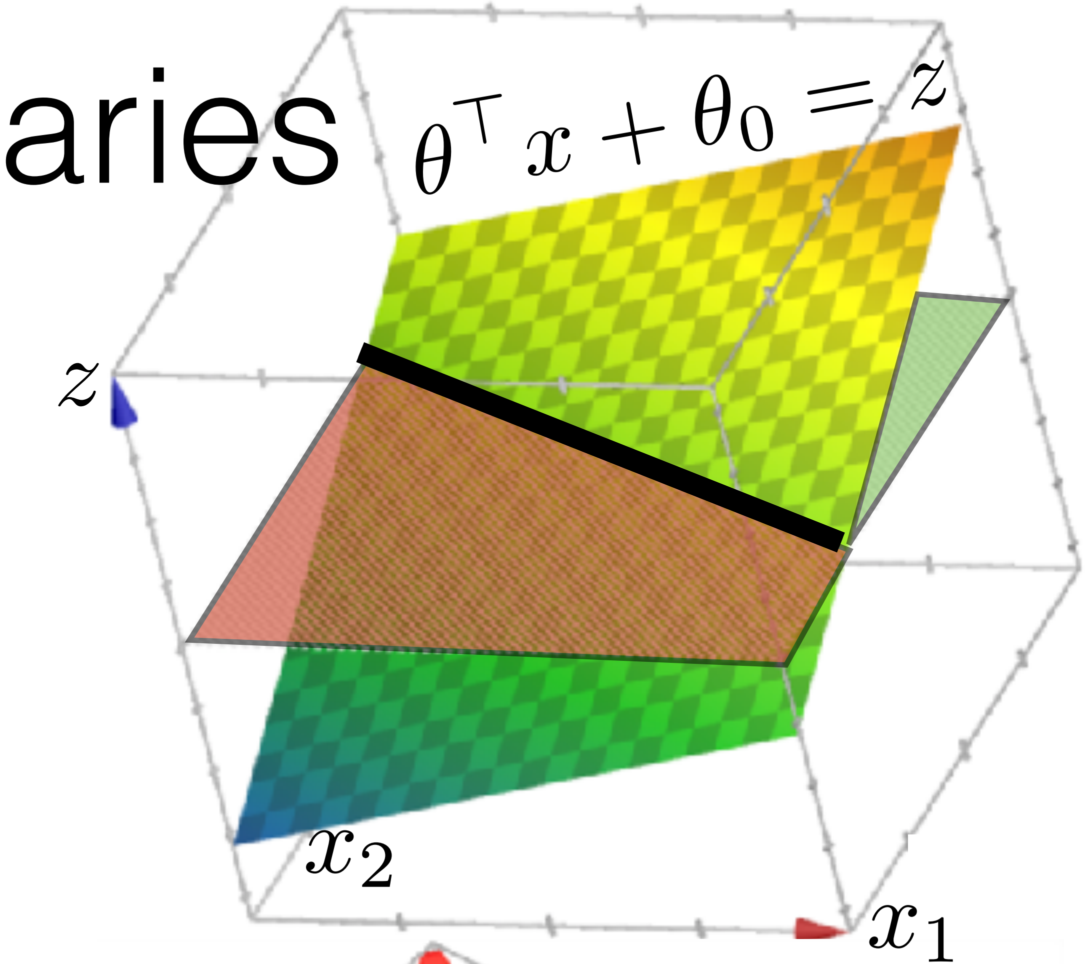


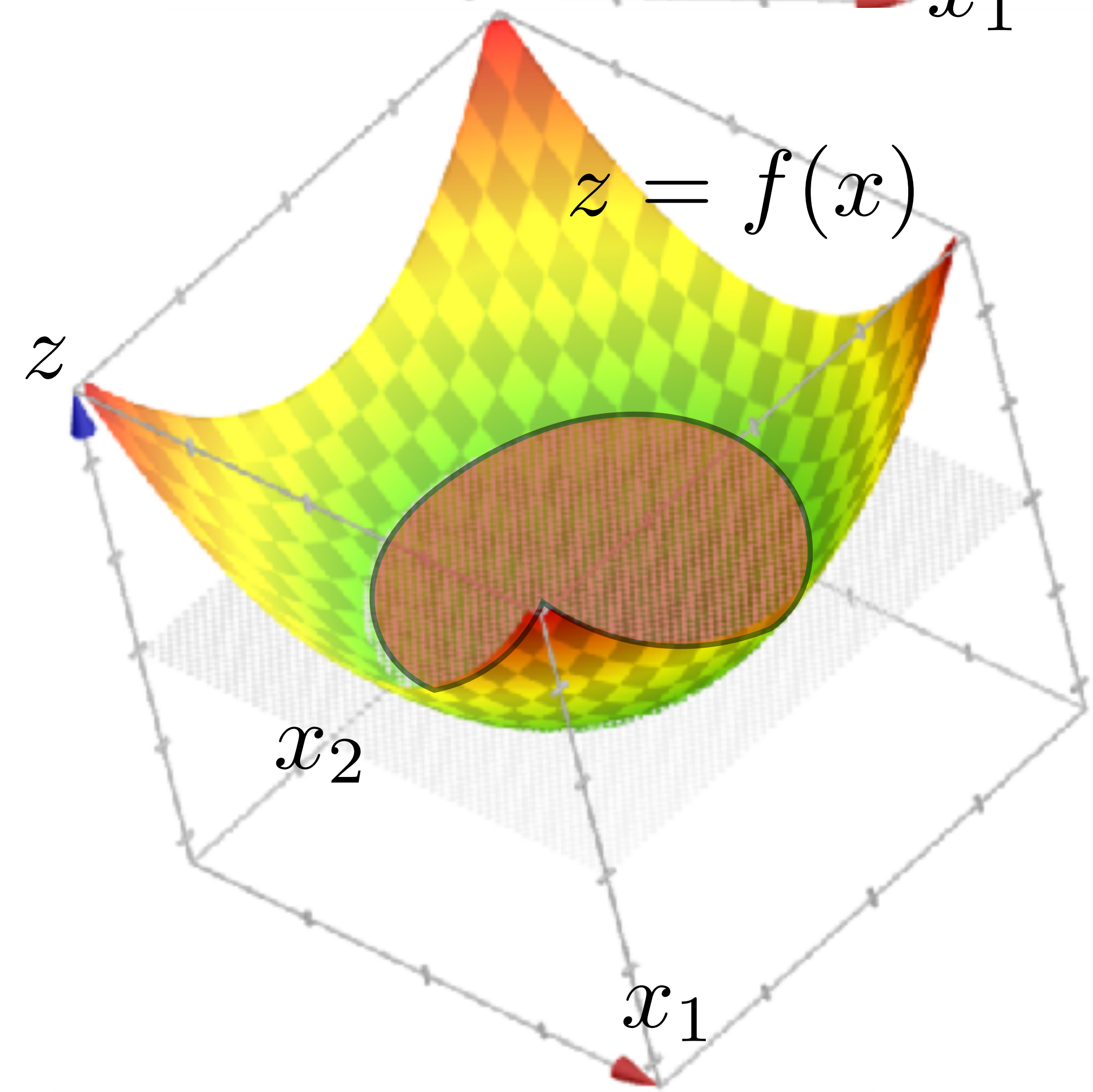




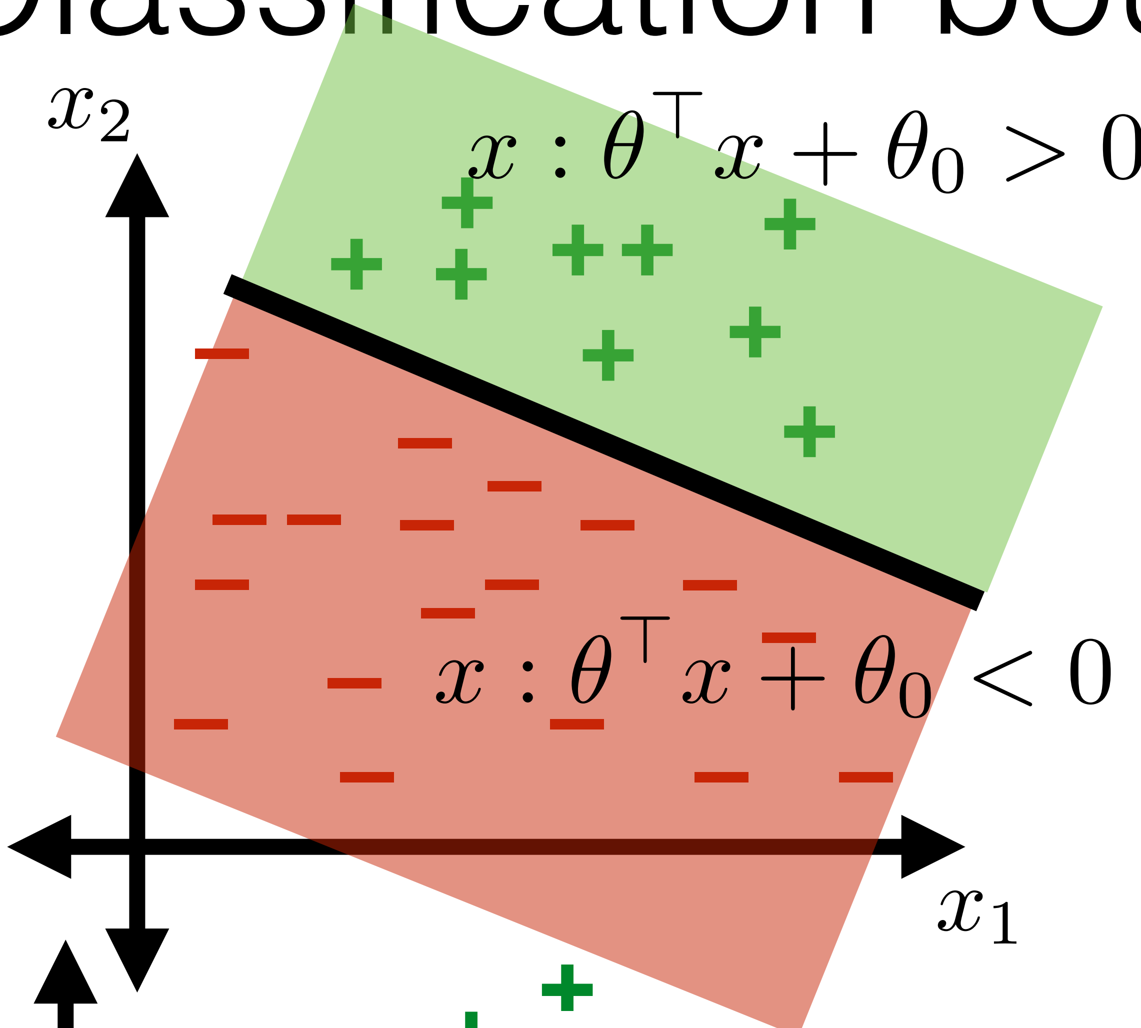
Classification boundaries

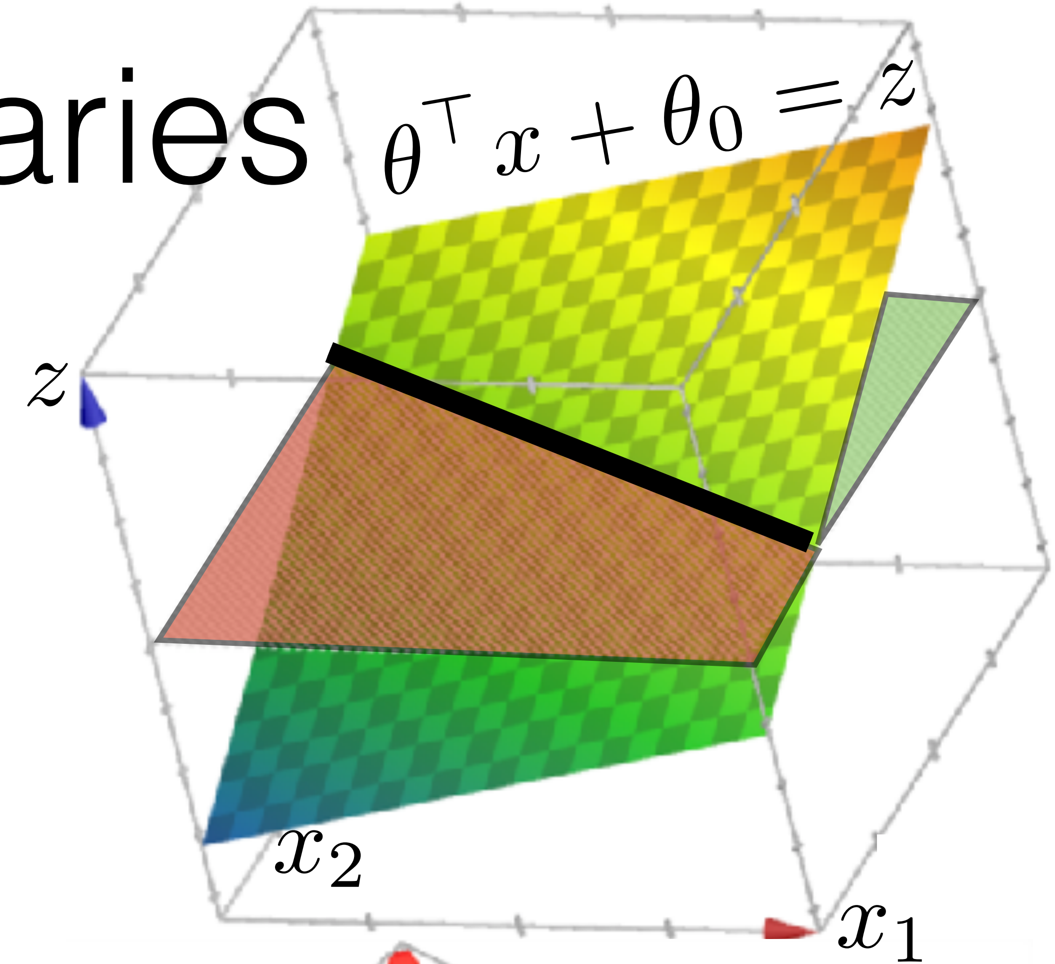


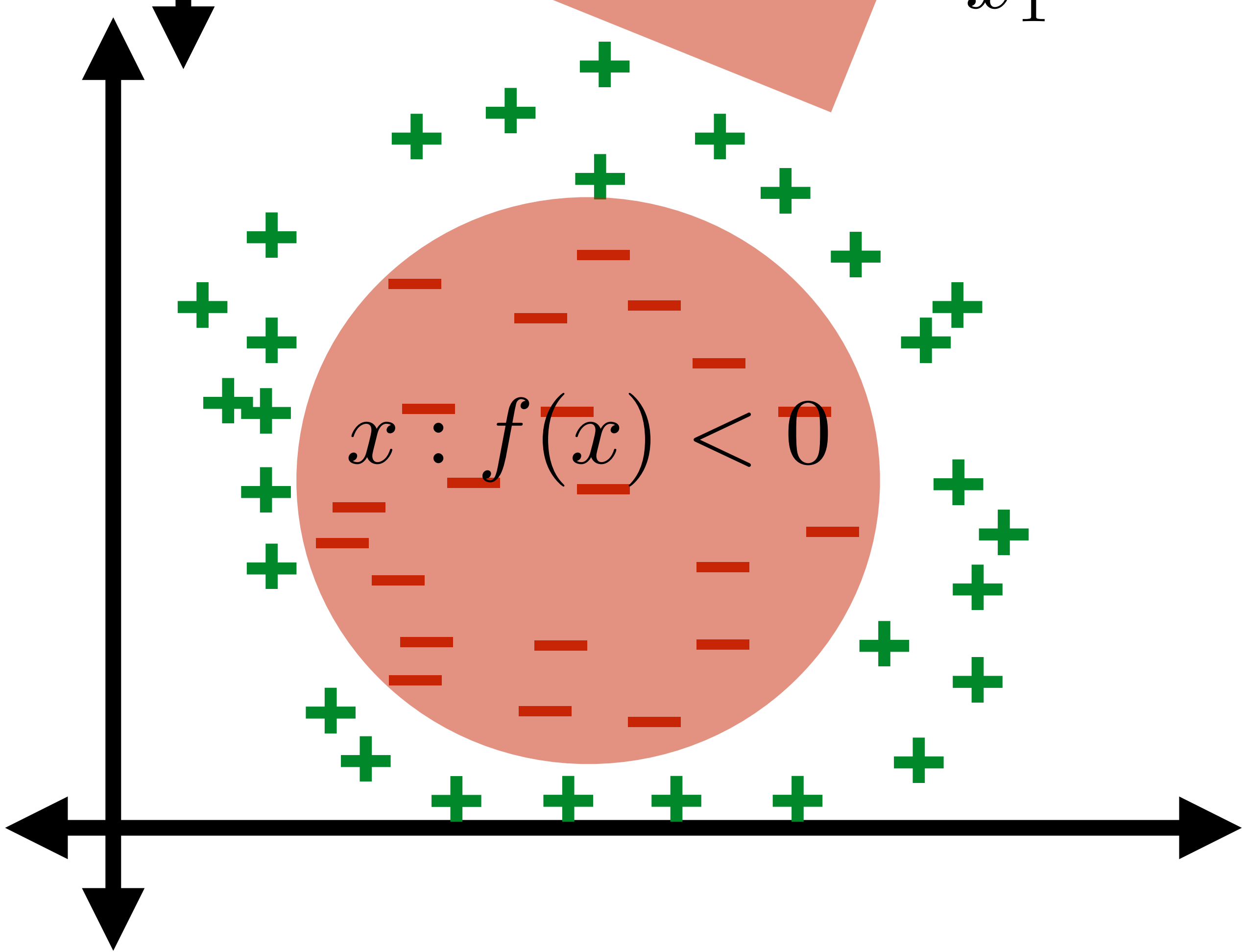


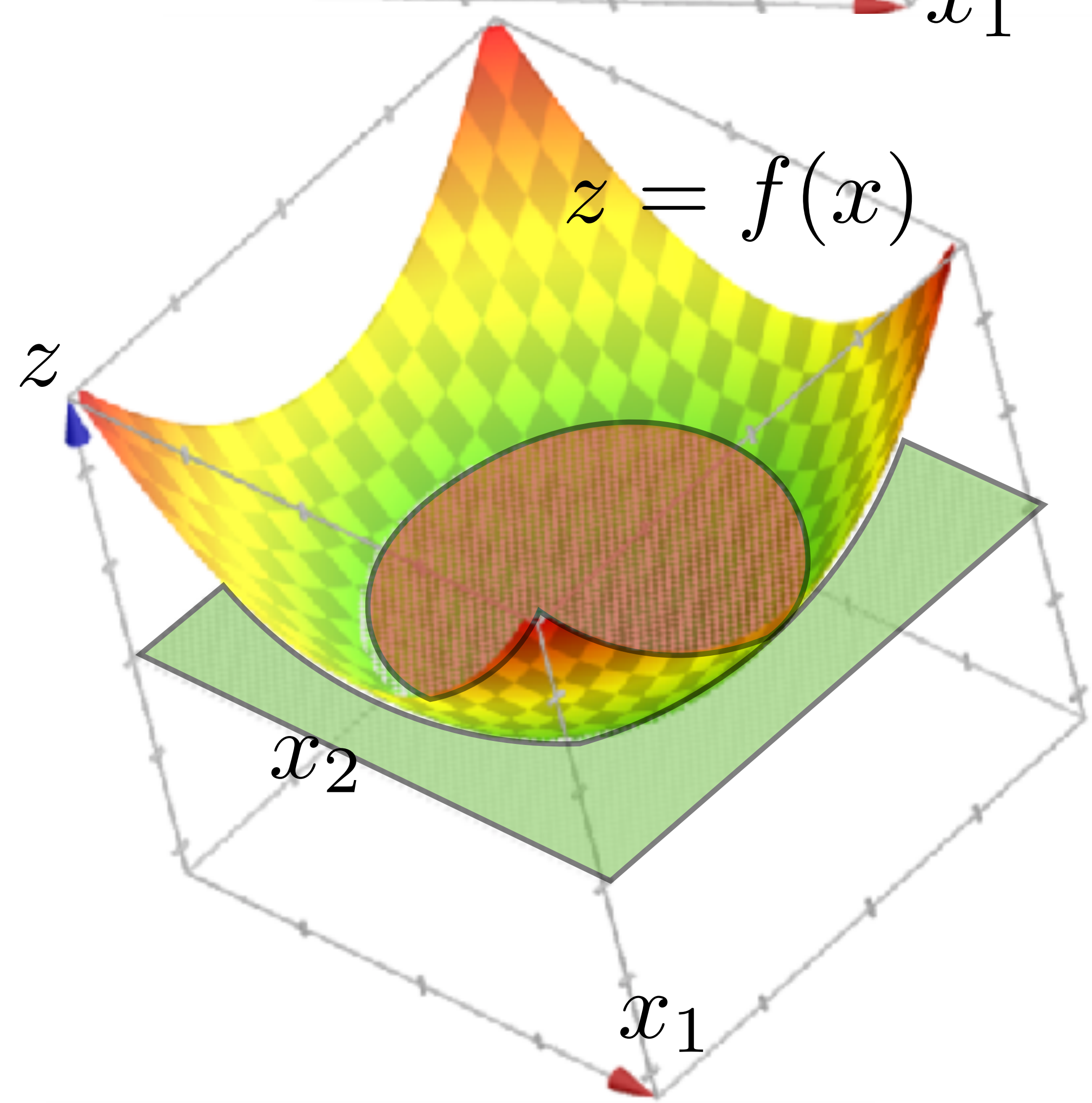


Classification boundaries

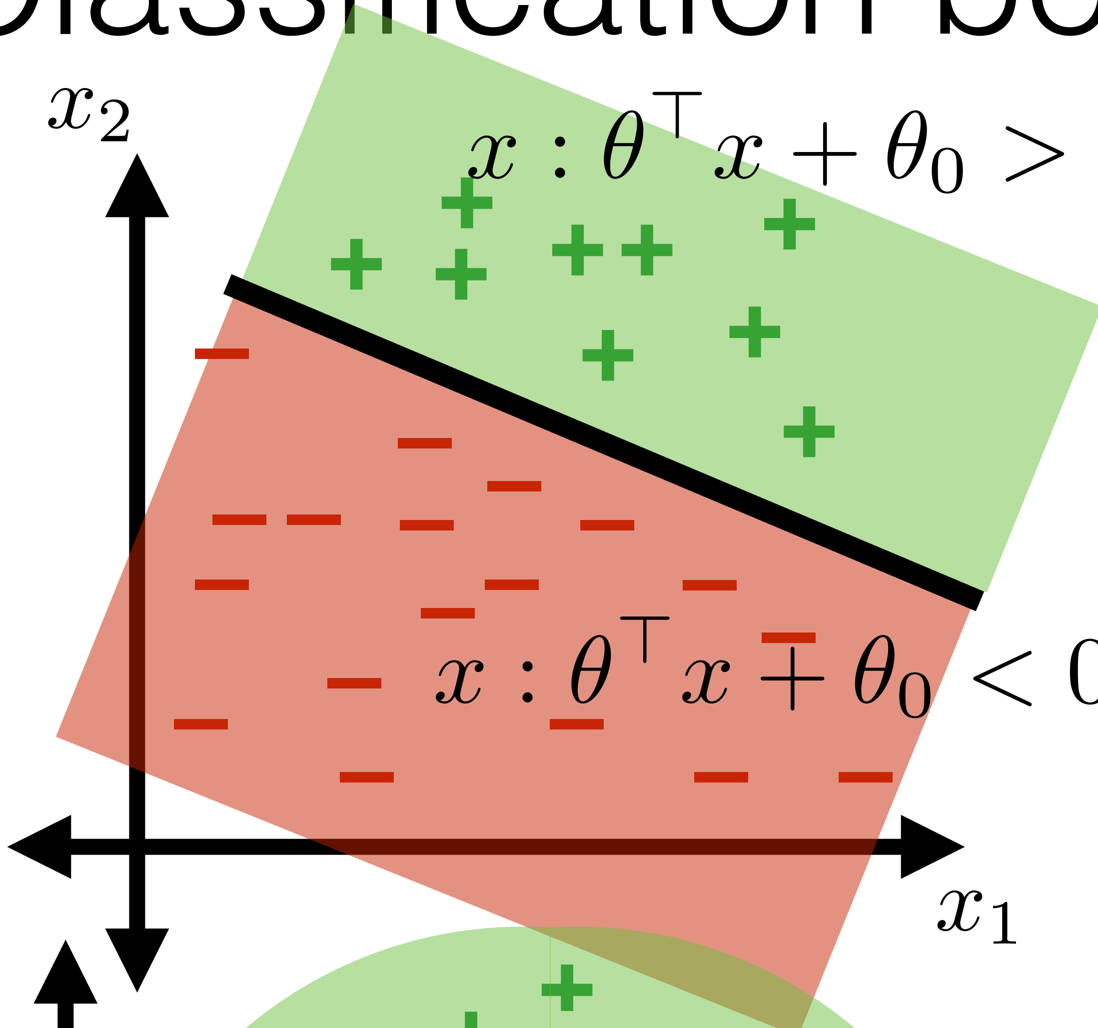


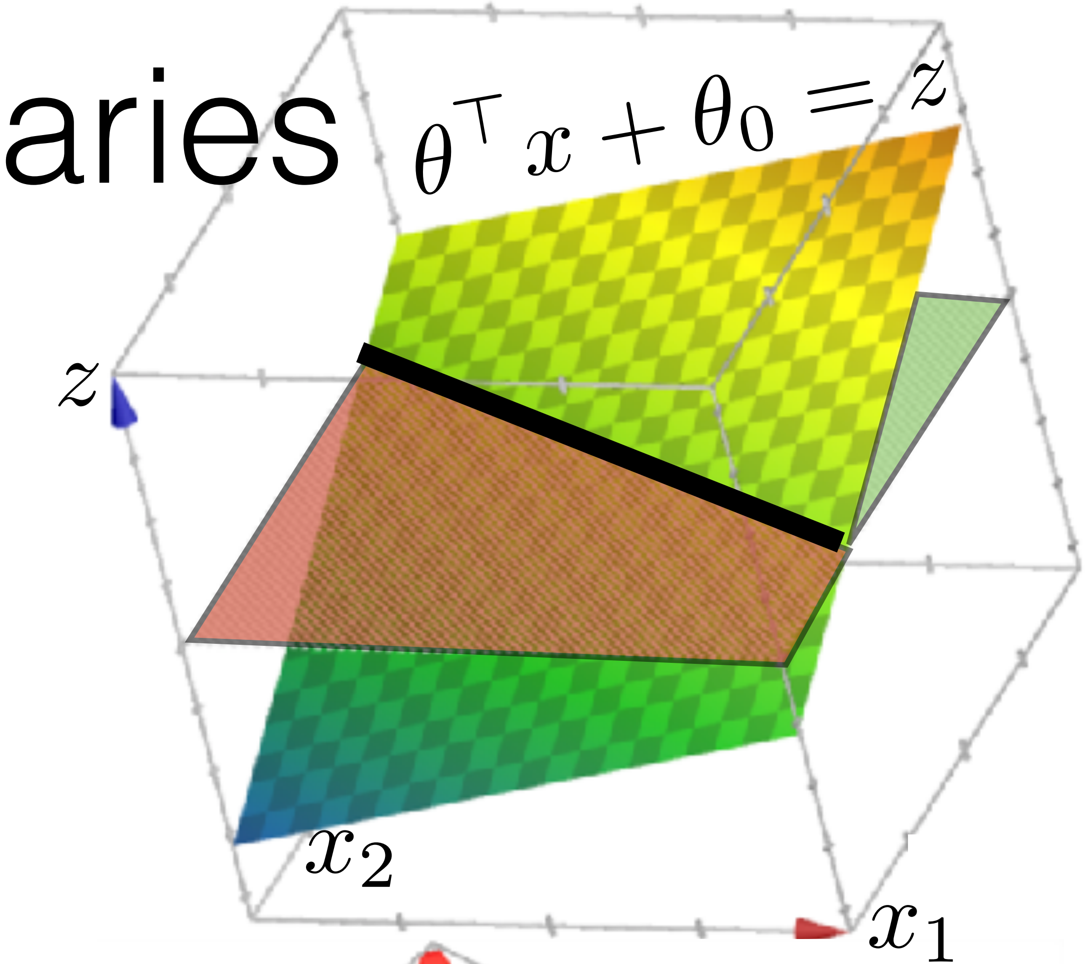


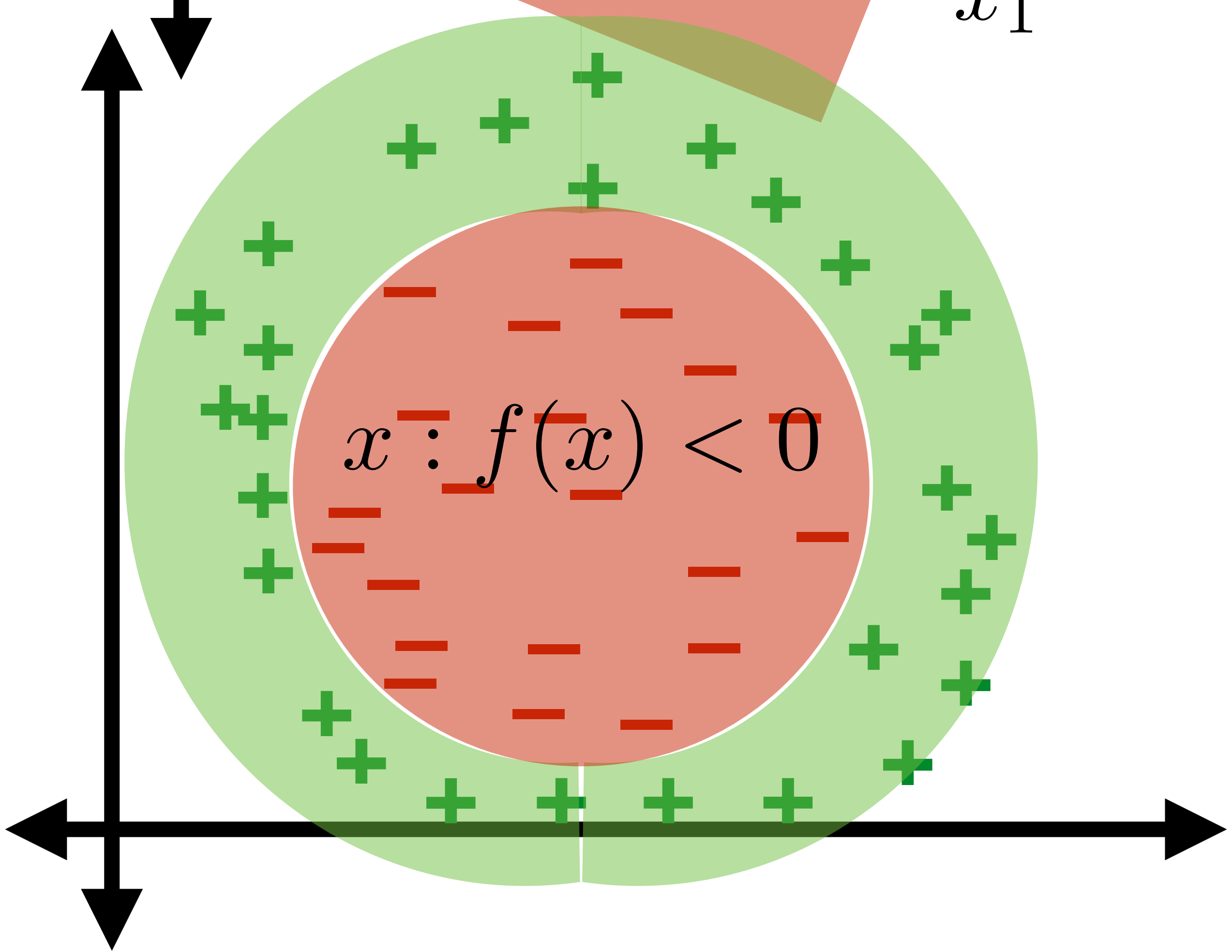


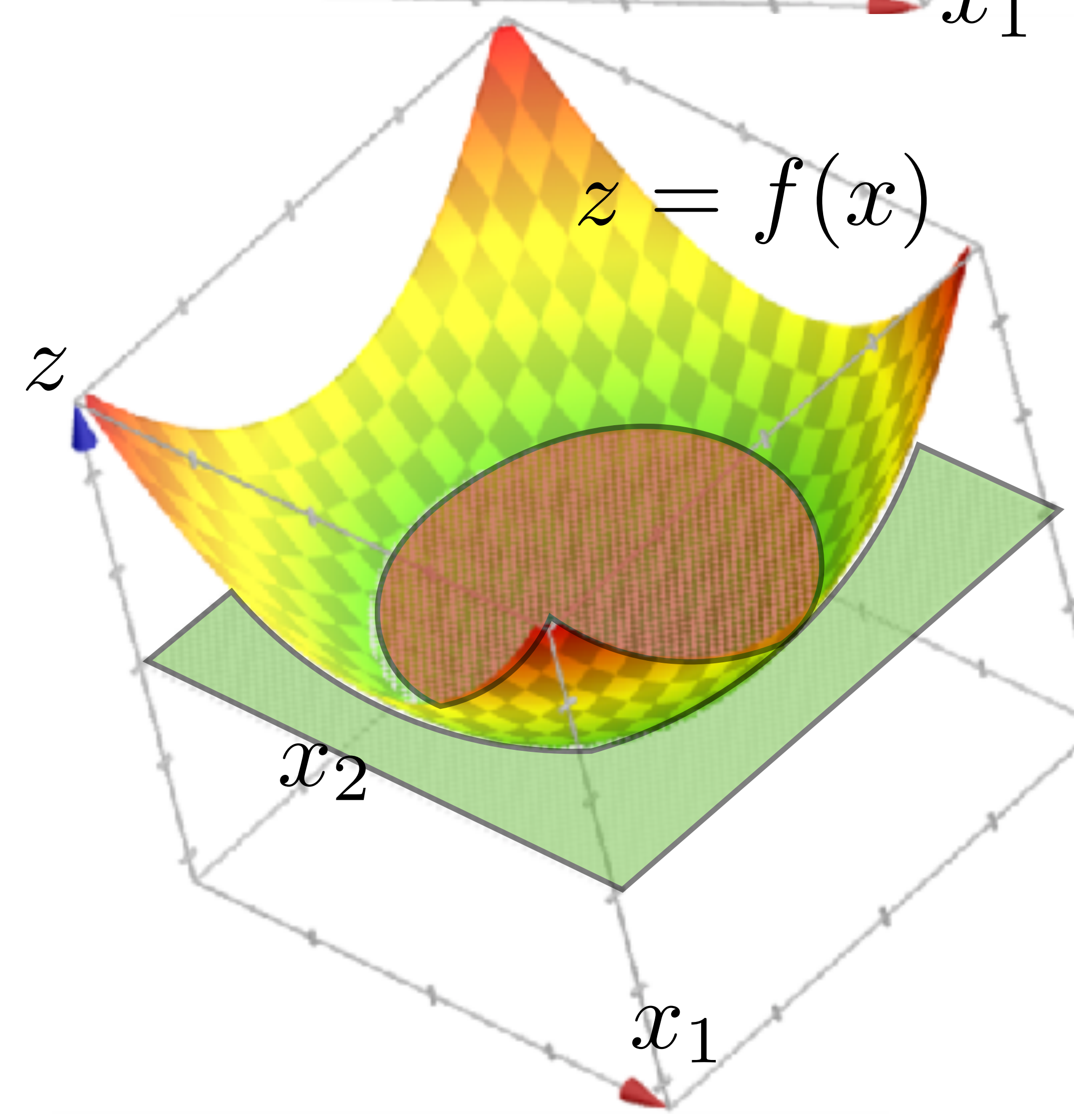


Classification boundaries

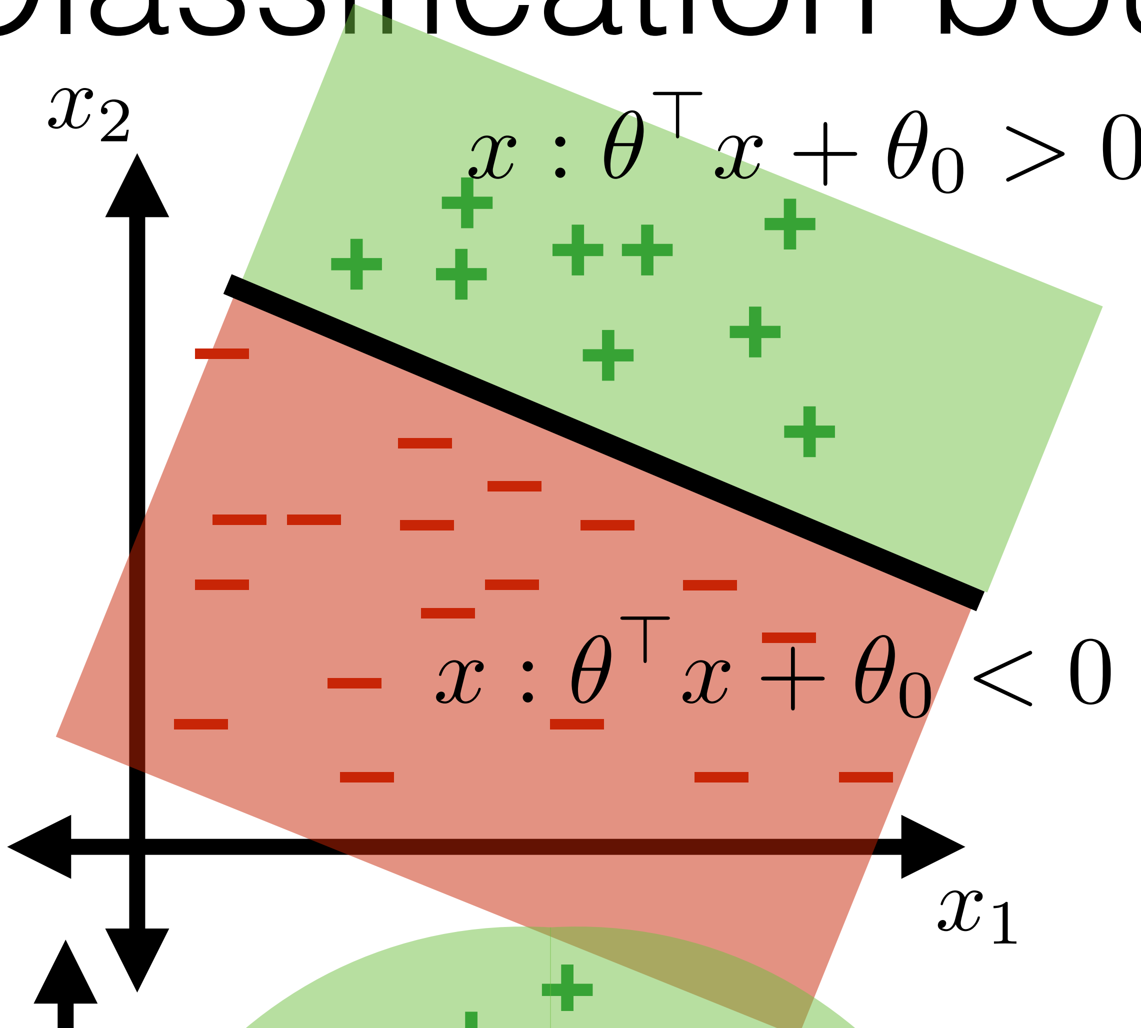


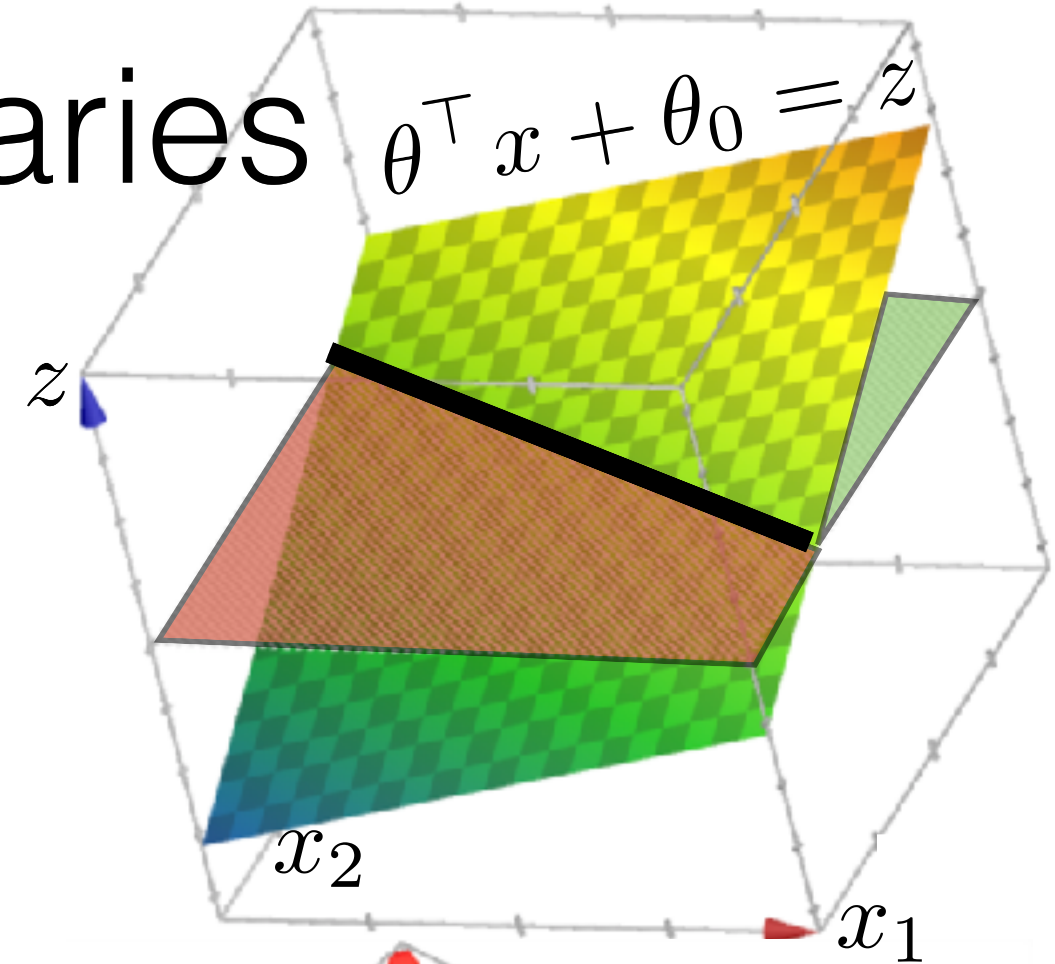


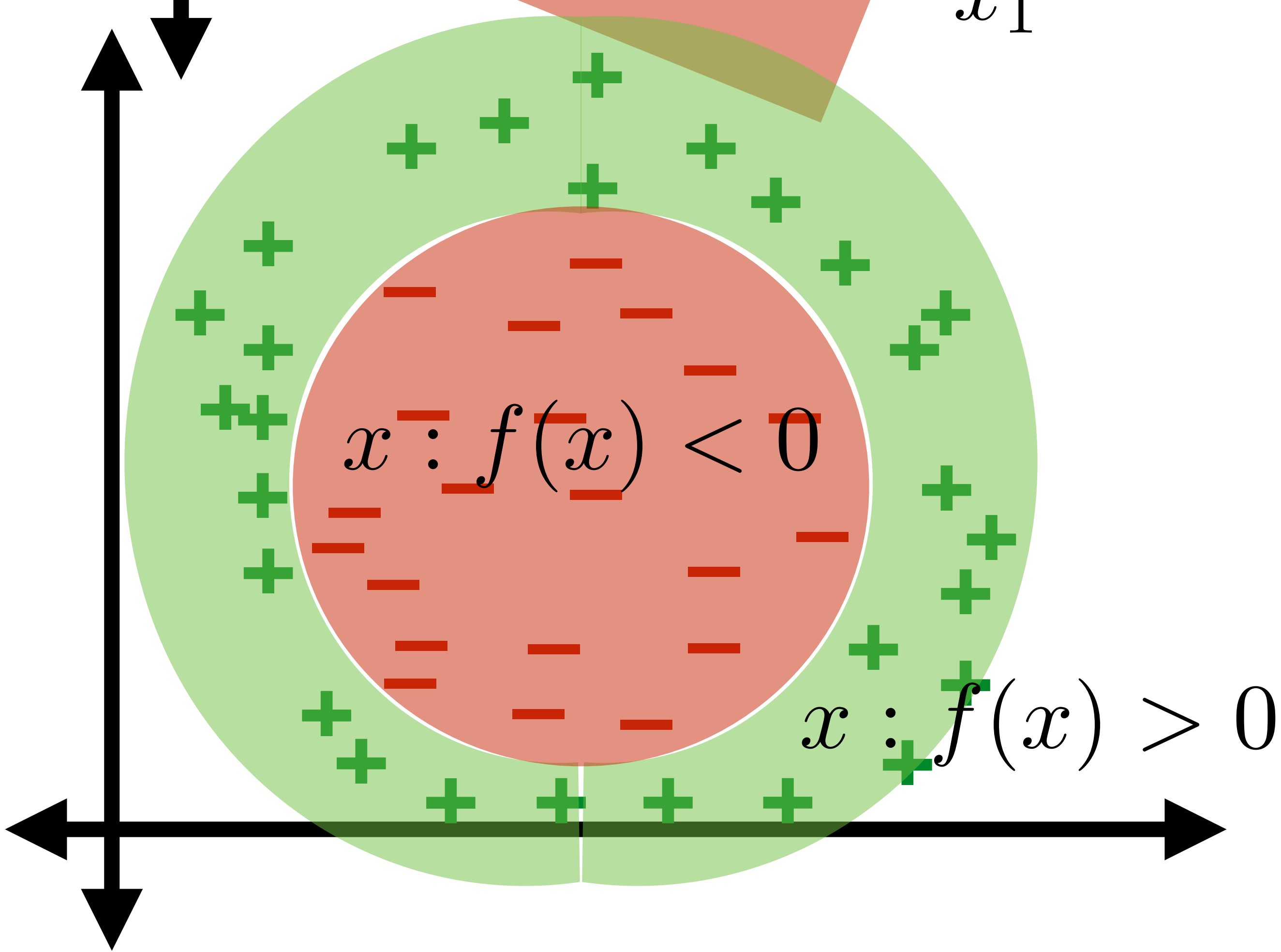


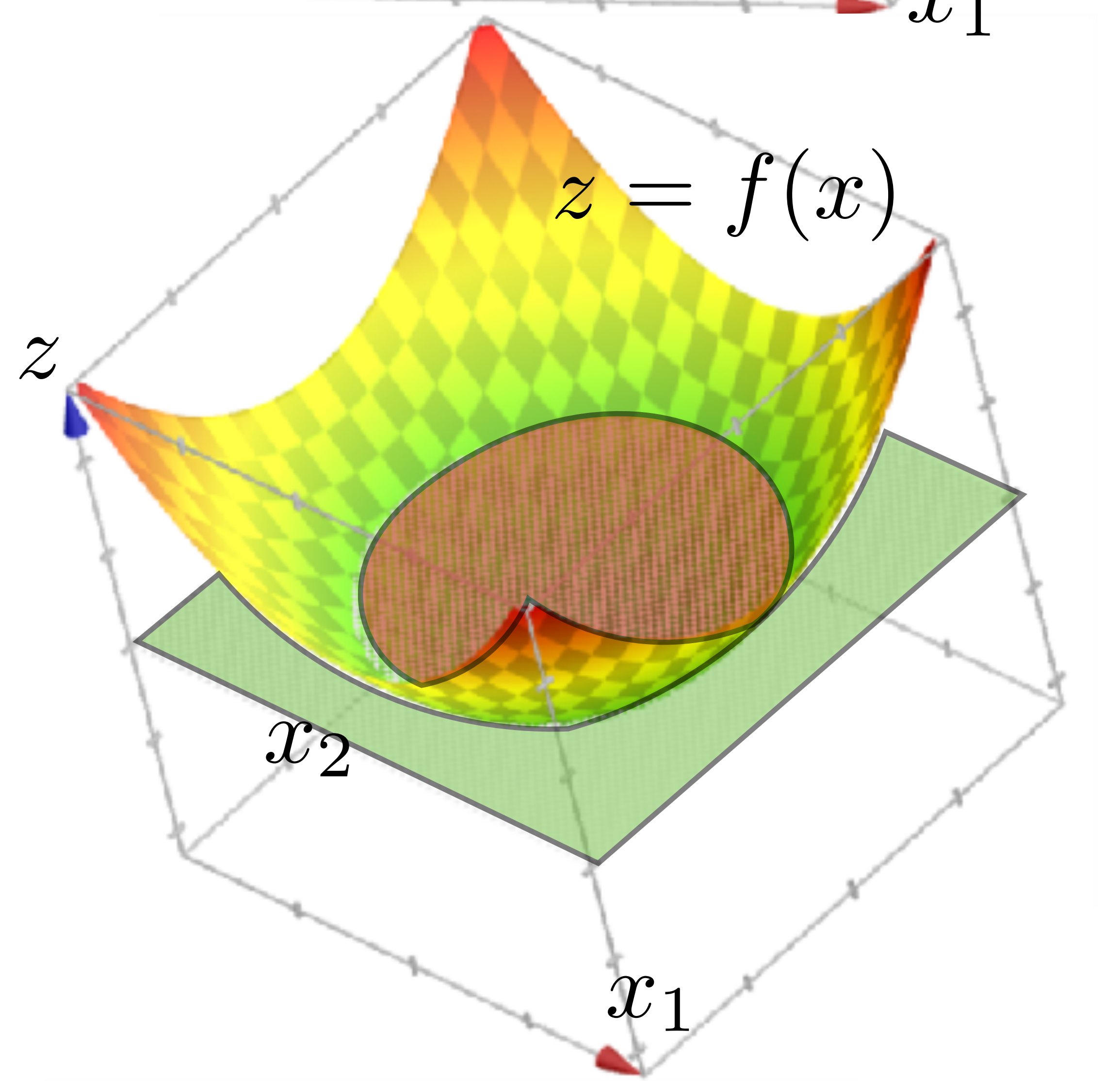


Classification boundaries









Nonlinear boundaries

Nonlinear boundaries

- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

Nonlinear boundaries

- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

order (*k*)	terms when *d*=1	terms for general d
0		
1		
2		
3		

Nonlinear boundaries

- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

order (*k*)	terms when *d*=1	terms for general d
0	$[1]$	
1		
2		
3		

Nonlinear boundaries

- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

order (*k*)	**terms when *d*=1**	**terms for general d**
0	$[1]$	
1	$[1, x_1]$	
2		
3		

Nonlinear boundaries

- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

order (*k*)	terms when *d*=1	terms for general d
0	$[1]$	
1	$[1, x_1]$	
2	$[1, x_1, x_1^2]$	
3		

Nonlinear boundaries

- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

order (*k*)	terms when *d*=1	terms for general d
0	$[1]$	
1	$[1, x_1]$	
2	$[1, x_1, x_1^2]$	
3	$[1, x_1, x_1^2, x_1^3]$	

Nonlinear boundaries

- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

order (*k*)	**terms when *d*=1**	**terms for general d**
0	$[1]$	$[1]$
1	$[1, x_1]$	
2	$[1, x_1, x_1^2]$	
3	$[1, x_1, x_1^2, x_1^3]$	

Nonlinear boundaries

- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

order (*k*)	terms when *d*=1	terms for general d
0	$[1]$	$[1]$
1	$[1, x_1]$	$[1, x_1, \ldots, x_d]$
2	$[1, x_1, x_1^2]$	
3	$[1, x_1, x_1^2, x_1^3]$	

Nonlinear boundaries

- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

order (*k*)	terms when *d*=1	terms for general d
0	$[1]$	$[1]$
1	$[1, x_1]$	$[1, x_1, \ldots, x_d]$
2	$[1, x_1, x_1^2]$	$[1, x_1, \ldots, x_d,$ $x_1^2, x_1 x_2, \ldots, x_{d-1} x_d, x_d^2]$
3	$[1, x_1, x_1^2, x_1^3]$	

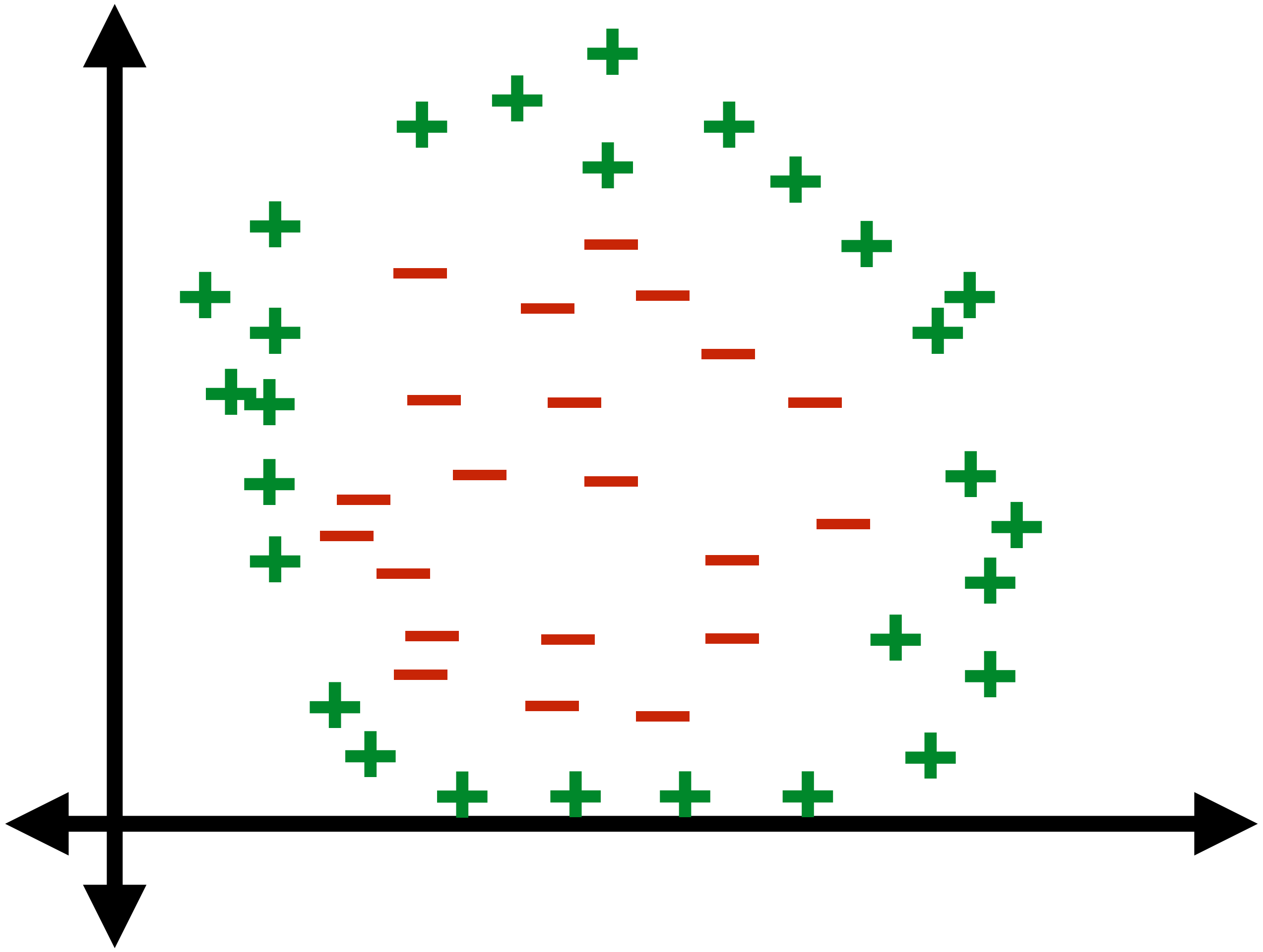
Nonlinear boundaries

- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

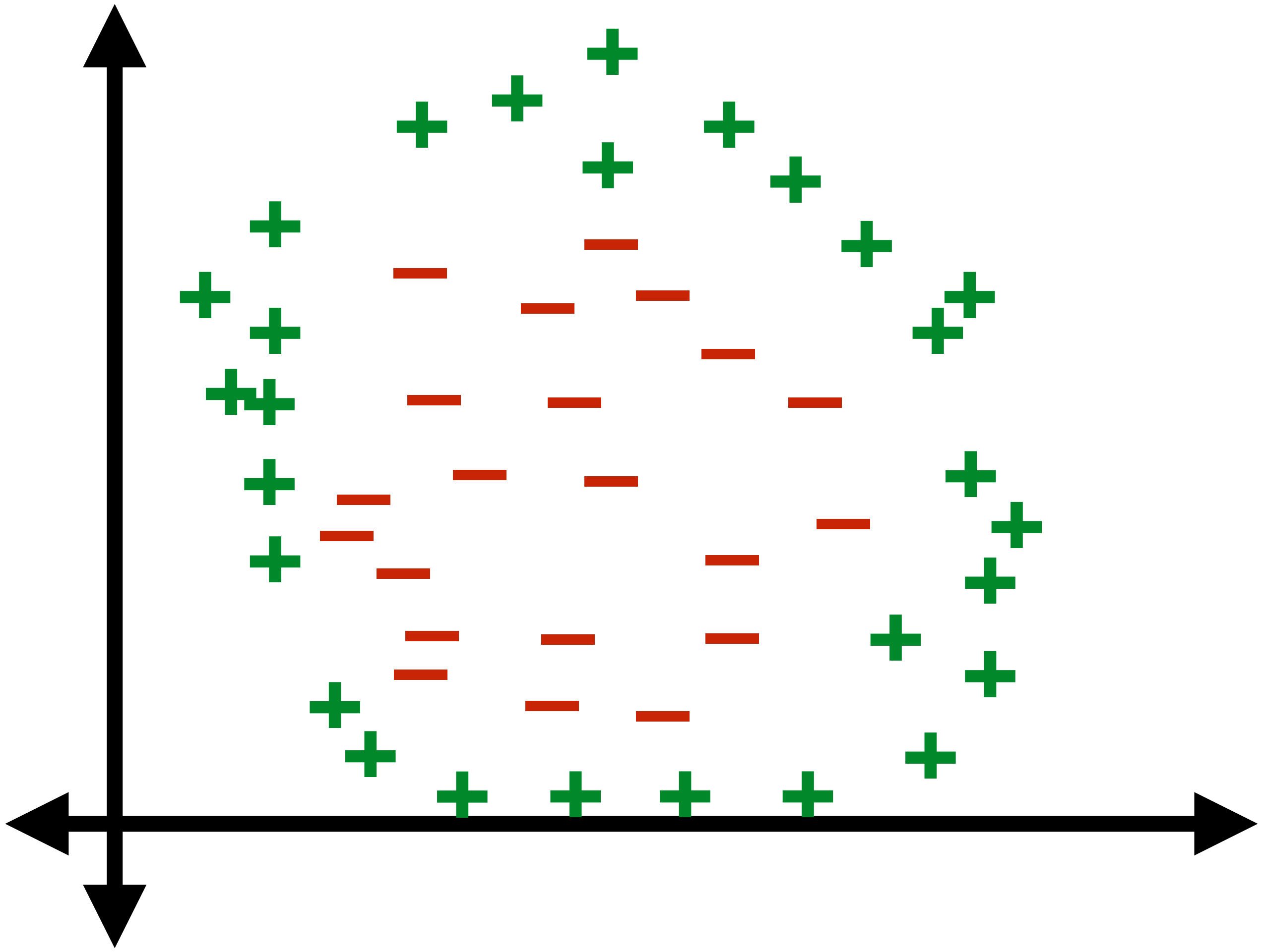
order (*k*)	terms when *d*=1	terms for general d
0	$[1]$	$[1]$
1	$[1, x_1]$	$[1, x_1, \ldots, x_d]$
2	$[1, x_1, x_1^2]$	$[1, x_1, \ldots, x_d,$ $x_1^2, x_1 x_2, \ldots, x_{d-1} x_d, x_d^2]$
3	$[1, x_1, x_1^2, x_1^3]$	$[1, x_1, \ldots, x_d,$ $x_1^2, x_1 x_2, \ldots, x_{d-1} x_d, x_d^2,$ $x_1^3, x_1^2 x_2, x_1 x_2 x_3, \ldots, x_d^3]$

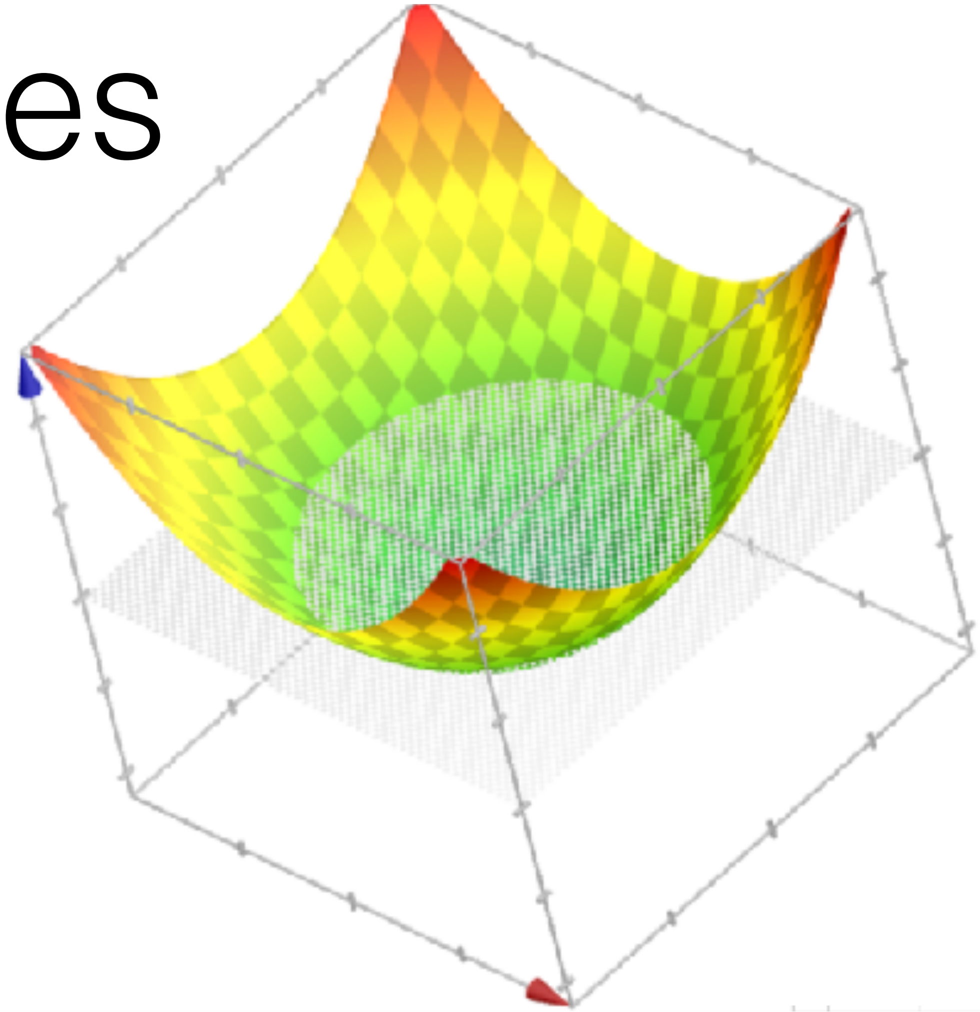
Nonlinear boundaries

Nonlinear boundaries

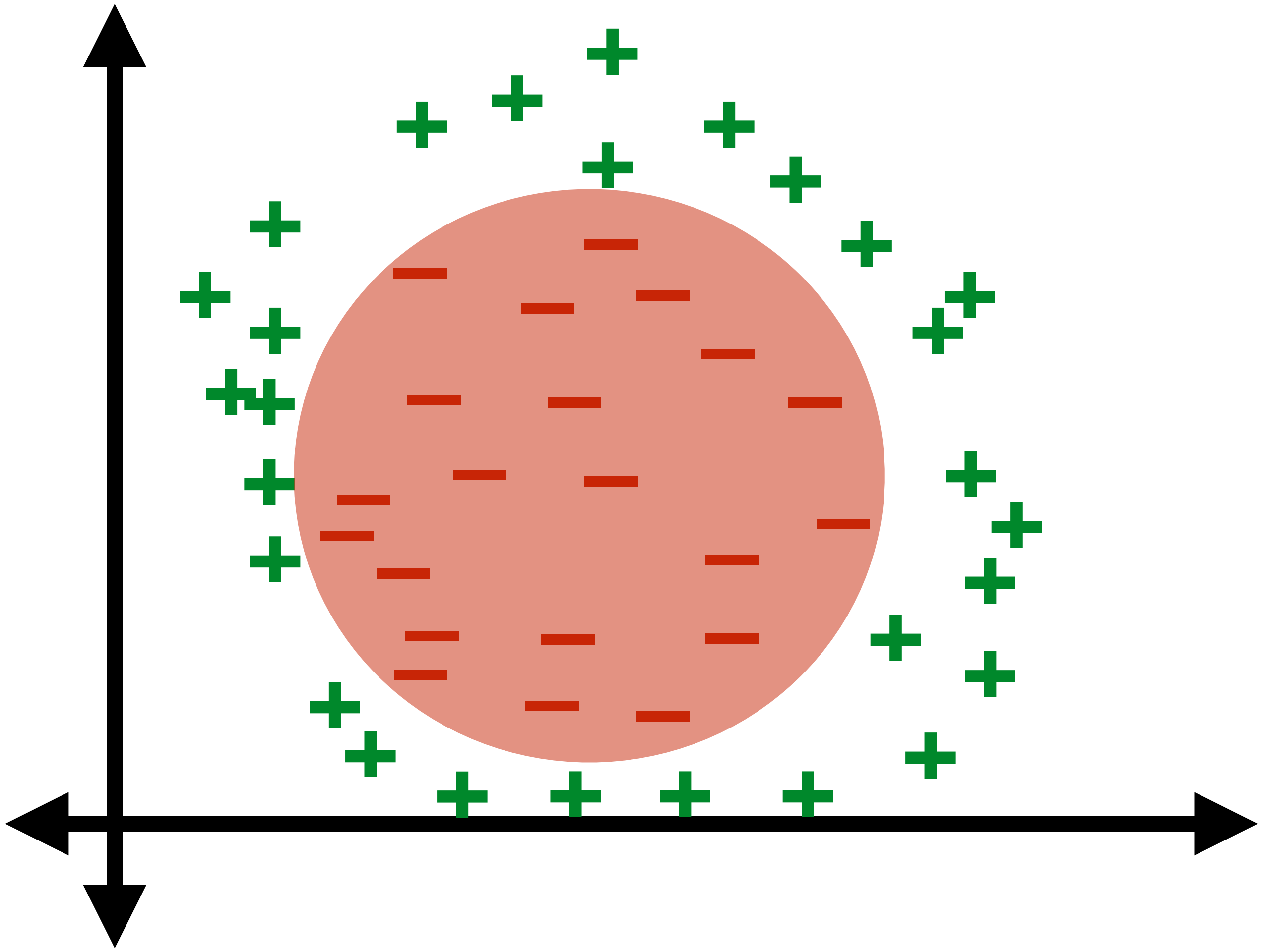


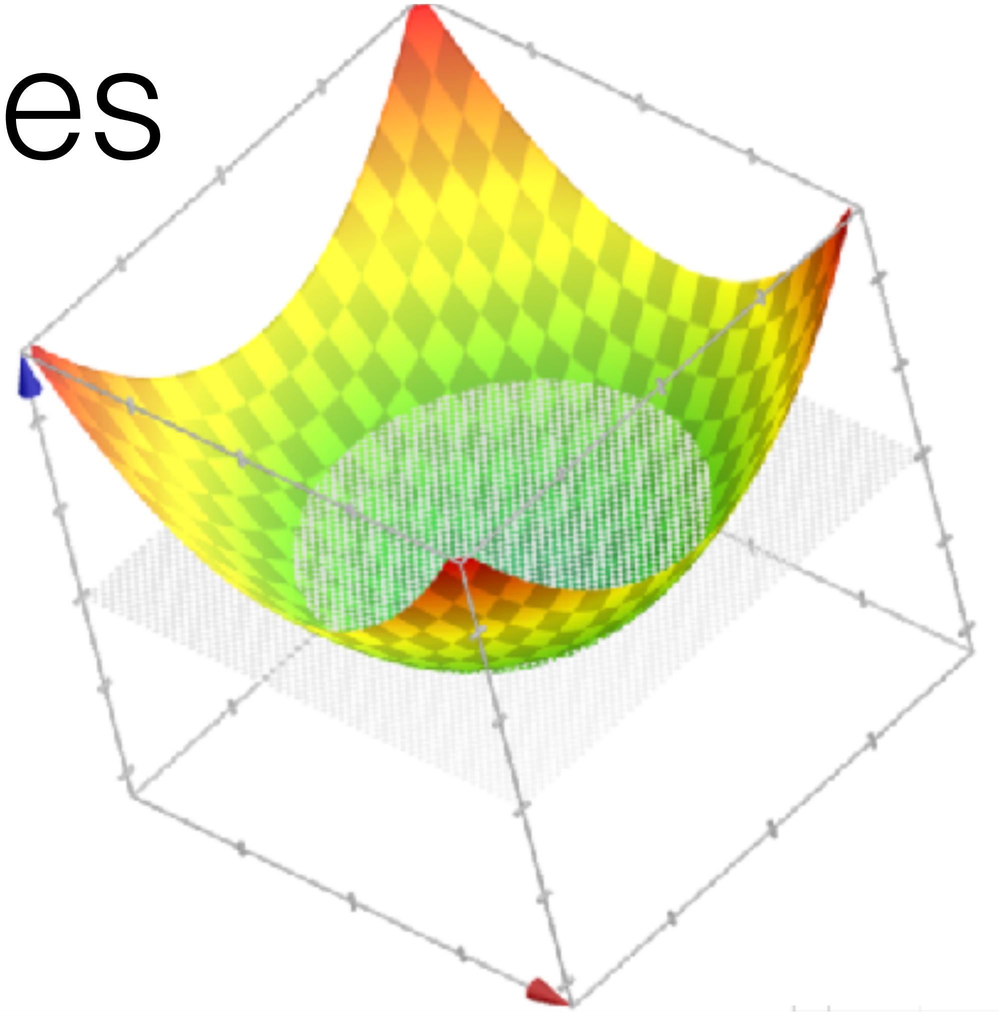
Nonlinear boundaries



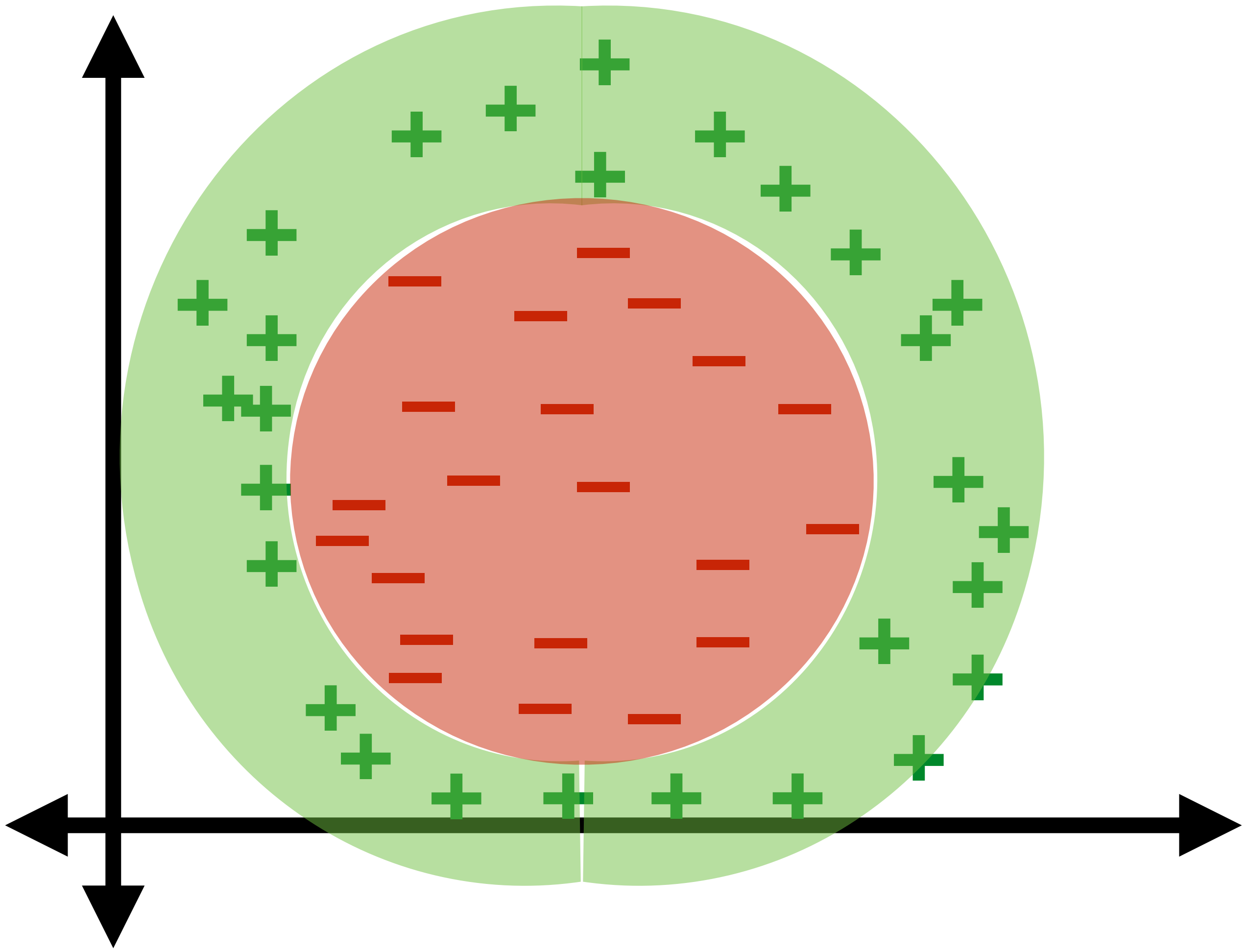


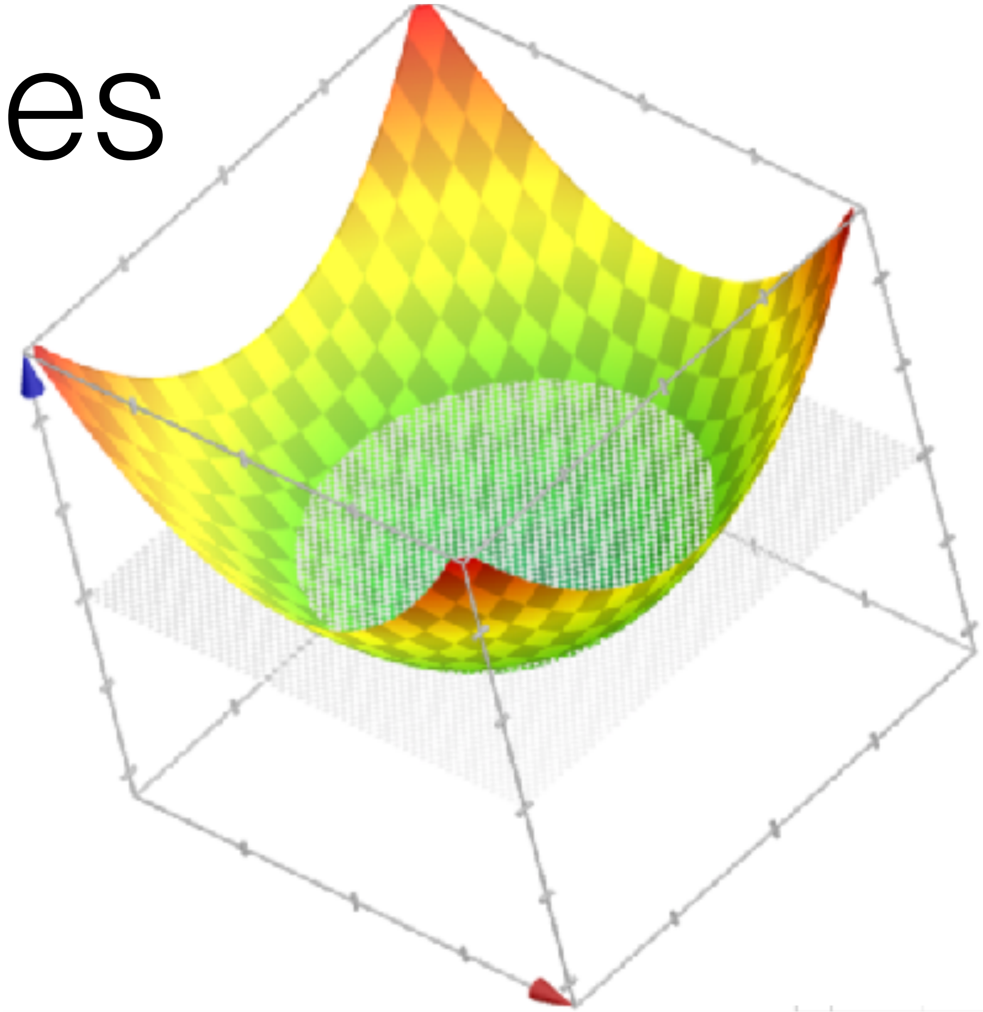
Nonlinear boundaries



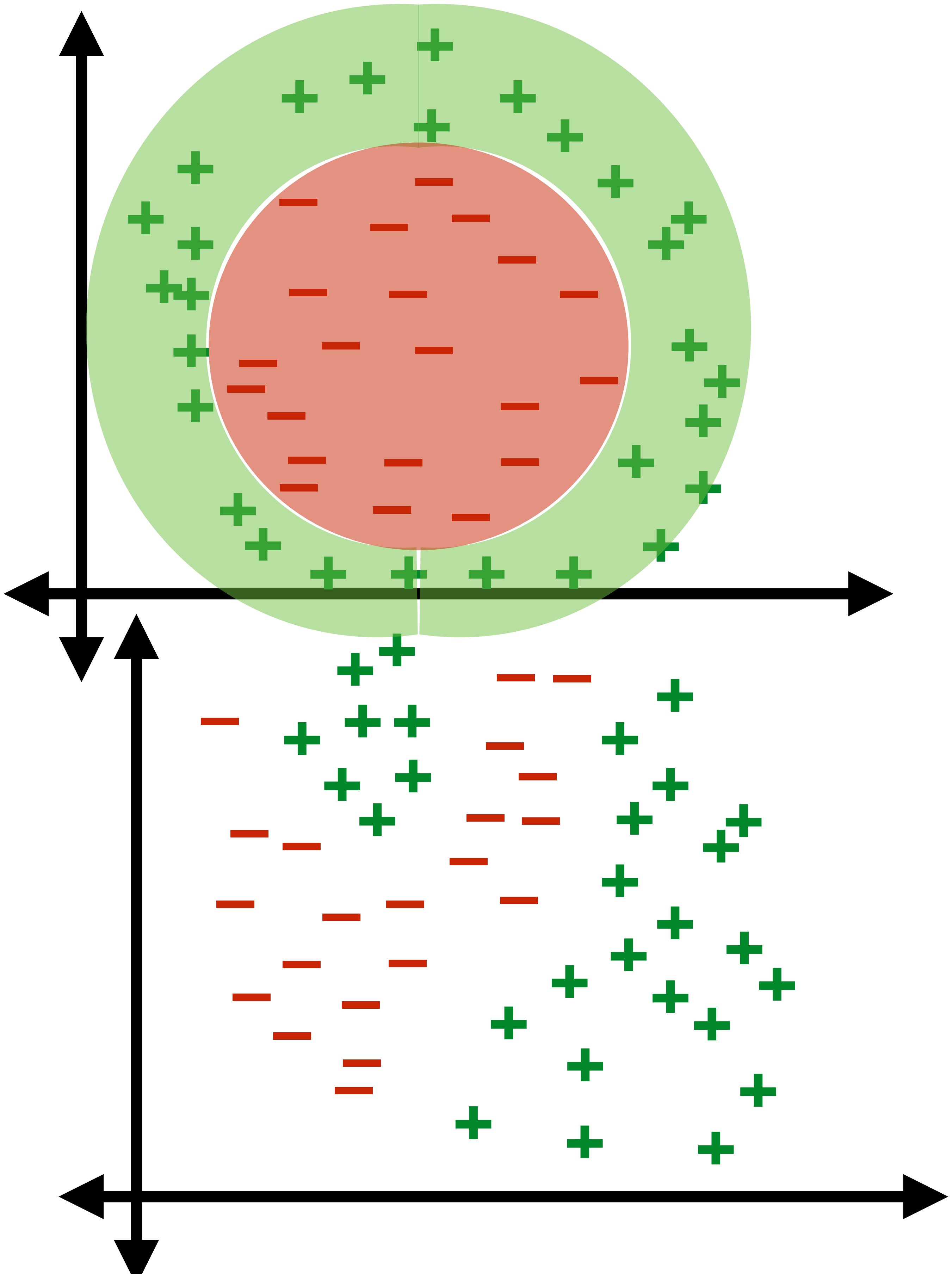


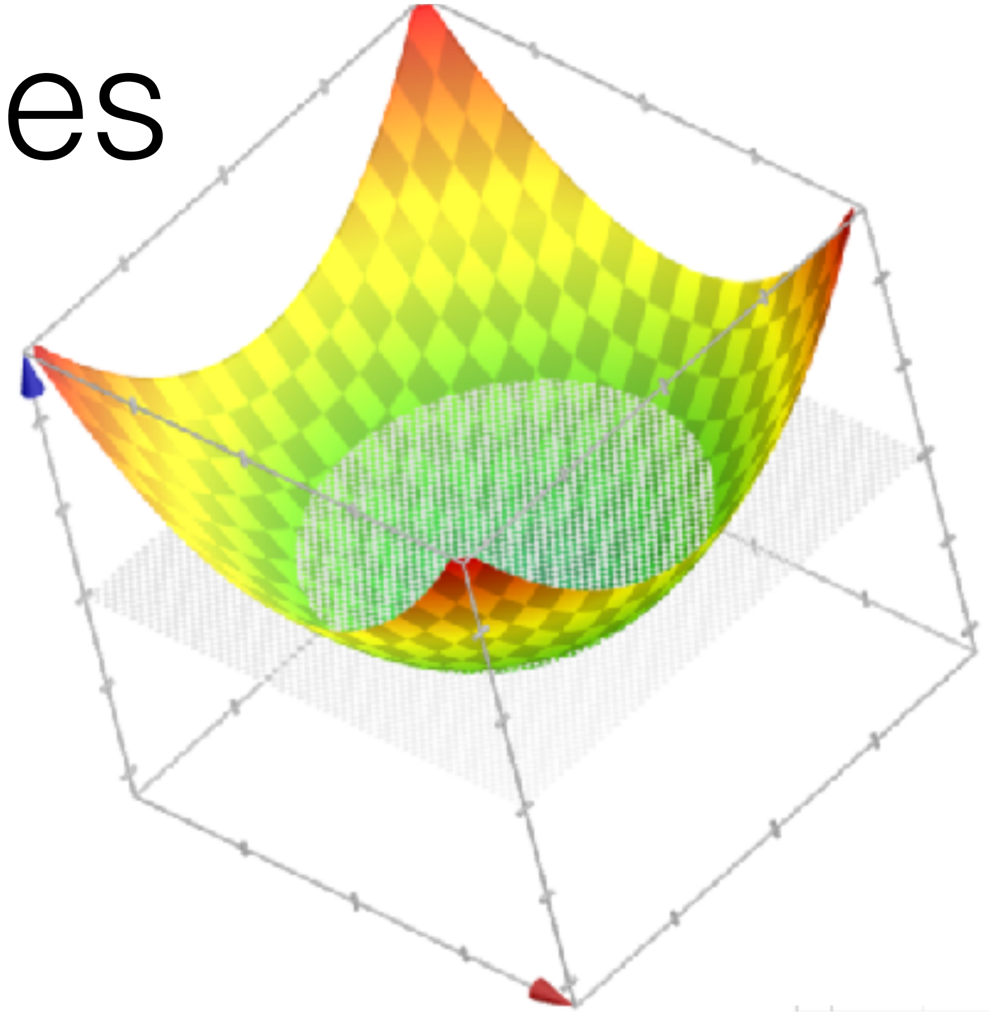
Nonlinear boundaries



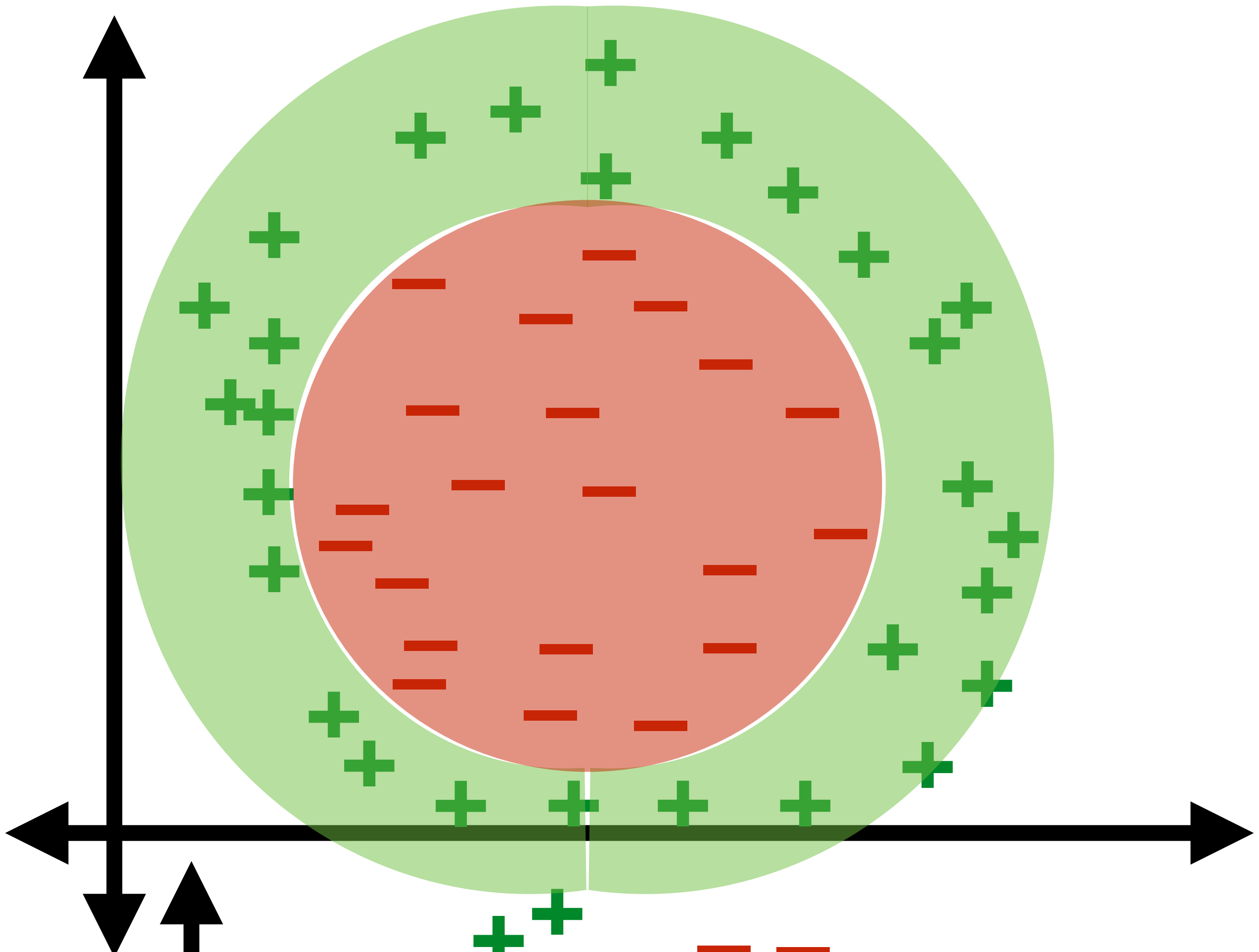


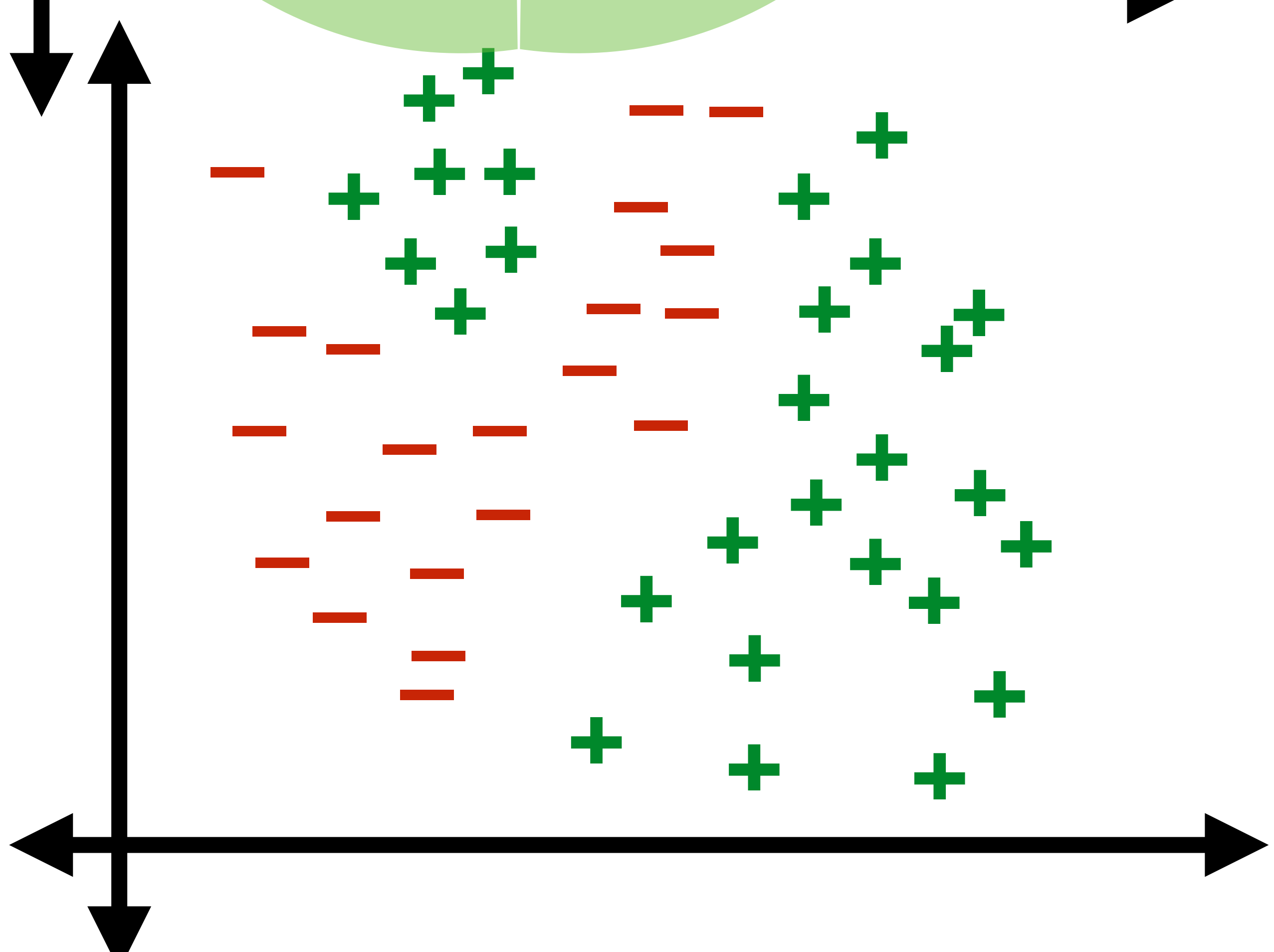
Nonlinear boundaries

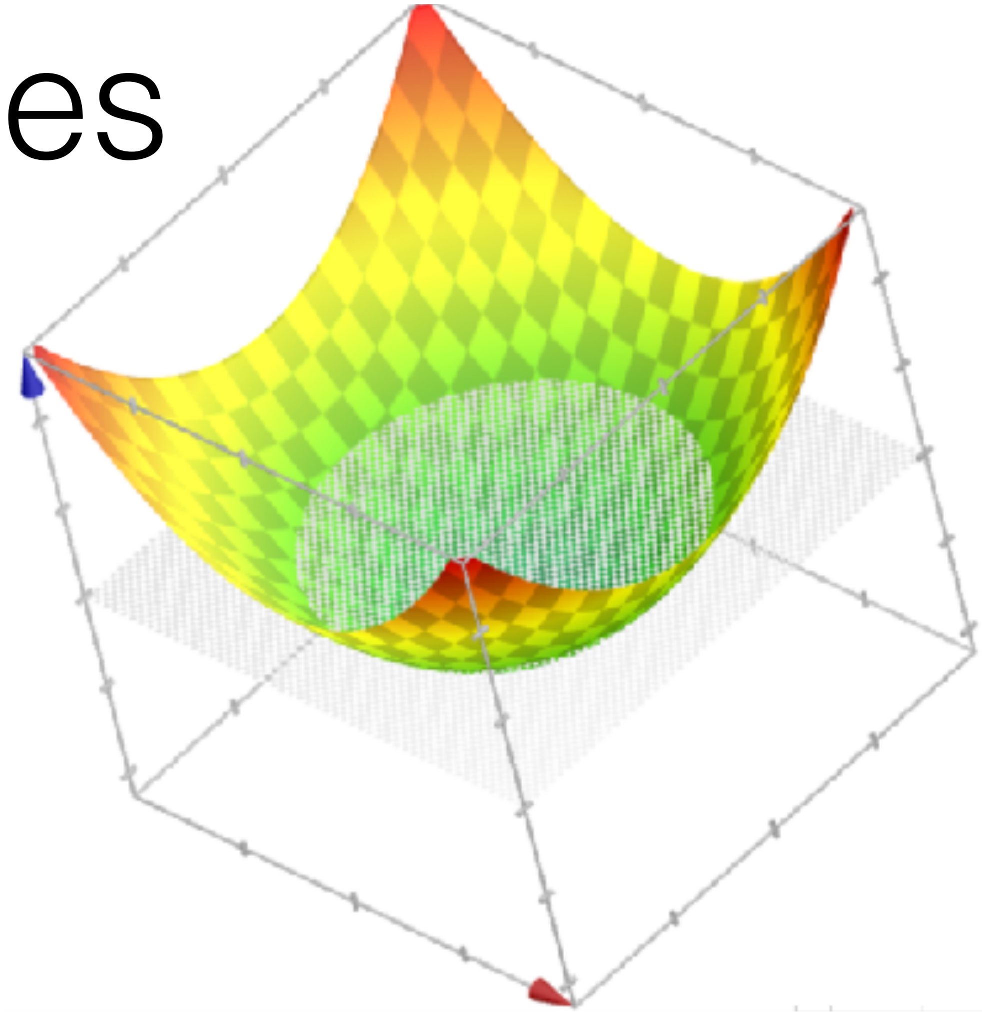


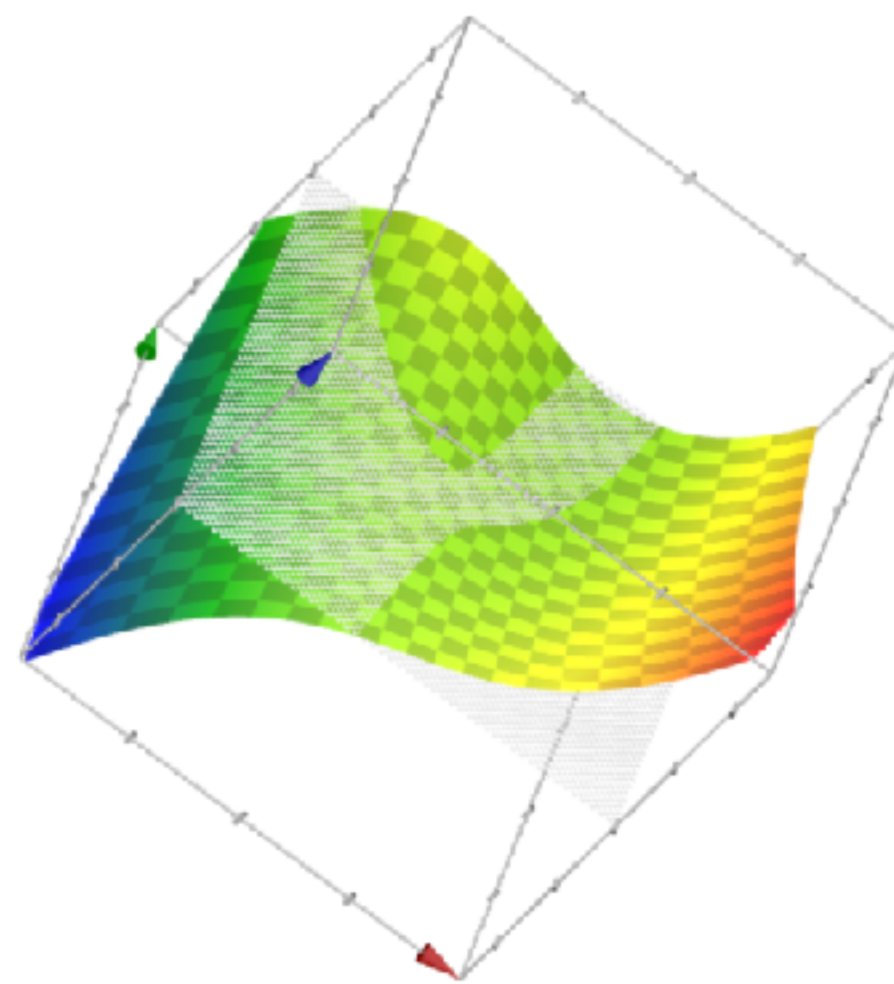


Nonlinear boundaries

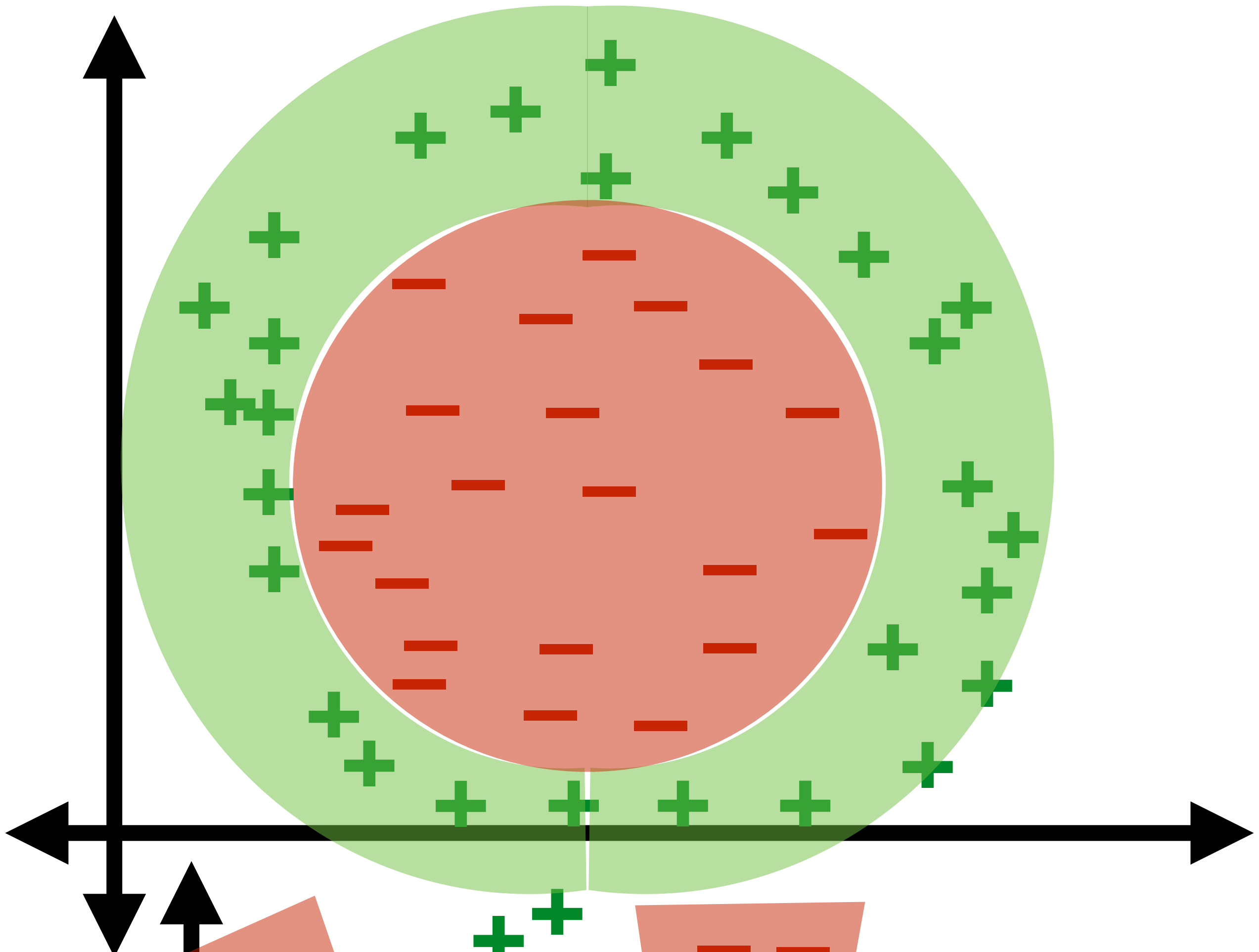


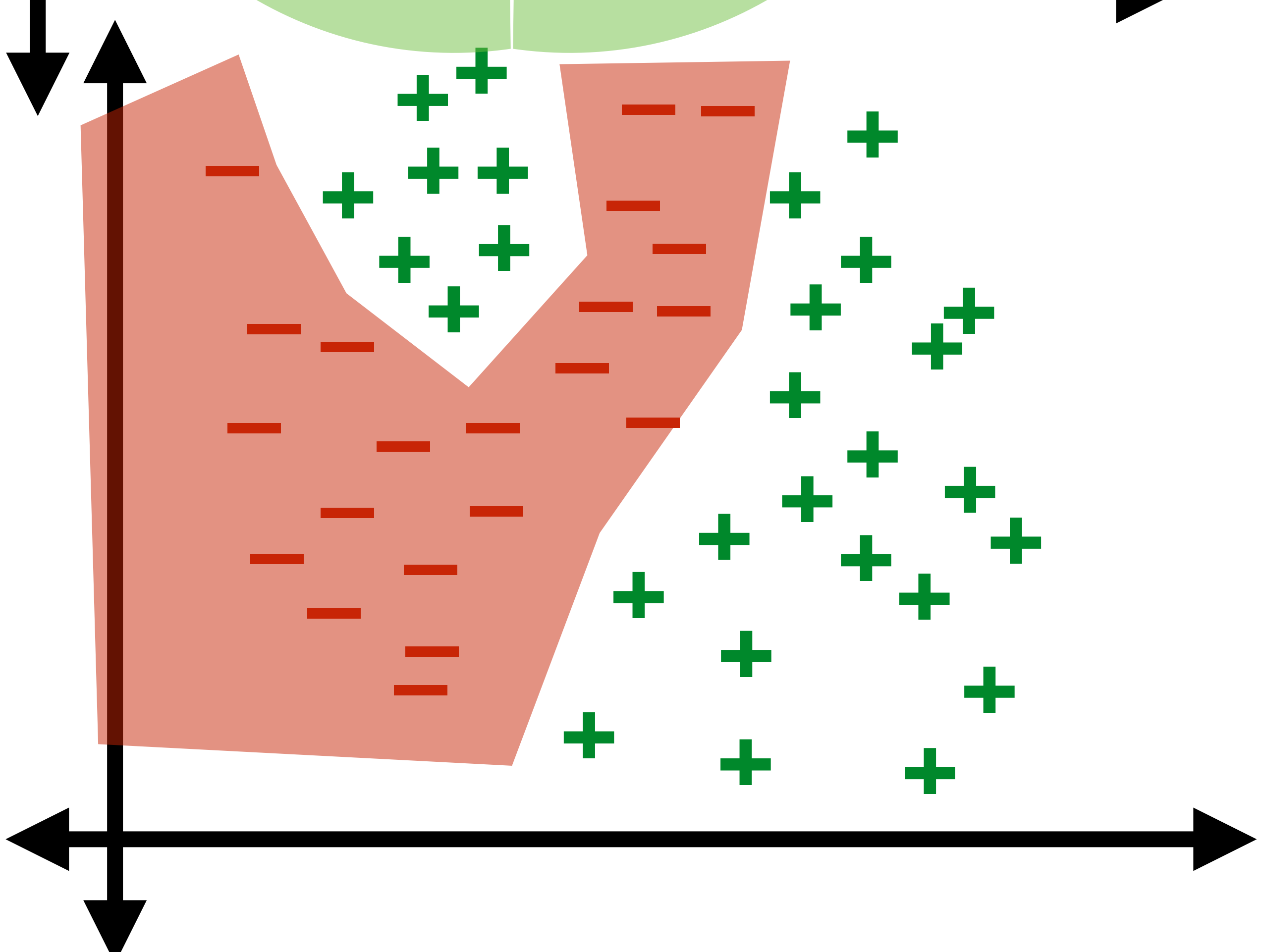


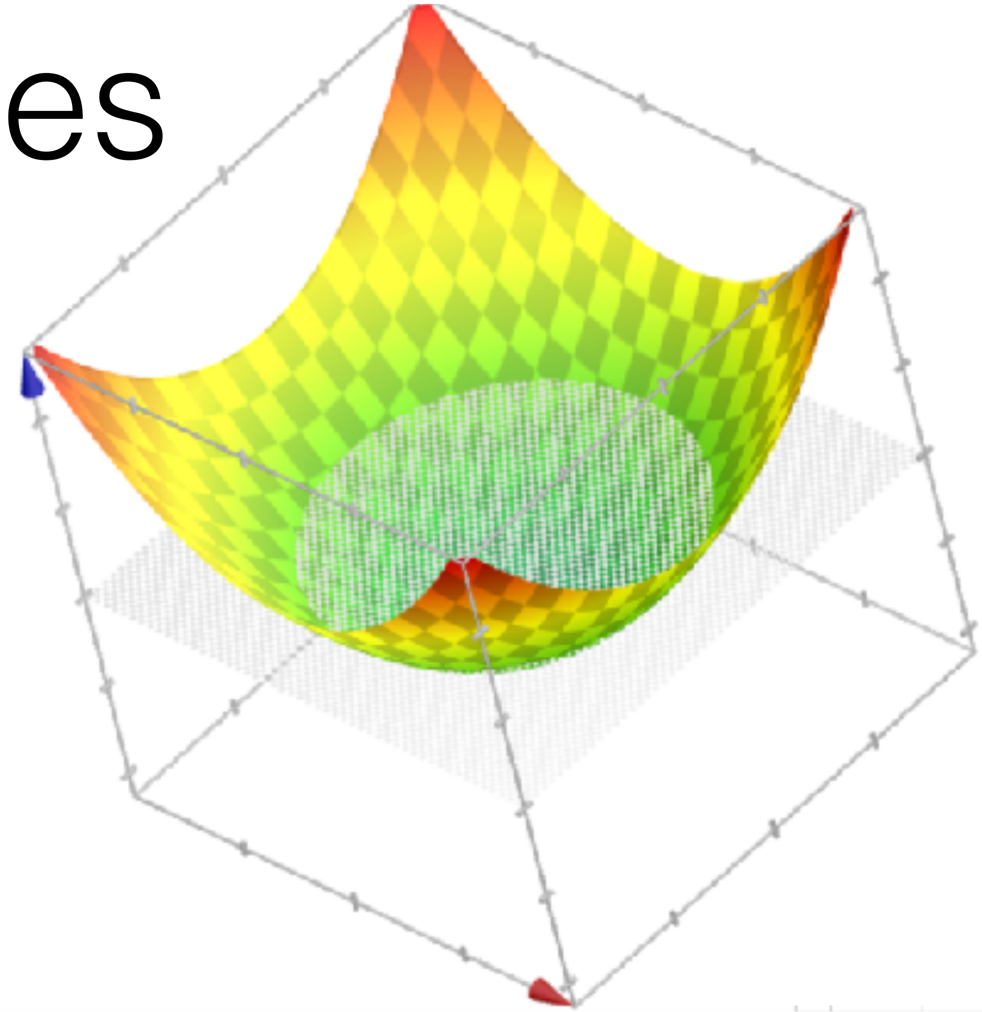


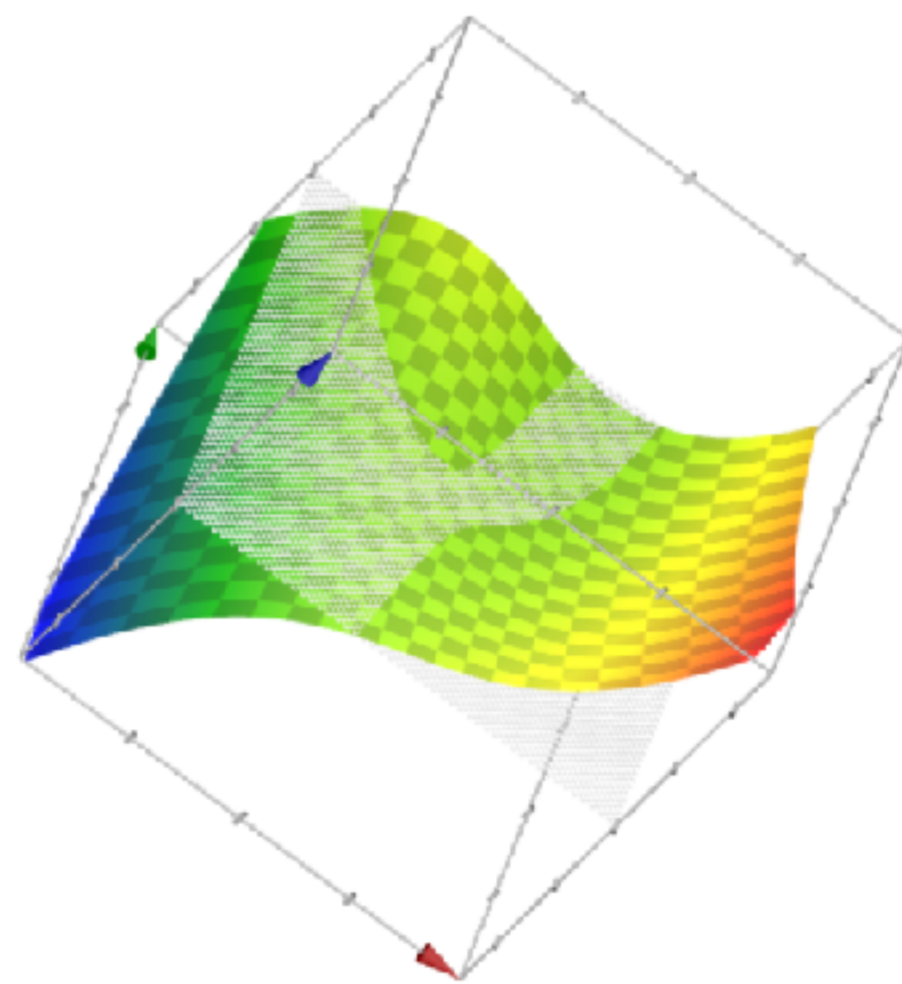


Nonlinear boundaries

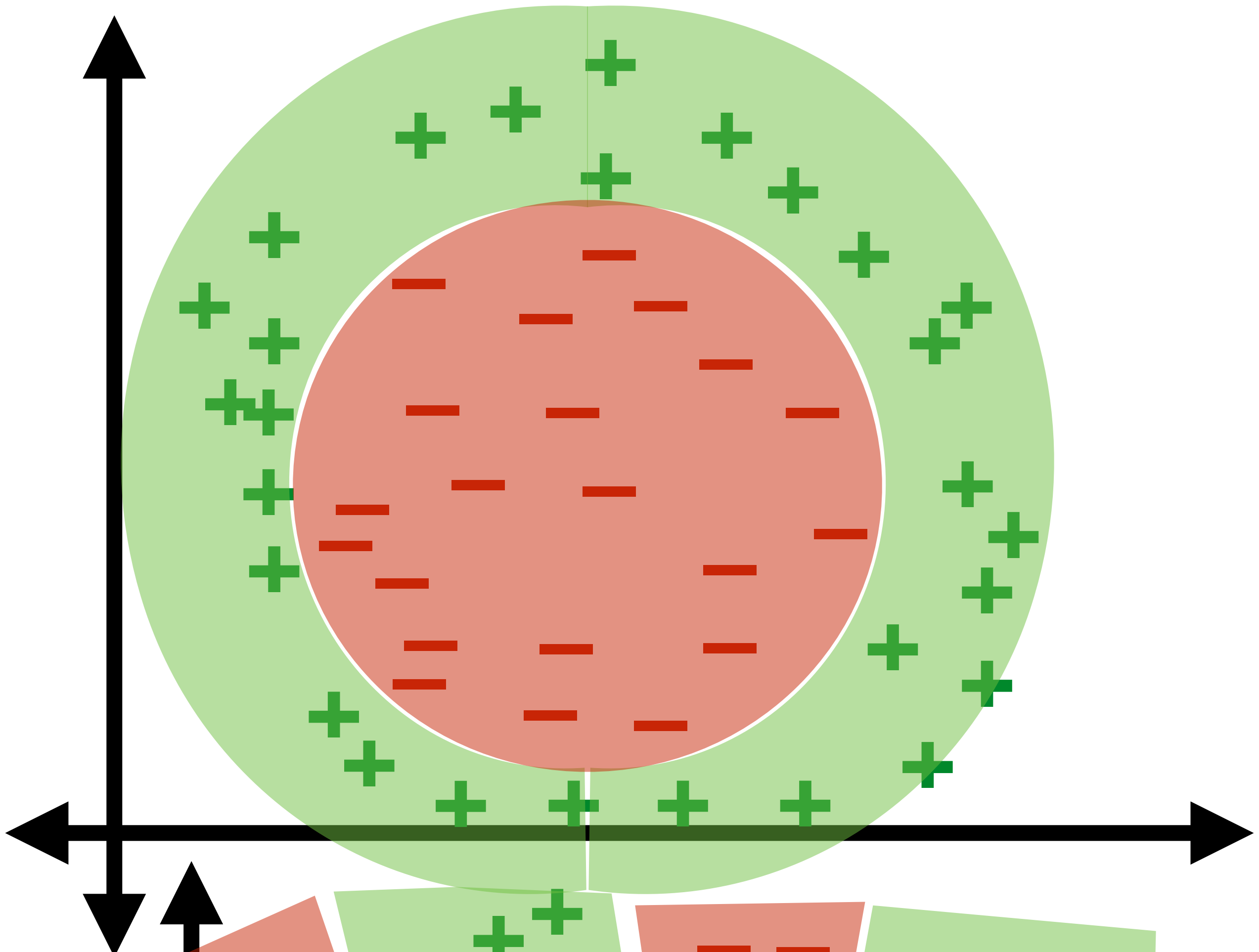


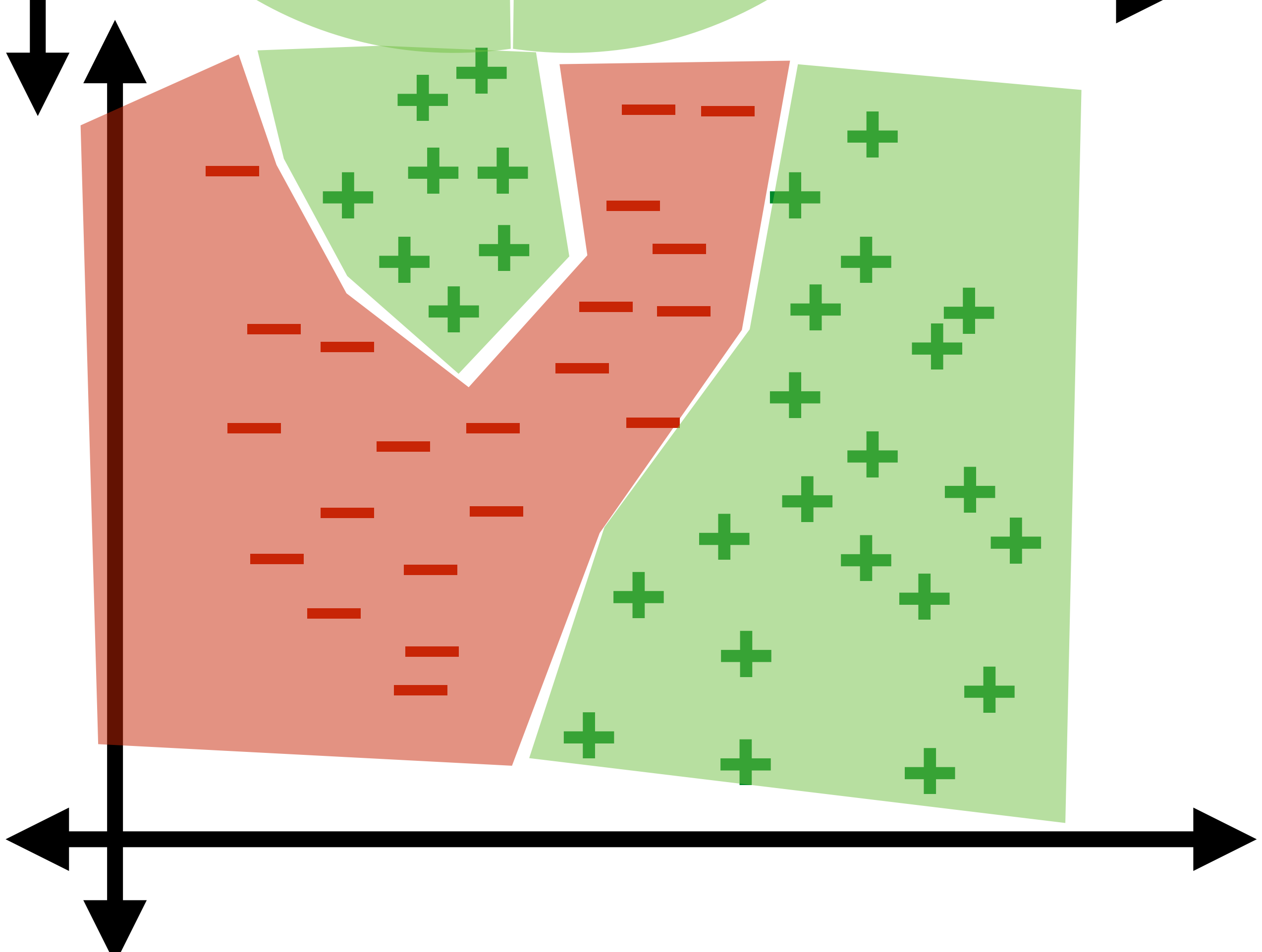


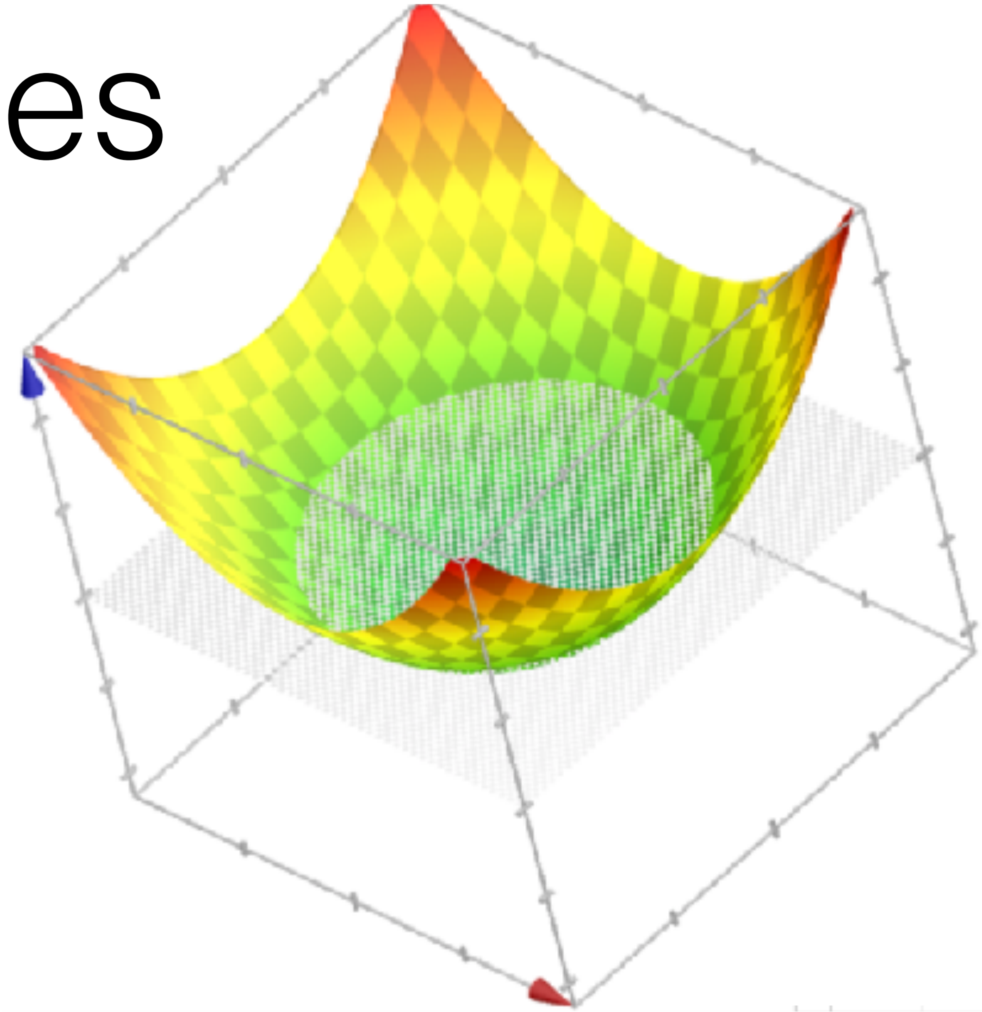


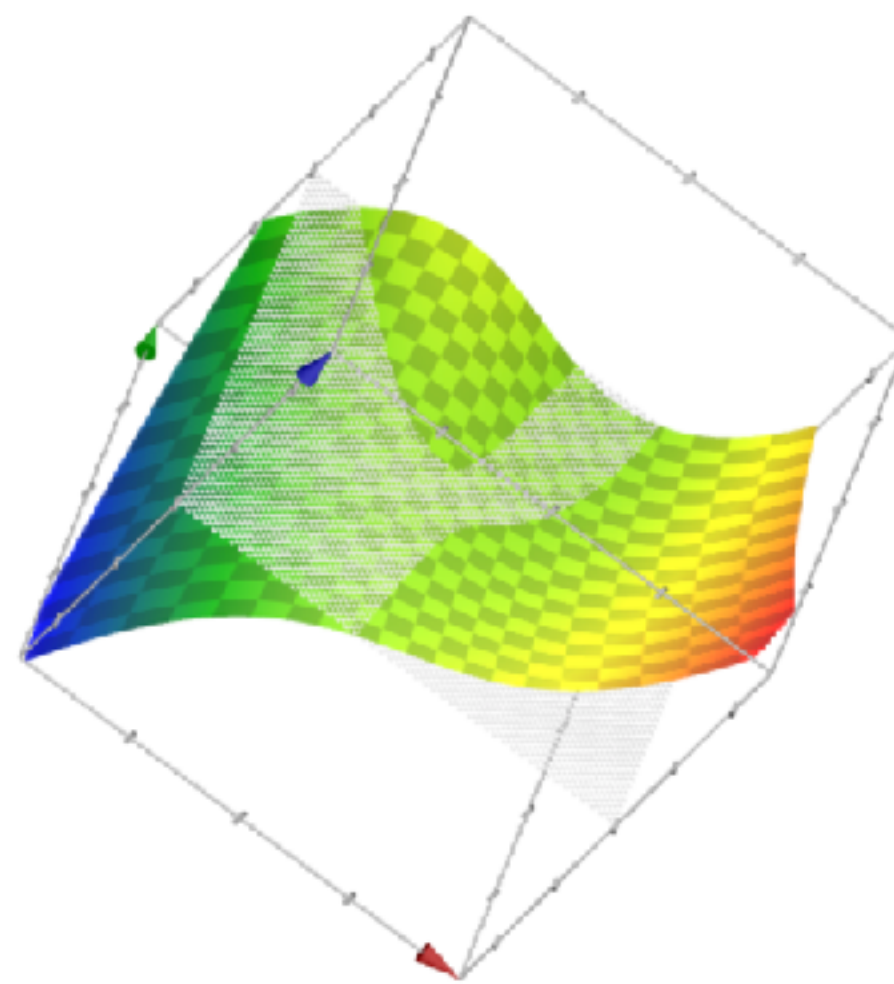


Nonlinear boundaries

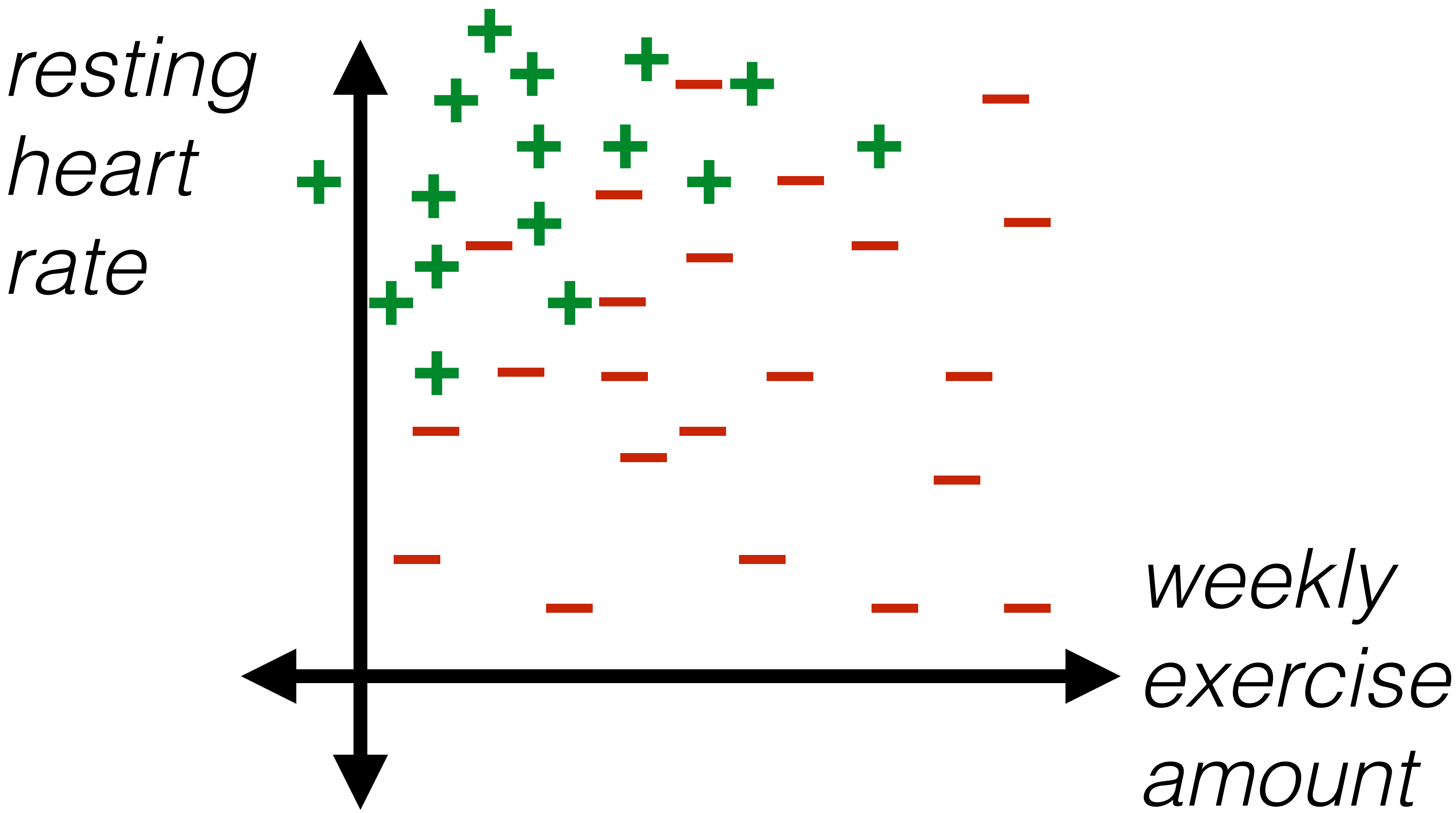




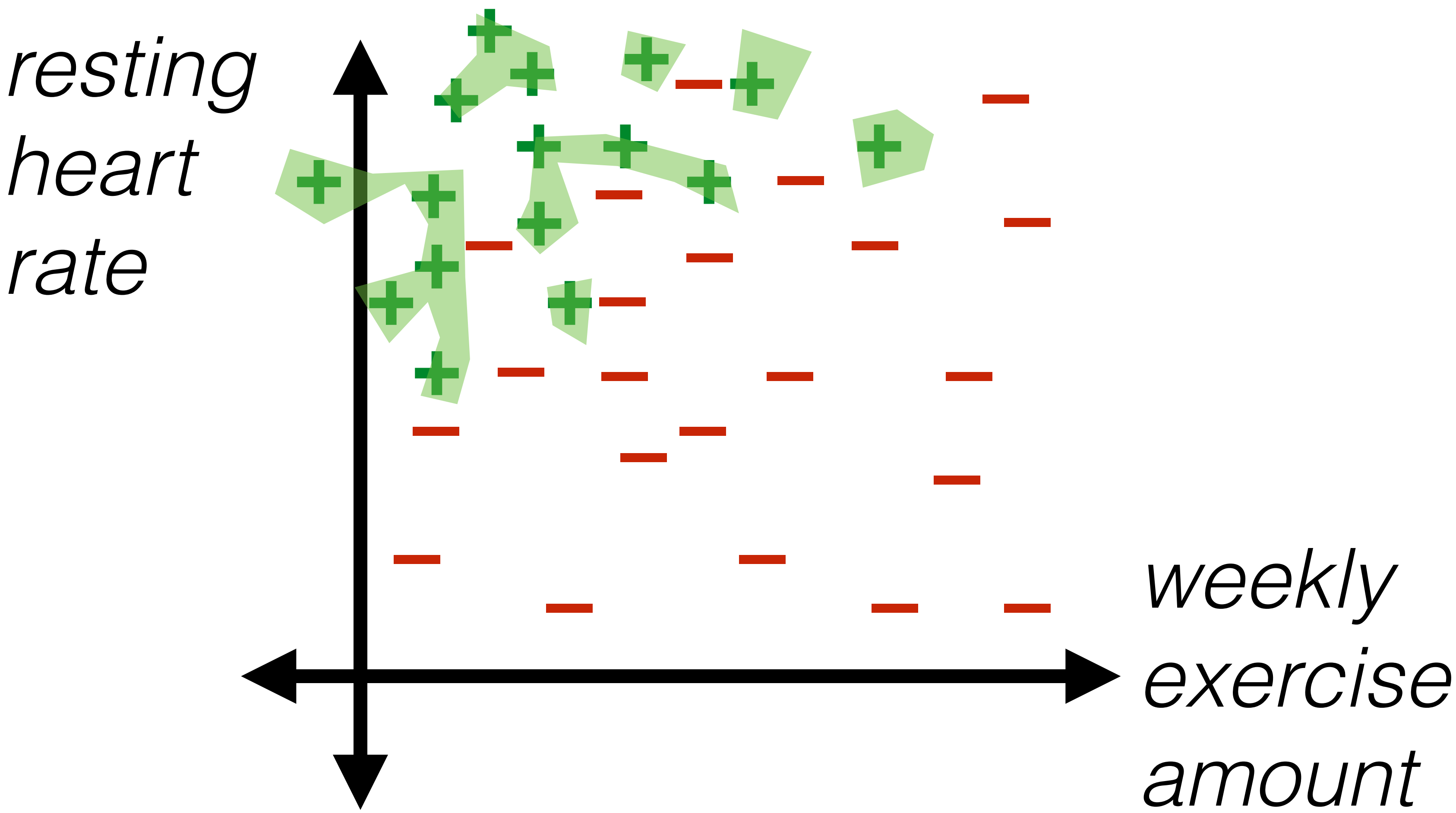




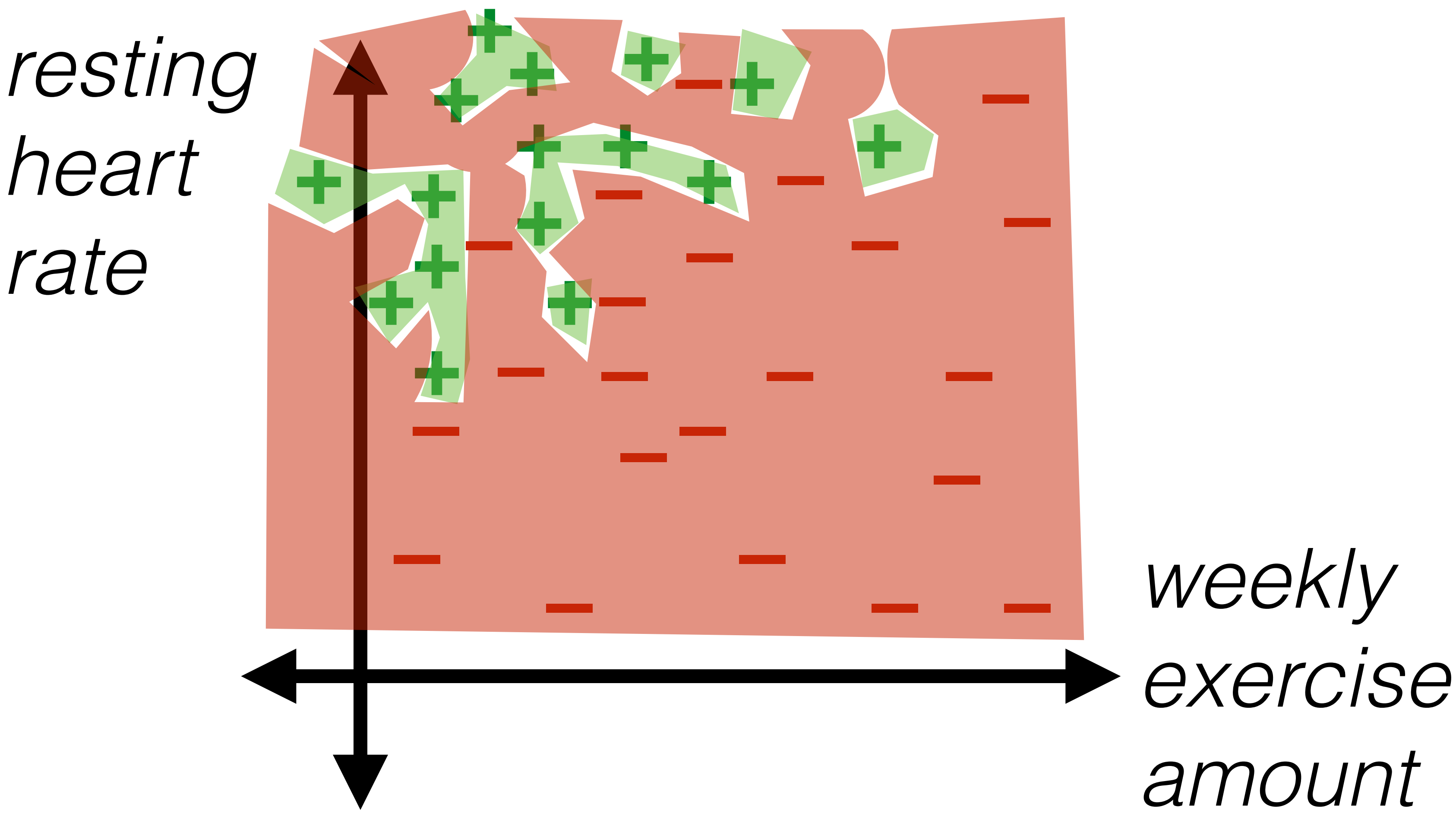
Nonlinear boundaries



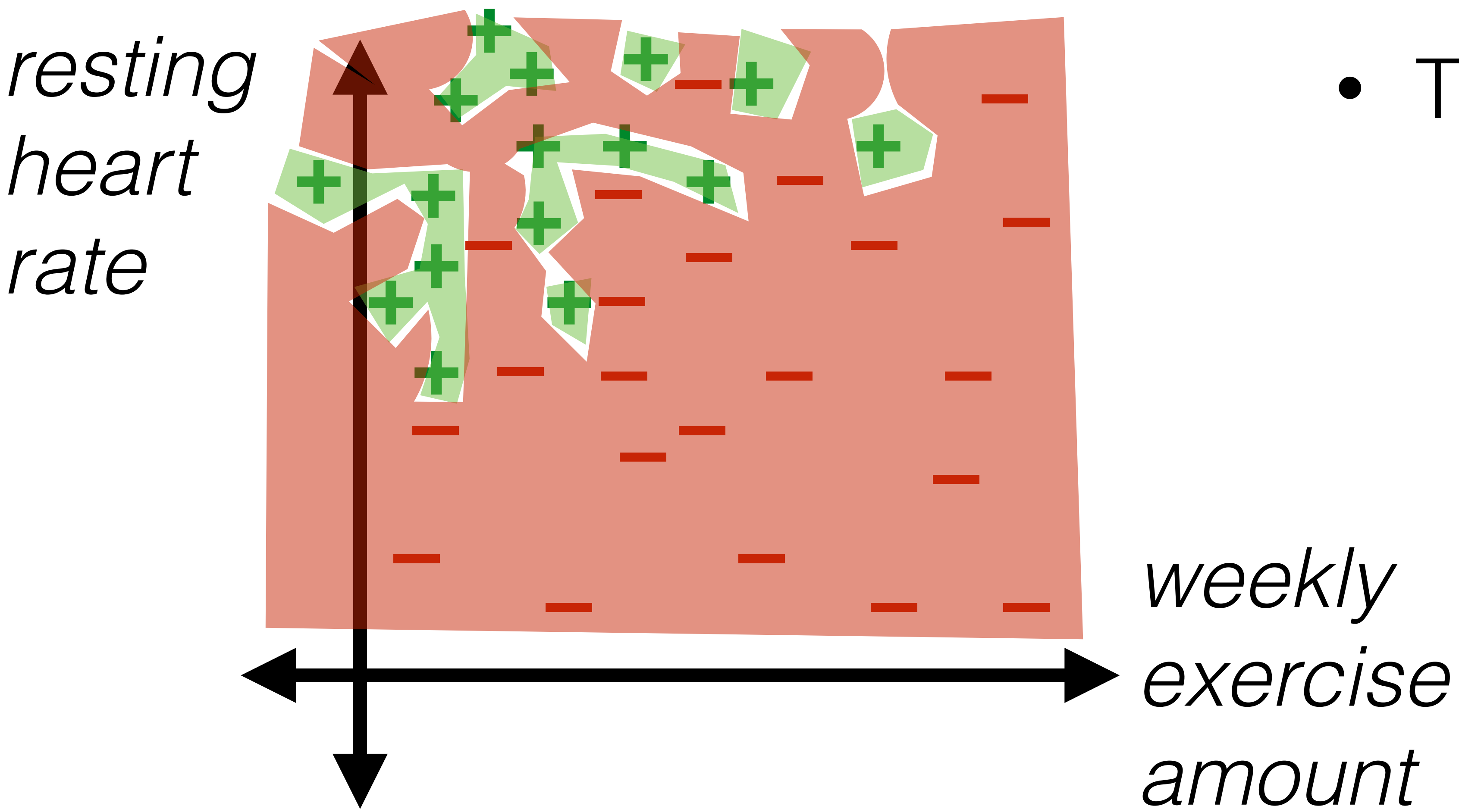
Nonlinear boundaries



Nonlinear boundaries

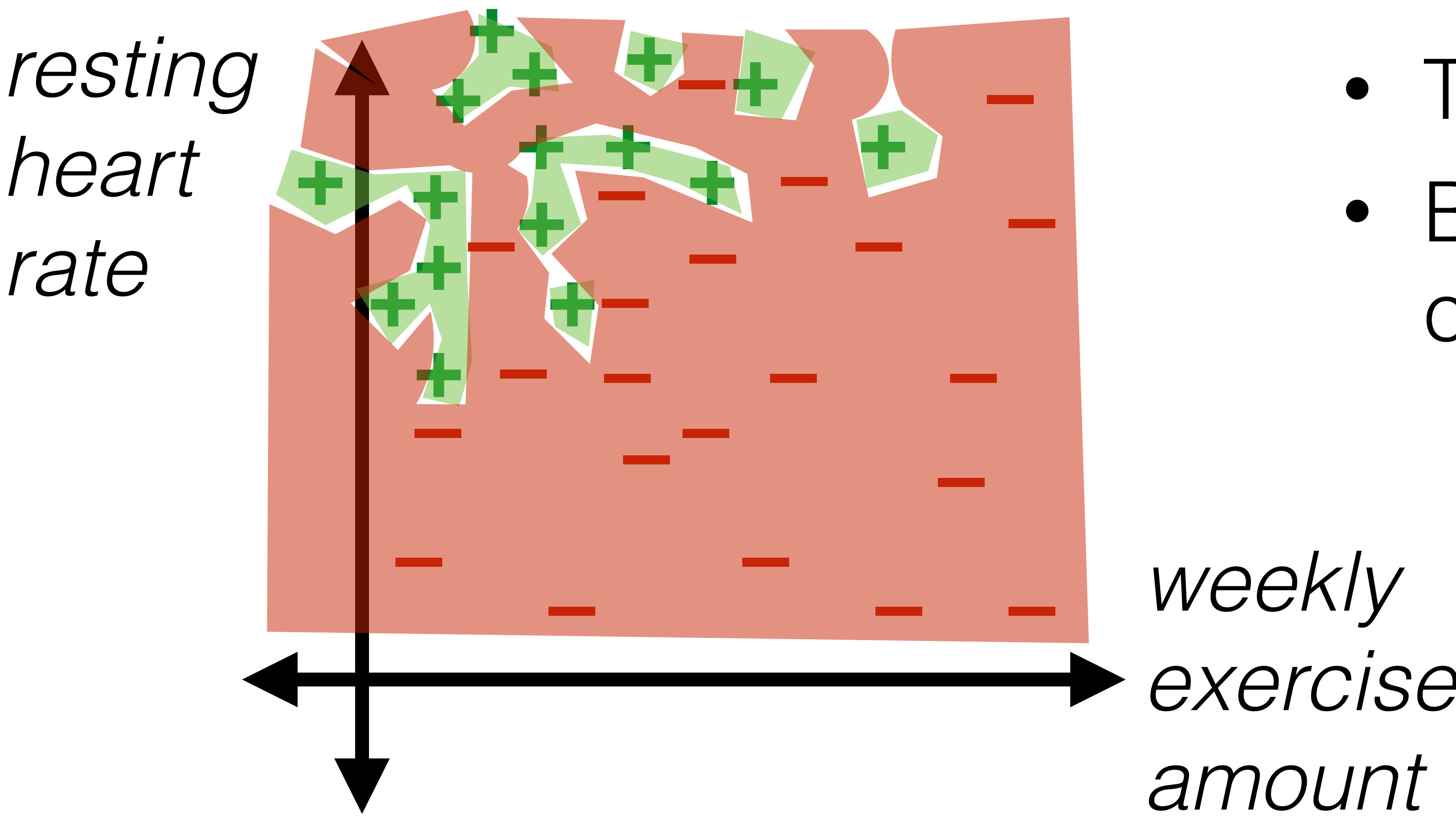


Nonlinear boundaries



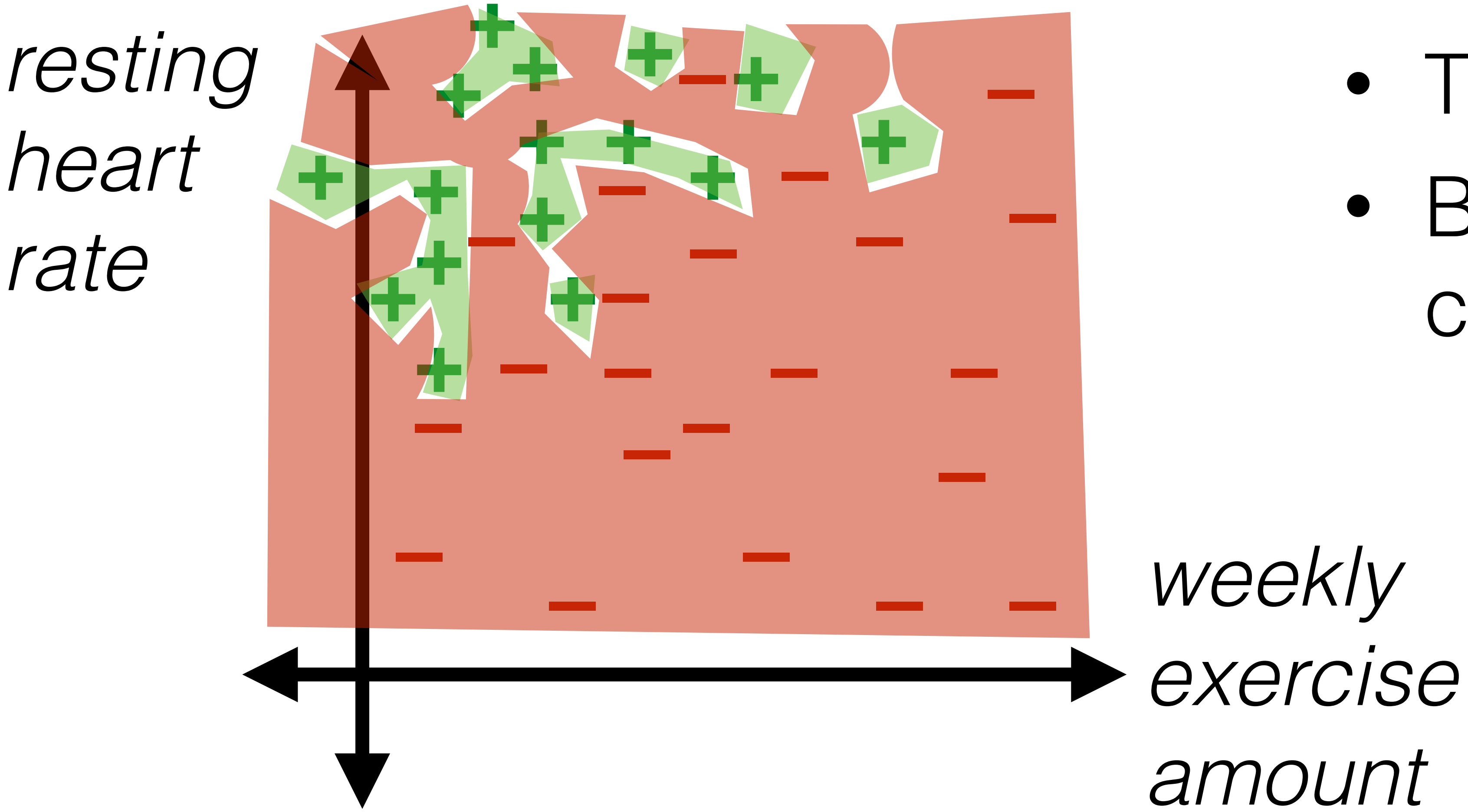
- Training error is 0!

Nonlinear boundaries



- Training error is 0!
- But seems like our classifier is overfitting

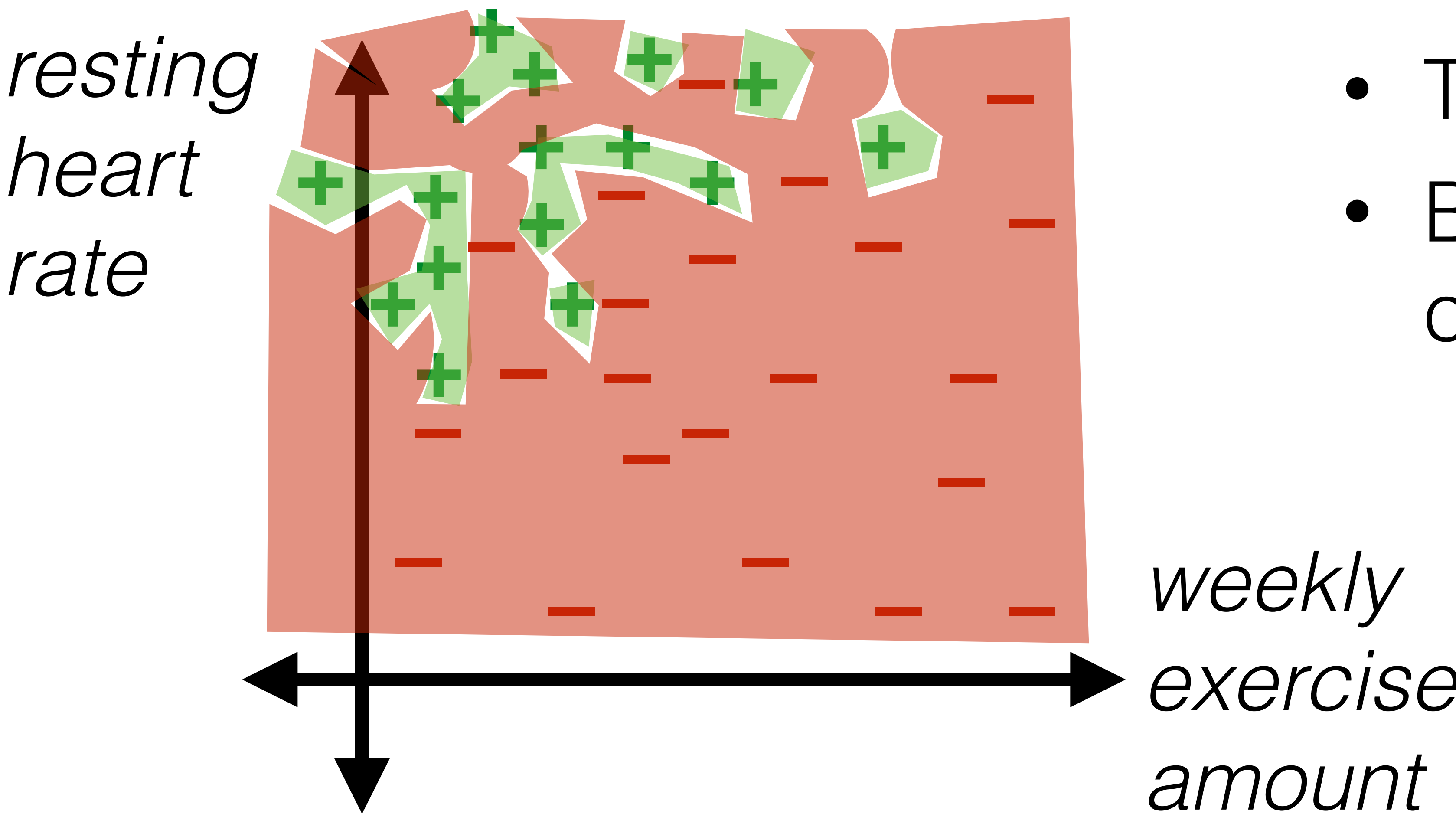
Nonlinear boundaries



- Training error is 0!
- But seems like our classifier is overfitting

- How can we detect overfitting?

Nonlinear boundaries



- Training error is 0!
- But seems like our classifier is overfitting

- How can we detect overfitting?
- How can we avoid overfitting?

Evaluation of a learning algorithm

- How good is our learning algorithm on data like ours?

Evaluation of a learning algorithm

- How good is our learning algorithm on data like ours?

- Idea: use full data for training and then report training error

Evaluation of a learning algorithm

- How good is our learning algorithm on data like ours?

- Idea: use full data for training and then report training error
- Idea: reserve some data for testing

Evaluation of a learning algorithm

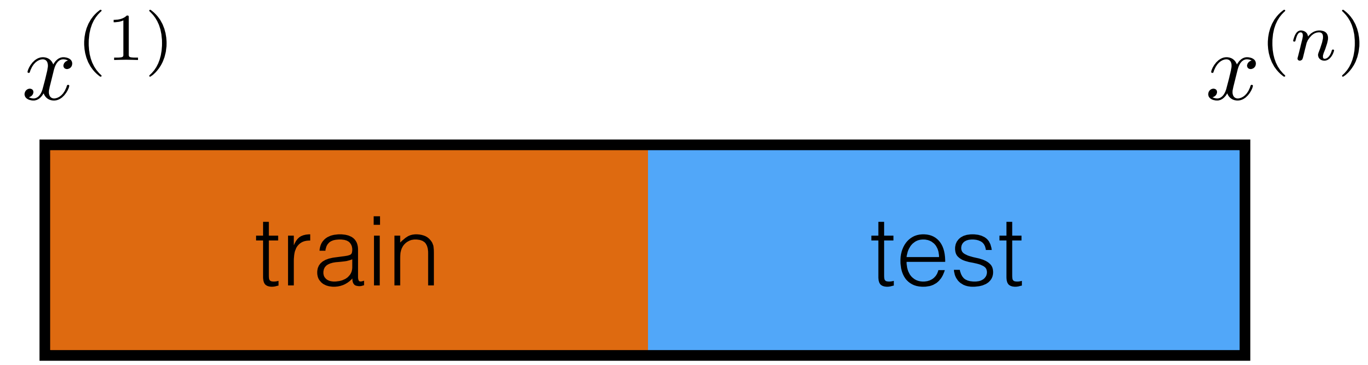
- How good is our learning algorithm on data like ours?
- Idea: use full data for training and then report training error
- Idea: reserve some data for testing

Evaluation of a learning algorithm

- How good is our learning algorithm on data like ours?

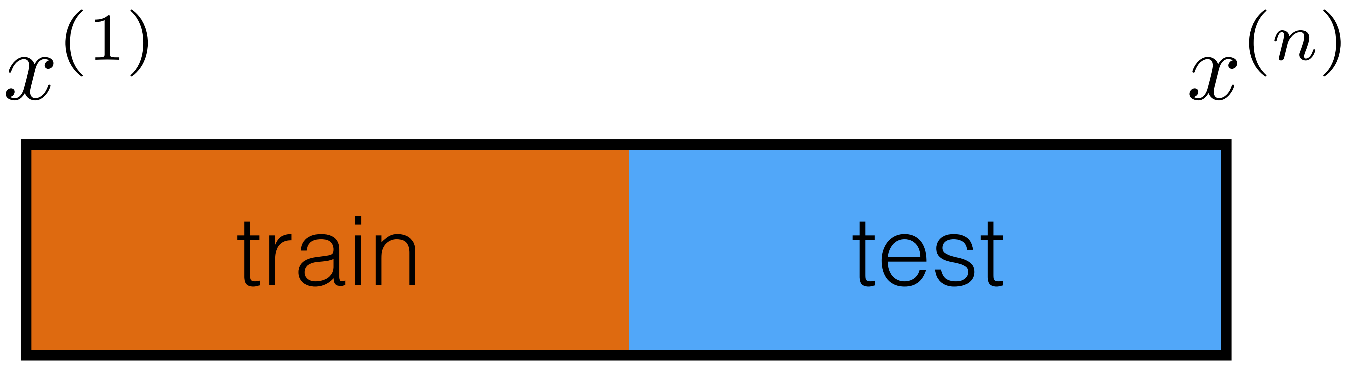
- Idea: use full data for training and then report training error
- Idea: reserve some data for testing

Evaluation of a learning algorithm



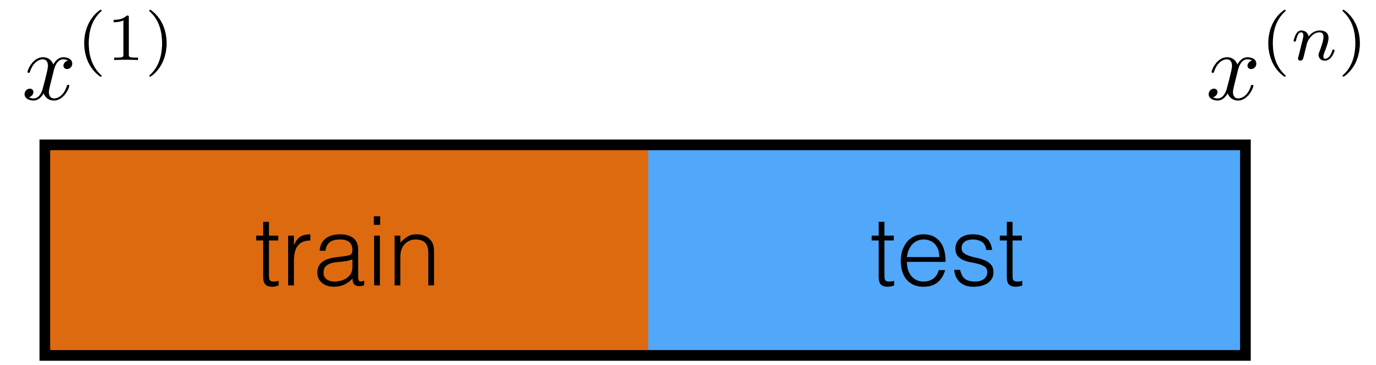
- How good is our learning algorithm on data like ours?
- Idea: use full data for training and then report training error
- Idea: reserve some data for testing

Evaluation of a learning algorithm



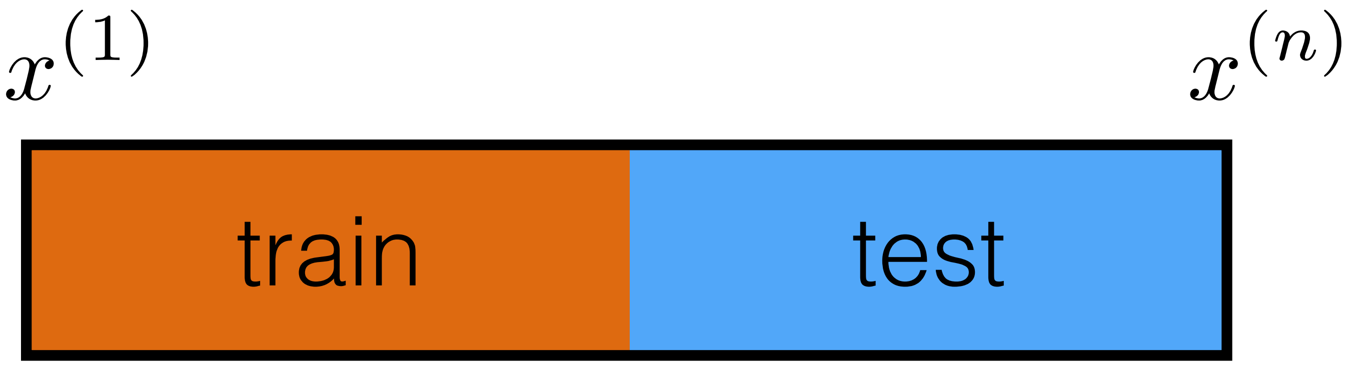
- How good is our learning algorithm on data like ours?
- Idea: use full data for training and then report training error
- Idea: reserve some data for testing
  - More training data: closer to training on full data

Evaluation of a learning algorithm



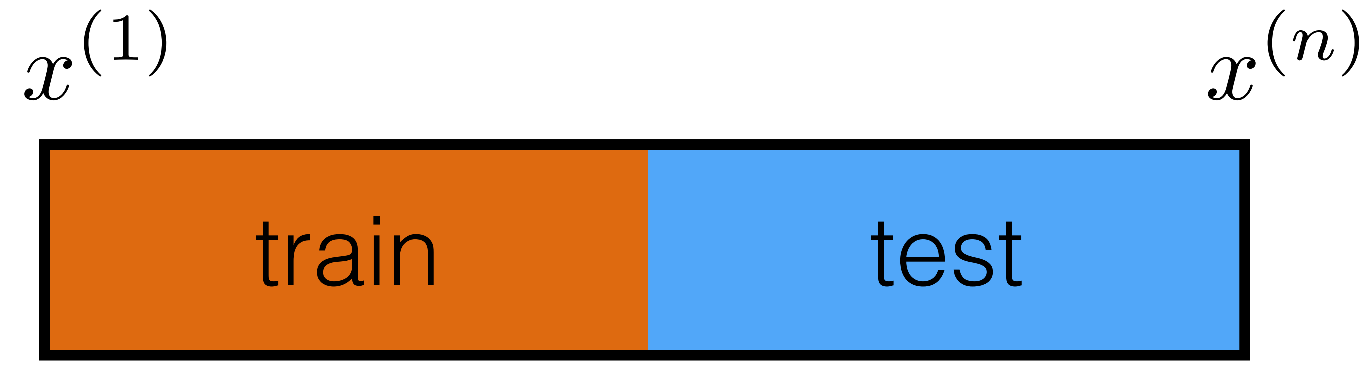
- How good is our learning algorithm on data like ours?
- Idea: use full data for training and then report training error
- Idea: reserve some data for testing
  - More training data: closer to training on full data
  - More testing data: less noisy estimate of performance

Evaluation of a learning algorithm



- How good is our learning algorithm on data like ours?
- Idea: use full data for training and then report training error
- Idea: reserve some data for testing
  - More training data: closer to training on full data
  - More testing data: less noisy estimate of performance
  - Only one classifier might not be representative

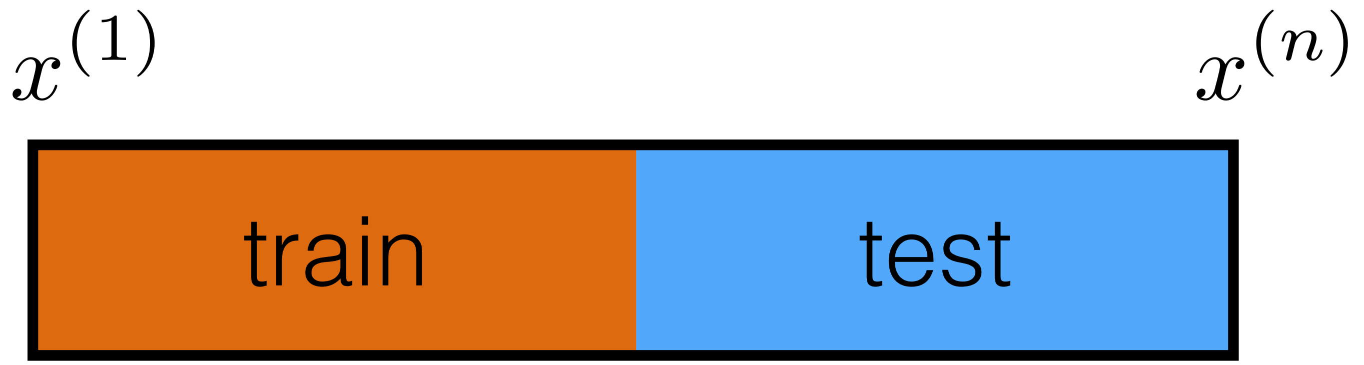
Evaluation of a learning algorithm



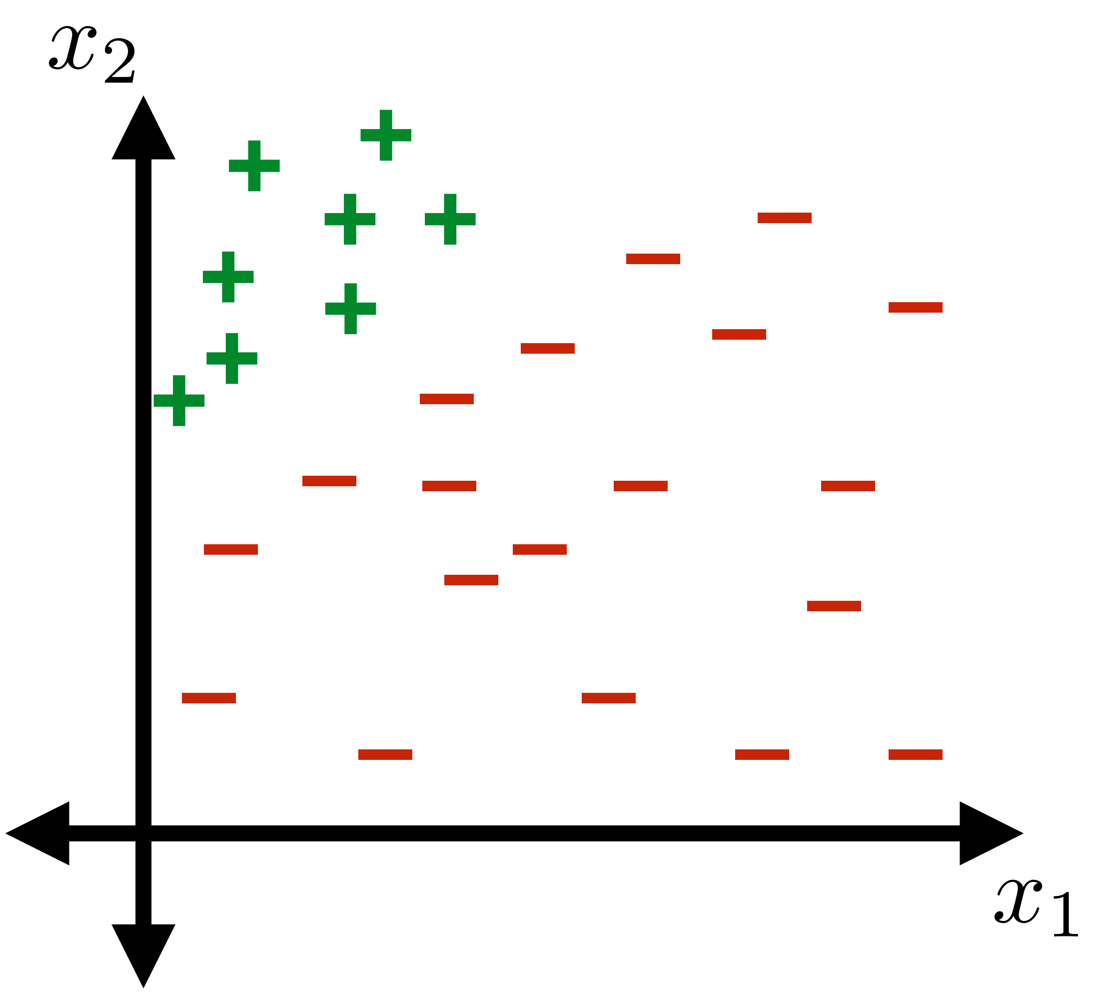
- How good is our learning algorithm on data like ours?
- Idea: use full data for training and then report training error
- Idea: reserve some data for testing
  - More training data: closer to training on full data
  - More testing data: less noisy estimate of performance
  - Only one classifier might not be representative
  - Good idea to shuffle order of data

Evaluation of a learning algorithm

- How good is our learning algorithm on data like ours?

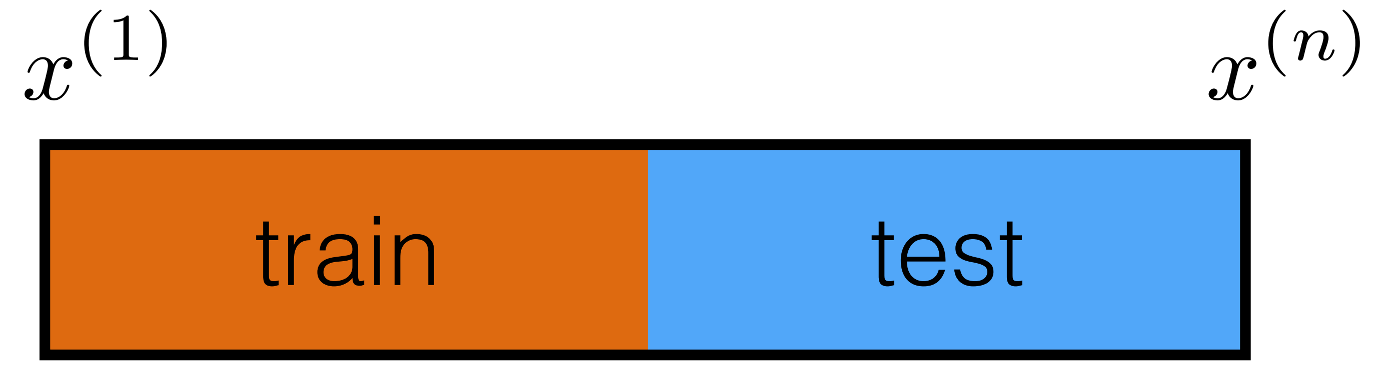


- Idea: use full data for training and then report training error
- Idea: reserve some data for testing
  - More training data: closer to training on full data
  - More testing data: less noisy estimate of performance
  - Only one classifier might not be representative
  - Good idea to shuffle order of data

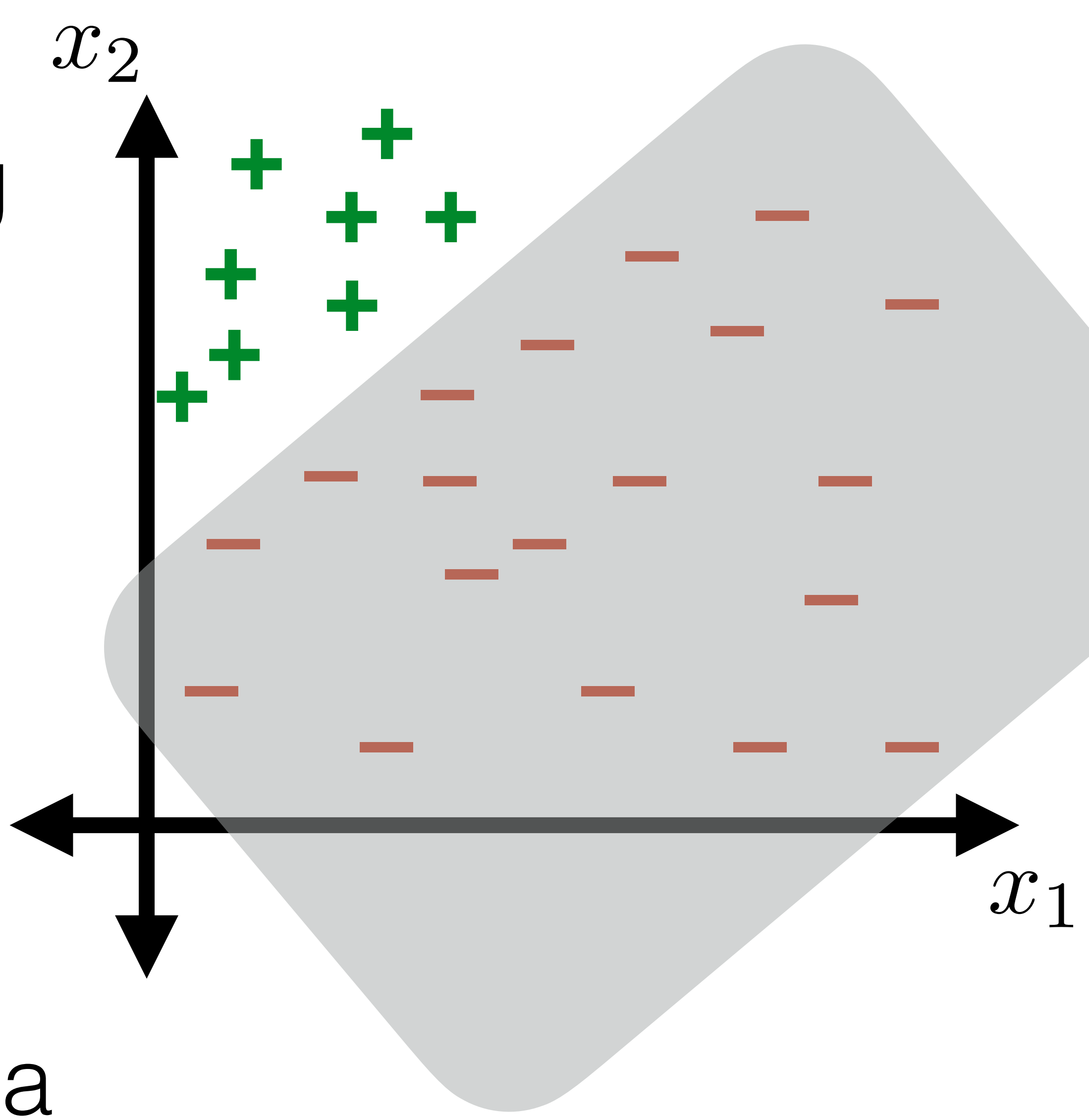


Evaluation of a learning algorithm

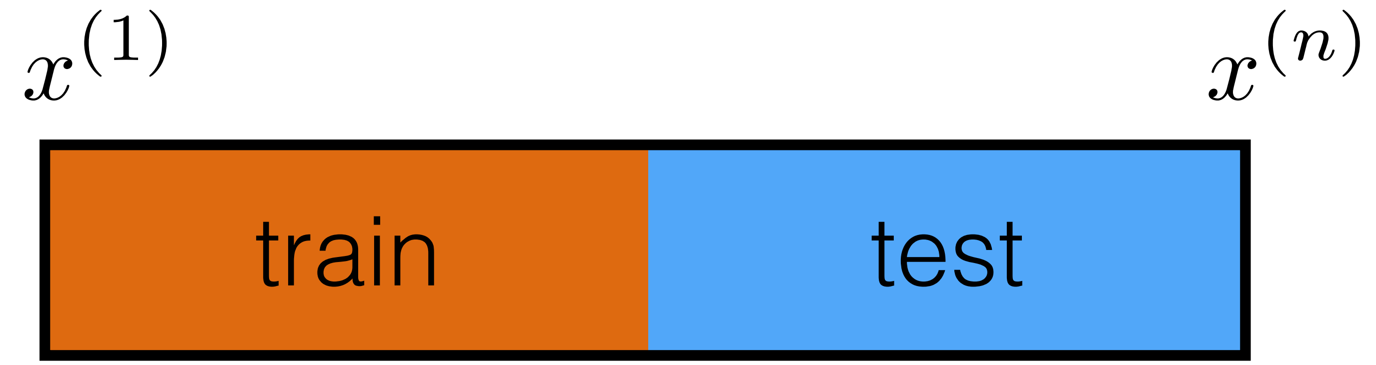
- How good is our learning algorithm on data like ours?



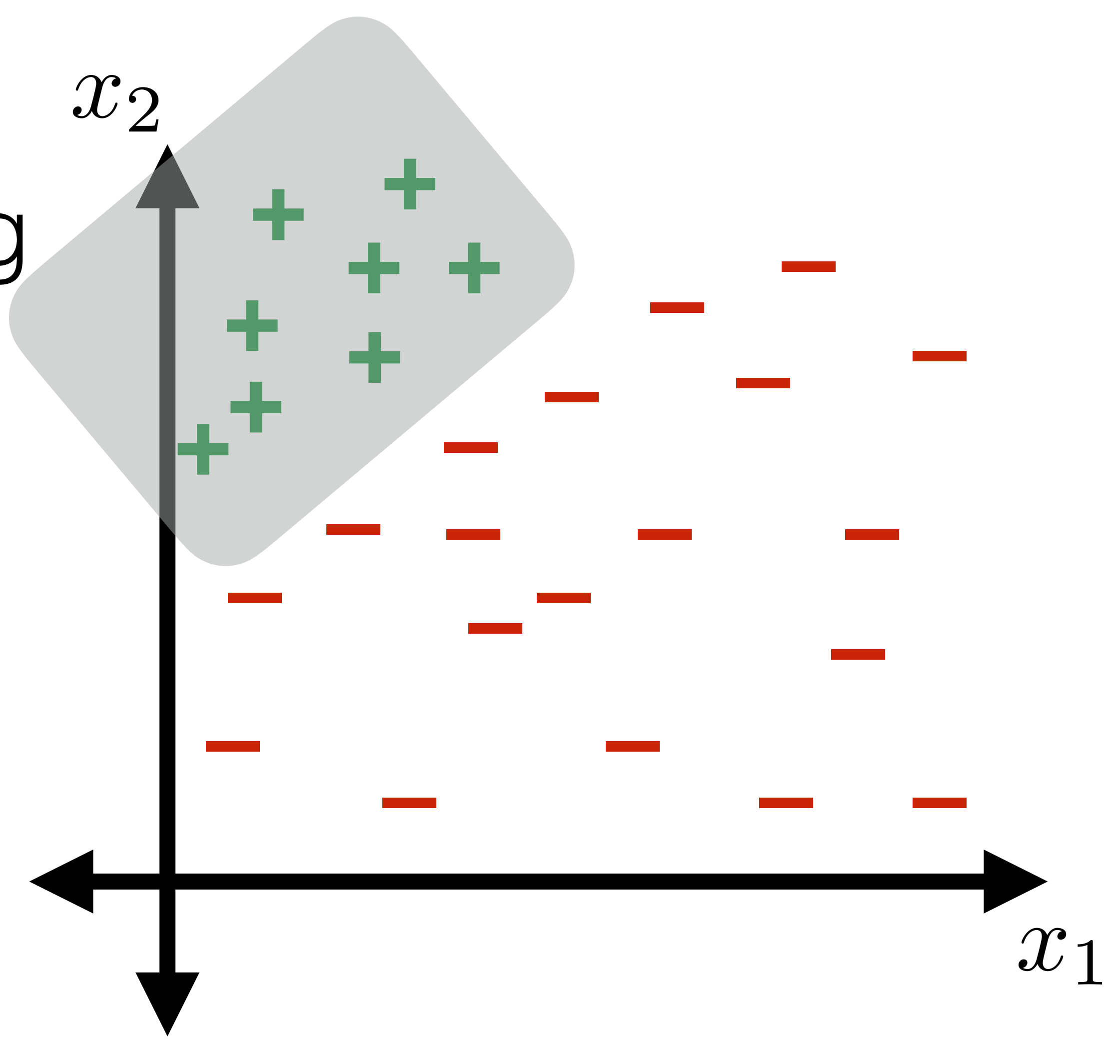
- Idea: use full data for training and then report training error
- Idea: reserve some data for testing
  - More training data: closer to training on full data
  - More testing data: less noisy estimate of performance
  - Only one classifier might not be representative
  - Good idea to shuffle order of data



Evaluation of a learning algorithm



- How good is our learning algorithm on data like ours?
- Idea: use full data for training and then report training error
- Idea: reserve some data for testing
    - More training data: closer to training on full data
    - More testing data: less noisy estimate of performance
    - Only one classifier might not be representative
    - Good idea to shuffle order of data



Evaluation of a learning algorithm

`Cross-validate(`$\mathcal{D}_n$ `,` k `)`

Evaluation of a learning algorithm

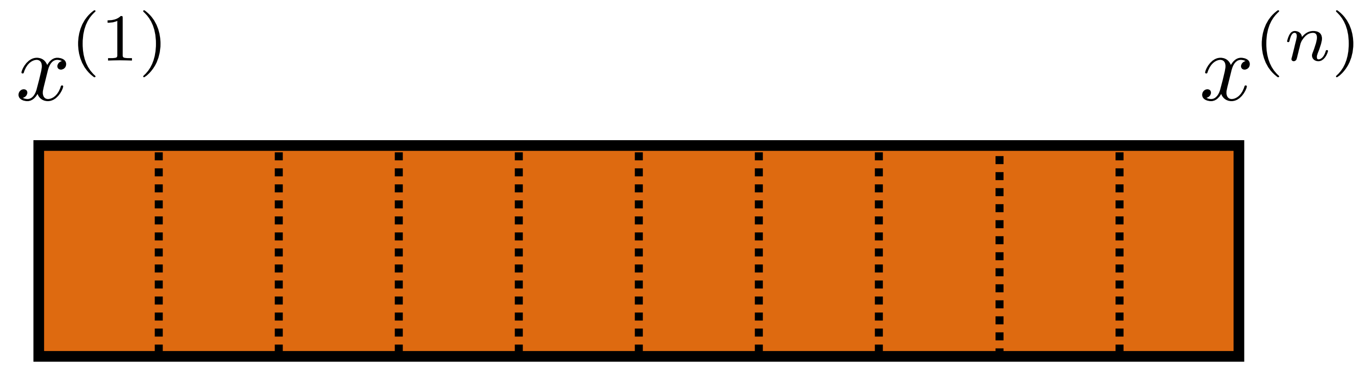
```
Cross-validate(D_n , k )
  Divide D_n into k chunks D_{n,1},...,D_{n,k} (of
  roughly equal size)
```

Evaluation of a learning algorithm

Cross-validate($\mathcal{D}_n$, k)

Divide $\mathcal{D}_n$ into k chunks $\mathcal{D}_{n,1}, \ldots, \mathcal{D}_{n,k}$ (of roughly equal size)

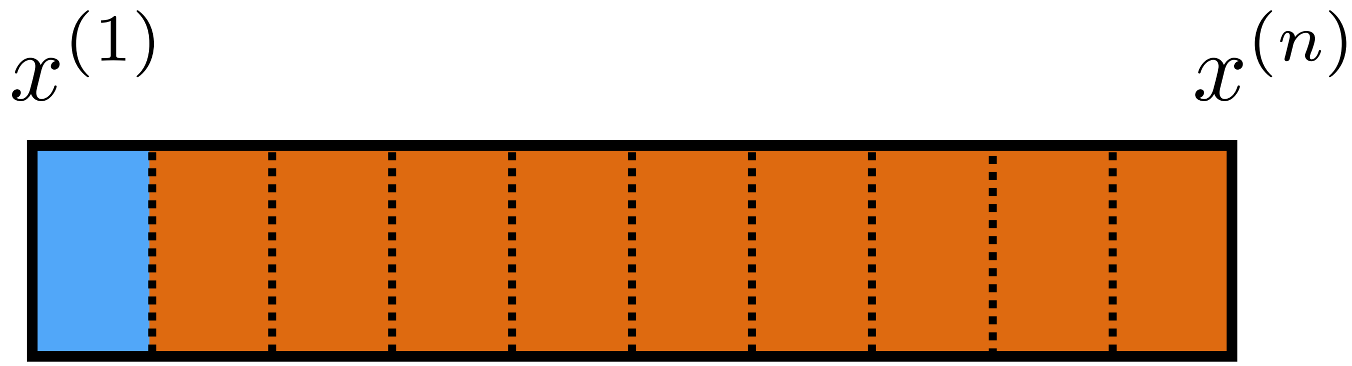
Evaluation of a learning algorithm



Cross-validate($\mathcal{D}_n$, k)

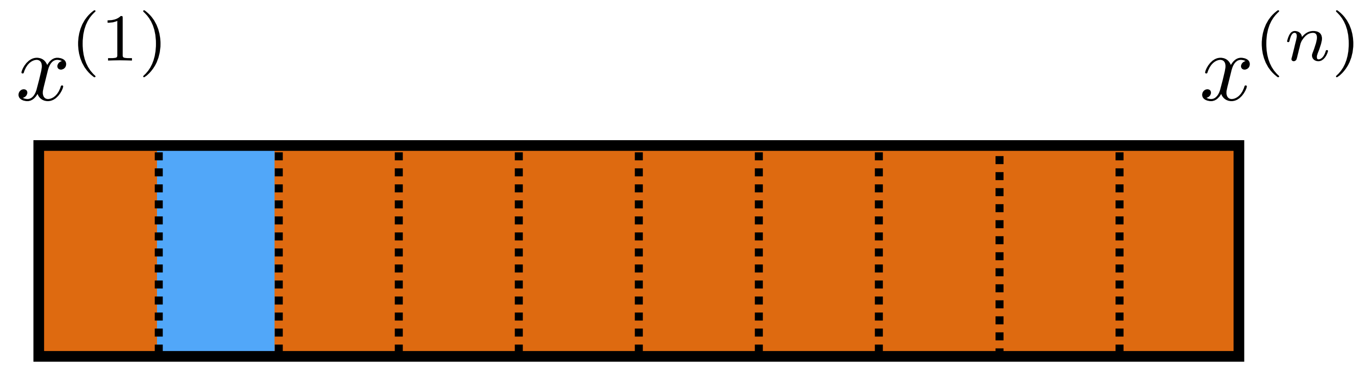
Divide $\mathcal{D}_n$ into k chunks $\mathcal{D}_{n,1}, \ldots, \mathcal{D}_{n,k}$ (of roughly equal size)

Evaluation of a learning algorithm



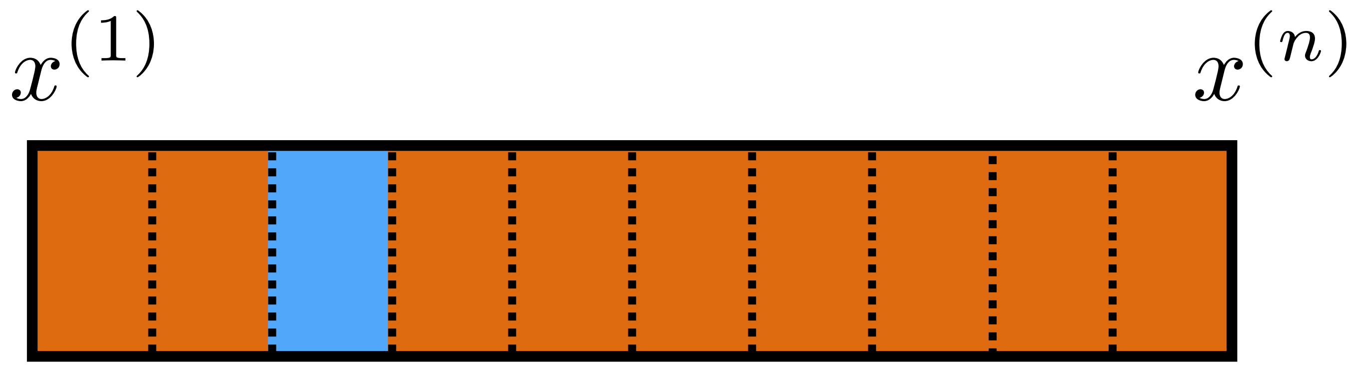

```
Cross-validate( $\mathcal{D}_n$ , $k$ )
  Divide $\mathcal{D}_n$ into $k$ chunks $\mathcal{D}_{n,1},\ldots,\mathcal{D}_{n,k}$ (of
  roughly equal size)
  for i = 1 to $k$
```

Evaluation of a learning algorithm



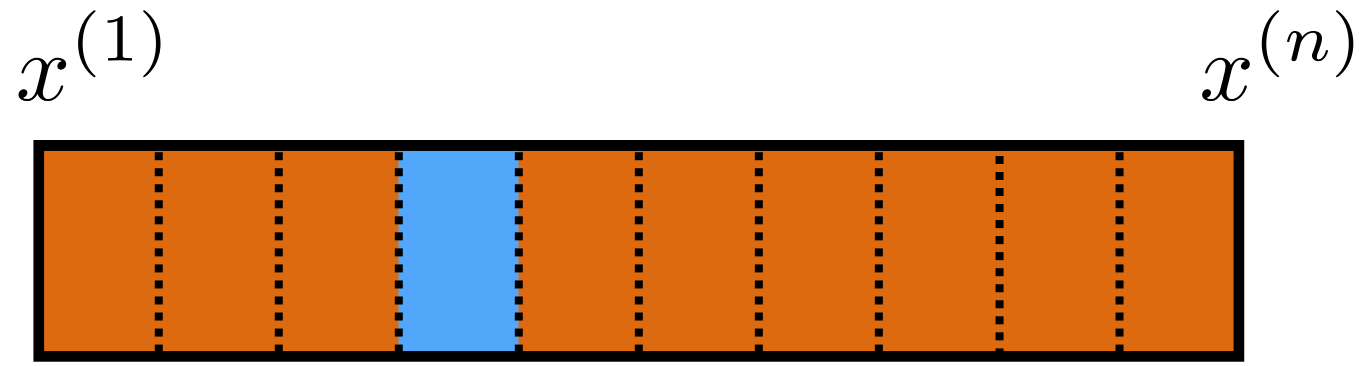

```
Cross-validate( $\mathcal{D}_n$ , $k$ )
  Divide $\mathcal{D}_n$ into $k$ chunks $\mathcal{D}_{n,1},\ldots,\mathcal{D}_{n,k}$ (of
  roughly equal size)
  for i = 1 to $k$
```

Evaluation of a learning algorithm



```
Cross-validate( $\mathcal{D}_n$ , $k$ )
  Divide $\mathcal{D}_n$ into $k$ chunks $\mathcal{D}_{n,1},\ldots,\mathcal{D}_{n,k}$ (of
  roughly equal size)
  for i = 1 to $k$
```

Evaluation of a learning algorithm



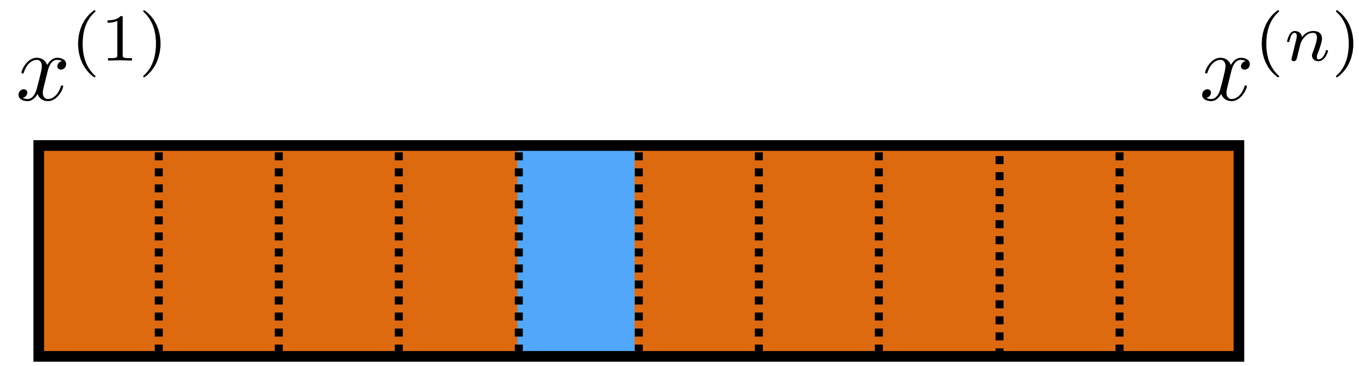

```
Cross-validate( 𝒟_n , k )
  Divide 𝒟_n into k chunks 𝒟_{n,1}, ..., 𝒟_{n,k} (of
  roughly equal size)
  for i = 1 to k
```

Cross-validate($\mathcal{D}_n$, k)

Divide $\mathcal{D}_n$ into k chunks $\mathcal{D}_{n,1}, \ldots, \mathcal{D}_{n,k}$ (of roughly equal size)

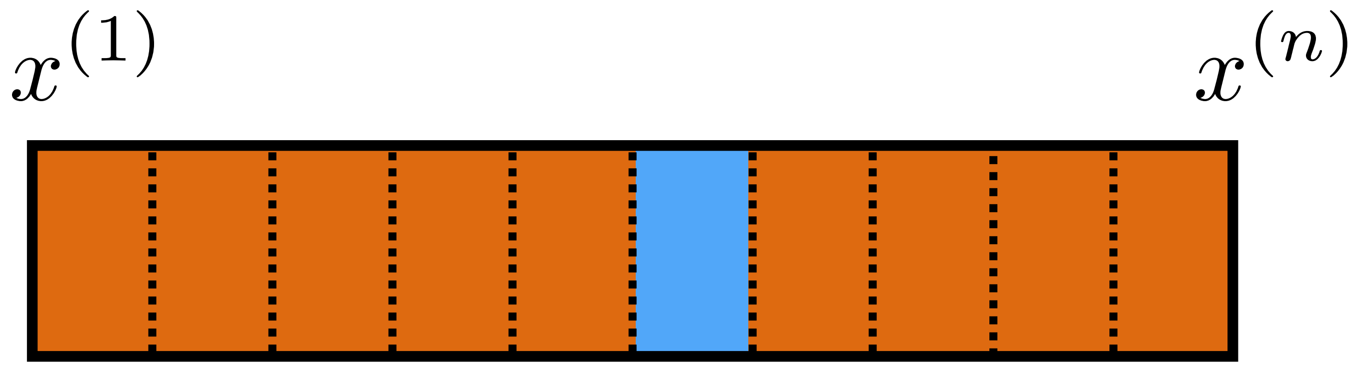
for i = 1 to k

Evaluation of a learning algorithm



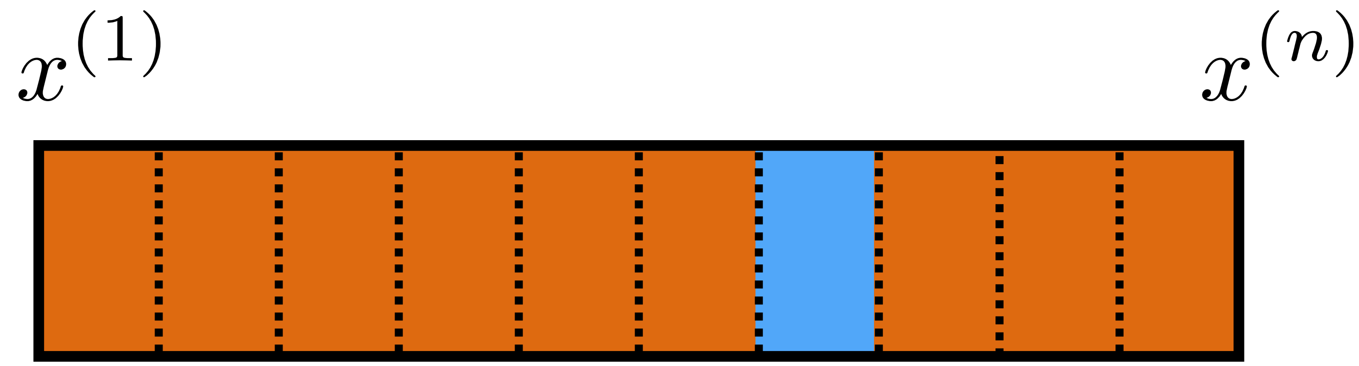

```
Cross-validate( 𝒟_n , k )
  Divide 𝒟_n into k chunks 𝒟_{n,1},...,𝒟_{n,k} (of
  roughly equal size)
  for i = 1 to k
```

Evaluation of a learning algorithm



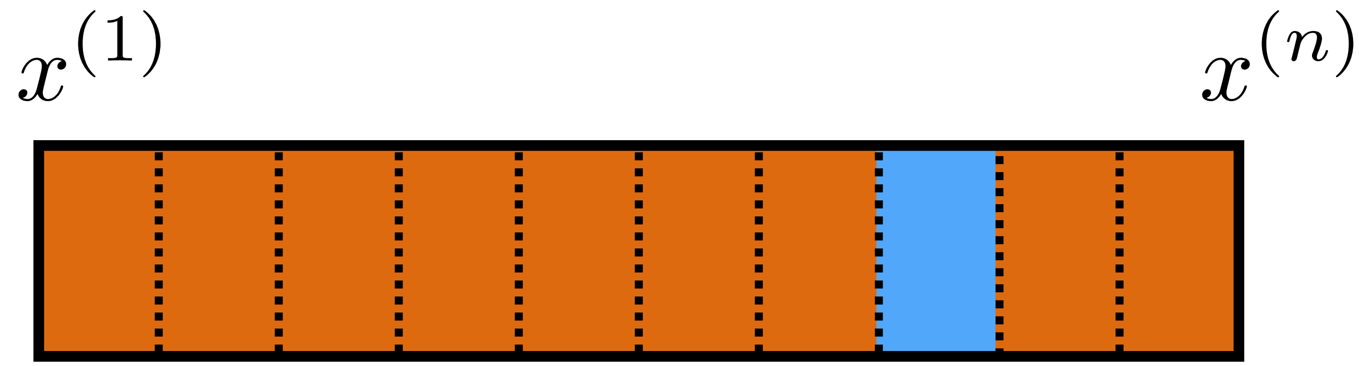
```
Cross-validate(𝒟_n , k )
  Divide 𝒟_n into k chunks 𝒟_{n,1},...,𝒟_{n,k} (of
  roughly equal size)
  for i = 1 to k
```

Evaluation of a learning algorithm




```
Cross-validate( $\mathcal{D}_n$ , $k$ )
  Divide $\mathcal{D}_n$ into $k$ chunks $\mathcal{D}_{n,1}, \ldots, \mathcal{D}_{n,k}$ (of
  roughly equal size)
  for i = 1 to $k$
```

Evaluation of a learning algorithm

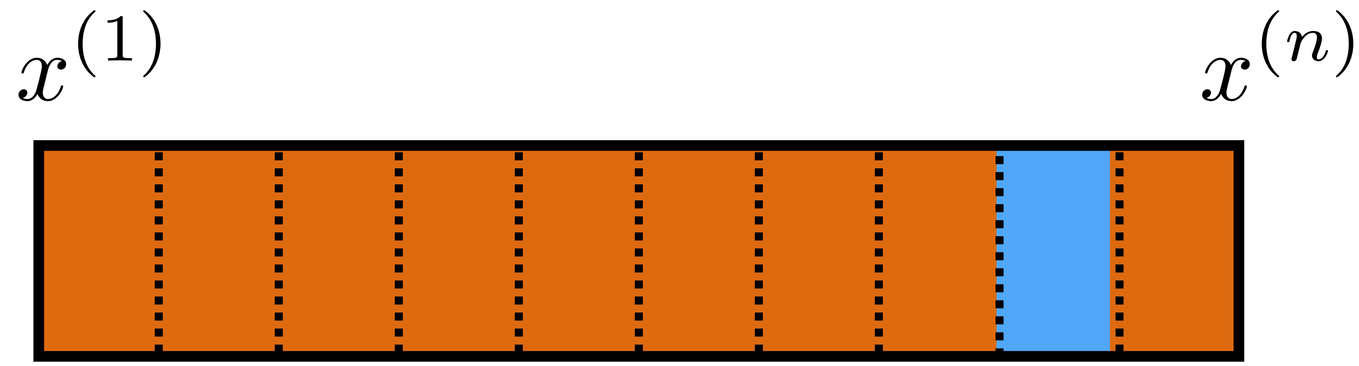


Cross-validate($\mathcal{D}_n$, k)

  Divide $\mathcal{D}_n$ into k chunks $\mathcal{D}_{n,1}, \ldots, \mathcal{D}_{n,k}$ (of roughly equal size)

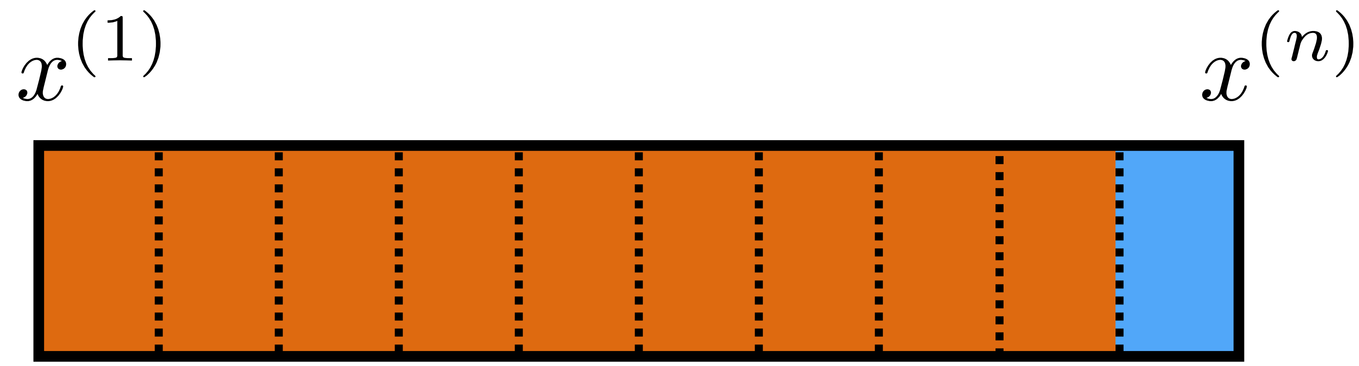
  for i = 1 to k

Evaluation of a learning algorithm



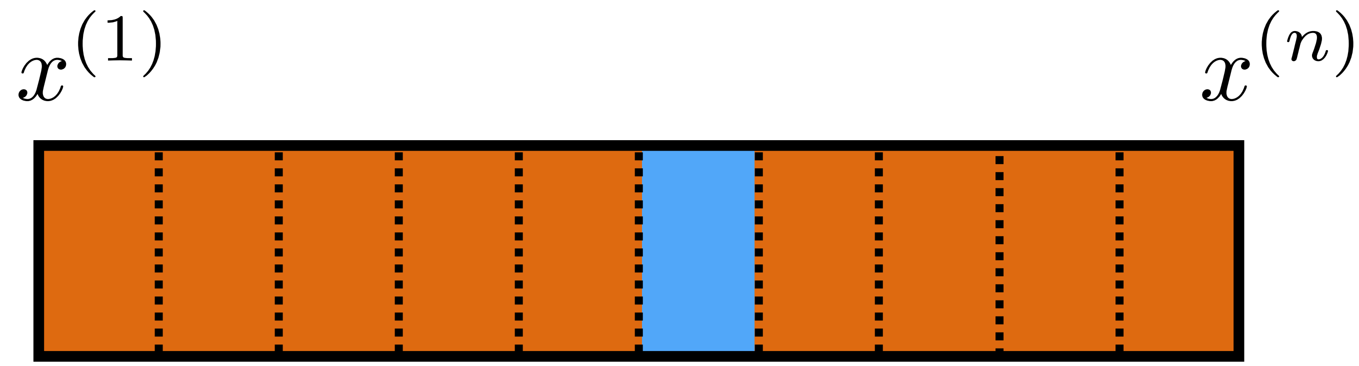

```
Cross-validate( $\mathcal{D}_n$ , $k$ )
  Divide $\mathcal{D}_n$ into $k$ chunks $\mathcal{D}_{n,1}, \ldots, \mathcal{D}_{n,k}$ (of
  roughly equal size)
  for i = 1 to $k$
```

Evaluation of a learning algorithm



```
Cross-validate( 𝒟ₙ , k )
  Divide 𝒟ₙ into k chunks 𝒟ₙ,₁,...,𝒟ₙ,ₖ (of
  roughly equal size)
  for i = 1 to k
```

Evaluation of a learning algorithm



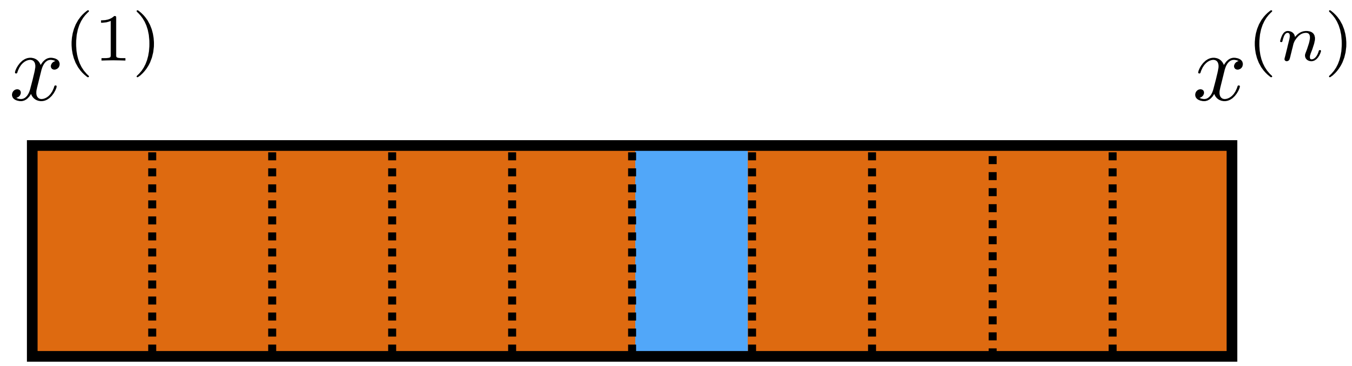

```
Cross-validate( 𝒟_n , k )
  Divide 𝒟_n into k chunks 𝒟_{n,1}, ..., 𝒟_{n,k} (of
  roughly equal size)
  for i = 1 to k
```

Cross-validate($\mathcal{D}_n$, k)

Divide $\mathcal{D}_n$ into k chunks $\mathcal{D}_{n,1}, \ldots, \mathcal{D}_{n,k}$ (of roughly equal size)

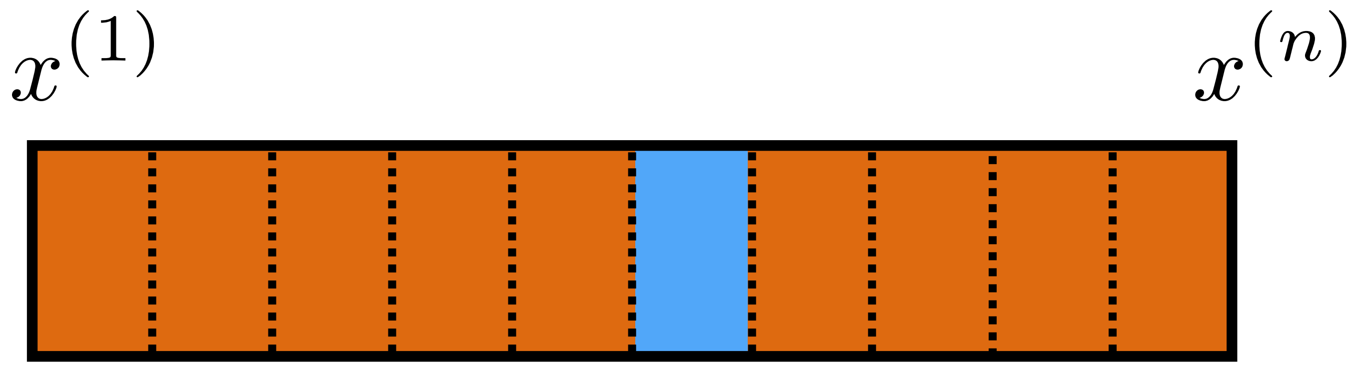
for i = 1 to k

Evaluation of a learning algorithm



```
Cross-validate( $\mathcal{D}_n$ , $k$ )
  Divide $\mathcal{D}_n$ into $k$ chunks $\mathcal{D}_{n,1},\ldots,\mathcal{D}_{n,k}$ (of
  roughly equal size)
  for i = 1 to $k$
    train $h_i$ on $\mathcal{D}_n \backslash \mathcal{D}_{n,i}$ (i.e. except chunk i)
```

Evaluation of a learning algorithm



Cross-validate($\mathcal{D}_n$, k)

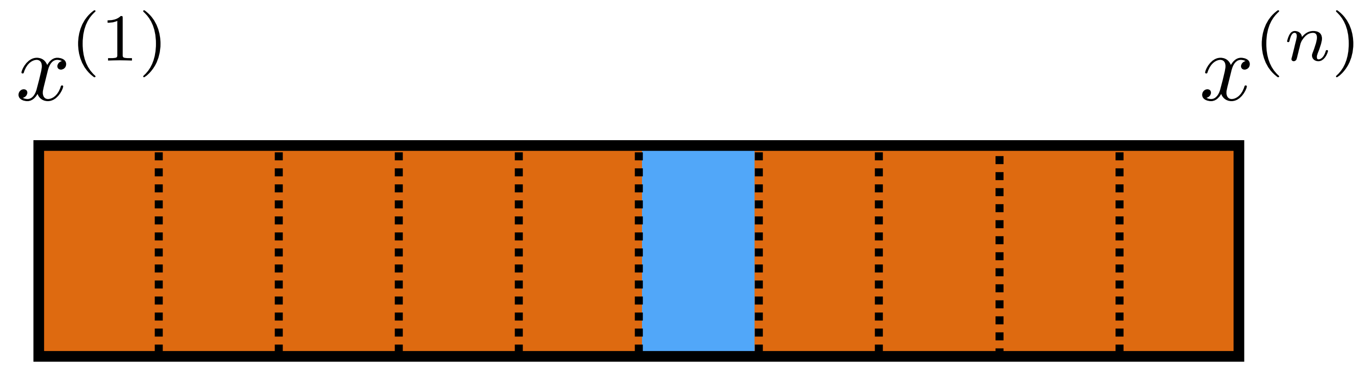
  Divide $\mathcal{D}_n$ into k chunks $\mathcal{D}_{n,1}, \ldots, \mathcal{D}_{n,k}$ (of roughly equal size)

  for i = 1 to k

    train h_i on $\mathcal{D}_n \backslash \mathcal{D}_{n,i}$ (i.e. except chunk i)

    compute "test" error $\mathcal{E}(h_i, \mathcal{D}_{n,i})$ of h_i on $\mathcal{D}_{n,i}$

Evaluation of a learning algorithm



Cross-validate($\mathcal{D}_n$, k)

  Divide $\mathcal{D}_n$ into k chunks $\mathcal{D}_{n,1}, \ldots, \mathcal{D}_{n,k}$ (of roughly equal size)

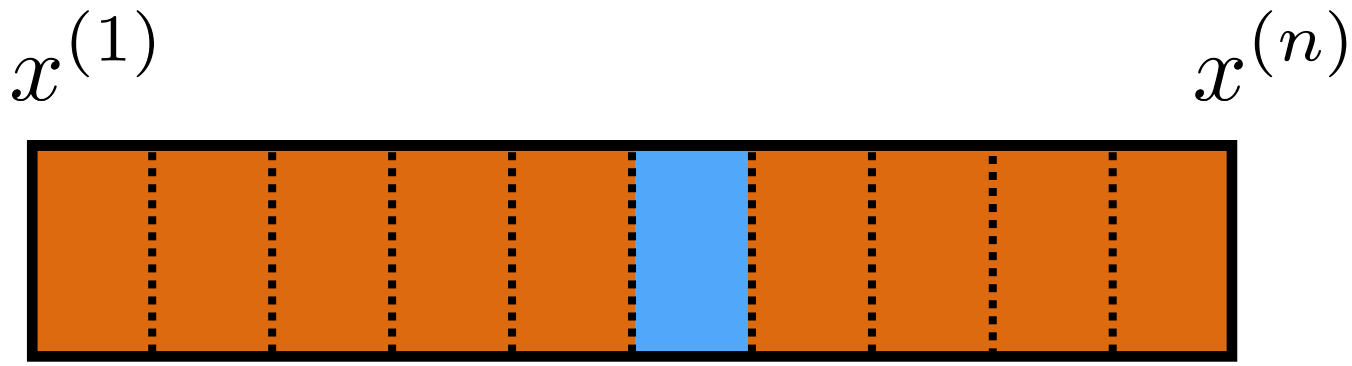
  for i = 1 to k

    train h_i on $\mathcal{D}_n \backslash \mathcal{D}_{n,i}$ (i.e. except chunk i)

    compute "test" error $\mathcal{E}(h_i, \mathcal{D}_{n,i})$ of h_i on $\mathcal{D}_{n,i}$

  Return $\frac{1}{k}\sum_{i=1}^{k} \mathcal{E}(h_i, \mathcal{D}_{n,i})$

Evaluation of a learning algorithm



```
Cross-validate( 𝒟ₙ , k )
  Divide 𝒟ₙ into k chunks 𝒟ₙ,₁,...,𝒟ₙ,ₖ (of
  roughly equal size)
  for i = 1 to k
    train hᵢ on 𝒟ₙ\𝒟ₙ,ᵢ (i.e. except chunk i)
    compute "test" error ℰ(hᵢ,𝒟ₙ,ᵢ) of hᵢ on 𝒟ₙ,ᵢ
  Return (1/k) Σᵢ₌₁ᵏ ℰ(hᵢ,𝒟ₙ,ᵢ)
```

Cross-validate($\mathcal{D}_n$, k)

Divide $\mathcal{D}_n$ into k chunks $\mathcal{D}_{n,1},\ldots,\mathcal{D}_{n,k}$ (of roughly equal size)

for i = 1 to k

train h_i on $\mathcal{D}_n \backslash \mathcal{D}_{n,i}$ (i.e. except chunk i)

compute "test" error $\mathcal{E}(h_i, \mathcal{D}_{n,i})$ of h_i on $\mathcal{D}_{n,i}$

Return $\frac{1}{k}\sum_{i=1}^{k}\mathcal{E}(h_i, \mathcal{D}_{n,i})$

- Again, good idea to shuffle order of data first