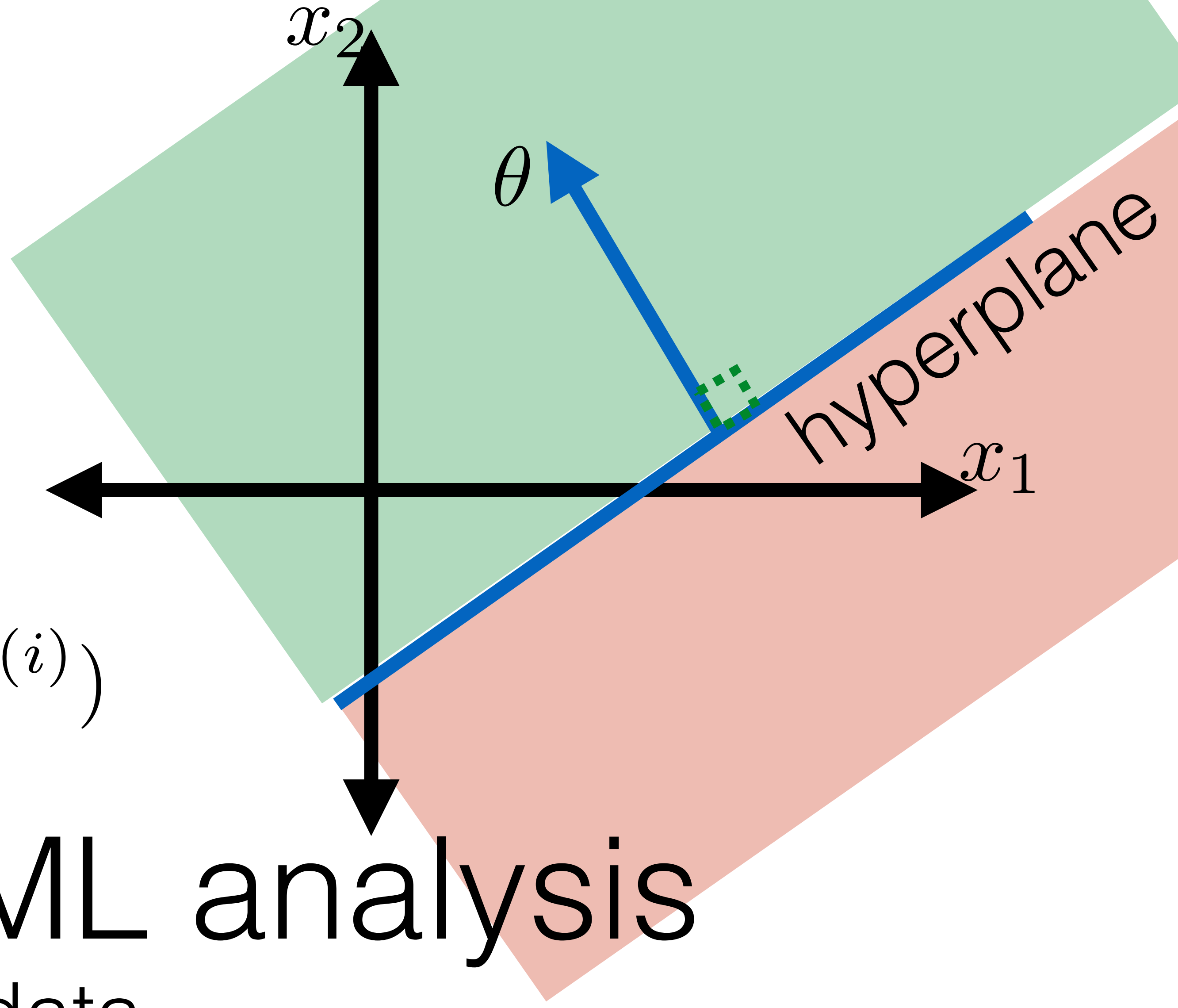
Recall

- 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. 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

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 $\{0, 1\}$
- Save mapping to recover predictions of new points

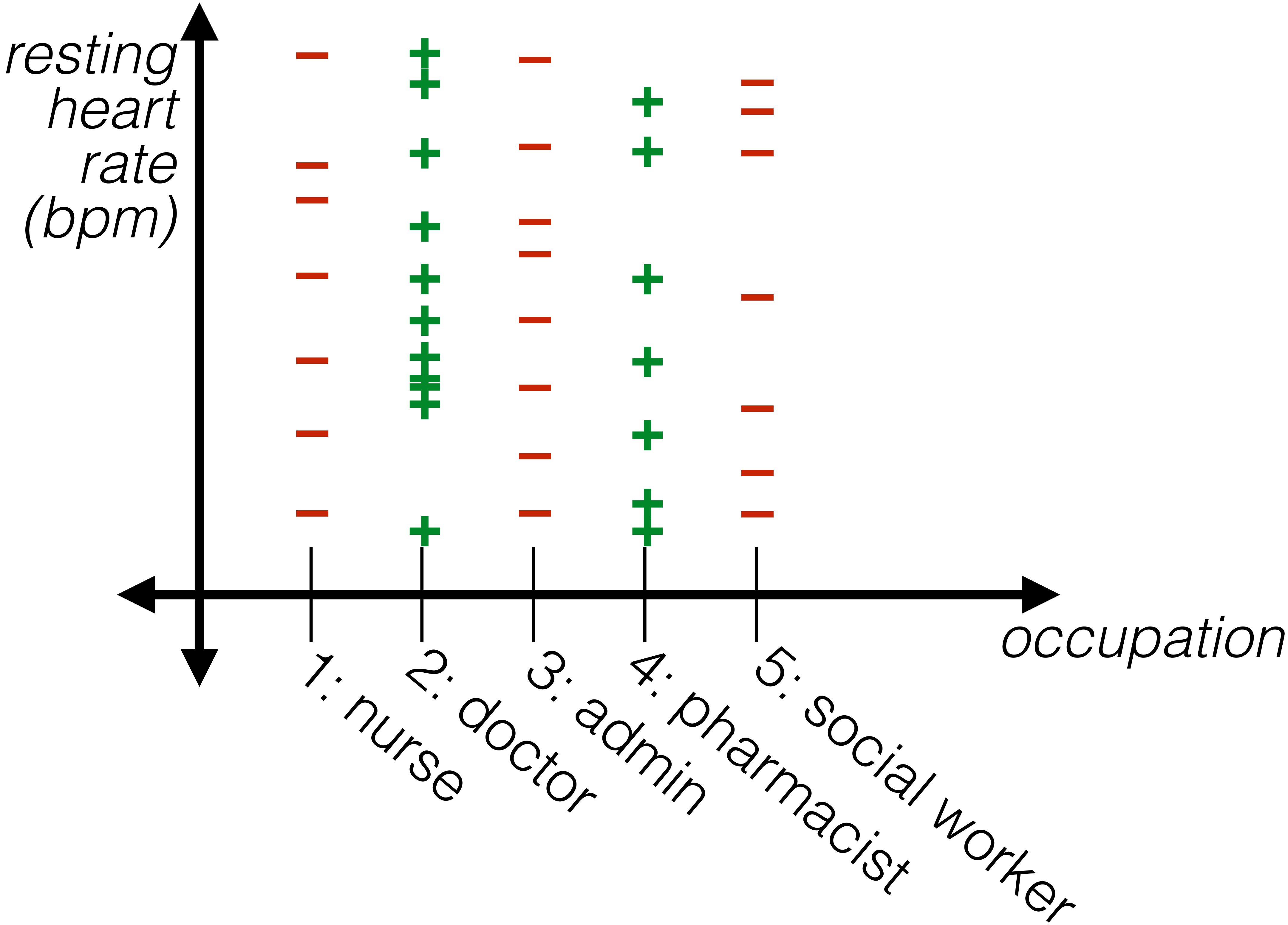
Encode data in usable form

- 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

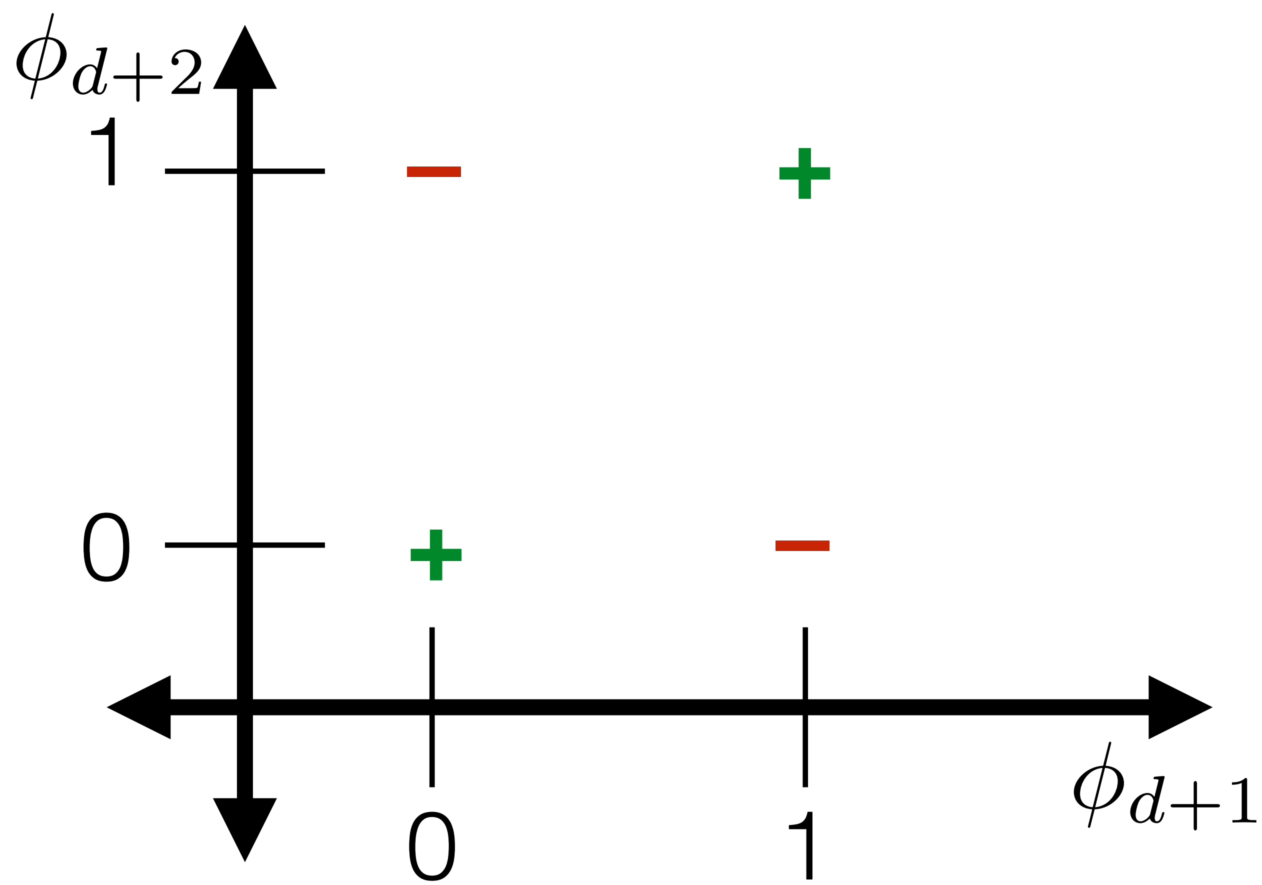
- Idea: turn each category into a unique natural number



Encode categorical data

- Idea: turn each category into a unique 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

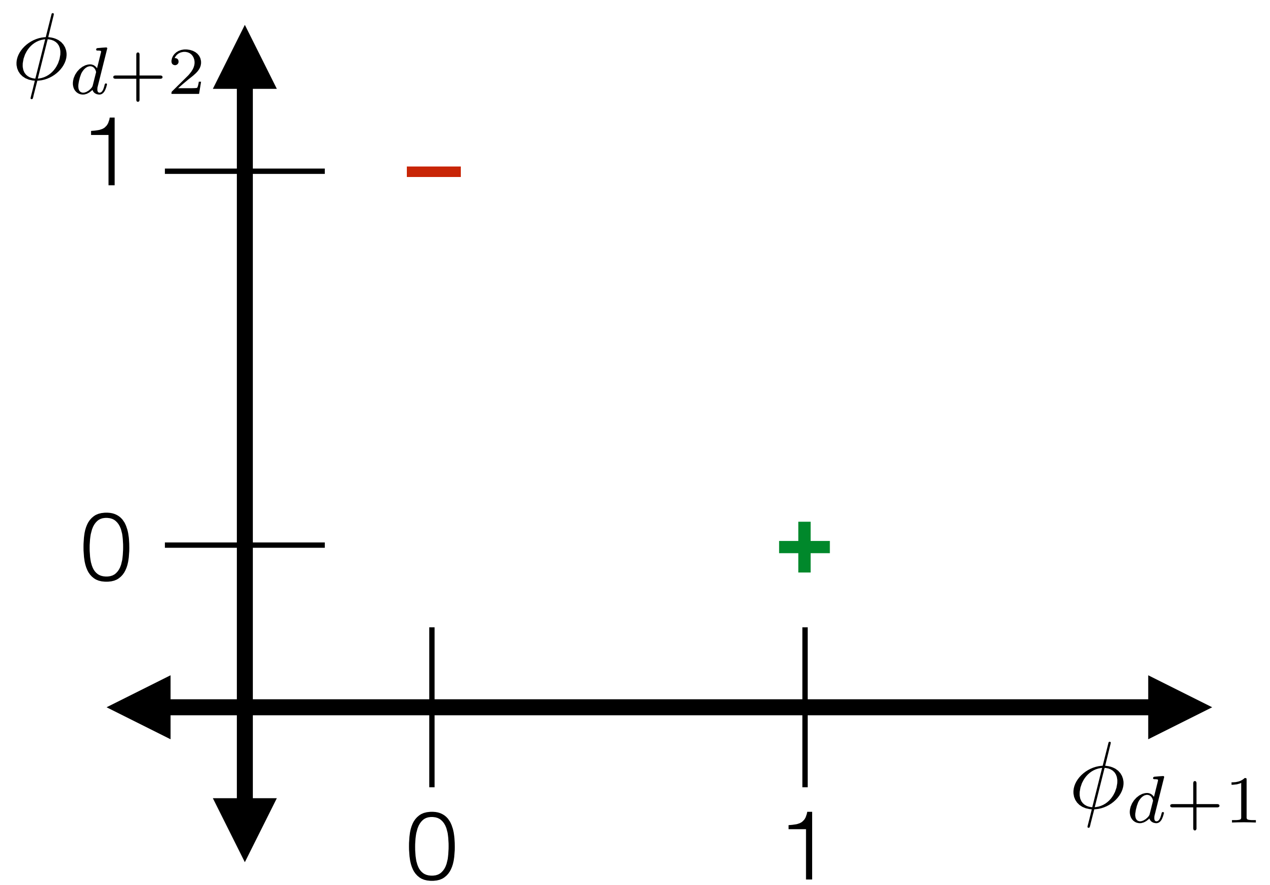


Encode categorical data

- Idea: turn each category into 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?	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

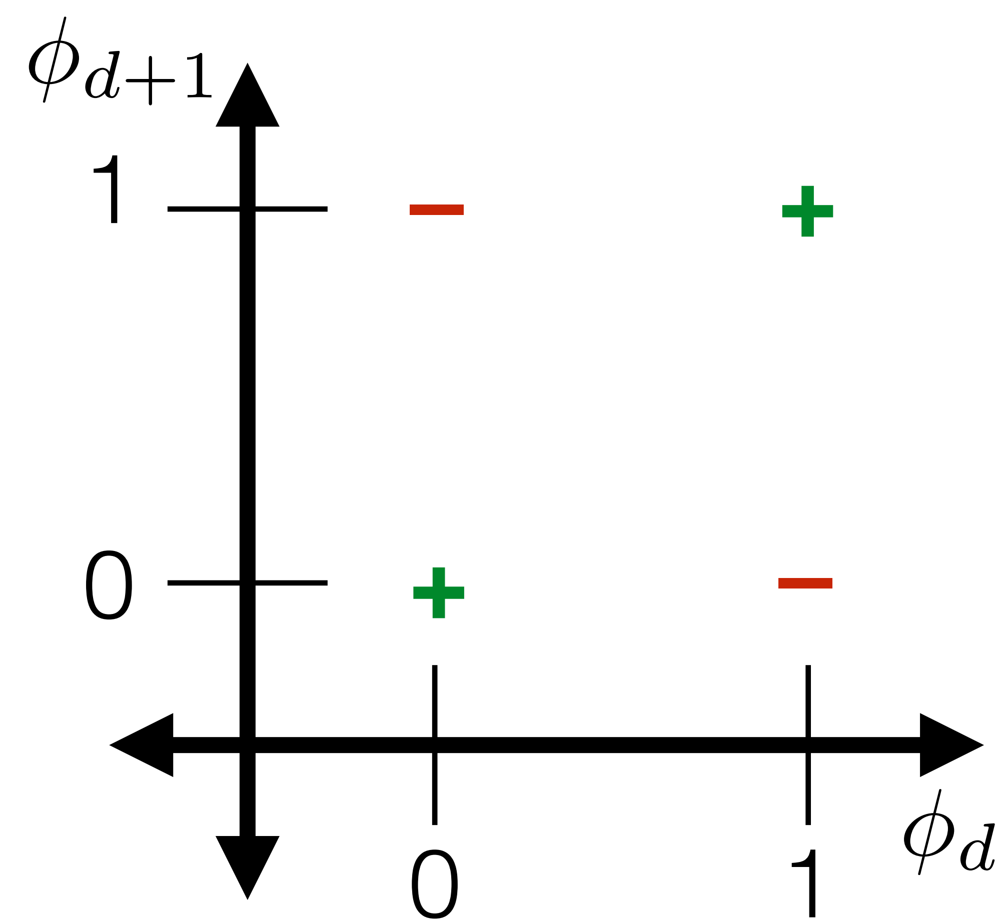
Encode categorical data

- 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



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

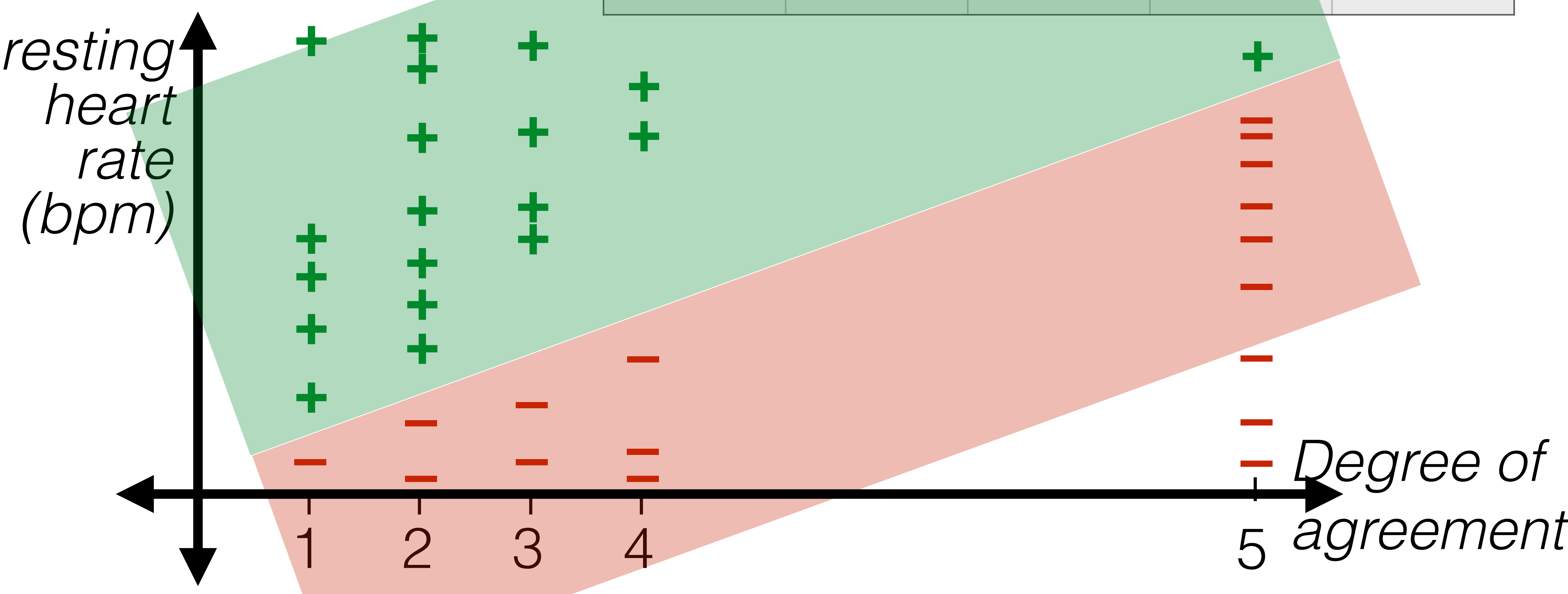
- Identify the features and encode as real numbers

	resting heart rate (bpm)	pain?	j1,j2,j3,j4,j5	m1, m2	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

- 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:

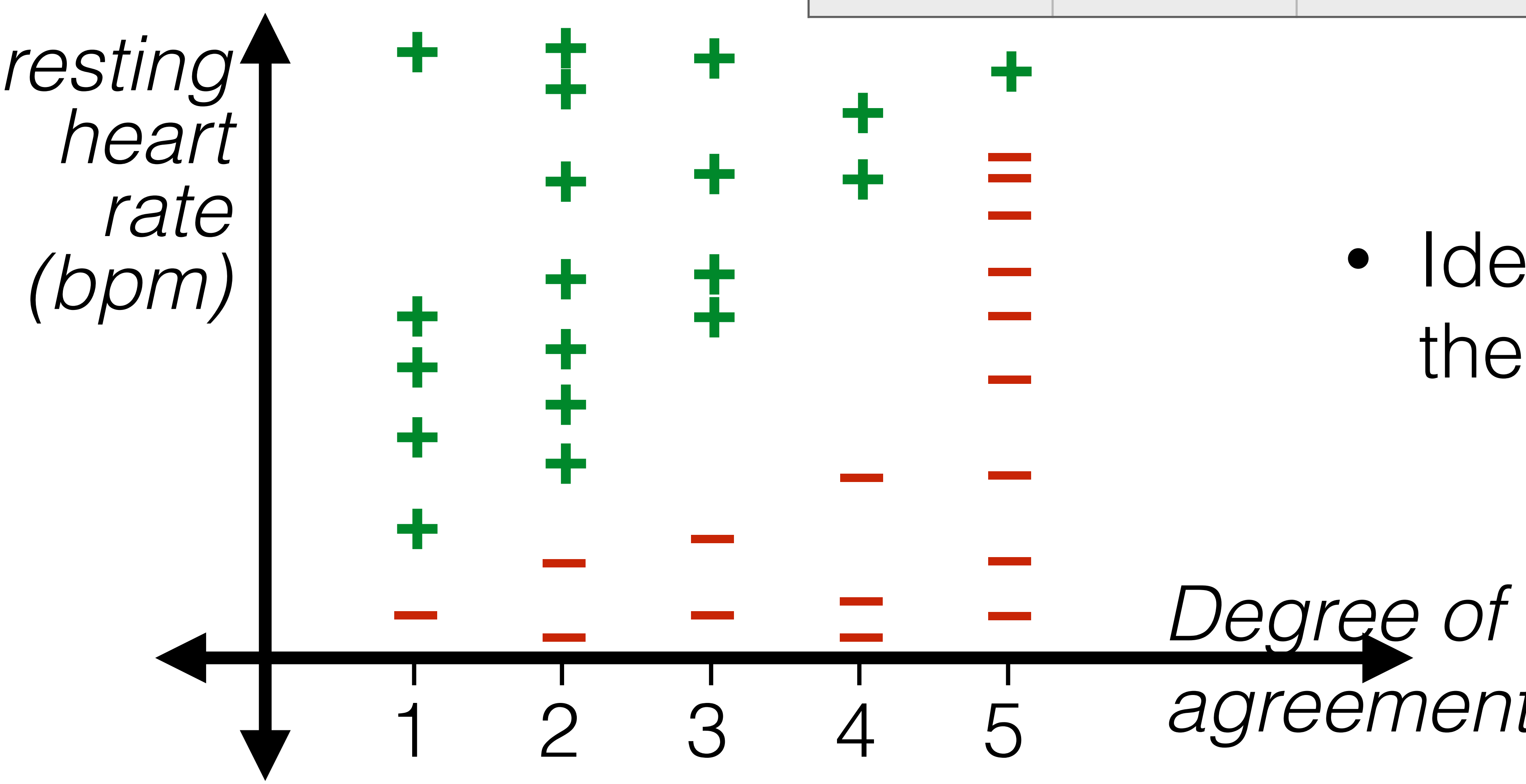
Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	2	3	4	5



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:

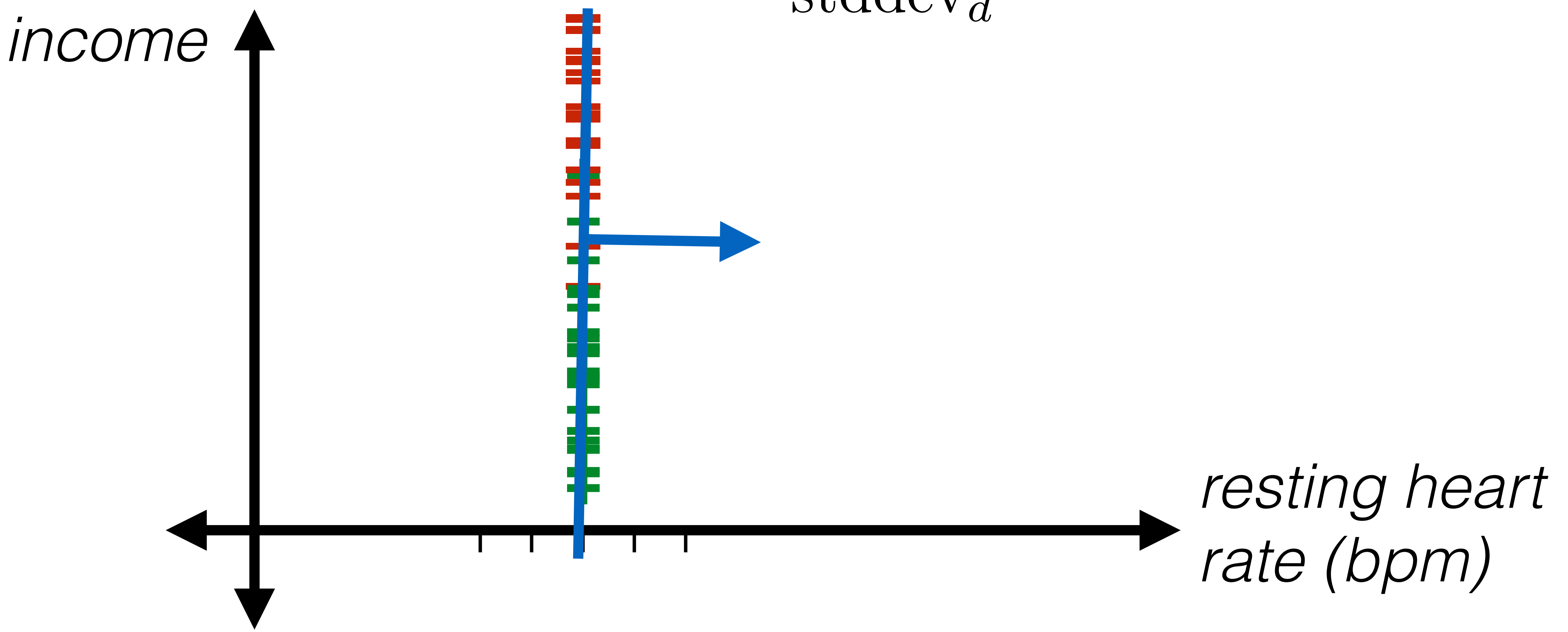
Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1,0,0,0,0	1,1,0,0,0	1,1,1,0,0	1,1,1,1,0	1,1,1,1,1



- Idea: Unary/ thermometer code

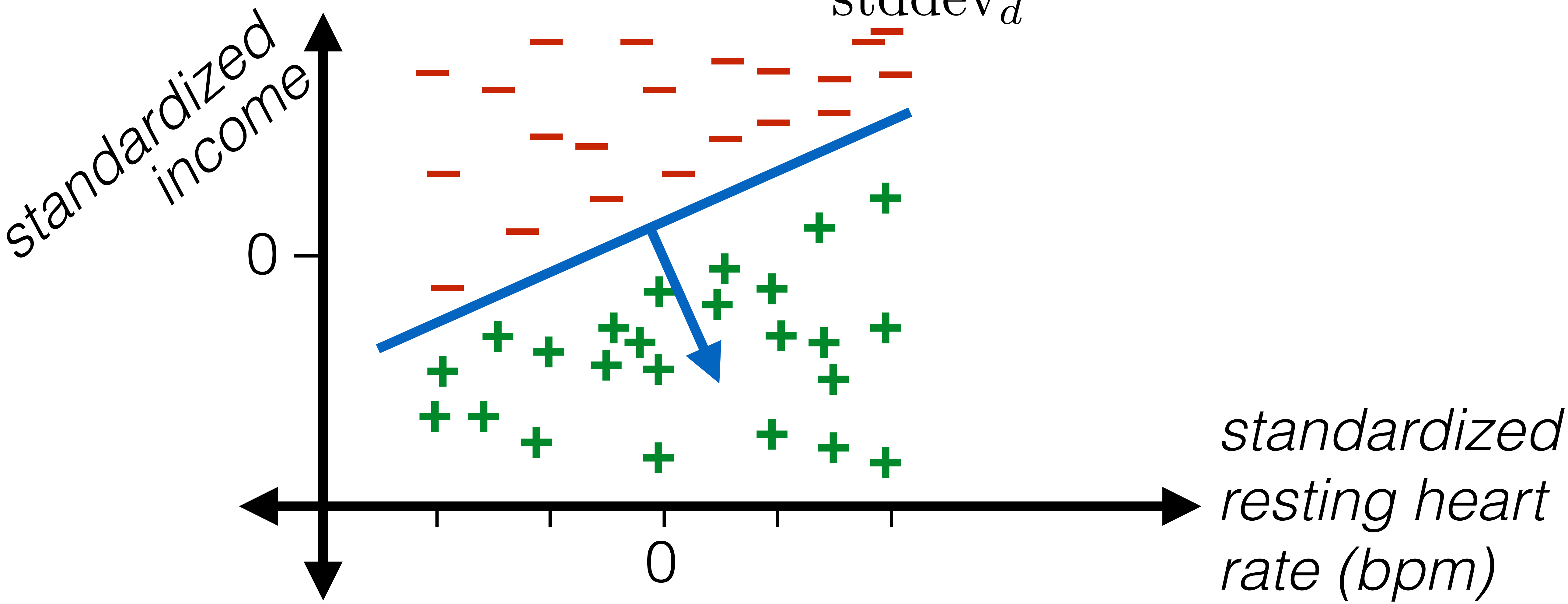
Encode numerical data

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



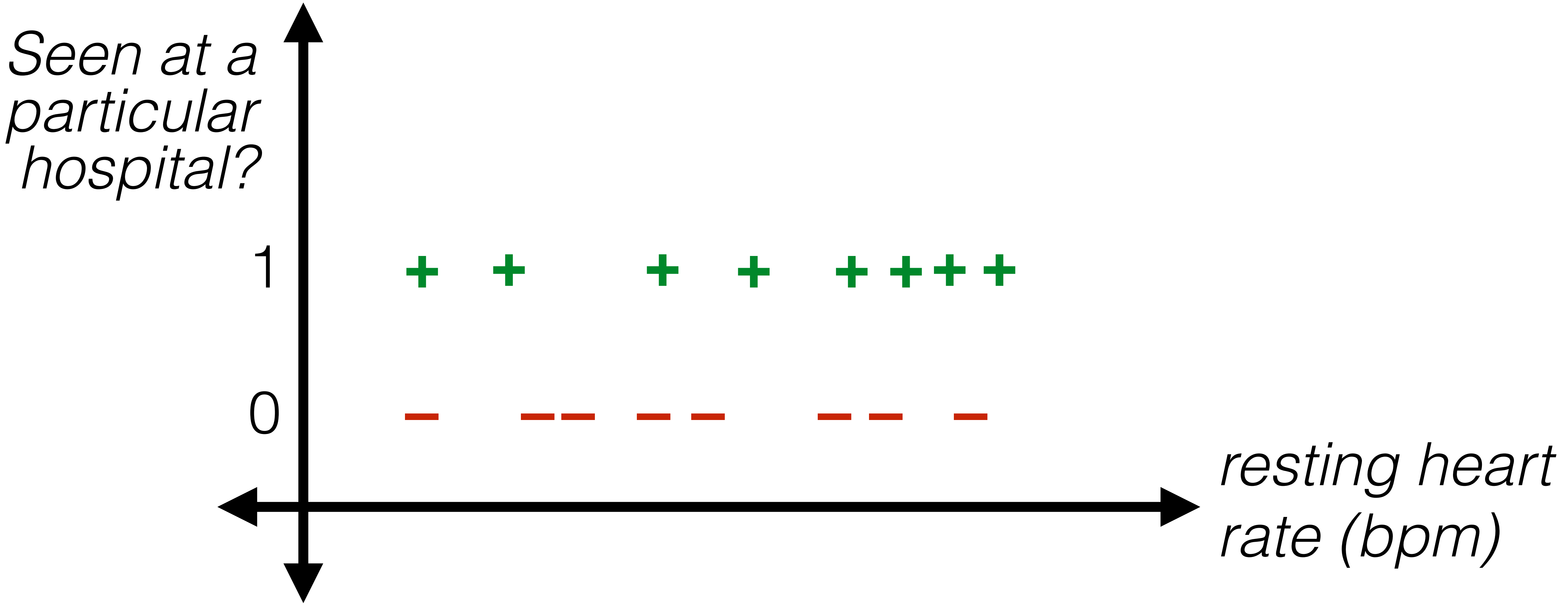
Encode numerical data

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



More benefits of plotting your data

- And talking to experts

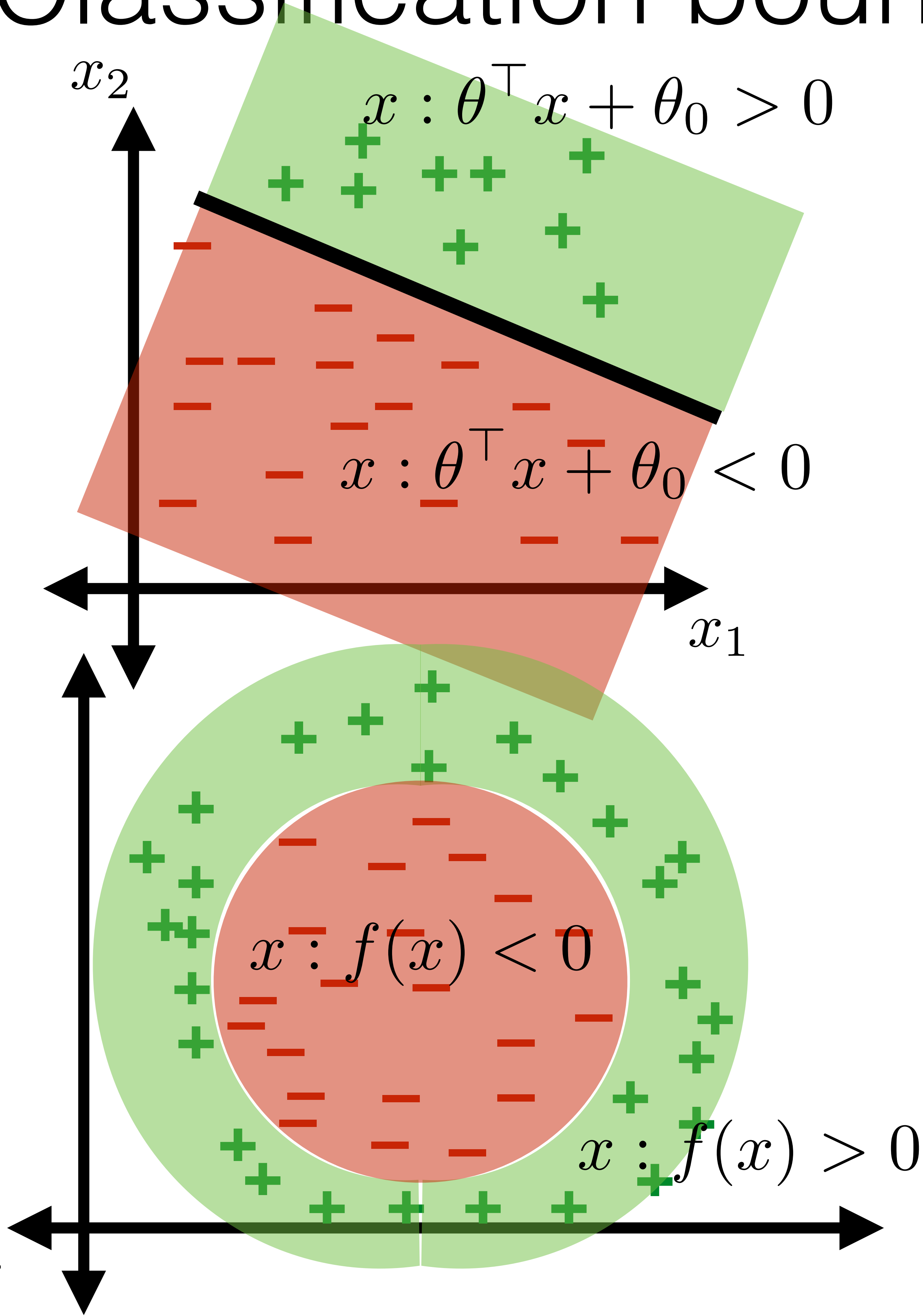


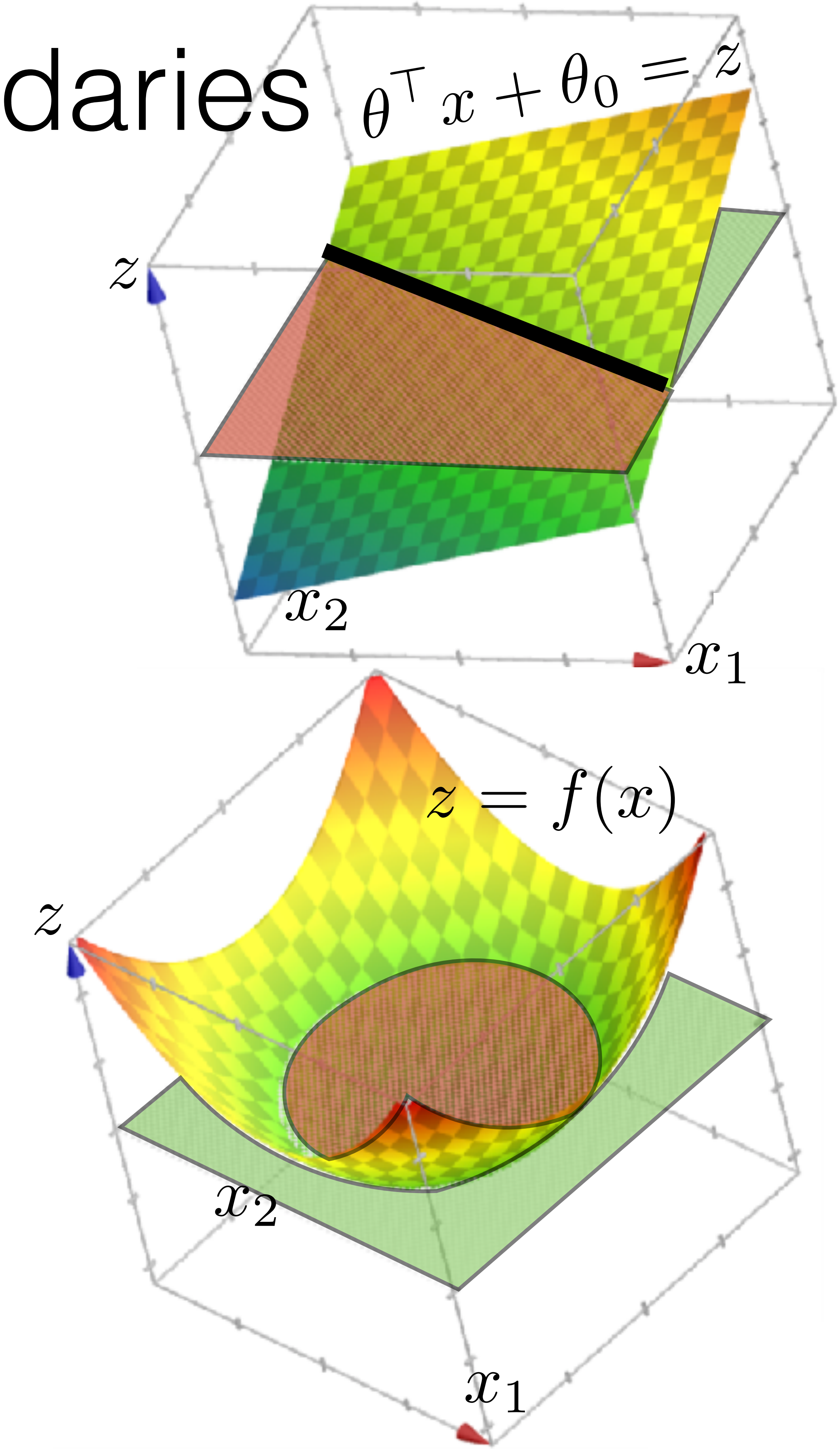
Encode data in usable form

- 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

Classification boundaries



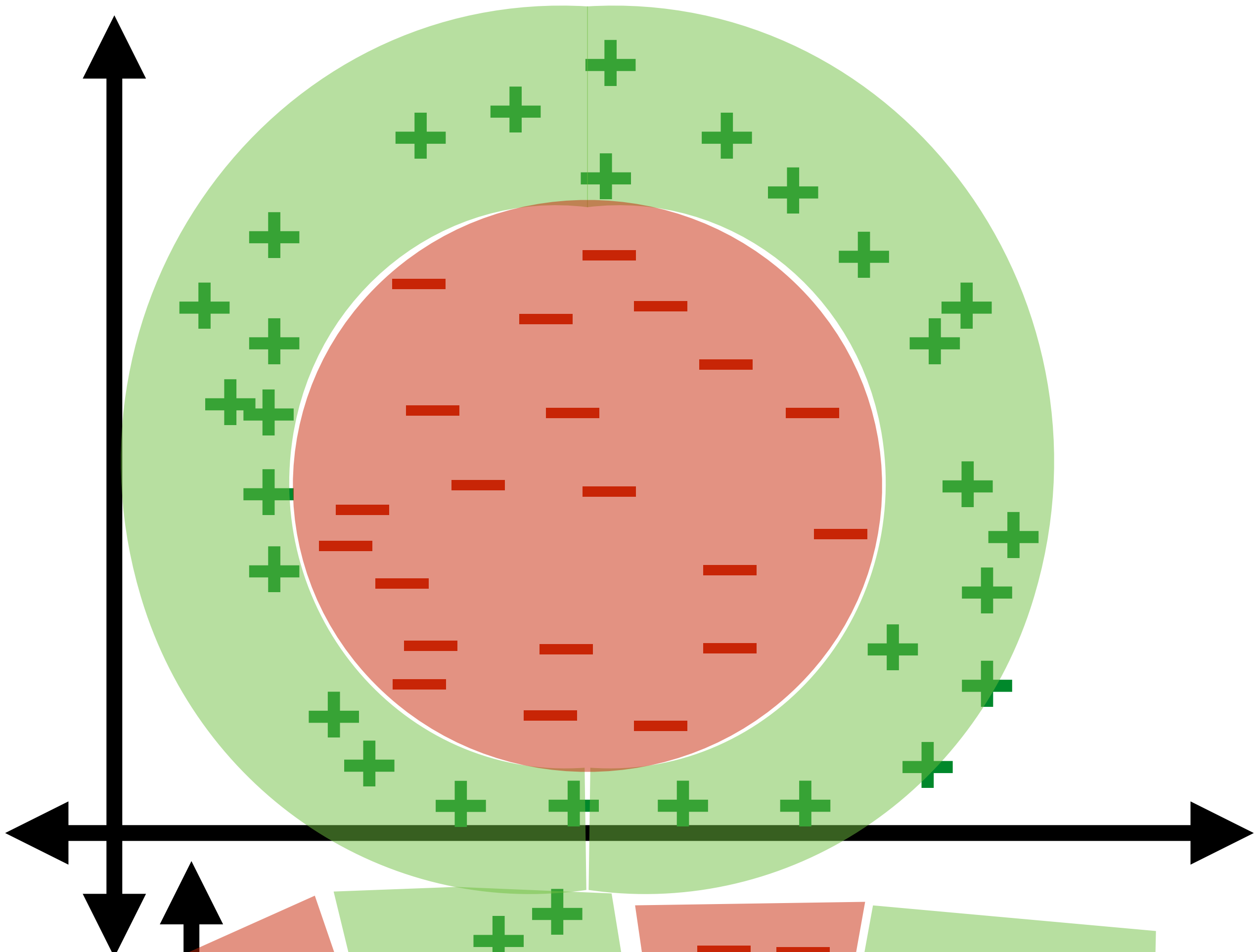


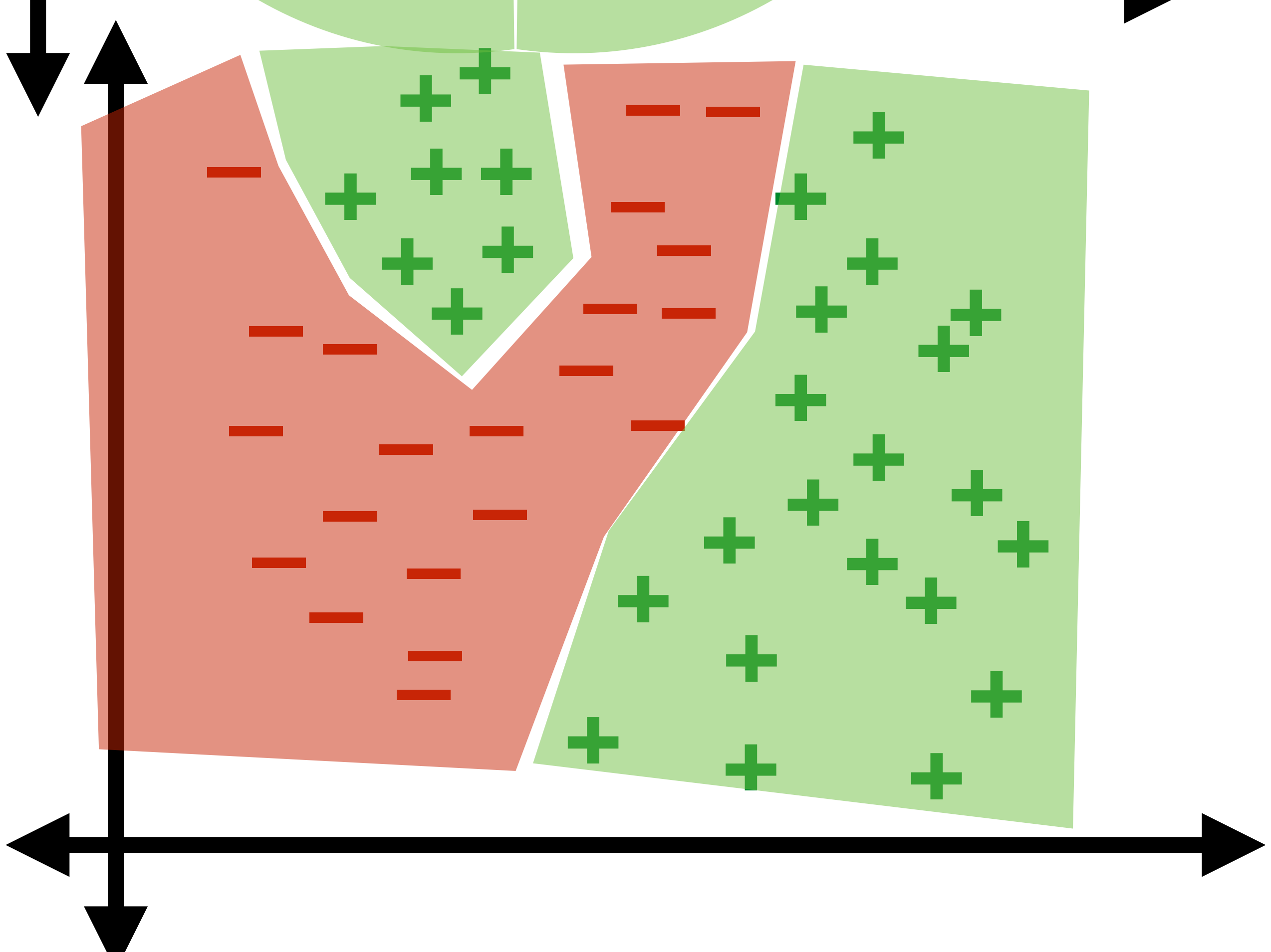
Nonlinear boundaries

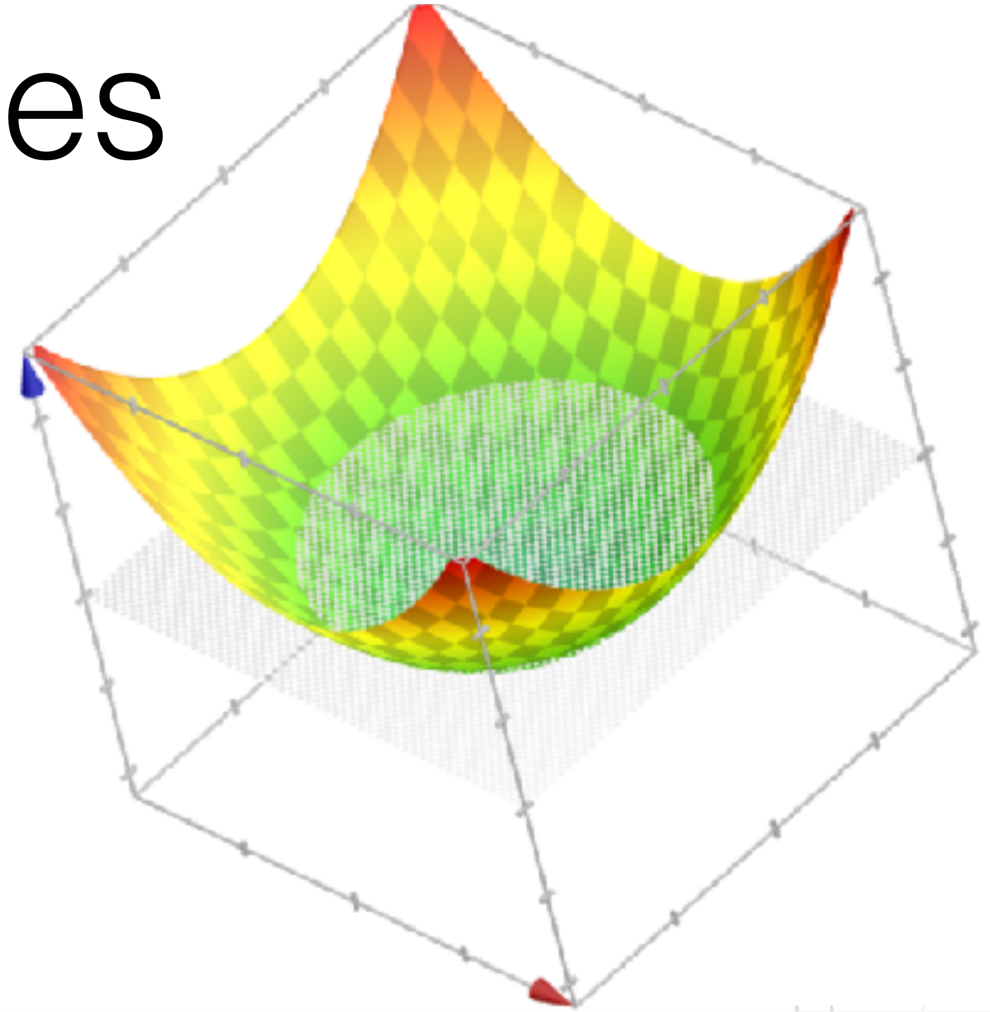
- Idea: can approximate a smooth function with a *k*th order Taylor polynomial (e.g. around 0)

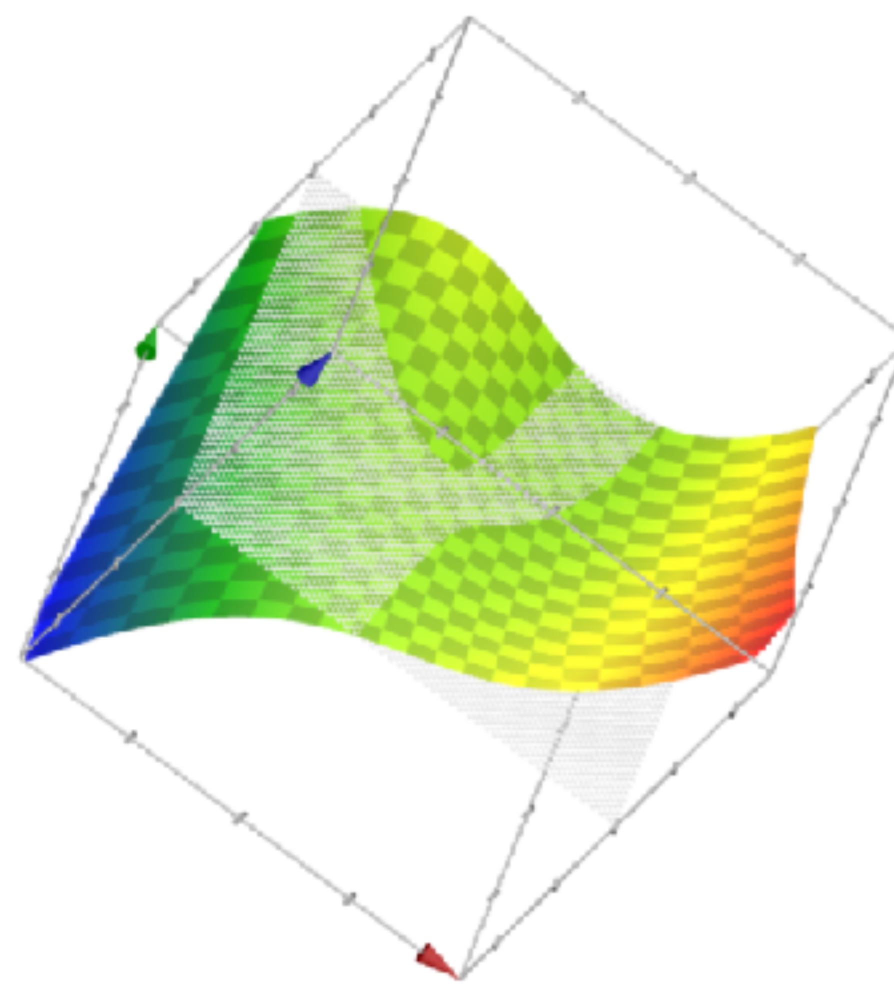
order (*k*)	terms when *d*=1	terms for general d
0	$[1]$	$[1]$
1	$[1, x_1]$	$[1, x_1, \ldots, x_d]$
2	$[1, x_1, x_1^2]$	$[1, x_1, \ldots, x_d,$ $x_1^2, x_1 x_2, \ldots, x_{d-1} x_d, x_d^2]$
3	$[1, x_1, x_1^2, x_1^3]$	$[1, x_1, \ldots, x_d,$ $x_1^2, x_1 x_2, \ldots, x_{d-1} x_d, x_d^2,$ $x_1^3, x_1^2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

Nonlinear boundaries

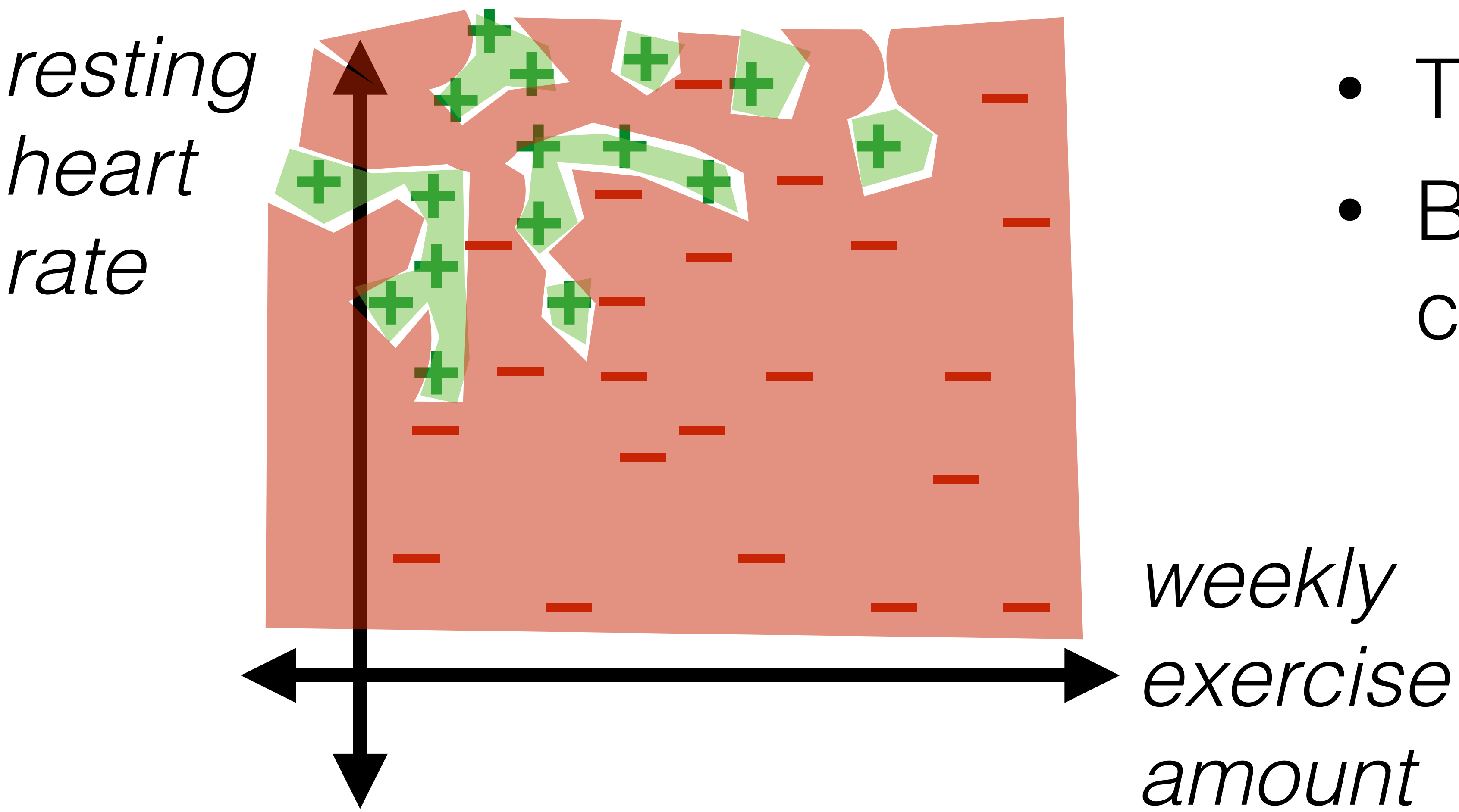








Nonlinear boundaries

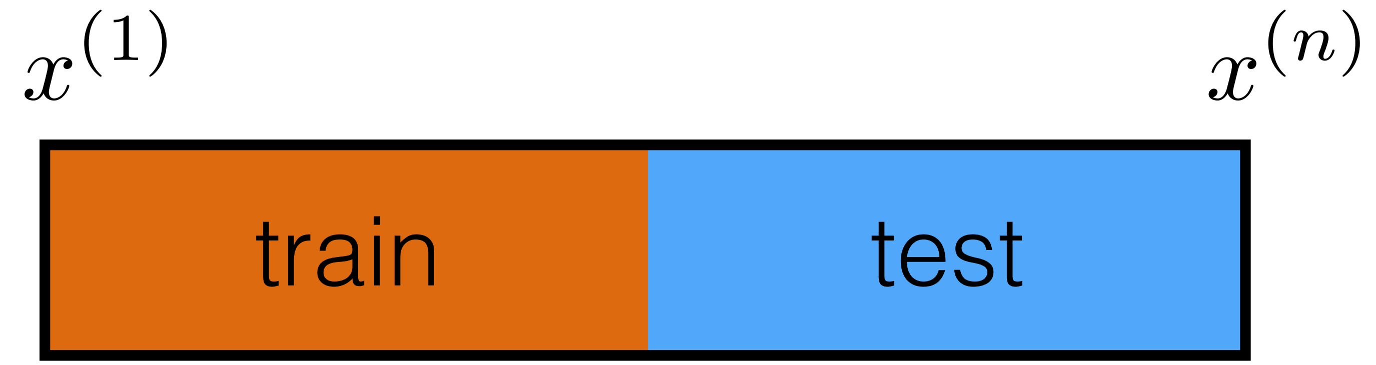


- Training error is 0!
- But seems like our classifier is overfitting

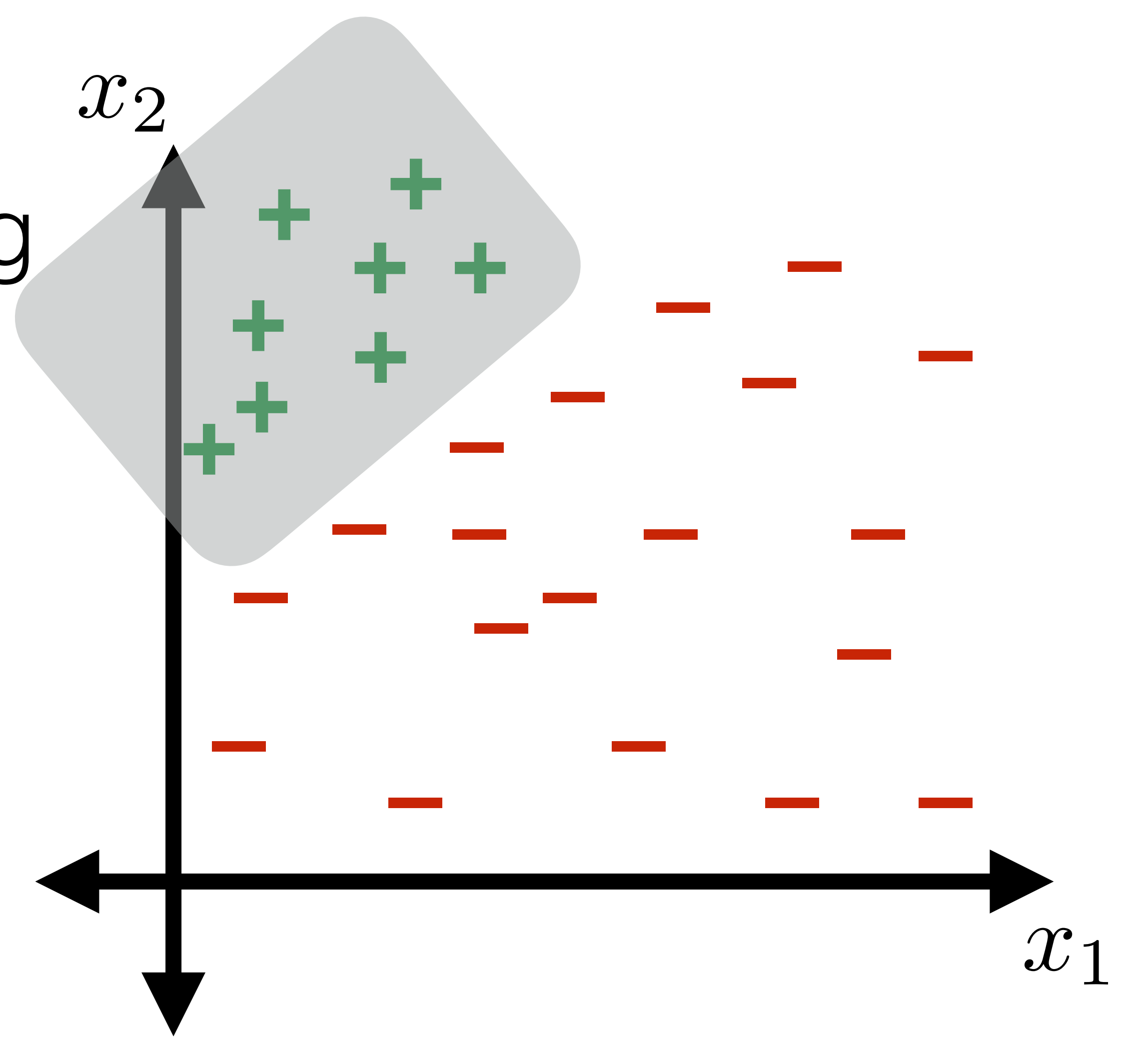
- How can we detect overfitting?
- How can we avoid overfitting?

Evaluation of a learning algorithm

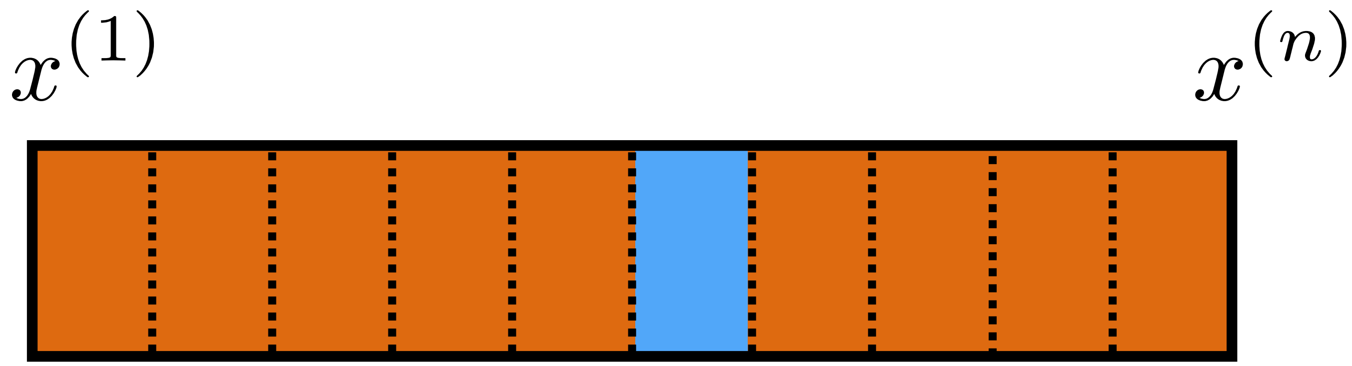
- How good is our learning algorithm on data like ours?



- 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( 𝒟_n , k )
  Divide 𝒟_n into k chunks 𝒟_{n,1}, ..., 𝒟_{n,k} (of
  roughly equal size)
  for i = 1 to k
    train h_i on 𝒟_n \ 𝒟_{n,i} (i.e. except chunk i)
    compute "test" error ℰ(h_i, 𝒟_{n,i}) of h_i on 𝒟_{n,i}
  Return (1/k) Σ_{i=1}^{k} ℰ(h_i, 𝒟_{n,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