Announcements

- Lab1 makeup due on gradescope tomorrow
- Pset1 out Tuesday, due in 2 weeks.

- No screens (laptops, tablets, phones) during the class.

Recall: Types of learning

Supervised

Unsupervised

Reinforcement

Recall: Supervised Learning

- Given a training set consisting of **inputs** and **outputs**, learn to map novel, unseen inputs to outputs

- The novel inputs are called a test set

Output

Categorical **Continuous**

Classification **Regression**

Recall: Supervised Learning

- Given a training set consisting of **inputs** and **outputs**, learn to map novel, unseen inputs to outputs

- The novel inputs are called a test set

Output

Categorical → **Classification**

Continuous → **Regression**

Today

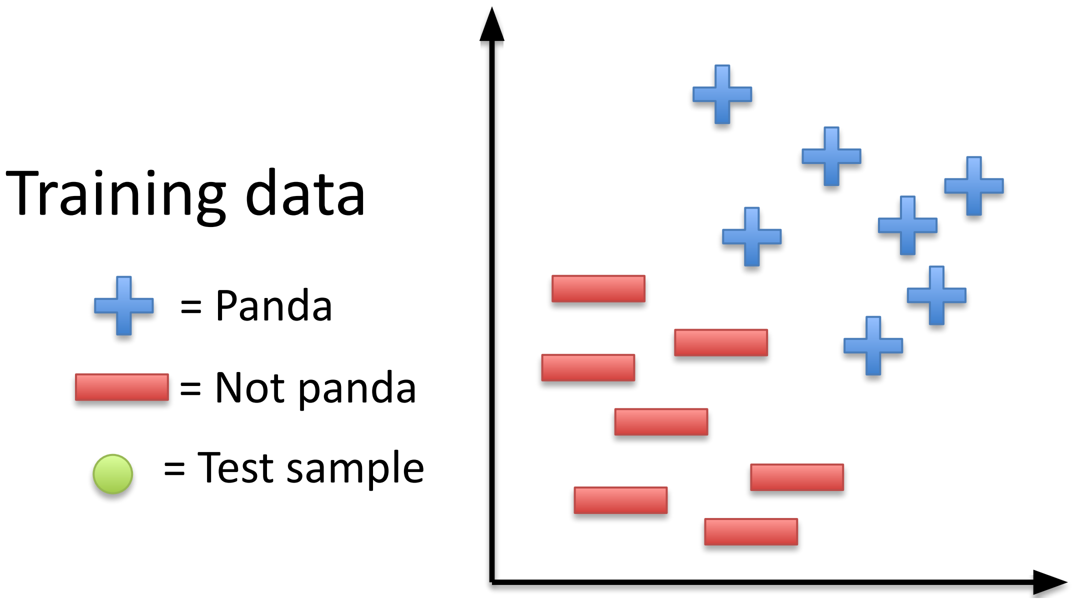
- Classification Intro
  - Nearest Neighbors
  - Learning to classify
    - Error Rates
- Maximum Likelihood

Many slides adapted from Kate Saenko and Relja Arandjelović

Classification

0: “Negative Class” (e.g., dog)

1: “Positive Class” (e.g., panda)



- Learning to separate **two classes**: Binary Classification

slido

What is learning to separate multiple classes called?

ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

What is learning to separate multiple classes called?

Multiple Choice Poll 69 votes 69 participants

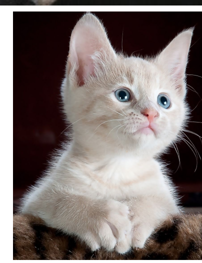
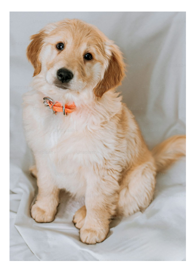
Multi-class classification - 56 votes — 81%

Multi-label classificaiton - 10 votes — 14%

Multi-class, multi-label classification - 3 votes — 4%

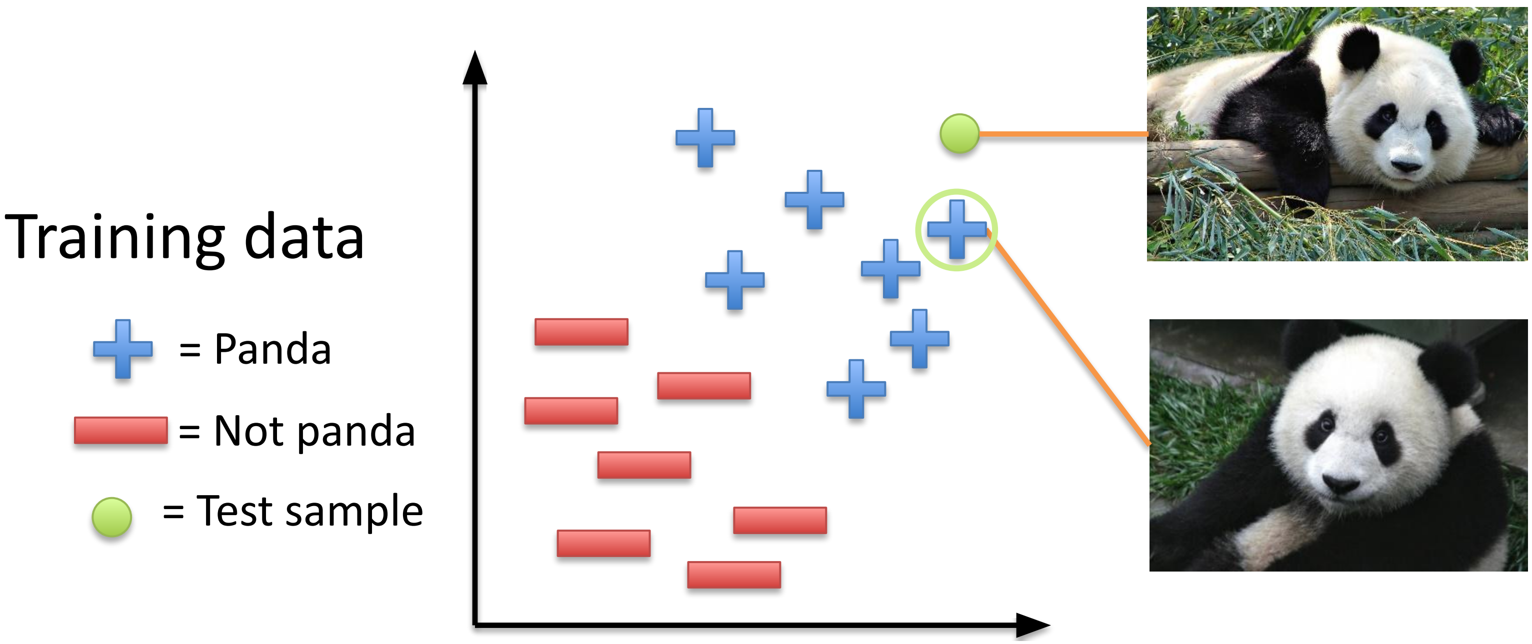
slido

Types of classification

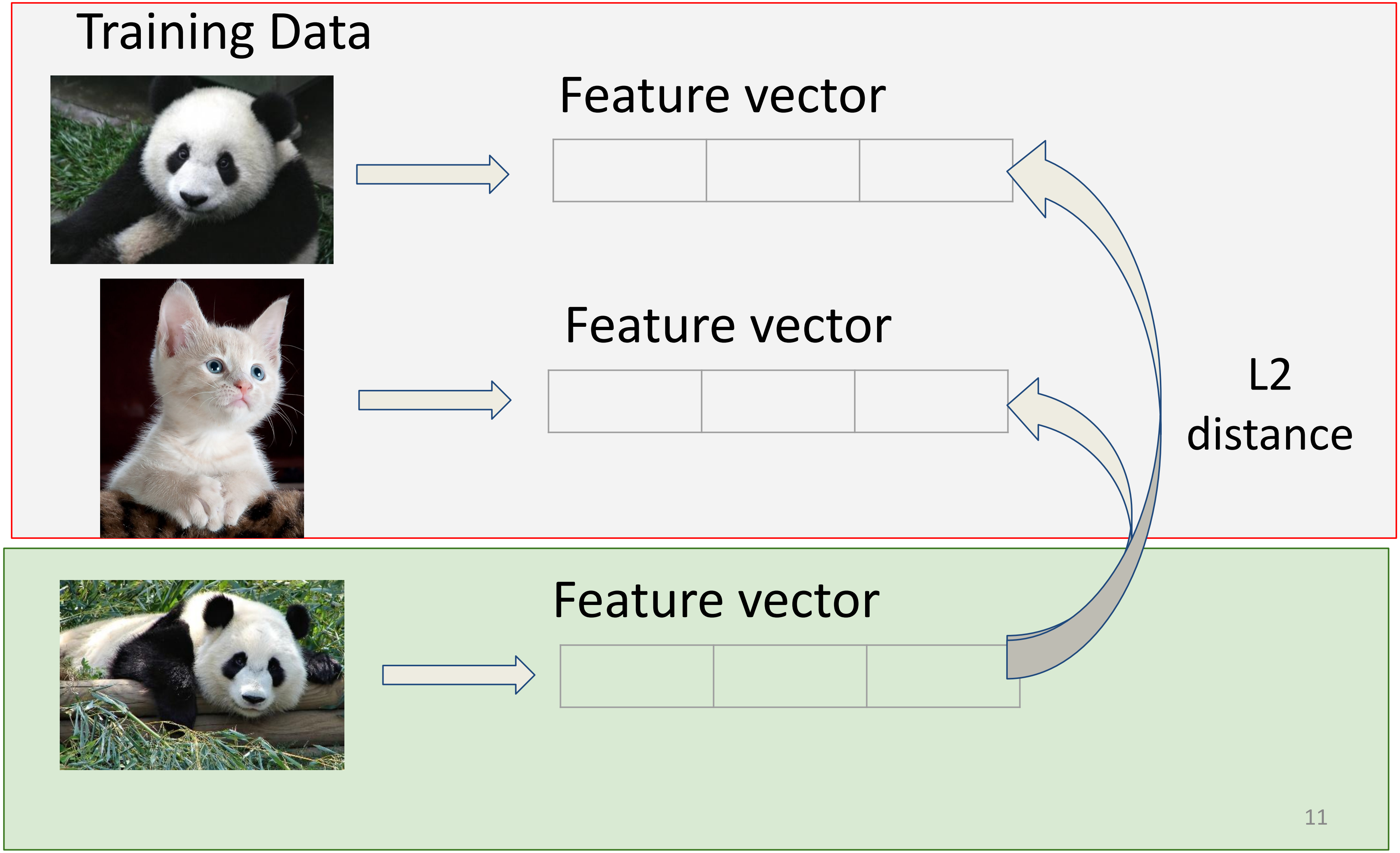
Binary	Multi-class	Multi-label
Panda Not panda 	Panda, Cat, Dog   	(Dog, cat) , (panda)   

Classification: one approach

- **Nearest Neighbor Classifier**
  - Use similarity (e.g., L2 distance) to labeled examples



Nearest Neighbor Classifier



slido

What is L2 distance between two variables x and y?

ⓘ Click **Present with Slido** or install our [Chrome extension] to activate this poll while presenting.

What is L2 distance between two variables x and y?

Multiple Choice Poll 64 votes 64 participants

Euclidean distance, eg: (x-y)^2 - 62 votes

97%

Manhattan distance |x-y| - 2 votes

3%

cos(x,y) - 0 votes

0%

slido

Nearest Neighbor Classifier

Training Data

Feature vector

Feature vector

$$\sum_{i=1}^{d}(x_i - y_i)^2$$

L2 distance

Feature vector

slido

What could be some challenges with the nearest neighbor classifier

Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

What could be some challenges with the nearest neighbor classifier

Multiple Choice Poll 70 votes 70 participants

Very sensitive to how we compute features - 0 votes

0%

Not robust to outliers - 2 votes

3%

Can be close to points from other classes - 2 votes

3%

Very sensitive to the distance metric - 1 vote

1%

All of the above - 65 votes

93%

Challenges: many nuisance parameters

Illumination

Object pose

Clutter

Occlusions

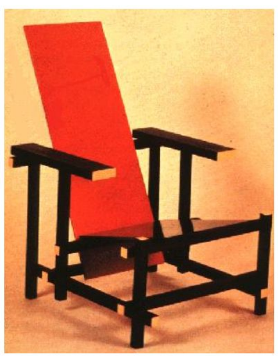
Intra-class appearance

Viewpoint

Slide credit: Kristen Grauman

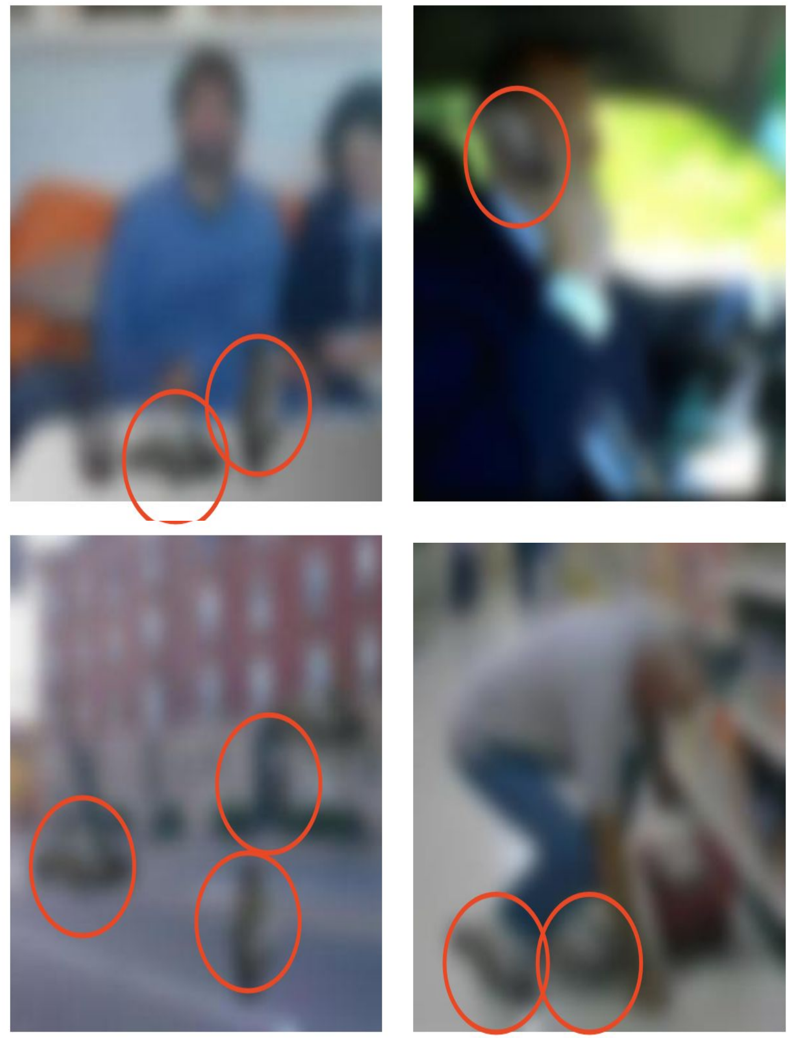
Challenges: intra-class variation

CMOA Pittsburgh

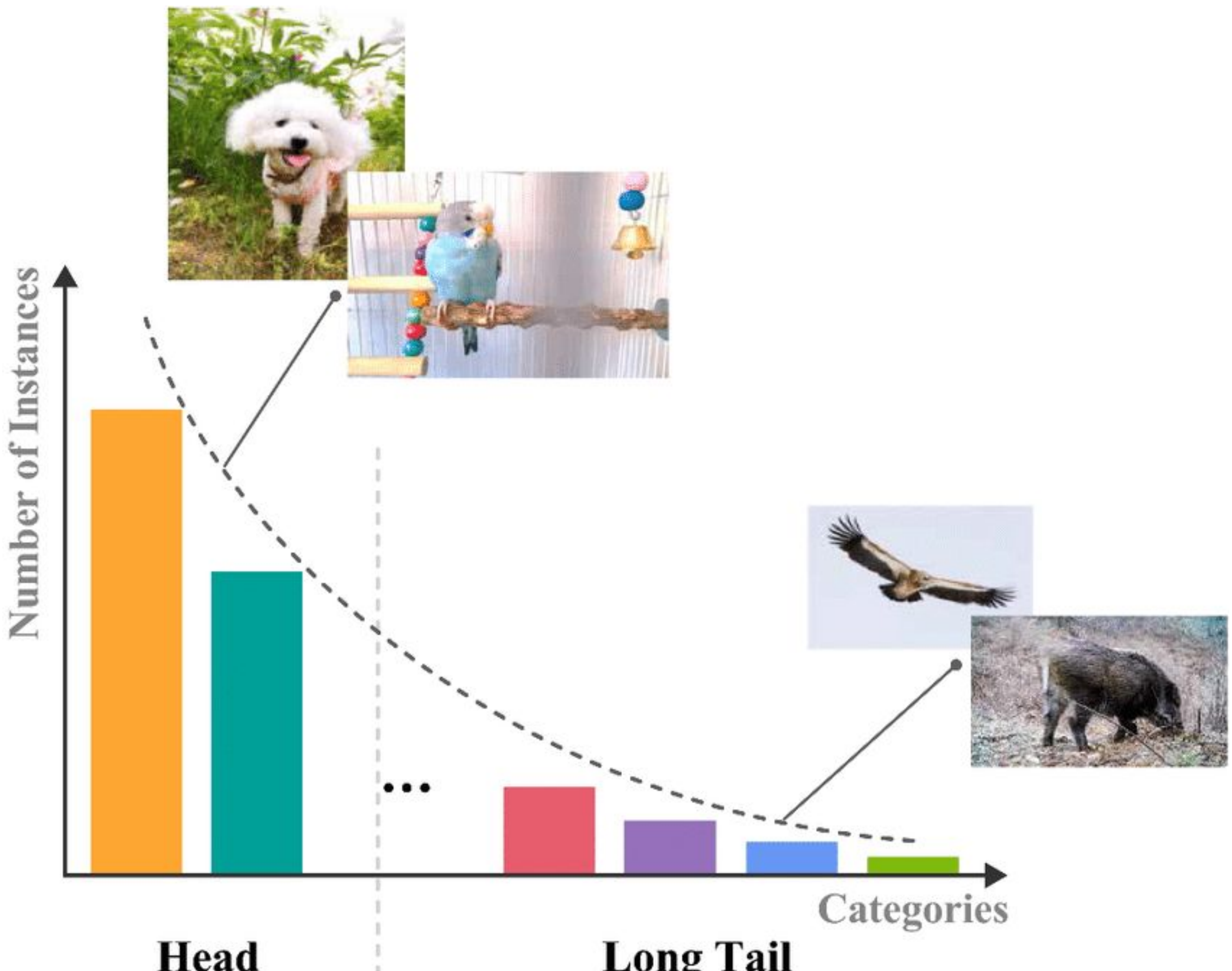


Slide credit: Fei-Fei, Fergus & Torralba

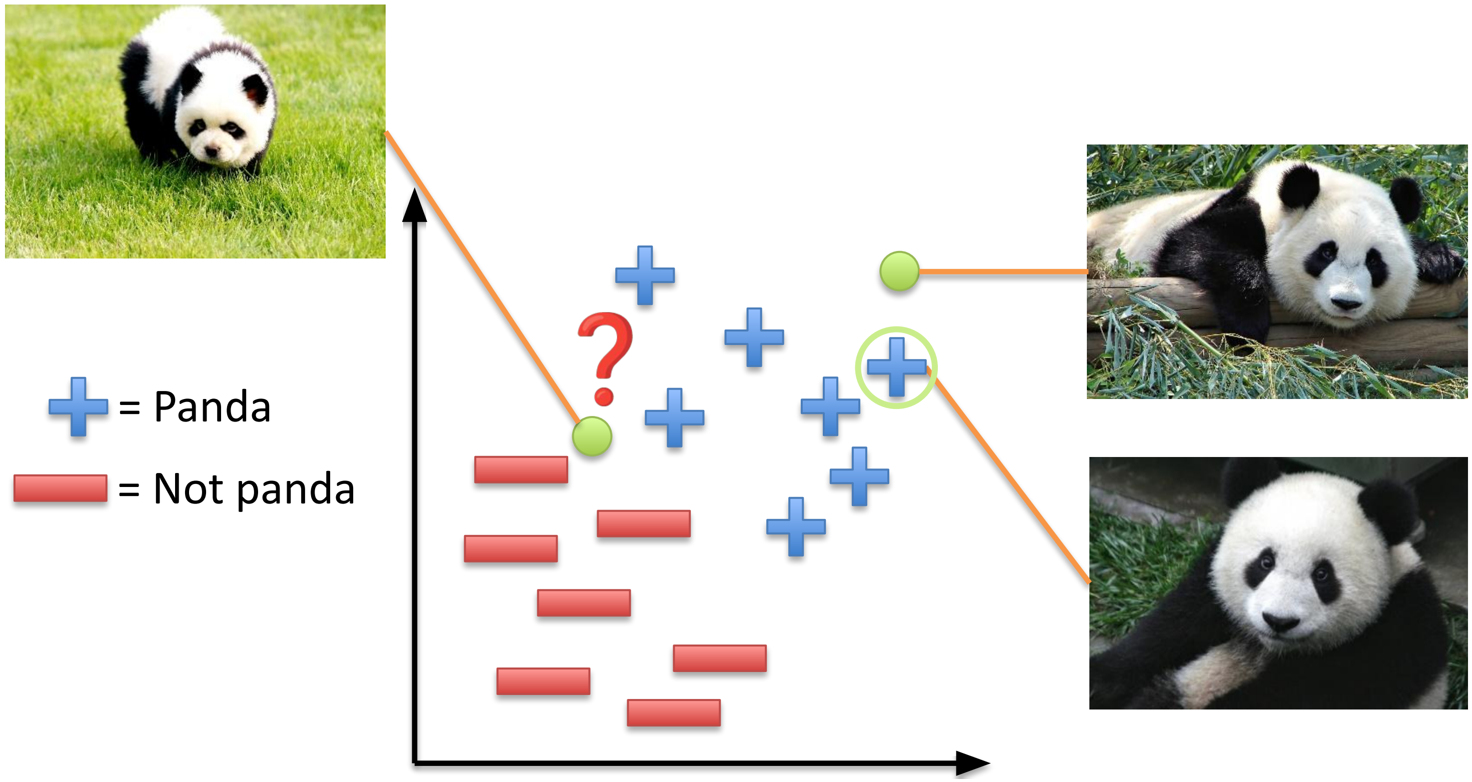
Challenges: Context



Challenges: Long-tailed data distribution



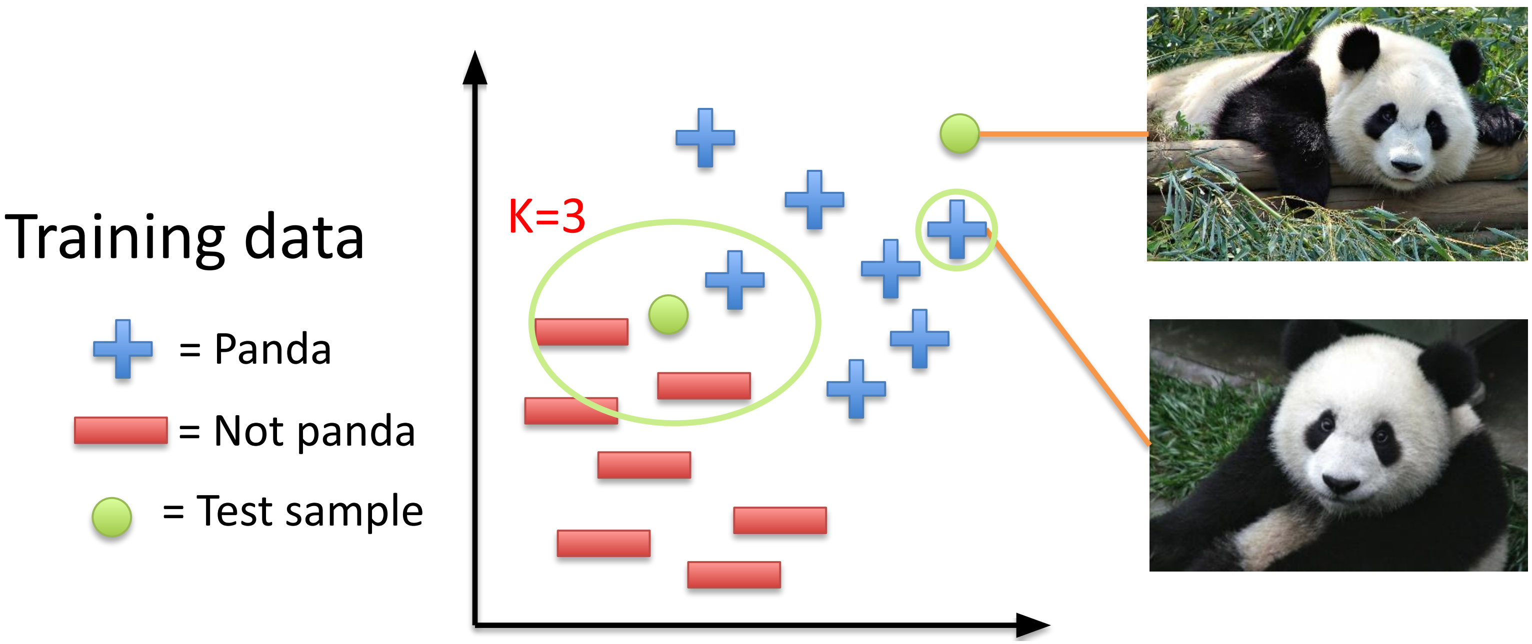
Challenges: Outliers



- Requires a large dataset to work learn a good model

Classification: one approach

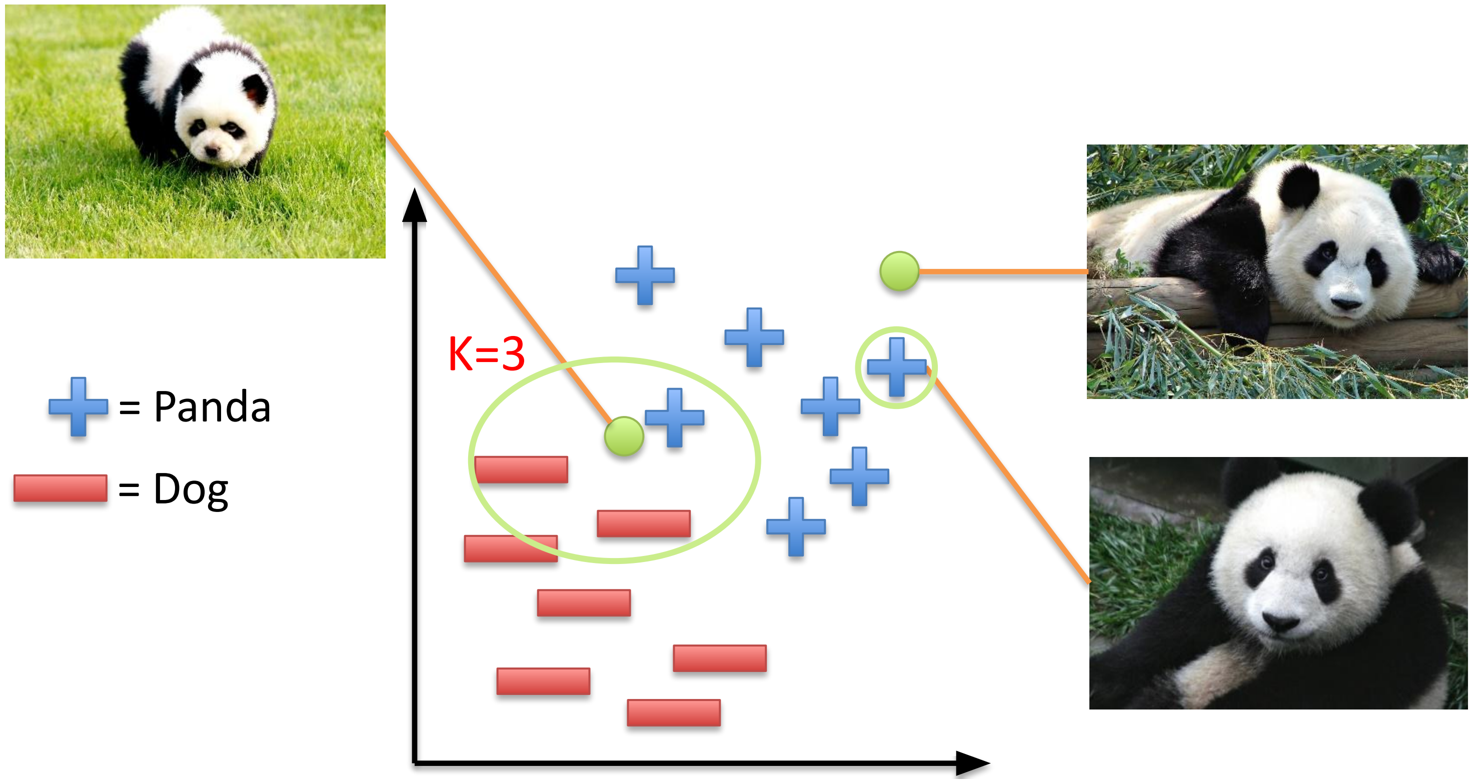
- **K- Nearest Neighbors Classifier**
  - Use similarity (e.g., L2 distance) to labeled examples



Idea for a simple classifier:

Use similarity (e.g., L2 distance) to labeled examples

i.e.,



Takeaways:

- Selecting K requires tuning, and the optimal value will vary
- Distance functions can also significantly affect performance

K-Nearest Neighbor Classifier

Training Data

Feature vector

Feature vector

$$\sum_{i=1}^{d}(x_i - y_i)^2$$

L2
distance

Feature vector

What is a disadvantage of nearest neighbor approach?

Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

What is a disadvantage of nearest neighbor approach?

Quiz question 72 answers 72 participants

Feature computation - 7 answers

10%

Distance computation for every test sample - 12 answers

17%

Choosing a value of K - 1 answer

1%

All of the above. - 52 answers

72%

How to speed NN up?

Issue	**Potential solution**
Feature dimensionality	***Feature Dimensionality reduction?*** ● Reasonable first step, but typically insufficient

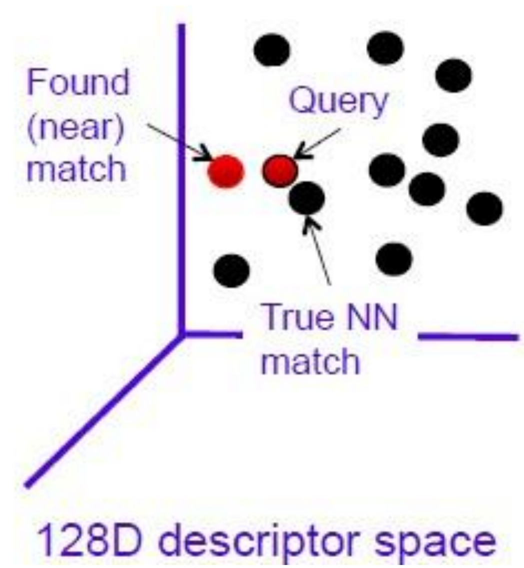
How to speed NN up?

Issue	Potential solution
Feature dimensionality	***Feature Dimensionality reduction?*** ● Reasonable first step, but typically insufficient
Pairwise distance computation	***Use GPUs?*** ● Adds lots of complexity ● Insufficient memory ● Overhead for memory copying between CPU & GPU
	Buy more machines to distribute computation? ● Costs money to buy, maintain, .. ● Adds lots of complexity ● For real-time systems: communication overhead ● Still often insufficient, e.g., if all pairwise distances are needed (e.g. building a neighbourhood graph, clustering, ..)

Finding ***approximate*** nearest neighbor vectors

- Approximations are not guaranteed to find the nearest neighbor

- Can be much faster, but comes at a cost of missing some nearest matches



slido

What are some real-world applications where nearest neighbors are used?

Click **Present with Slido** or install our [Chrome extension] to activate this poll while presenting.

What are some real-world applications where nearest neighbors are used?

Quiz question · 72 answers · 72 participants

Ranking of social media posts - 28 answers ✓ — 39%

Product searches - 68 answers ✓ — 94%

GPS routing - 26 answers — 36%

Identifying if an object is panda or not - 62 answers — 86%

slido

Approximate Nearest Neighbors (ANN)

Is finding only approximate nearest neighbors acceptable?

?

Big Ben

Often times yes!

How approximate NN helps

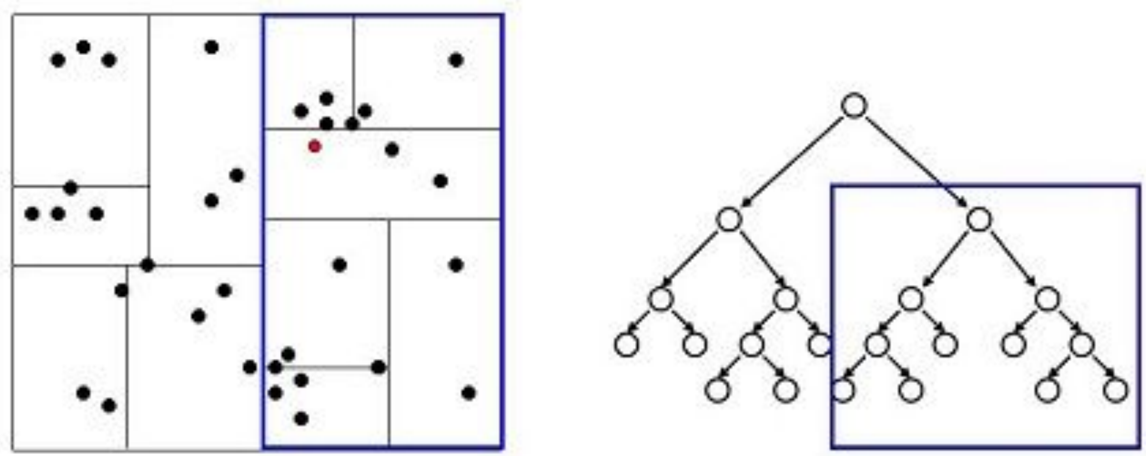
Issue	**Potential solution**
Feature representation	● Vector Quantization ○ eg: binary codes instead of floating point numbers ● Spectral Hashing ○ represent such that simple hamming distance can be used. ● Locality sensitive hashing (LSH) ○ Similar data points are hashed together.

How approximate NN helps

<table>
<tr><th>Issue</th><th>Potential solution</th></tr>
<tr><td>Feature representation</td><td>Vector Quantization
eg: binary codes instead of floating point numbers
Spectral Hashing
represent such that simple hamming distance can be used.
Locality sensitive hashing (LSH)
Similar data points are hashed together.</td></tr>
<tr><td>Pairwise distance computation (approximate)</td><td>(Randomized) K-d trees
Goal: reduce the number of times distance metric is computed.
Idea: If we know that two data points are close to each other, calculate distance to only one of them.</td></tr>
</table>

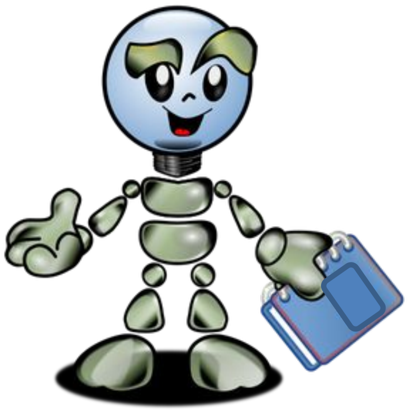
How approximate NN helps

Nearest Neighbor with KD Trees



Examine nearby points first: Explore the branch of the tree that is closest to the query point first.

24



Learning to Classify

Intro

Example: Temperature Prediction

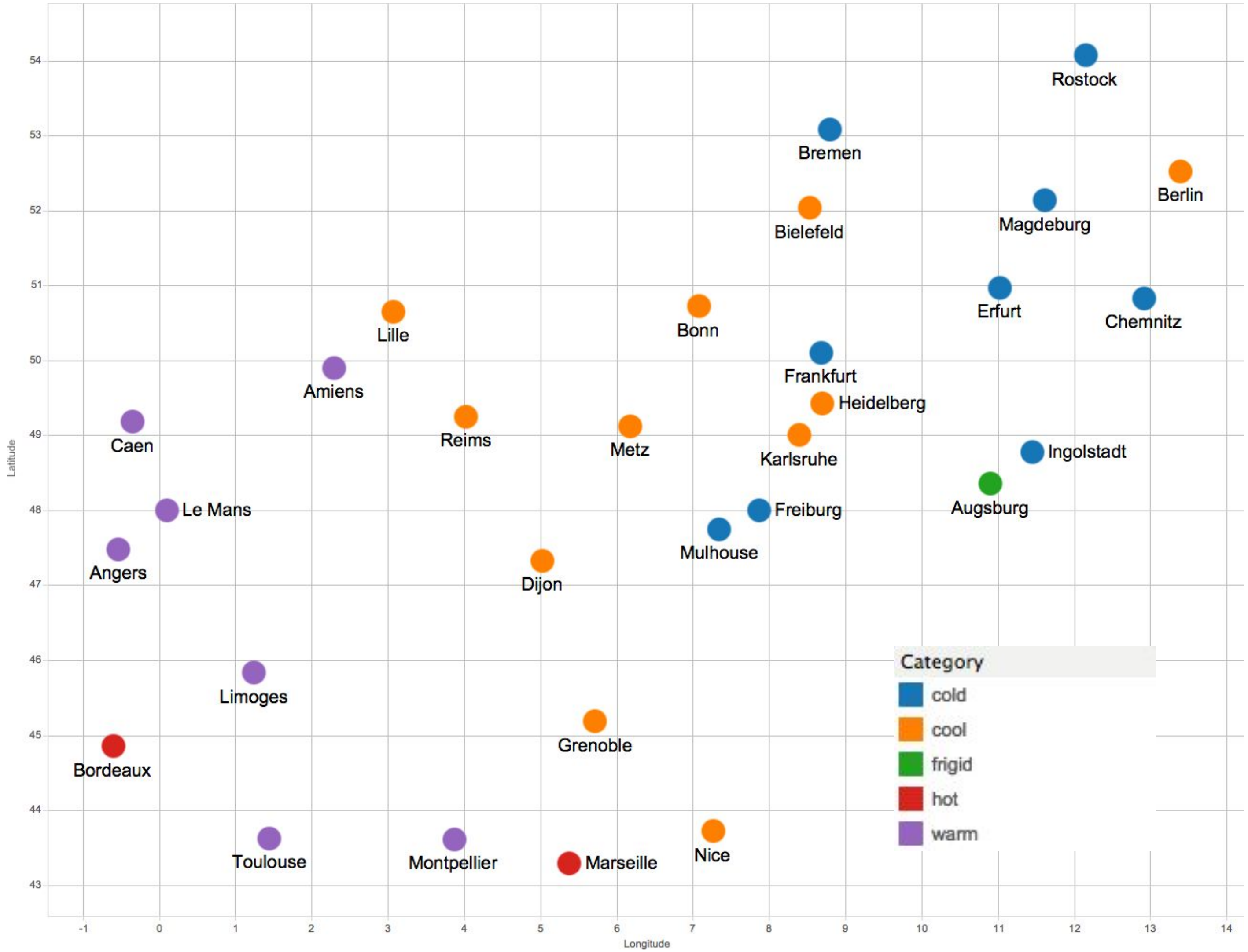
- City temperatures – France and Germany
- Features: longitude, latitude
- Labels: frigid, cold, cool, warm, hot

Nice (7.27, 43.72) cool
Toulouse (1.45, 43.62) warm
Frankfurt (8.68, 50.1) cold
......

Predict temperature category from longitude and latitude

Credit: Jennifer Widom

Example: Temperature Prediction



Credit: Jennifer Widom

Training set

Training set:

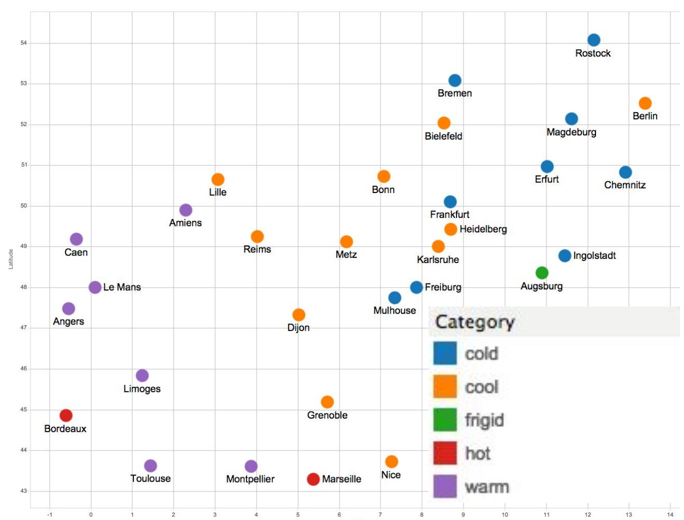
7.27, 43.72 (Nice)	cool
1.45, 43.62 (Toulouse)	warm
8.68, 50.1 (Frankfurt)	cold
…	…

Supervised Learning

Predict: Is the city cold?

What should the learner be??

Want: input x → ? → output y

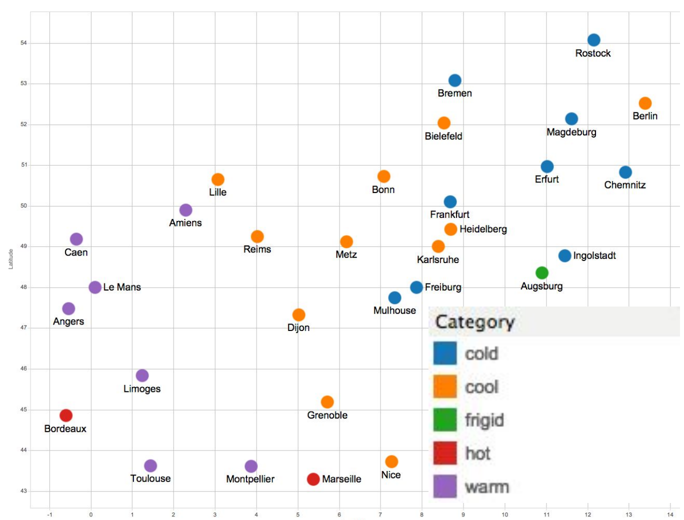


Hypothesis γ

γ : function parametrized by θ, e.g.,

$$\gamma(x) = \text{sign}(\underbrace{\boxed{a}}_{\theta_{0,1}}x + \underbrace{\boxed{b}}_{\theta_2})$$

Want: input x $\longrightarrow$ $\boxed{\gamma_{a,b}{}^*}$ $\longrightarrow$ output y

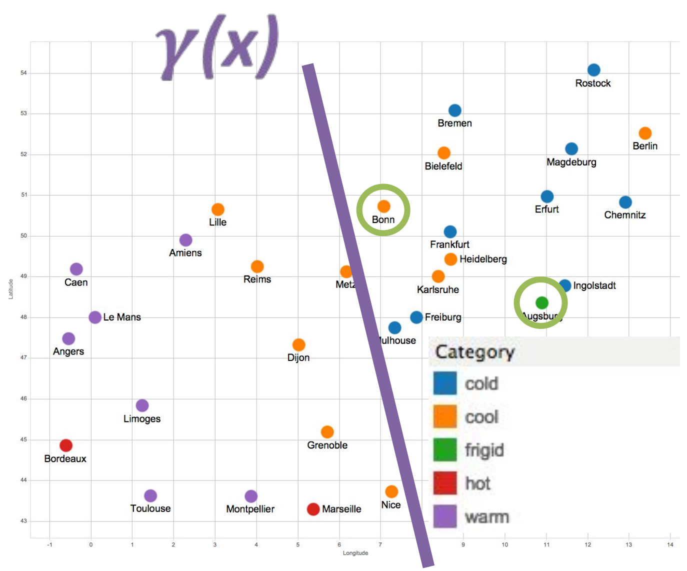


How to learn a,b?

But what if $\gamma(x_i) \neq y_i$?

Given:

Want: input x $\longrightarrow$ $\boxed{\gamma_{a,b}{}^*}$ $\longrightarrow$



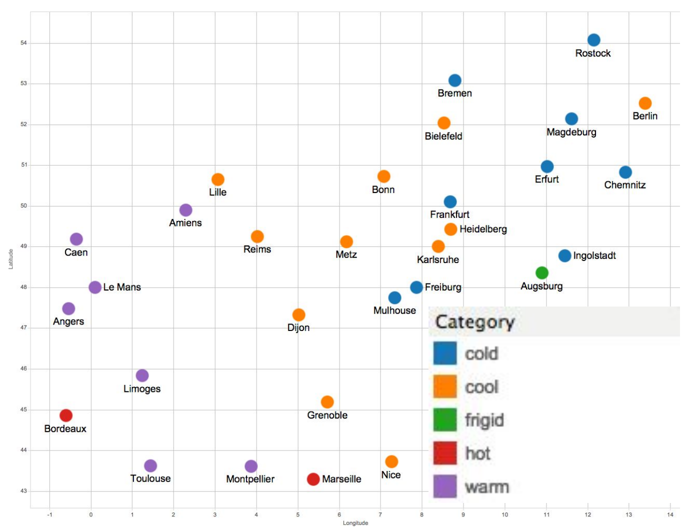
Cost function

Given: Training Set $\{x_i, y_i\}$

Cost/Error function $\text{Cost}(\gamma(x_i), y_i)$

learning == minimizing cost

Want: input $x \longrightarrow \boxed{\gamma_{\mathrm{a,b}}{}^{*}} \longrightarrow$ output y



Supervised learning in one slide

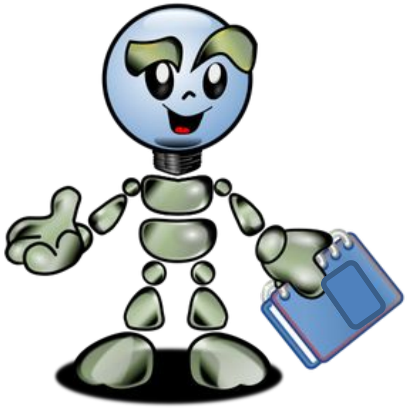
Given:

Cost function $\mathrm{Cost}(\gamma(x_i), y_i)$

learning == minimizing cost

Learn a,b*: $\min_{a,b} \mathrm{Cost}(\gamma_{a,b}(x_i), y_i)$

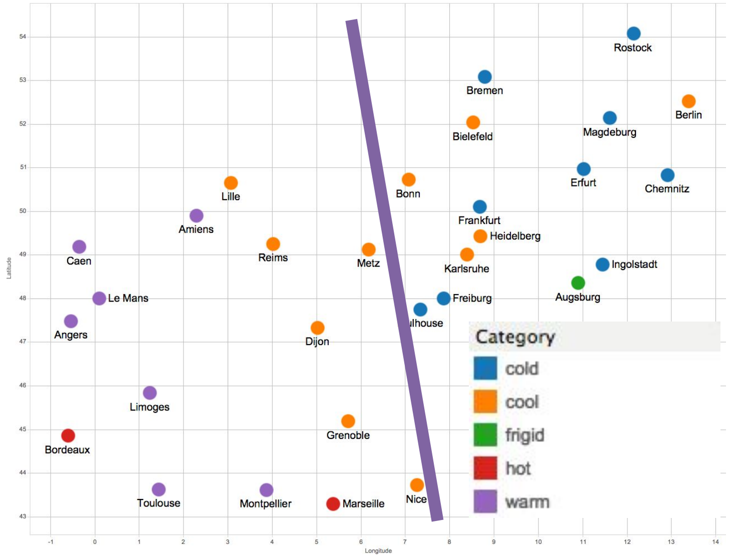
Result: $\longrightarrow \boxed{\gamma_{a,b}{}^*} \longrightarrow$ output y_i



Learning to Classify

Error Rates

How do we know if γ is good?



Linear hypothesis:

$$\gamma_{a,b}(x) = \text{sign}(ax + b)$$

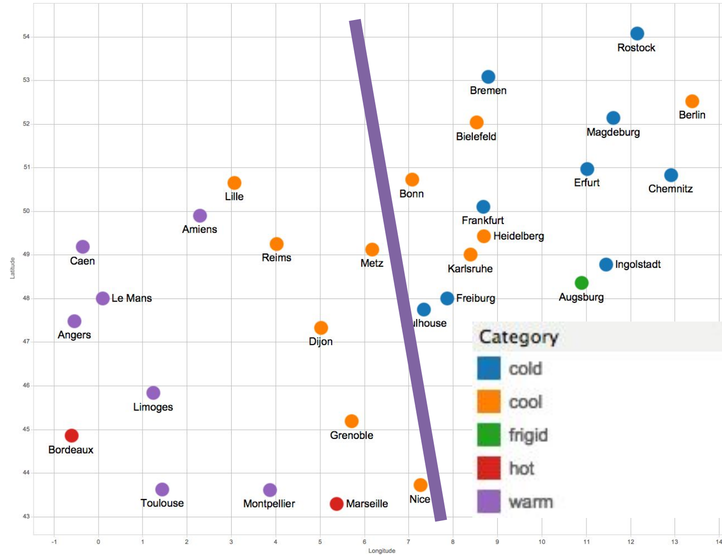
Error Function:

Portion of incorrect predictions

$$Error(\gamma_{a,b}, D\{x, y\}) = \frac{1}{N} \sum_{i=1}^{N} \gamma_{a,b}(x_i) \neq y_i$$

Goal: minimize $Error(\gamma_{a,b}, D\{x, y\})$

What is a good baseline to compare to?



Current hypothesis:

$$\gamma_{a,b}(x) = \operatorname{sign}(ax + b)$$

Random Baseline γ_{rand} (a know nothing strategy):

- Assign a random class label to each datapoint

slido

$$Error(\gamma_{a,b}, D\{x,y\}) > Error(\gamma_{rand}, D\{x,y\})$$

What does it mean to have a higher error rate than a random baseline?

ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

What does it mean to have a higher error rate than a random baseline?

Quiz question | 70 answers | 70 participants

Our hypothesis is performing very well - 2 answers — 3%

Our hypothesis is performing very poorly - 68 answers ✓ — 97%

slido

Diving deeper into model analysis

	Predicted "1"	Predicted "0"
GT Label "1"	**True Positive (TP)**	False Negative (FN)
GT Label "0"	False Positive (FN)	True Negative (TN)

Diving deeper into model analysis

Confusion Matrix Example (Table 1.1 in Forsyth)

Predicted labels

True labels

	0	1	2	3	4	Class error
0	151	7	2	3	1	7.9%
1	32	5	9	9	0	91%
2	10	9	7	9	1	81%
3	6	13	9	5	2	86%
4	2	3	2	6	0	100%

Diving deeper into model analysis

	Predicted “1”	Predicted “0”
GT Label “1”	**True Positive (TP)**	**False Negative (FN)**
GT Label “0”	False Positive (FP)	True Negative (TN)

$$Precision = \frac{TP}{TP + FP}$$

Predict

True	0	1	2	3	4	Class error
0	151	7	2	3	1	7.9%
1	32	5	9	9	0	91%
2	10	9	7	9	1	81%
3	6	13	9	5	2	86%
4	2	3	2	6	0	100%

Diving deeper into model analysis

	Predicted “1”	Predicted “0”
GT Label “1”	**True Positive (TP)**	**False Negative (FN)**
GT Label “0”	False Positive (FP)	True Negative (TN)

$$Recall = \frac{TP}{TP + FN}$$

Predicted labels

True labels

	0	1	2	3	4	Class error
0	151	7	2	3	1	7.9%
1	32	5	9	9	0	91%
2	10	9	7	9	1	81%
3	6	13	9	5	2	86%
4	2	3	2	6	0	100%

Precision and Recall

- **Precision:**
  a. The percentage of predictions that are correct
  b. The ability to identify **only** relevant data points

$$Precision = \frac{TP}{TP + FP}$$

- **Recall:**
  a. The percentage of relevant data points that are correctly identified
  b. The ability to identify **all** relevant data points

$$Recall = \frac{TP}{TP + FN}$$

Diving deeper into model analysis

Predicted labels

True labels

	0	1	2	3	4	Class error
0	151	7	2	3	1	7.9%
1	32	5	9	9	0	91%
2	10	9	7	9	1	81%
3	6	13	9	5	2	86%
4	2	3	2	6	0	100%

Diving deeper into model analysis

Predicted labels

True labels

	0	1	2	3	4	Class error
0	151	7	2	3	1	7.9%
1	32	5	9	9	0	91%
2	10	9	7	9	1	81%
3	6	13	9	5	2	86%
4	2	3	2	6	0	100%

What else do you notice in this confusion matrix?

Predicted labels

True labels

	0	1	2	3	4	Class error
0	151	7	2	3	1	7.9%
1	32	5	9	9	0	91%
2	10	9	7	9	1	81%
3	6	13	9	5	2	86%
4	2	3	2	6	0	100%

How to address the data distribution?

How to address the data distribution?

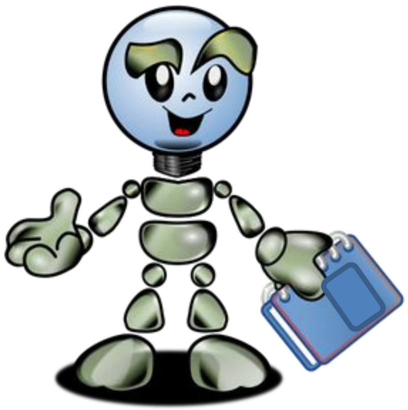
Quiz question · 65 answers · 65 participants

Sample more data from classes 3,4 - 60 answers ✓ — 92%

Reduce the samples of class 0 - 42 answers — 65%

Make replicas of the data from 3,4 - 24 answers ✓ — 37%

slido



Maximum Likelihood Principle

Recall: Cost function

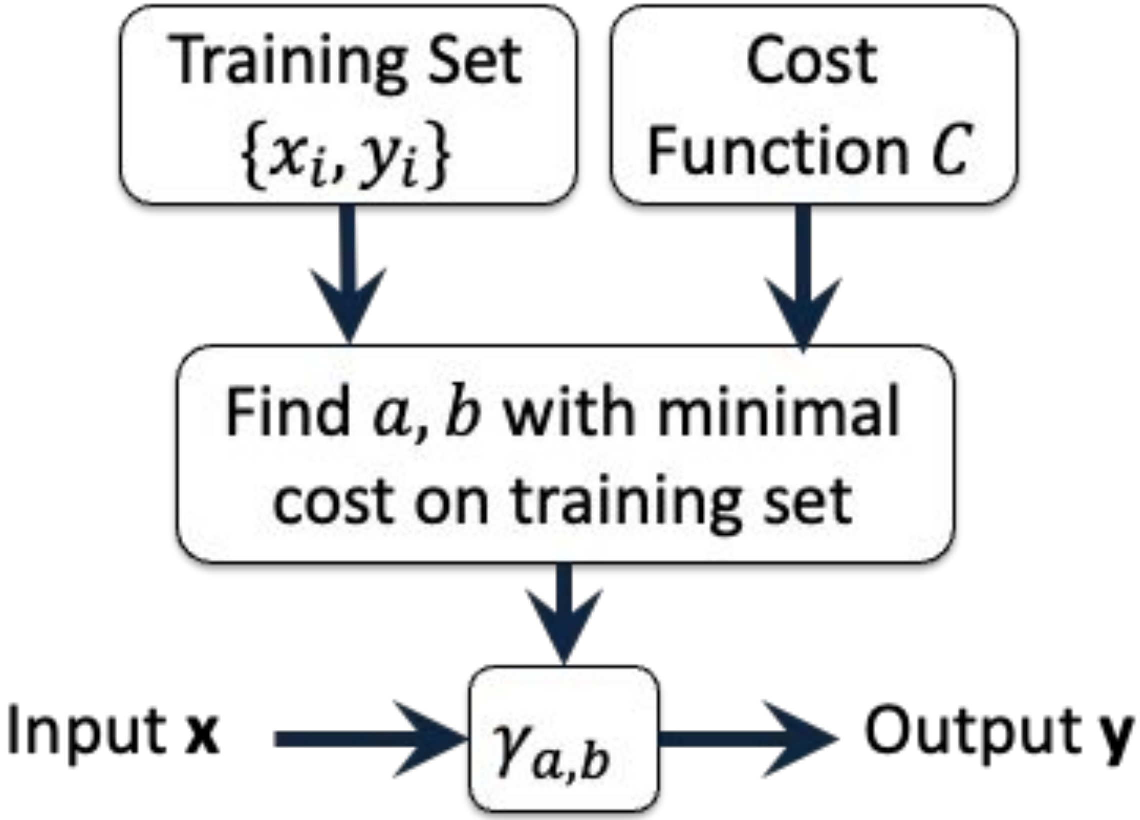
Given: Training Set $\{x_i, y_i\}$

Cost/Error function $\text{Cost}(\gamma(x_i), y_i)$

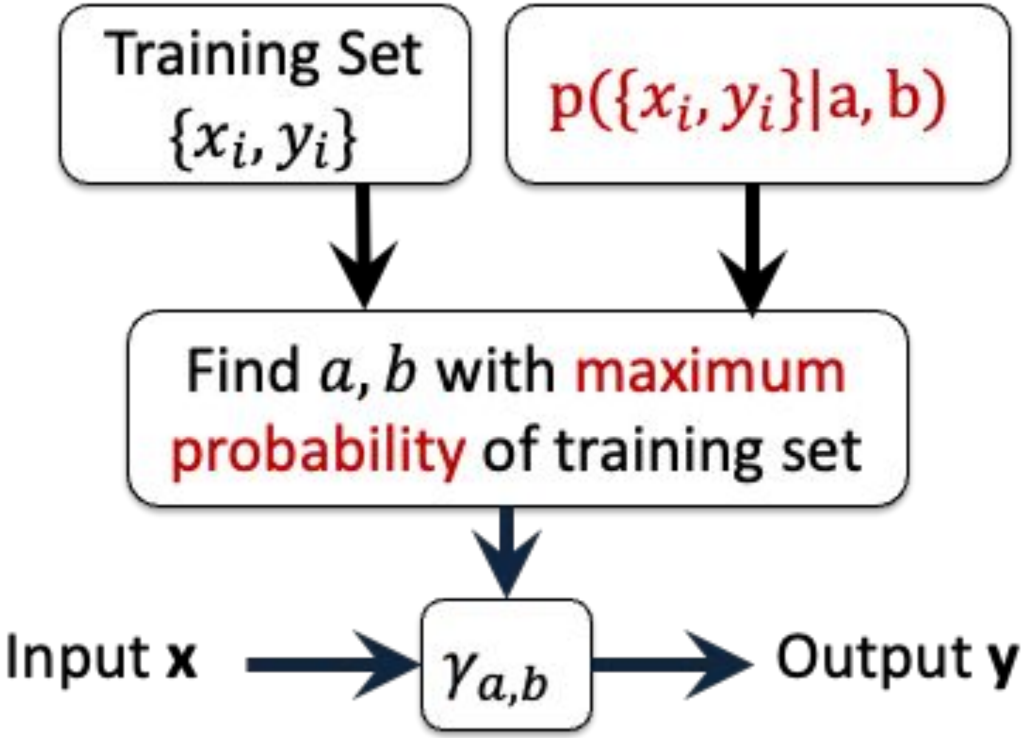
learning == minimizing cost

Want: input x $\longrightarrow$ $\boxed{\gamma_{\text{a,b}}{}^{*}}$ $\longrightarrow$ output y

Recall: Cost Function



Alternative View:
"Maximum Likelihood"



Maximum Likelihood: Example

- Intuitive example: Estimate a coin toss

  I have seen 3 flips of heads, 2 flips of tails, what is the chance of head (or tail) of my next flip?

- Model:

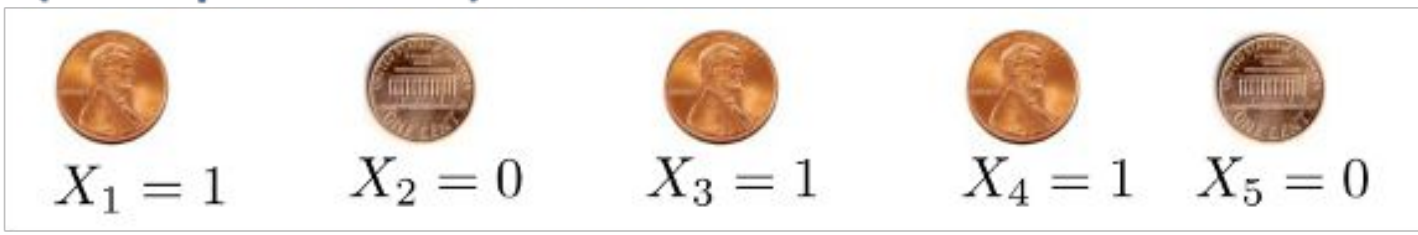
  Each flip is a Bernoulli random variable X

  X can take only two values: 1 (head), 0 (tail)

  $$p(X = 1) = \theta, \quad p(X = 0) = 1 - \theta$$

- θ is a parameter to be identified from data

- 5 (independent) trials



- Likelihood of all 5 observations:

$$p(X_1, \dots, X_5 | \theta) = \theta^3 (1 - \theta)^2$$

- Intuition

ML chooses θ such that likelihood is maximized

- 5 (independent) trials

- Likelihood of all 5 observations:

$$p(X_1, \dots, X_5 | \theta) = \theta^3 (1 - \theta)^2$$

- Solution (left as exercise)

$$\theta_{ML} = \frac{3}{(3+2)}$$

i.e. fraction of heads in total number of trials

IID Observations

- independently identically distributed random variables
- If u^i are i.i.d. r.v.s, then

$$p(u^1, u^2, \dots, u^m) = p(u^1)p(u^2)\dots p(u^m)$$

- A reasonable assumption about many datasets, but not always

Maximum likelihood way of estimating model parameters θ

In general, assume data is generated by some distribution

$$U \sim p(U|\theta)$$

Observations (i.i.d.)

$$\mathrm{D} = \{u^{(1)}, u^{(2)}, \ldots, u^{(m)}\}$$

Maximum likelihood way of estimating model parameters θ

In general, assume data is generated by some distribution

$$U \sim p(U|\theta)$$

Observations (i.i.d.)

$$\mathrm{D} = \{u^{(1)}, u^{(2)}, \dots, u^{(m)}\}$$

Maximum likelihood estimate

Likelihood

$$\mathcal{L}(\mathrm{D}) = \boxed{\prod_{i=1}^{m} p(u^{(i)}|\theta)}$$

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\mathrm{D})$$

Maximum likelihood way of estimating model parameters θ

In general, assume data is generated by some distribution

$$U \sim p(U|\theta)$$

Observations (i.i.d.)

$$\mathrm{D} = \{u^{(1)}, u^{(2)}, \dots, u^{(m)}\}$$

Maximum likelihood estimate

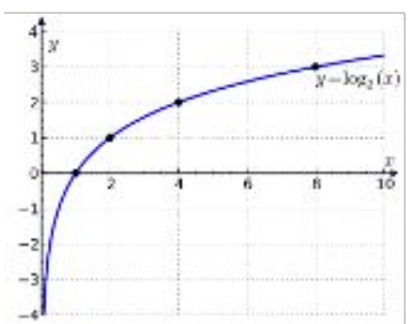
Likelihood

$$\mathcal{L}(\mathrm{D}) = \boxed{\prod_{i=1}^{m} p(u^{(i)}|\theta)}$$

Log likelihood

$$\theta_{ML} = \operatorname*{argmax}_{\theta} \mathcal{L}(\mathrm{D})$$

$$= \operatorname*{argmax}_{\theta} \boxed{\sum_{i=1}^{m} \log p(u^{(i)}|\theta)}$$



log(f(x)) is monotonic/increasing, same argmax as *f(x)*

Next Class

Classification II:

Overfitting, cross validation, naive bayes, support vector machines intro

Reading: Forsyth Ch 1.3, 2.1-2.1.2