Welcome

- Course webpage
  - http://ttic.edu/intromlss2018
- Daily schedule (most days)
  - Lectures
    - 9am-noon
    - Room 526, 6045 S Kenwood Ave
    - Instructors: Suriya Gunasekar, Karl Stratos
  - Lunch
    - Noon-1pm
  - Invited talks + Coding assignments
    - 1-5pm
      Room 526+530, 6045 S Kenwood Ave

Introduction to Machine Learning Summer School
June 18, 2018 - June 29, 2018, Chicago

TOYOTA TECHNOLOGICAL INSTITUTE AT CHICAGO

NSF

THE UNIVERSITY OF CHICAGO

Today's schedule

- 9am-9:30pm Course setup
- 9:30:00-11:00am Lecture 1.a
  - Introduction, supervised learning
- 11:00-12:30pm Lecture 1.b
  - Linear regression
- 12:30-2:00pm Lunch
  - group delivery order, or
  - walk to nearby places - Press Café, Booth School of Business, Hutchinson Commons, Build Coffee
- 2:00-5pm Programming

Teaching Assistants

Rebecca Kotsonis

Pedro Savarese

Kevin Stangl

Emails TAs **ta-intromlss2018@ttic.edu**

Day 1: Introduction, Linear Regression

Introduction to Machine Learning Summer School
June 18, 2018 - June 29, 2018, Chicago

Instructor: Suriya Gunasekar, TTI Chicago

18 June 2018

TOYOTA TECHNOLOGICAL INSTITUTE AT CHICAGO

THE UNIVERSITY OF CHICAGO

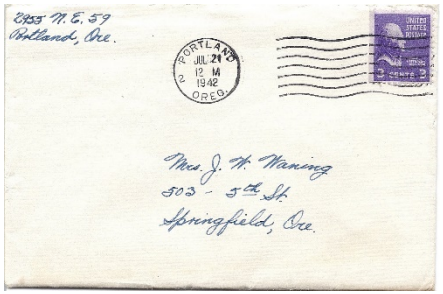
NSF

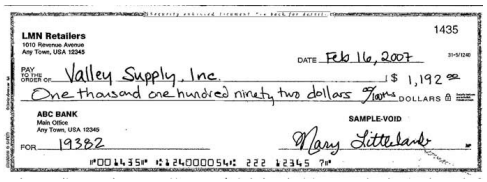
What is machine learning?

Let us first see some examples…

Optical character recognition

→ “b”





Advantages of data driven approach

- Less programming
- Easily adaptive
- Less dependent on expert knowledge

Slide credit: Nati Srebro

Spam detection

Gmail

Suriya Gunasekar <suriyag88@gmail.com>

Don't forget to take your free ride!

Divvy Bikes <support@divvybikes.com>
Reply-To: Divvy Bikes <support@divvybikes.com>
To: suriyag88@gmail.com

Fri, May 25, 2018 at 12:24 PM

First ride FREE!

DIVVY

Your free ride awaits.

Download the new Divvy app to redeem.

You keep saying you're going to "explore your city more," and now we're giving you a super easy way to do it. With your first 30-minute Single Ride FREE, you'll actually hit up all those summer hot spots. Download the new Divvy app and enter code **FIRSTRIDEFREE** before this offer rides away.

GET MY FREE RIDE

Offer expires 6/17

- less programming
- less expert knowledge
- adaptive

sender info, recipients info, length of text/subject, timestamp, subject line, words/pairs of words in text, previous interactions with senders/fellow recipients

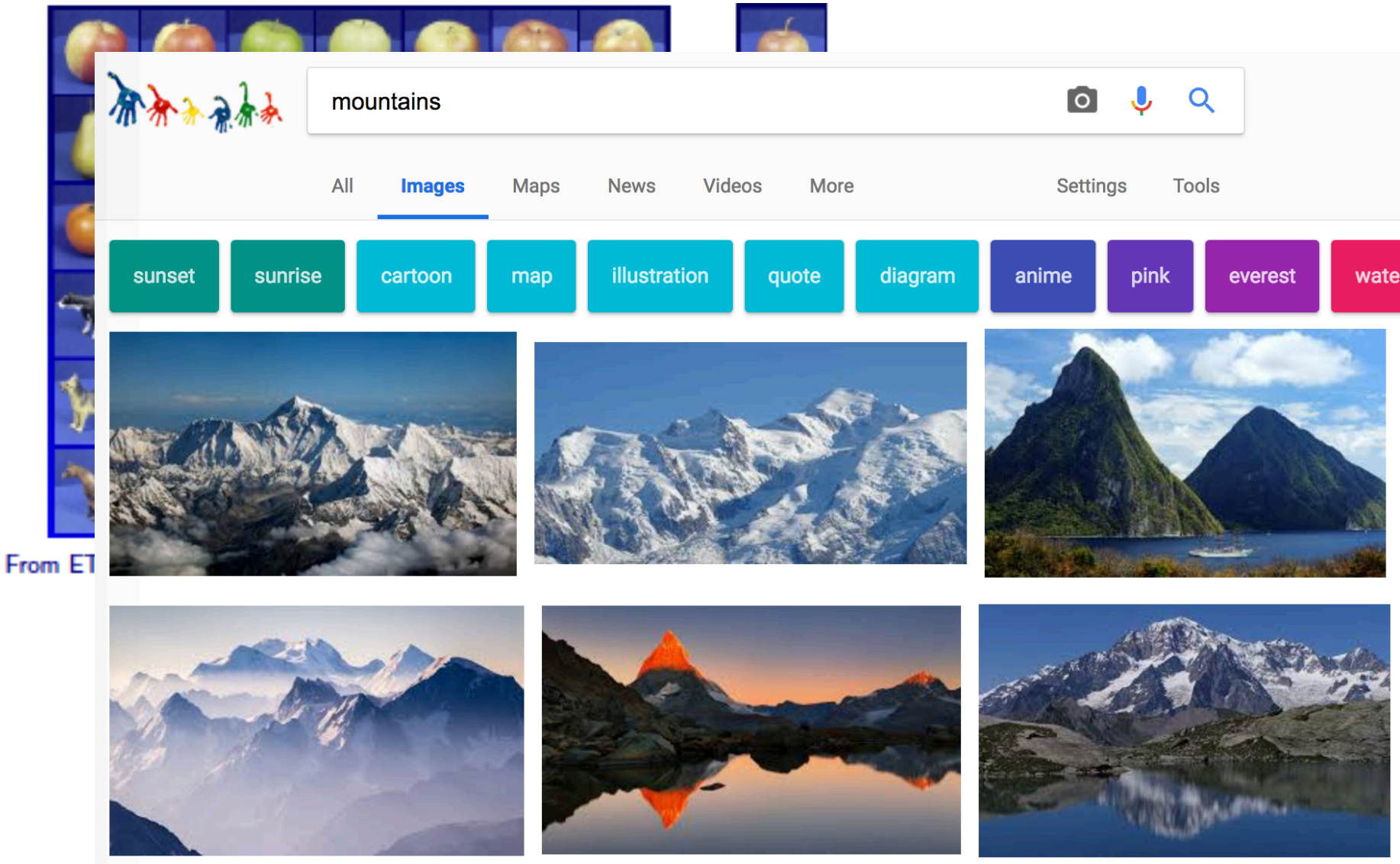
spam or no-spam

Object recognition

From ETH database of object categories, [Leibe & Schiele 2003]

Slide credit: Greg Shaknarovich

Object recognition



Slide credit: Greg Shaknarovich

Face detection/recognition

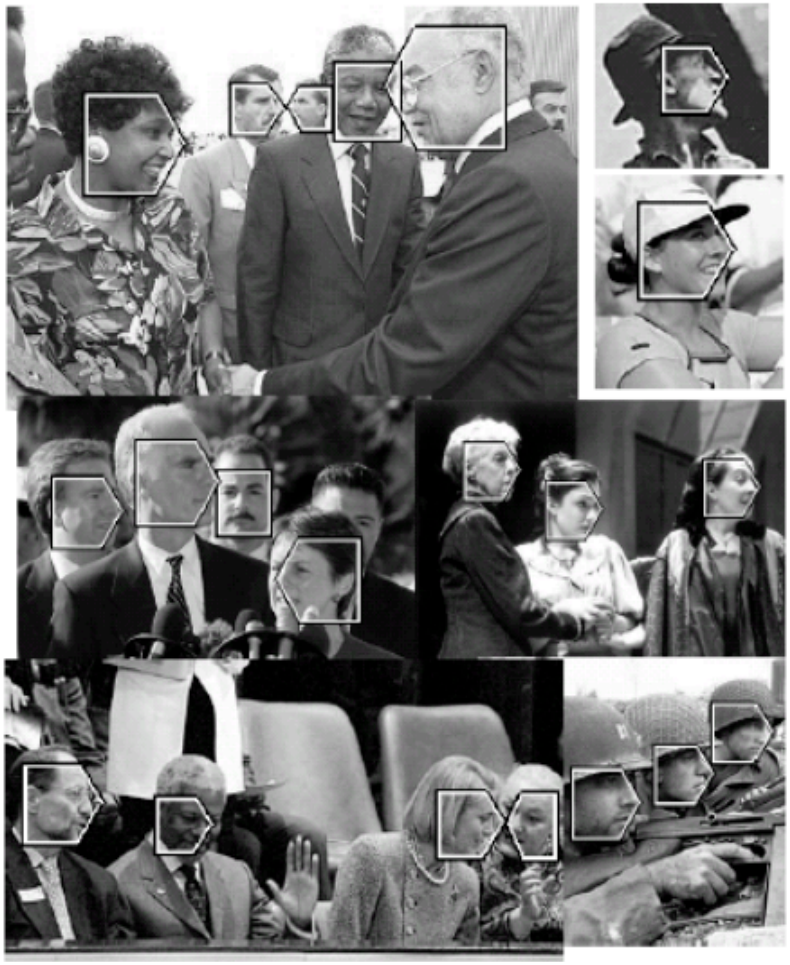
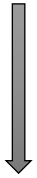


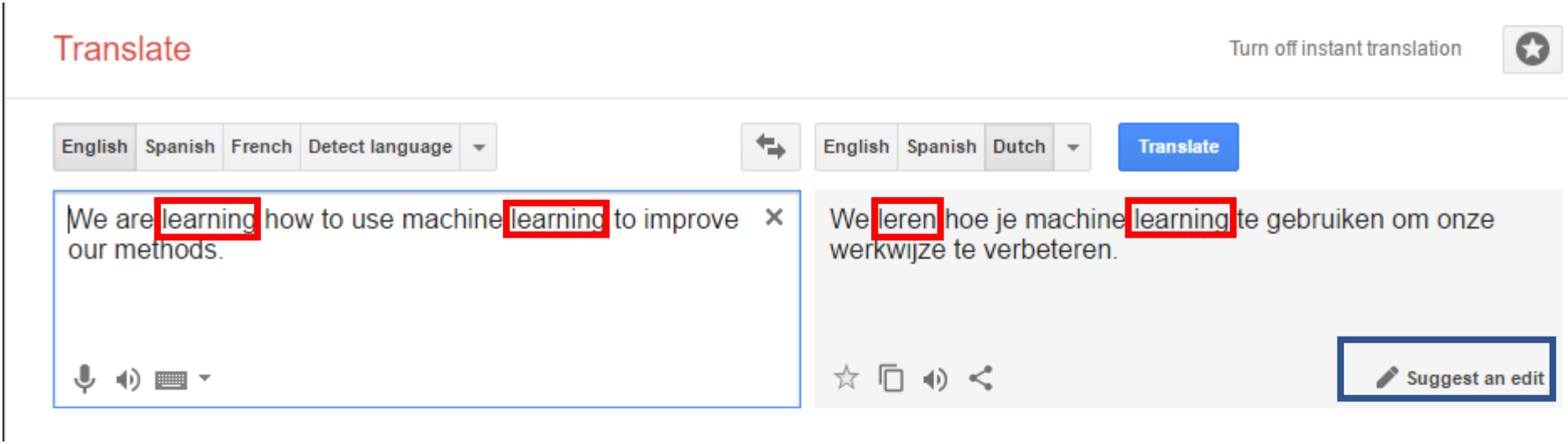
image pixels



detect faces
recognize faces
detect orientation
A is talking B

Slide credit: Carlos Guestrin, David Sontag

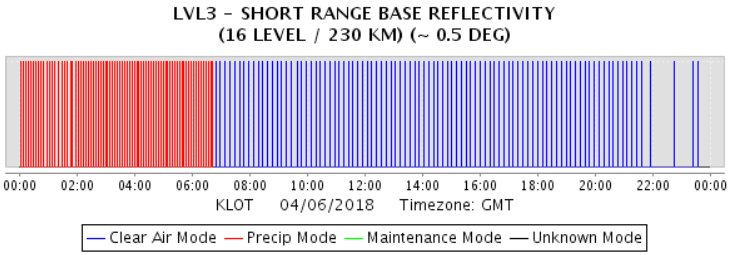
Machine translation

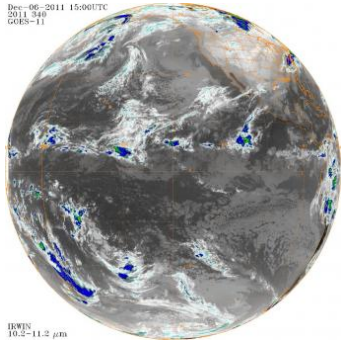


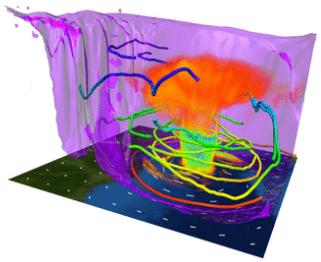
- Complex output
- Learn from: - annotated translations
  - matching text
  - text (in a single language)
  - corrections

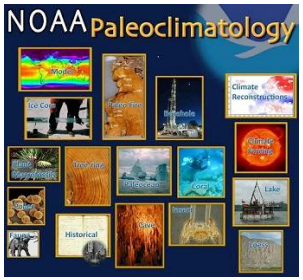
Slide and example credit: Nati Srebro

Weather forecasting







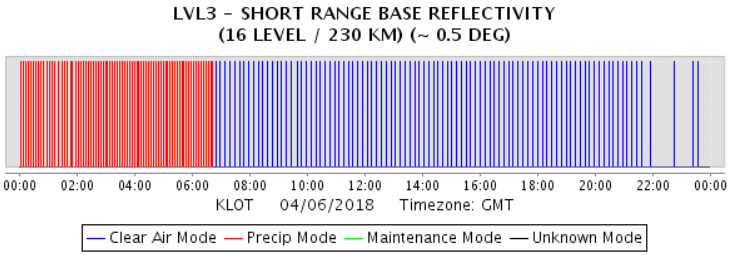


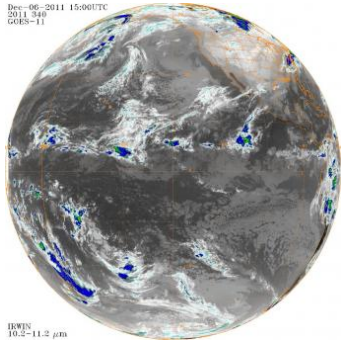
Measurements from land observatories, weather balloons, radar equipment, meteorological satellites, marine observatories, simulations, paleoclimatology data, and the historical records of these measurements

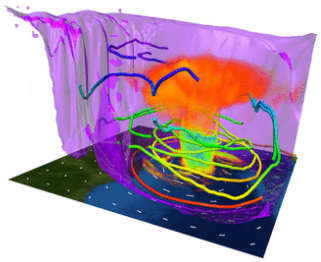
Temperature

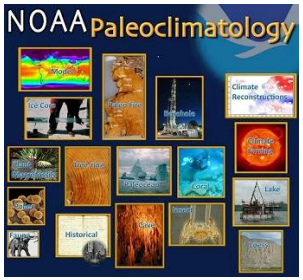
72° F

Weather forecasting



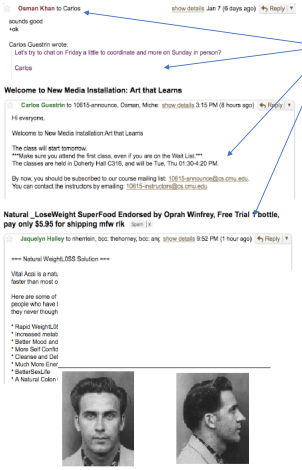






Still largely done by expert build systems

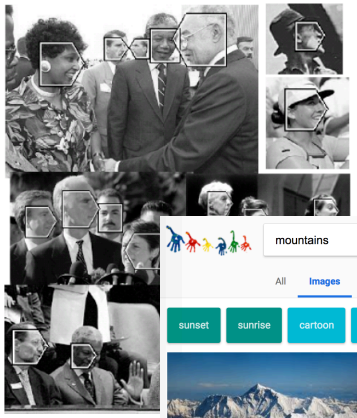
Predict output y from input x using examples of (x, y)



sender info, recipients info, length of text/subject, timestamp, subject line, words/pairs of words in text, previous interactions with senders/fellow recipients

↓

spam or no-spam

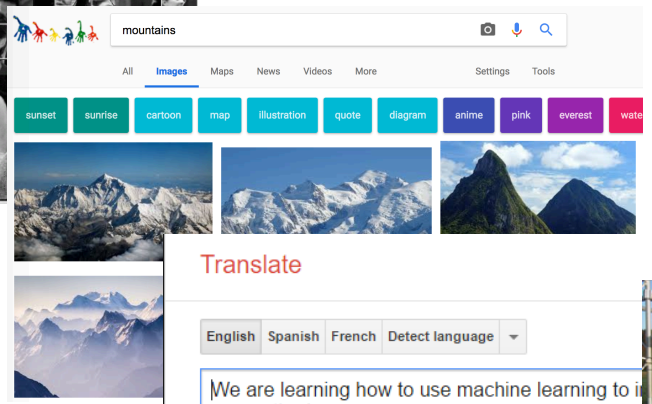


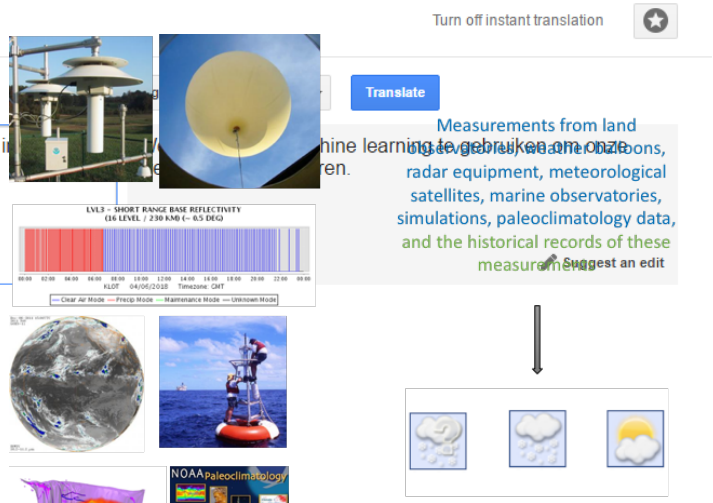
Example training images for each orientation

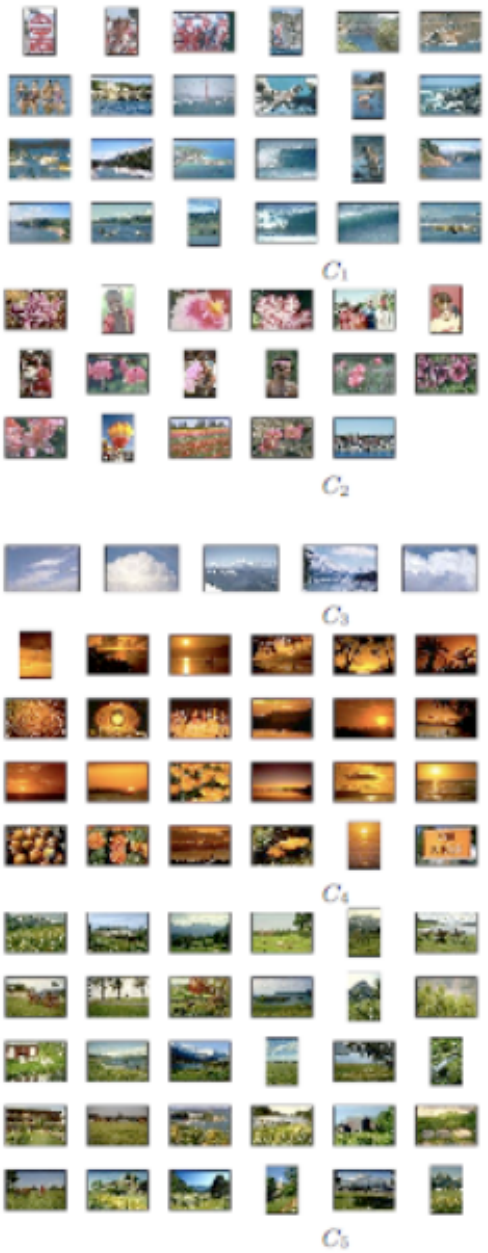
image pixels

↓

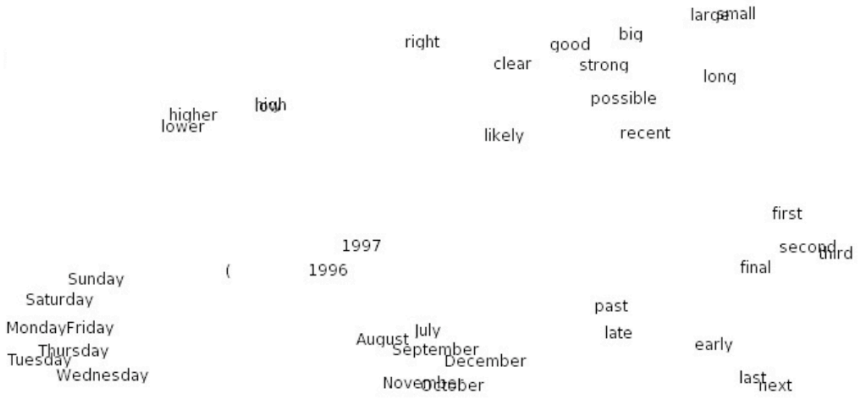
detect/recognize faces
detect orientation



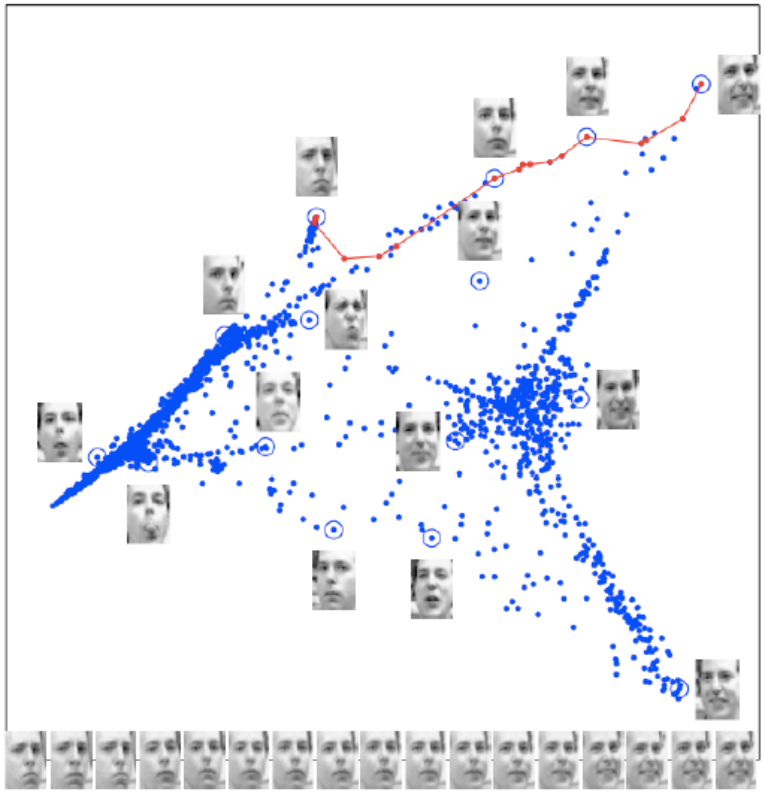




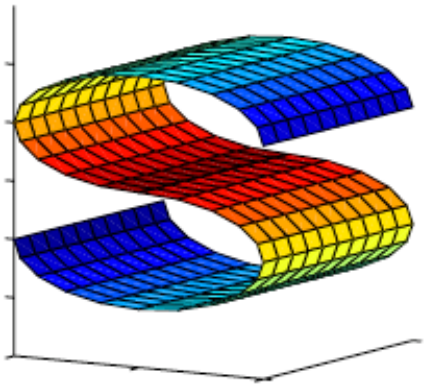
[Goldberger et al.]



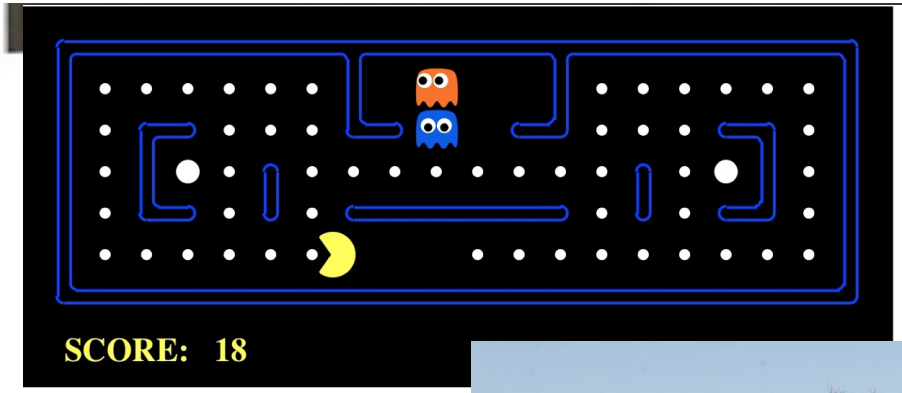
t-SNE visualization from Turian et al. (2010)

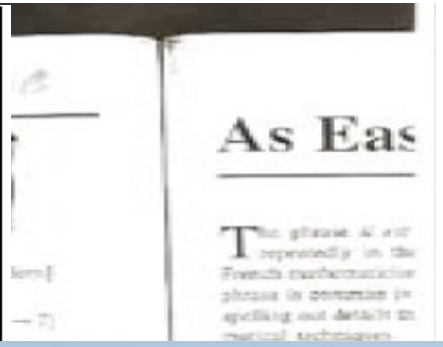


[Saul & Roweis '03]



Slide credit: David Sontag





Slide and examples credit: Luke Zettlemoyer

So what is machine learning?

What is (machine) learning?

- T. Hastie, R. Tibshirani, J. Friedman: “extract important patterns and trends, and understand ‘what the data says’”
- T. Mitchell: “The field of Machine Learning seeks to answer the question “How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?””
- S. Shalev-Shwartz and S. Ben-David: “automated detection of meaningful patterns in the data”
- K. Murphy: “set of methods that can automatically detect patterns in data, and then to use the uncovered patterns to predict future data or other kinds of decision making under uncertainty”
- M. Mohri, A. Rostamizadeh, and A. Talwalkar “computational methods using experience to improve performance or to make accurate predictions”

Slide credit: Greg Shaknarvich

What is (machine) learning?

- T. Hastie, R. Tibshirani, J. Friedman: "extract important patterns and trends, and understand 'what the **data** says'"

- T. Mitchell: "The field of Machine Learning seeks to answer the question "How can we build computer systems that automatically improve with **experience**, and what are the fundamental laws that govern all learning processes?""

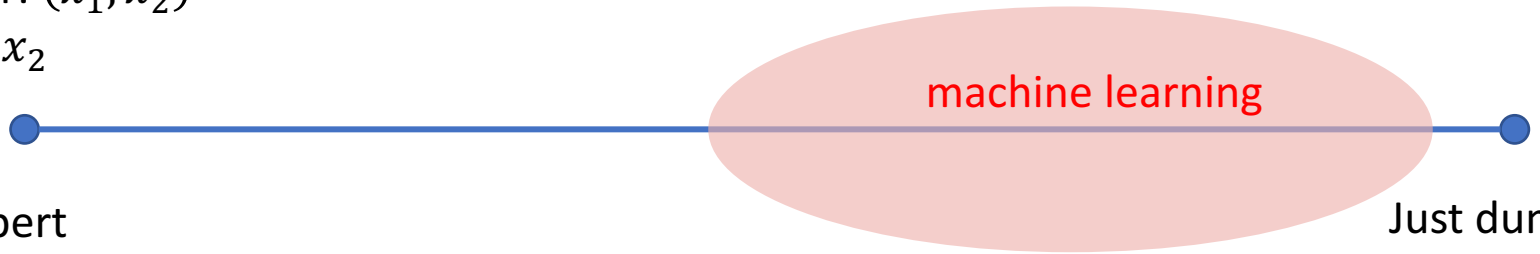
- S. Shalev-Shwartz and S. Ben-David: "automated detection of meaningful patterns in the **data**"

- K. Murphy: "set of methods that can automatically detect patterns in **data**, and then to use the uncovered patterns to predict future data or other kinds of decision making under uncertainty"

- M. Mohri, A. Rostamizadeh, and A. Talwalkar "computational methods using **experience** to improve performance or to make accurate predictions"

Expert designed → data driven

Adder: $(x_1, x_2) \rightarrow x_1 + x_2$



Expert designed systems

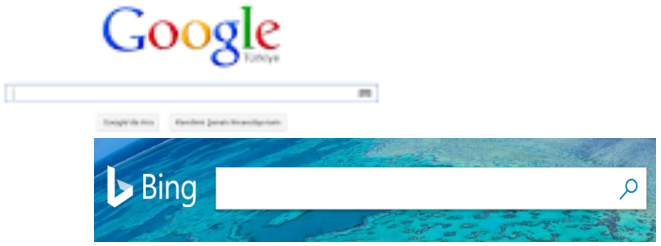
Just dump all data into the machine

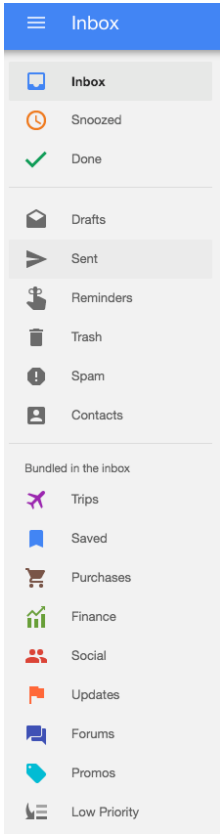
C. M. Bishop: "…a training set is used to tune the parameters of an adaptive model"

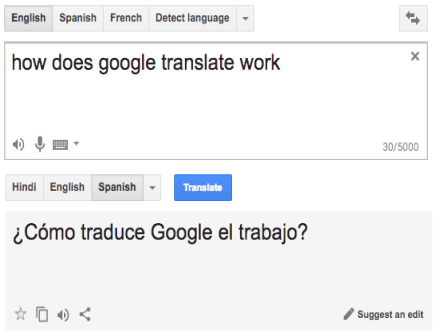
Machine learning everywhere

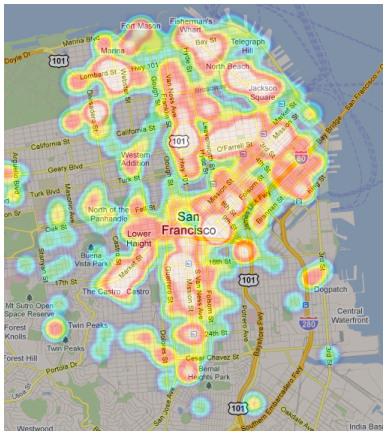
- **Email categorization,**
- **Web search**
- **Machine translation**
- **Question answering, dialog systems**
- **Automatic speech recognition, speech synthesis**
- **Surveillance**
- **Recommendations**
- **Fraud protection**
- **Traffic routing**

……









Why machine learning?

Why automation?

- Save human effort
  - automate boring tasks: surveillance, scanning license plates, driving!
  - automate dangerous tasks: robots for rescue operations, driving!
  - automate tasks requiring expert skills: translation
- Machines may perform tasks better than (expert) humans
  - detecting anomalies in radiology measurements
  - more accurate and consistent sensors!

Why learning?

- Machines can operate at scale and adapt
  - personalizing medicine, recommendations, etc.
  - learning based systems can adapt to new data
- Some systems are simply too complex to design manually
  - autonomous driving
  - weather prediction
- We have a lot of data and lot of computation
- Automate programming/automation

Data driven fields

Statistics

How long ago did cats and dogs diverge?
- About 55 MY, with 95% confidence interval [51,60]

traditionally more focused on explaining and summarizing data rather than prediction
- less computational considerations

Data mining

$$(Rotation\ time)^2 \propto (avg\ radius)^3$$

detect previously unknown patterns/knowledge discovery

Machine learning

Ultra deep network

Hello Karl!

perform tasks and improve at performing tasks by processing data
- "glorified statistics" R. Tibshirani

Slide credit: Nati Srebro

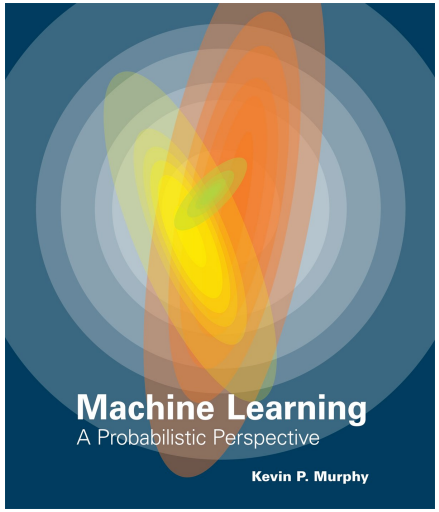
Goals of the course

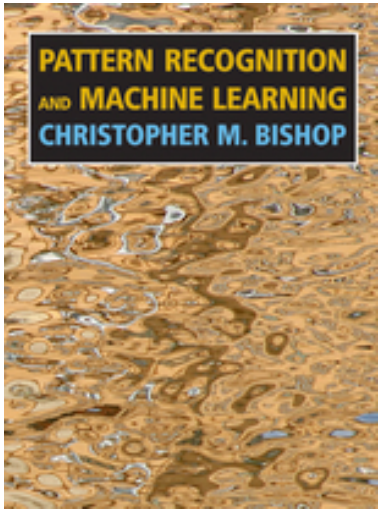
- Understand basic models in machine learning, mainly supervised learning
- Familiarize with mathematical tools to formalize machine learning problems
- Gain hands-on experience in systematically implementing, diagnosing, and testing machine learning models
- (Somewhat) understand how/when/why machine learning works

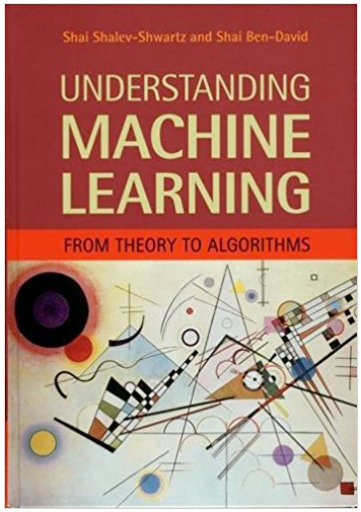
Course overview

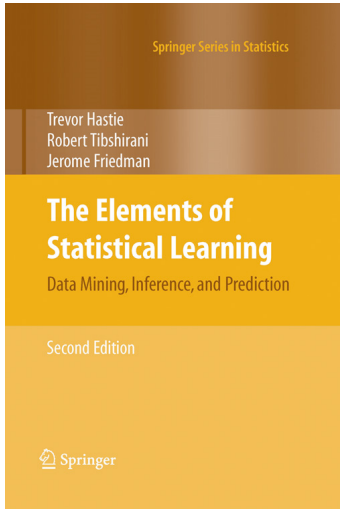
- Week 1
  - Basic models for supervised learning
  - Regression: linear regression
  - Classification: logistic regression, maximum margin classifiers, generative models for classification, structured classification
- Week 2
  - Introduction to neural networks, design and algorithmic choices in neural networks
  - Unsupervised learning

Resources


Machine Learning
A Probabilistic Perspective
Kevin P. Murphy


PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP


Shai Shalev-Shwartz and Shai Ben-David
UNDERSTANDING
MACHINE
LEARNING
FROM THEORY TO ALGORITHMS


Springer Series in Statistics
Trevor Hastie
Robert Tibshirani
Jerome Friedman
The Elements of
Statistical Learning
Data Mining, Inference, and Prediction
Second Edition
Springer

Supervised learning

learn a function that maps input $\boldsymbol{x}$ to output y
from examples of $(\boldsymbol{x}, y)$

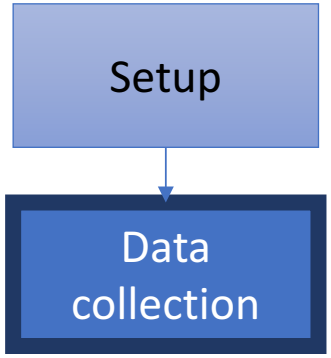
Supervised learning: problem setup

Setup

- Input $x \in \mathcal{X}$
  - Example 1. meteorological measurements – $\mathcal{X}$ numeric data in $\mathbb{R}^d$ + text_reports
  - Example 2. RGB pixel values of an image $\mathcal{X} = \mathbb{R}^{h \times w \times 3}$
- Output/label/target $y \in \mathcal{Y}$
  - Example 1. average temperature for tomorrow $\mathcal{Y} = \mathbb{R}$
  - Example 2. face or no-face $\mathcal{Y} = \{\text{face}, \text{noface}\}$ *or* $\mathcal{Y} = \{0,1\}$
- Goal: learn a function to predict y from x

$$f: \mathcal{X} \rightarrow \mathcal{Y}$$

Supervised learning: data collection



- Collect a set of labeled examples
  $S = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})\}$
  - Example 1. weather prediction -> historic data
  - Example 2. face detection -> ask humans to label images with faces
- Desirable properties

→ want examples to be accurate – minimize noise

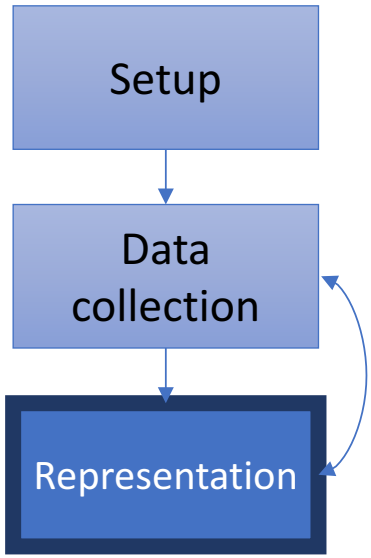
don't want ( , noface)

→ want examples to be diverse

don't want { ( , face), ( , face), ( , face) }

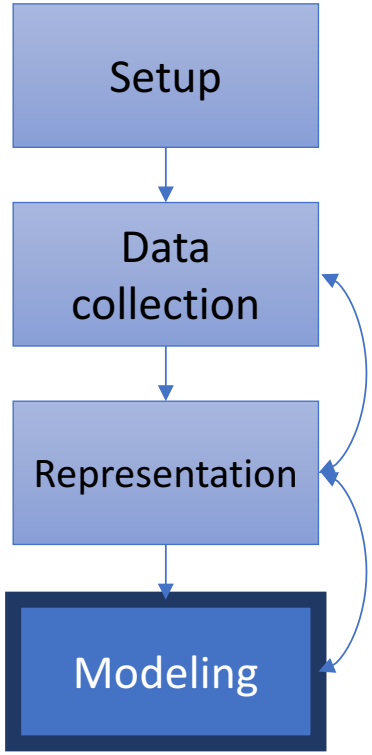
→ want many examples

Supervised learning: representation



- Preprocess features $x \leftarrow \text{preprocess}(x)$
  - e.g., convert non-numeric data to numbers
  - what units to use for various measurements? e.g., previous day temperature in $°C$ *or* $°F$? does (should) it matter?
  - depends on next step!!
- Extract more sophisticated features $x \leftarrow \text{advanced\_features}(x)$
  - e.g., pass raw pixels of an image through an edge detector
  - again depends on next step!!

Supervised learning: modeling

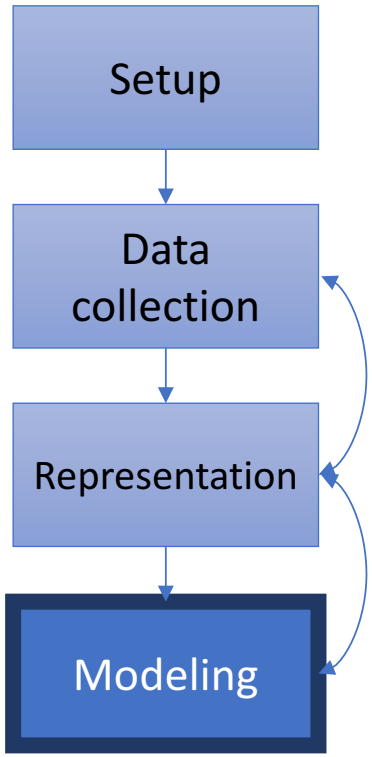


- Pick a set of candidates $\mathcal{H}$ for $f: \mathcal{X} \to \mathcal{Y}$?
- Parameterize f by $w \in \mathbb{R}^P$, $\mathcal{H} = \{f_w : w \in \mathbb{R}^P\}$ then somehow pick "correct" w or "correct" f_w (next step)
  - Example 1. $f(x)$ is a linear combination of numeric measurements in x

    $f_{\boldsymbol{w}}(x) = w_1 x_1 + w_2\ x_2 + \cdots w_d x_d = \boldsymbol{w}.\boldsymbol{x_{num}}$
  - Example 2. $f(x)$ is a linear combination of edge features

    $f_{\boldsymbol{w}}(x) = \boldsymbol{w}.\boldsymbol{x_{edge}} = \langle \boldsymbol{w}, \boldsymbol{x_{edge}} \rangle = \boldsymbol{w}^{\top}\boldsymbol{x_{edge}}$

Supervised learning: modeling

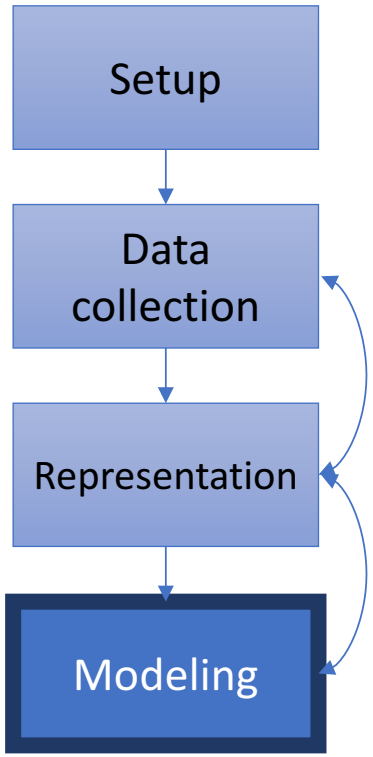


- Pick a set of candidates $\mathcal{H}$ for $f: \mathcal{X} \to \mathcal{Y}$?
- Parameterize f by $w \in \mathbb{R}^P$ $\mathcal{H} = \{f_w : w \in \mathbb{R}^P\}$
  then somehow pick "correct" w or "correct" f_w (next step)
  - Example 1. $f(x)$ is a linear combination of numeric measurements in x

    $$f_w(x) = w_1 x_1 + w_2\ x_2 + \cdots w_d x_d = \boldsymbol{w}.\boldsymbol{x_{num}}$$
  - Example 2. $f(x)$ is a linear combination of edge features

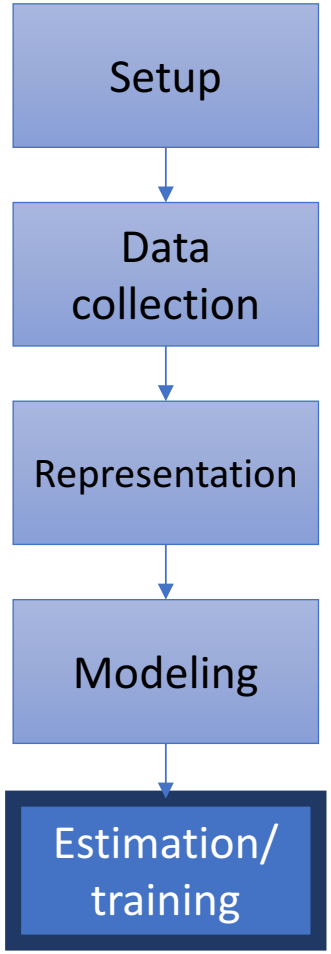
    $$f_w(x) = \boldsymbol{w}.\boldsymbol{x_{edge}} = \langle \boldsymbol{w}, \boldsymbol{x_{edge}} \rangle = \boldsymbol{w}^\top \boldsymbol{x_{edge}}$$
- Can also be non-parametric,
  - e.g., to compute $f(x)$ first find "closest" $x^{(i)}$ in training data and then return corresponding label $y^{(i)}$

Supervised learning: modeling



- Pick a set of candidates $\mathcal{H}$ for $f: \mathcal{X} \to \mathcal{Y}$?
- Parameterize f by $w \in \mathbb{R}^P$ $\mathcal{H} = \{f_w : w \in \mathbb{R}^P\}$
  then somehow pick "correct" w or "correct" f_w (next step)
  - Example 1. $f(x)$ is a linear combination of numeric measurements in x
    $$f_w(x) = w_1 x_1 + w_2\ x_2 + \cdots w_d x_d = \boldsymbol{w}.\boldsymbol{x_{num}}$$
  - Example 2. $f(x)$ is a linear combination of edge features
    $$f_w(x) = \boldsymbol{w}.\boldsymbol{x_{edge}} = \langle \boldsymbol{w}, \boldsymbol{x_{edge}} \rangle = \boldsymbol{w}^\top \boldsymbol{x_{edge}}$$
- Can also be non-parametric,
  - e.g., to compute $f(x)$ first find "closest" $x^{(i)}$ in training data and then return corresponding label $y^{(i)}$
- Some main considerations
  - Does $\mathcal{H} = \{f_w : \boldsymbol{w} \in \mathbb{R}^P\}$ capture what "correct" f is?
  - Do we have enough data to find the "correct" w? (next step)
  - What is the cost of estimating "correct" w? (next step)
  - Can $f(x) \notin \mathcal{Y}$? what if f returns -1 for no-faces instead of 0?

Supervised learning: training

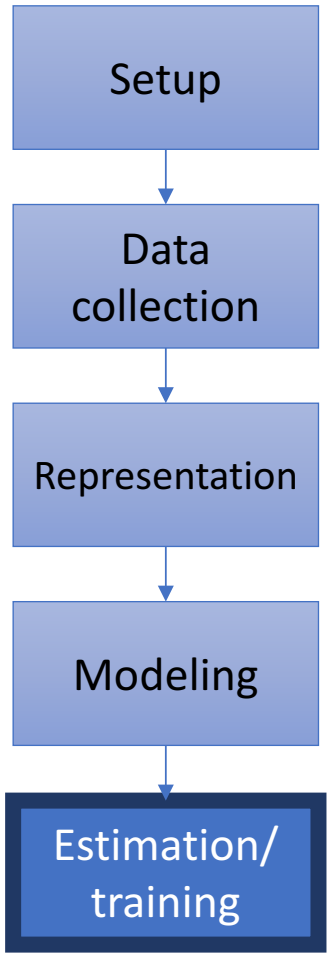


- Pick/estimate the "correct" $\hat{f}$ from $\mathcal{H}$
- Option 1: Find $\hat{f} \in \mathcal{H}$ such that for all training examples $\left(x^{(i)}, y^{(i)}\right) \in S$ we get correct prediction,

$$\hat{f}\left(x^{(i)}\right) = y^{(i)} \text{ for } i = 1,2, \ldots, N$$

What happens if there is no such $f \in \mathcal{H}$? or many such $f \in \mathcal{H}$?

Supervised learning: training



- Pick/estimate the "correct" $\hat{f}$ from $\mathcal{H}$
- Option 1: Find $\hat{f} \in \mathcal{H}$ such that for all training examples $\left(x^{(i)}, y^{(i)}\right) \in S$ we get correct prediction,

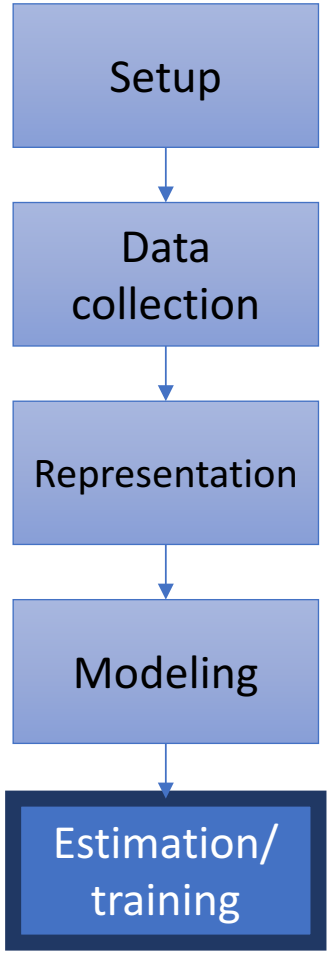
$$\hat{f}\left(x^{(i)}\right) = y^{(i)} \text{ for } i = 1,2, \ldots, N$$

  What happens if there is no such $f \in \mathcal{H}$? or many such $f \in \mathcal{H}$?

- Option 2: Find $\hat{f} \in \mathcal{H}$ that "fits/explains" most of data?
  - find $\hat{f}$ that minimizes some "loss" ℓ between prediction $f\left(x^{(i)}\right)$ and $y^{(i)}$

$$\hat{f} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{N} \ell\left(f\left(x^{(i)}\right), y^{(i)}\right) \text{ or } \widehat{w} = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{N} \ell\left(f_w\left(x^{(i)}\right), y^{(i)}\right)$$

Supervised learning: training



- Pick/estimate the "correct" $\hat{f}$ from $\mathcal{H}$
- Option 1: Find $\hat{f} \in \mathcal{H}$ such that for all training examples $\left(x^{(i)}, y^{(i)}\right) \in S$ we get correct prediction,

  $$\hat{f}\left(x^{(i)}\right) = y^{(i)} \text{ for } i = 1,2, \ldots, N$$

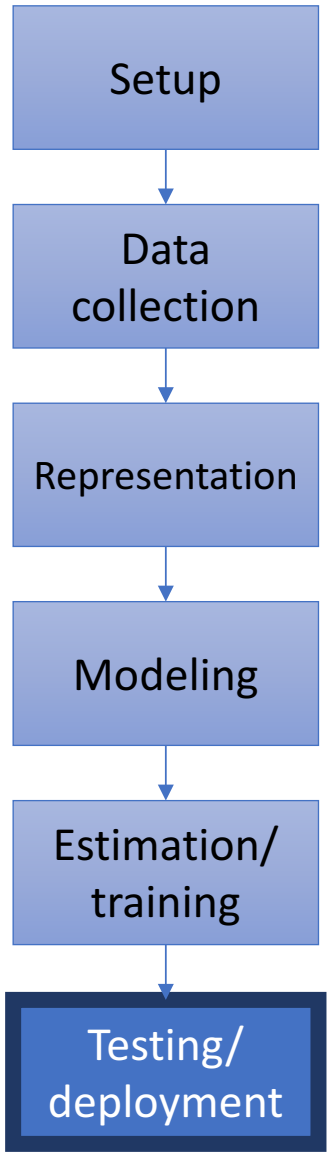
  What happens if there is no such $f \in \mathcal{H}$? or many such $f \in \mathcal{H}$?
- Option 2: Find $\hat{f} \in \mathcal{H}$ that "fits/explains" most of data?
  - find $\hat{f}$ that minimizes some "loss" ℓ between prediction $f(x_i)$ and y_i

    $$\hat{f} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{N} \ell\left(f\left(x^{(i)}\right), y^{(i)}\right) \text{ or } \hat{w} = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{N} \ell\left(f_w\left(x^{(i)}\right), y^{(i)}\right)$$

  - what loss ℓ to use?
  - how to solve the minimization? – computational considerations
  - Example 1: $\ell\left(f\left(x^{(i)}\right), y^{(i)}\right) = \left|f\left(x^{(i)}\right) - y^{(i)}\right|$ or $\left(f\left(x^{(i)}\right) - y^{(i)}\right)^2$
  - Example 2: $\ell\left(f\left(x^{(i)}\right), y^{(i)}\right) = \begin{cases} 0 \text{ if } f\left(x^{(i)}\right) = y^{(i)} \\ 1 \text{ if } f\left(x^{(i)}\right) \neq y^{(i)} \end{cases}$

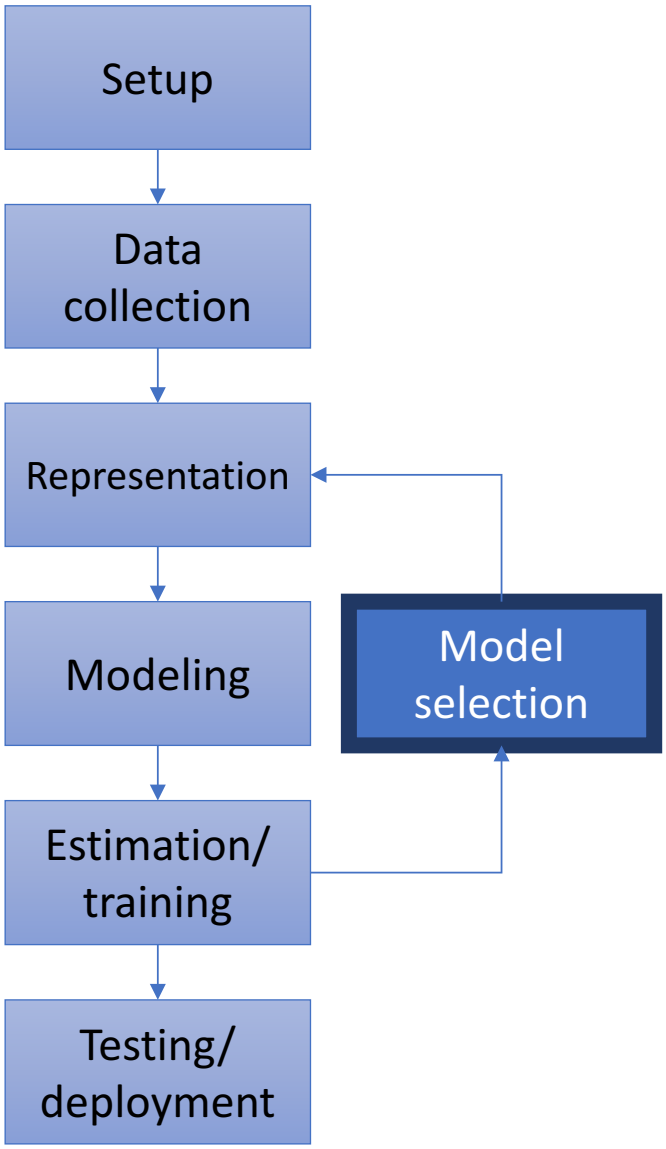
→ Empirical Risk Minimization (ERM)

Supervised learning: testing



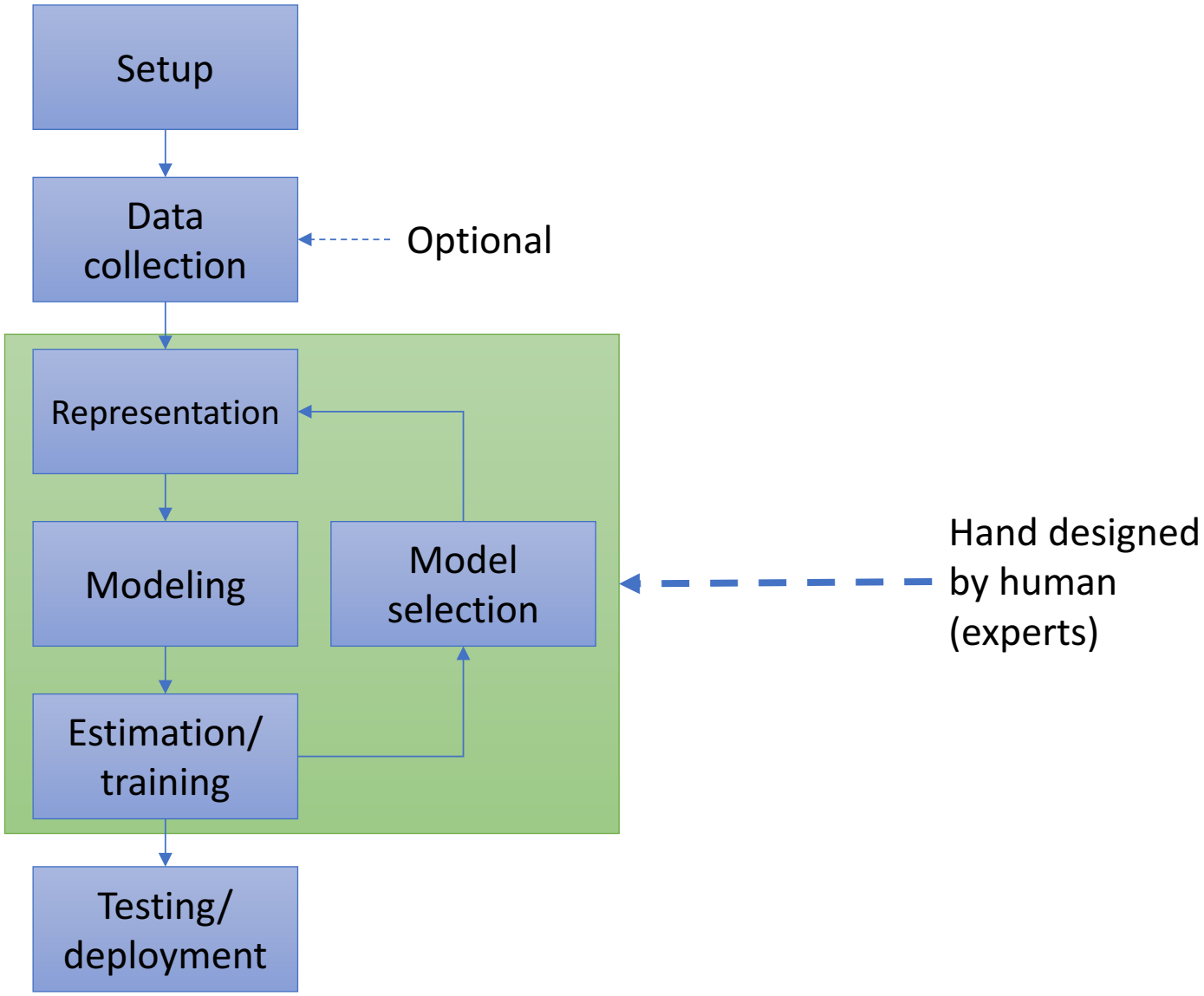
- Test how well $\hat{f}$ performs the task
- Can we check $\sum_i \ell\left(\hat{f}\left(x^{(i)}\right), y^{(i)}\right)$?
  - e.g., $\left\{\left(\text{  }, \text{face}\right), \left(\text{  }, \text{face}\right), \left(\text{  }, \text{noface}\right)\right\}$
    and $\hat{f}$ always outputs "noface" for any non-Karl image
- What we want is how well $\hat{f}$ performs on new examples!
- Deploy in real world and see how well $\hat{f}$ performs
- Collect more data S_{test} but do not use for training
- Split original data and do training on only S_{train}

Supervised learning: Model selection

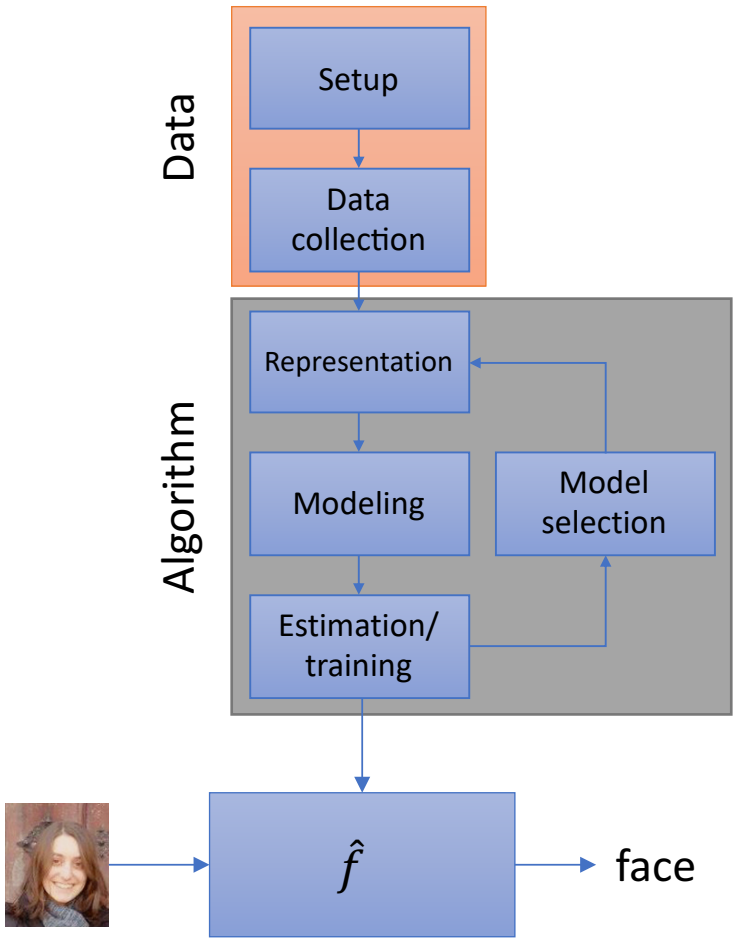


- What if $\hat{f}$ performs poorly on S_{test}?
  - Can we again use same S_{test}?
- Can we "detect" how well $\hat{f}$ will perform while testing?
- Mimic "testing" during model selection
  - Keep a small number of training example to mimic the testing process – validation or development data
  - Split S into S_{train}, S_{val} and S_{test}
  - Train on S_{train}, get mock-test score by checking on S_{val}
  - Change models and repeat
  - DO NOT use S_{test}

Rule based learning



Supervised learning – key questions



- Data: what kind of data can we get? how much data can we get?
- Model: what is the correct model for my data? – want to minimize the effort put into this question!
- Training: what resources - computation/memory - does the algorithm need to estimate the model $\hat{f}$?
- Testing: how well will $\hat{f}$ perform when deployed? what is the computational/memory requirement during deployment?

Other learning paradigms

- Unsupervised learning (will cover some)
  - Only examples of x are seen but no y
  - There might not be any specific y
  - Goal is to make sense of the data
- Hybrid learning
  - semi-supervised: small number of (x, y) examples are available, but lot more examples of just x
  - weakly supervised: $(x, \tilde{y})$ examples are available where $\tilde{y}$ is a very noisy estimate of y, or has partial information, e.g., fruit instead of apple
- Reinforcement learning
  - Learning to plan and execute long term actions
  - e.g., navigate a maze, win a soccer match