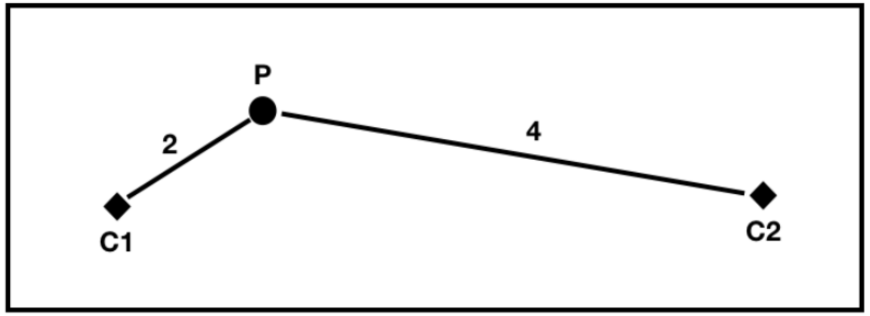
Exercise 1

Exercise 1

Given is a point cloud with the two cluster centers $C1$ *and* $C2$*. For a clearer representation, only a single data point* P *is mapped.*



Furthermore, the two distances $d(P, C1) = 2$ *and* $d(P, C2) = 4$ *as well as the to be minimized objective function are given:*

$$J(X, B, U) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{w} d^2(\overrightarrow{\beta_i}, \overrightarrow{x_j})$$

Exercise 1

X is the set of data points, B is the set of cluster prototypes, and U is a fuzzy partition matrix. As fuzzifier $w = 2$ was chosen.

In each of the following cases, calculate the resulting value of the objective function J when, for the given data point P, the following degrees of membership have been calculated:

(i) $\overrightarrow{U_1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

(ii) $\overrightarrow{U_2} = \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix}$

(iii) $\overrightarrow{U_3} = \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}$

Exercise 1

Recap:

cluster centers

data points

cluster prototype

$$J(X, B, U) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{w} d^2(\overrightarrow{\beta_i}, \overrightarrow{x_j})$$

distance function

Exercise 1

$$J(X,B,U) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{w} d^2(\overrightarrow{\beta_i}, \overrightarrow{x_j})$$

(i) $\overrightarrow{U_1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$$d(P, C_1) = 2$$

$$d(P, C_2) = 4$$

$$J = 1^2 * 2^2 + 0^2 * 4^2 = 4$$

Exercise 1

$$J(X,B,U) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{w} d^2(\overrightarrow{\beta_i}, \overrightarrow{x_j})$$

(ii) $\overrightarrow{U_2} = \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix}$

$$d(P, C_1) = 2$$

$$d(P, C_2) = 4$$

$$J = 0.9^2 * 2^2 + 0.1^2 * 4^2 = 3.4$$

Exercise 1

$$J(X,B,U)=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^{w}d^{2}(\overrightarrow{\beta_i},\overrightarrow{x_j})$$

(iii) $\overrightarrow{U_3}=\begin{pmatrix}0.8\\0.2\end{pmatrix}$

$$d(P,C_1)=2$$

$$d(P,C_2)=4$$

$$J=0.8^2*2^2+0.2^2*4^2=3.2$$

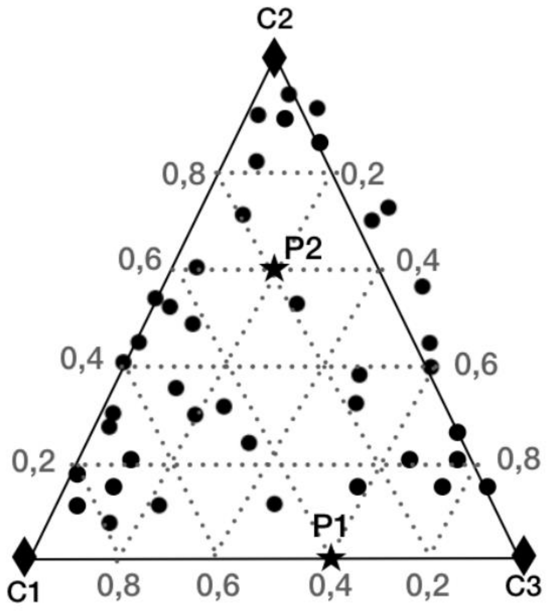
Exercise 1

Which of the given degrees of membership from a) is to be preferred for the minimization of the objective function?

$$\overrightarrow{U_3} = \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}$$

Exercise 1

c) Now let us assume that we have a data set represented in coefficient space with auxiliary lines drawn in for membership weighting as well as the three cluster centers C1, C2, C3 and the two marked points P1 and P2.

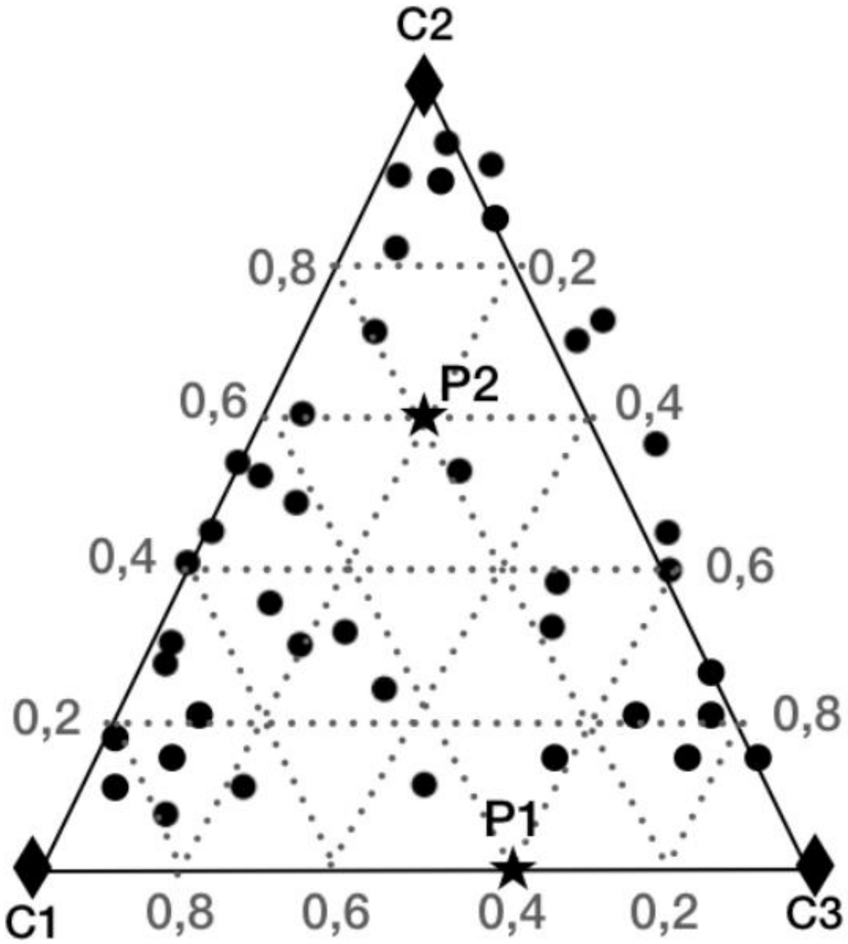


Exercise 1

Fill in the following table by determining the fuzzy affiliations of the two points P1 and P2 to the cluster centers C1, C2, C3.

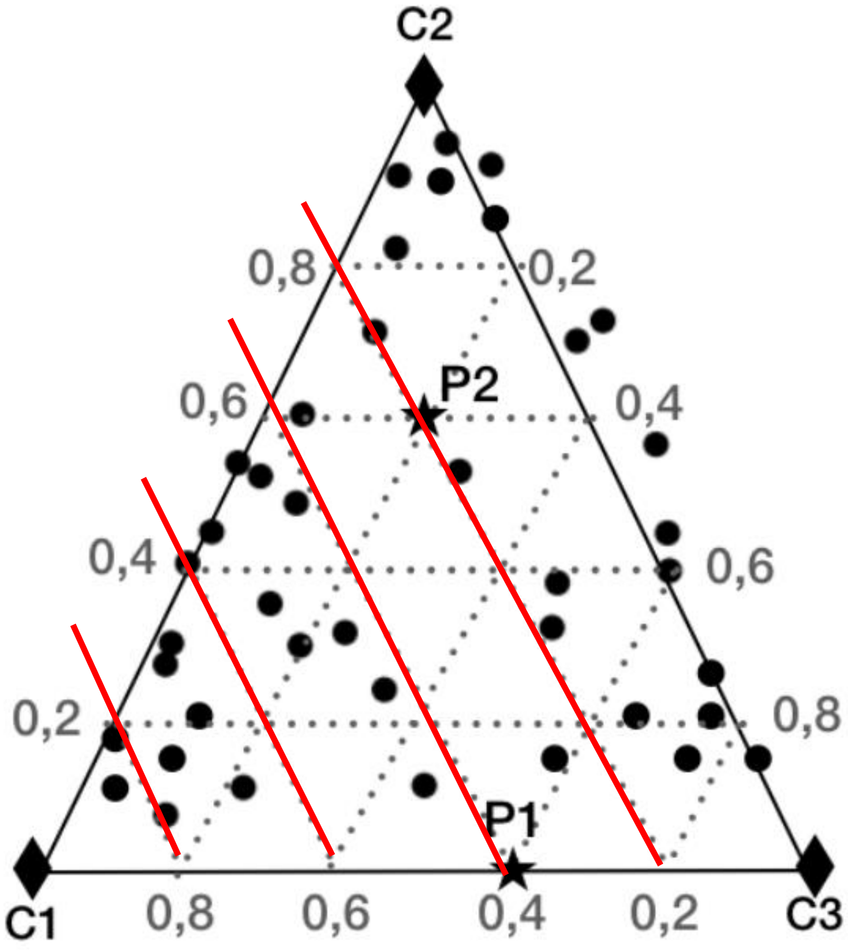
	C1	C2	C3
P1			
P2			

Exercise 1



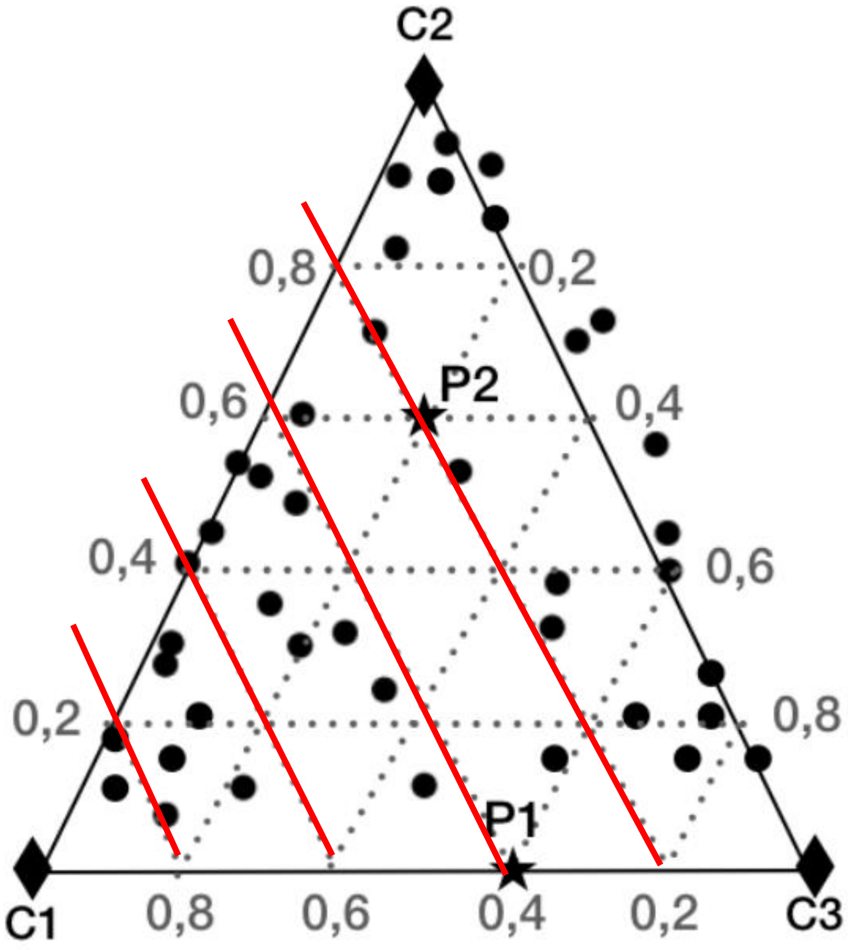
	C1	C2	C3
P1			
P2			

Exercise 1



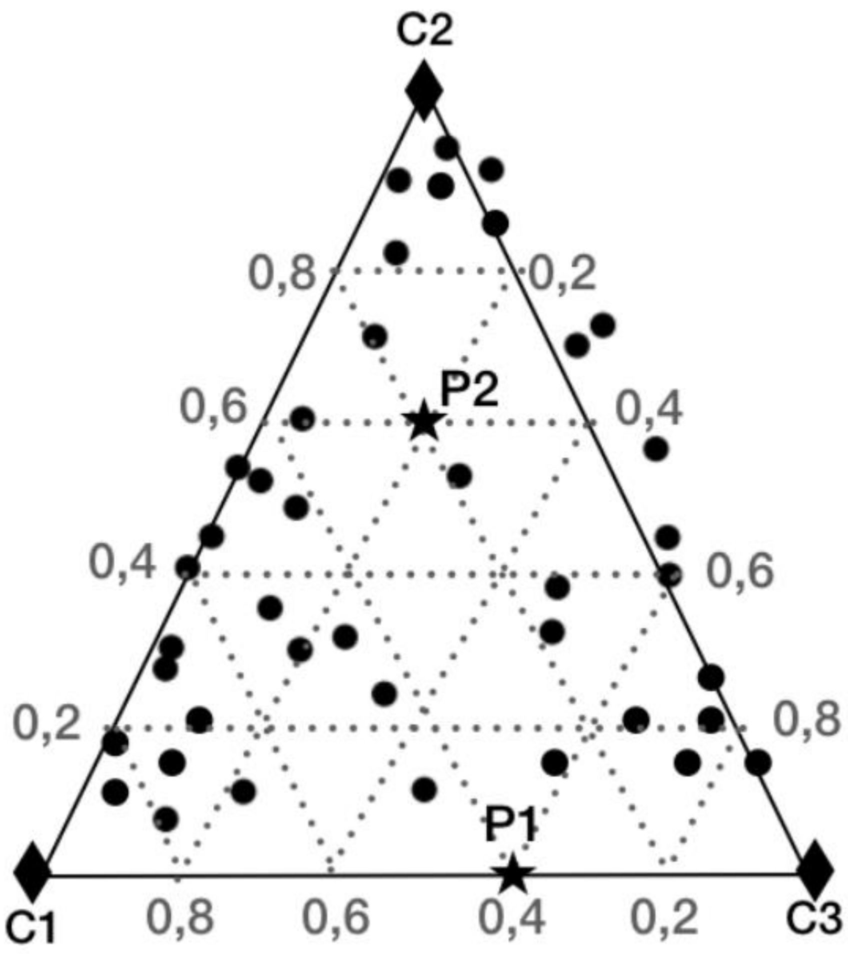
	C1	C2	C3
P1			
P2			

Exercise 1



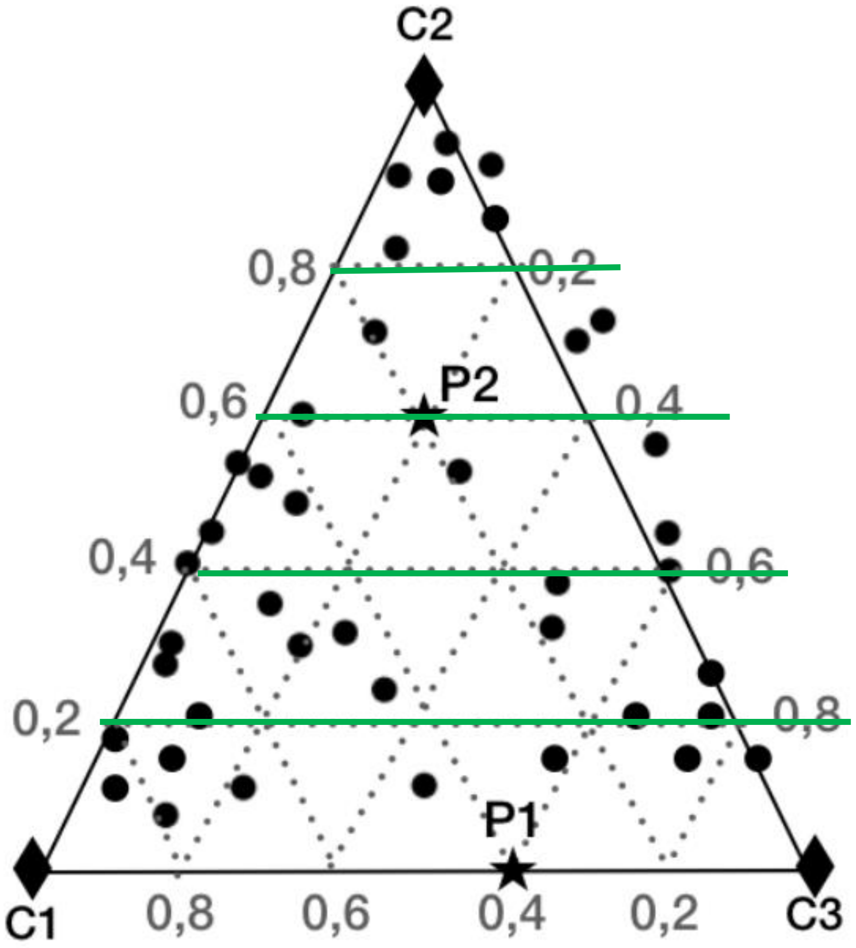
	C1	C2	C3
P1	0.4		
P2	0.2		

Exercise 1



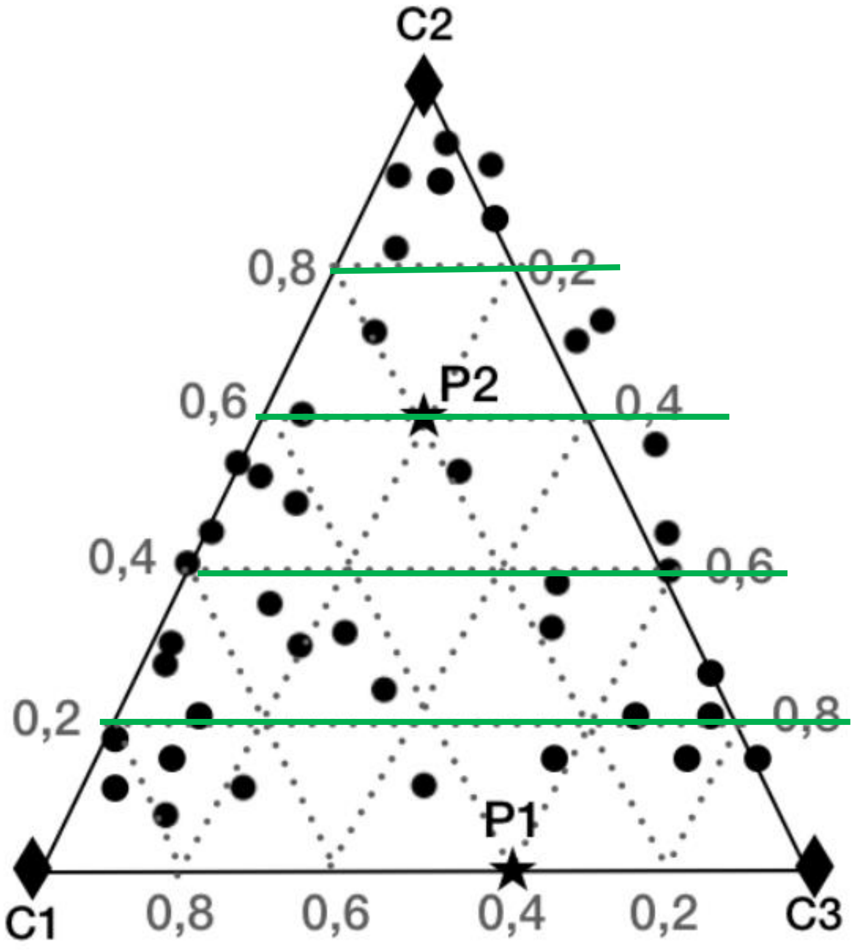
	C1	C2	C3
P1	0.4		
P2	0.2		

Exercise 1



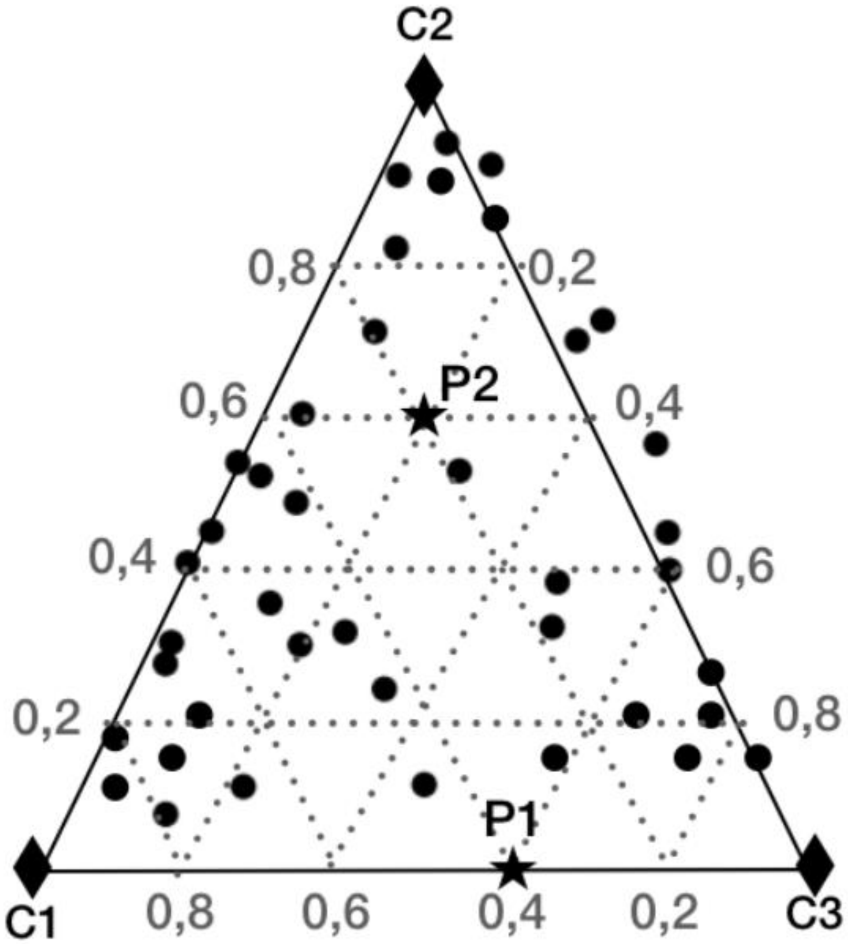
	C1	C2	C3
P1	0.4		
P2	0.2		

Exercise 1



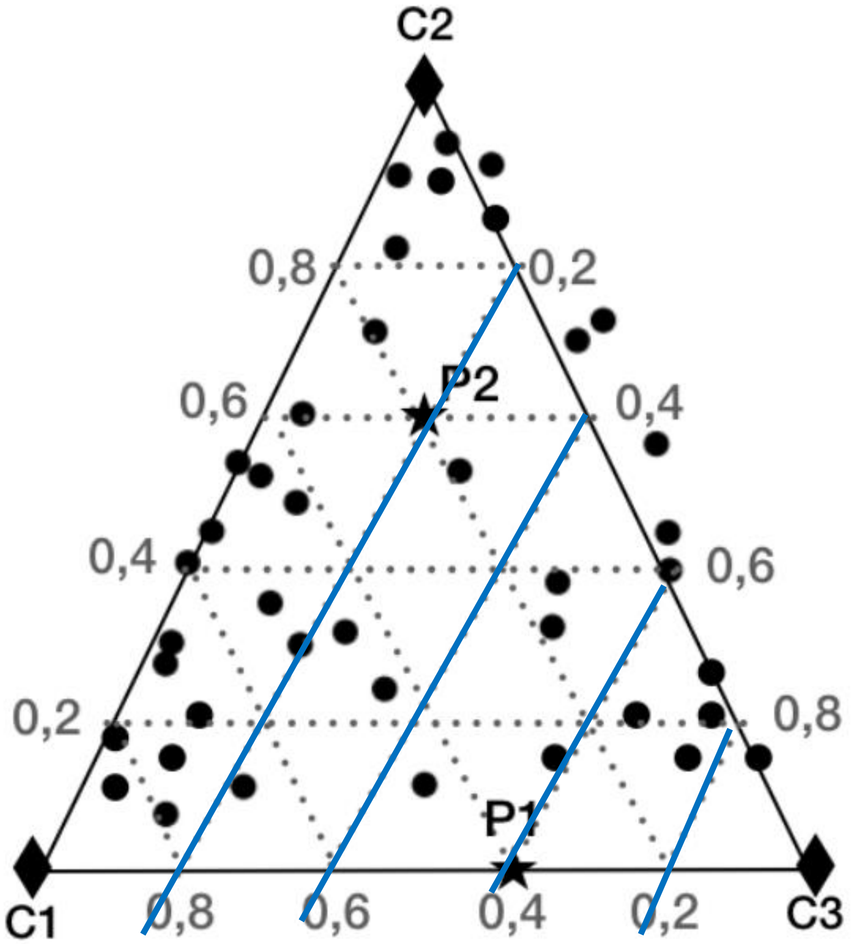
	C1	C2	C3
P1	0.4	0	
P2	0.2	0.6	

Exercise 1



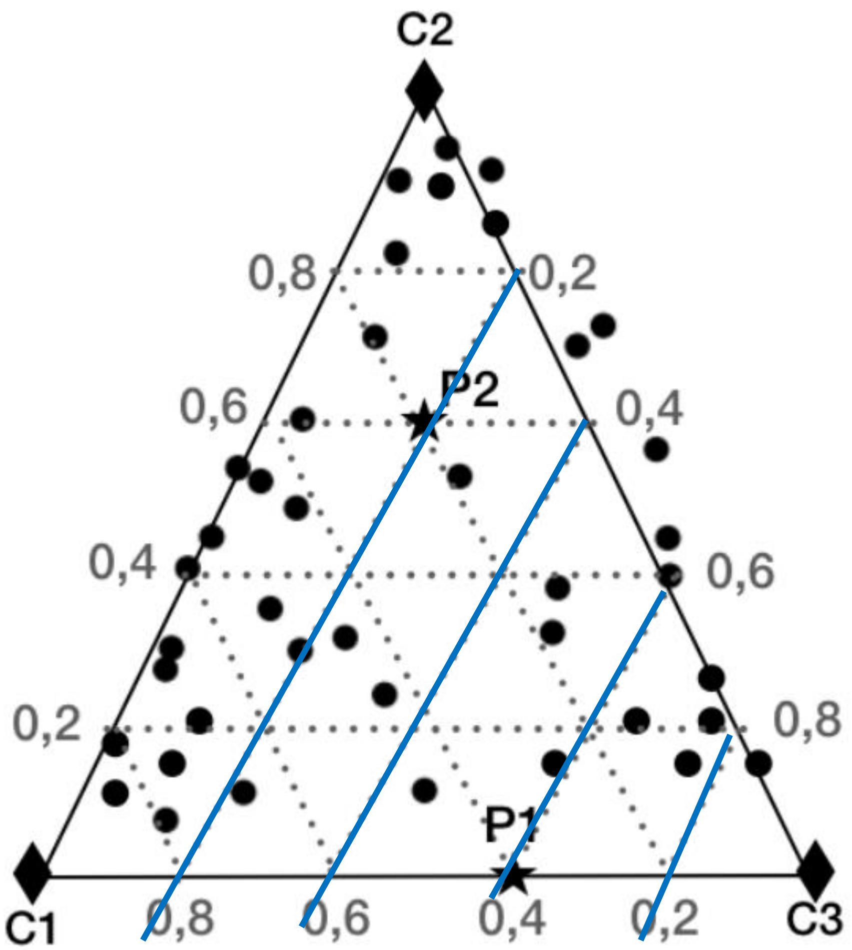
	C1	C2	C3
P1	0.4	0	
P2	0.2	0.6	

Exercise 1



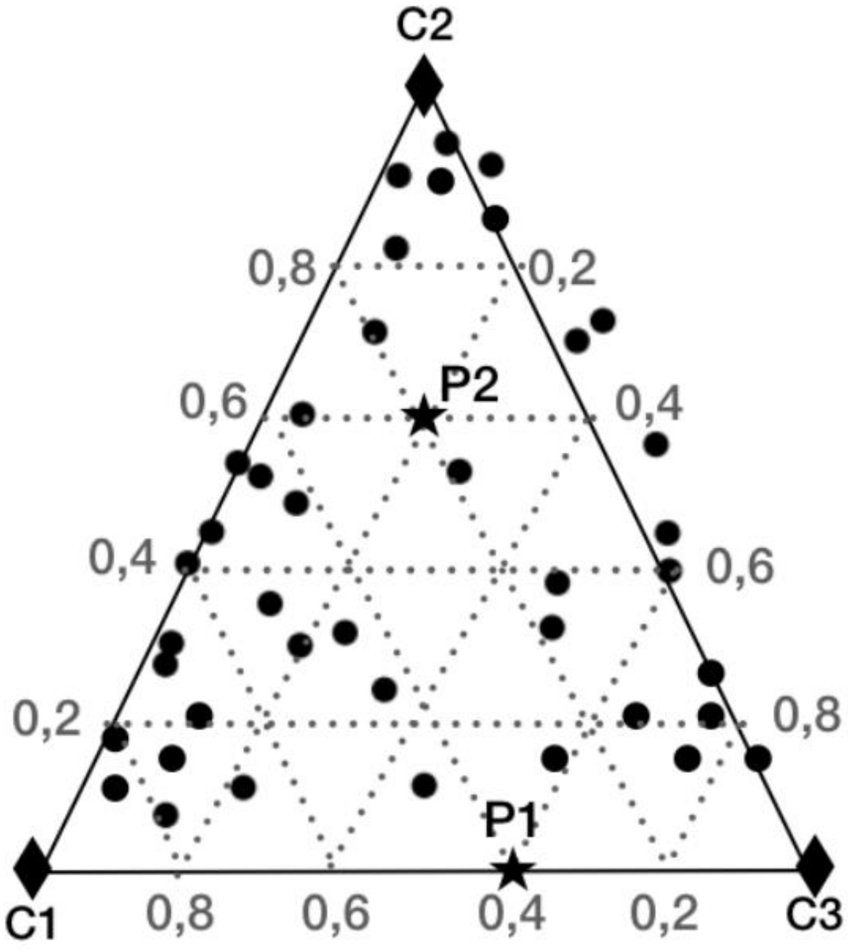
	C1	C2	C3
P1	0.4	0	
P2	0.2	0.6	

Exercise 1



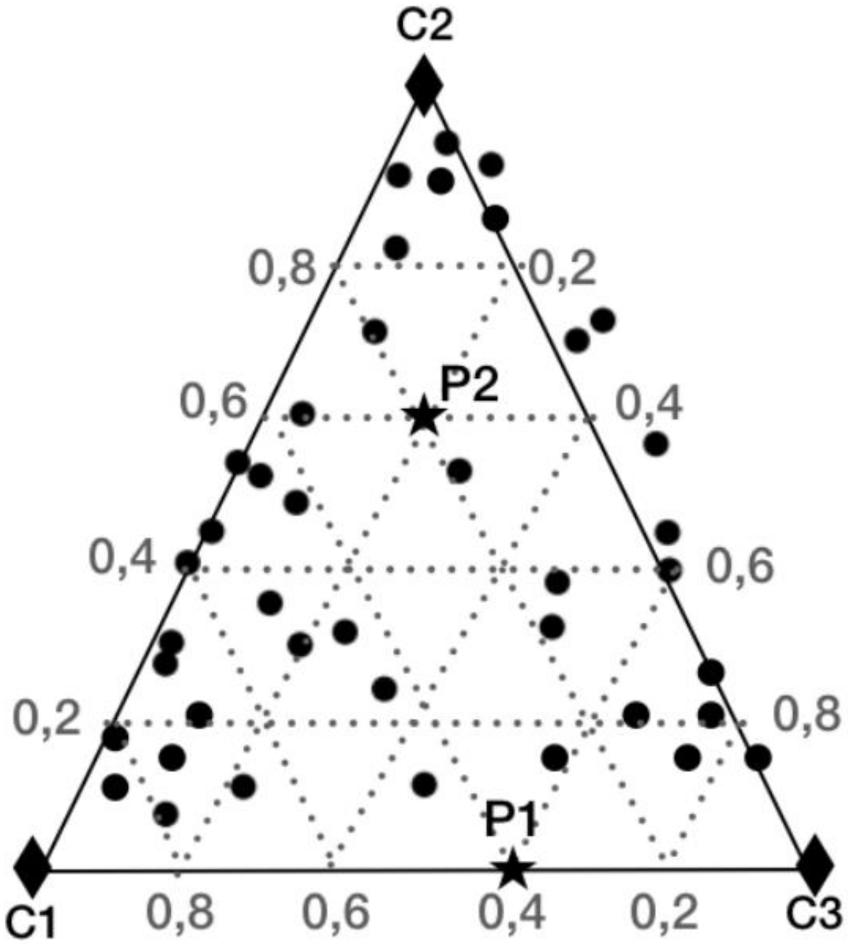
	C1	C2	C3
P1	0.4	0	0.6
P2	0.2	0.6	0.2

Exercise 1



	C1	C2	C3
P1	0.4	0	0.6
P2	0.2	0.6	0.2

Exercise 1



$$J(X, B, U) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{w} d^2(\vec{\beta_i}, \vec{x_j})$$

	C1	C2	C3
P1	0.4	0	0.6
P2	0.2	0.6	0.2

Exercise 2

Autoregressive vs. Masked Transformers

Autoregressive Transformer:

- Predict the next token conditioned on the previous tokens

This is a → sentence

- Unidirectional context
- Commonly used for: Text generation

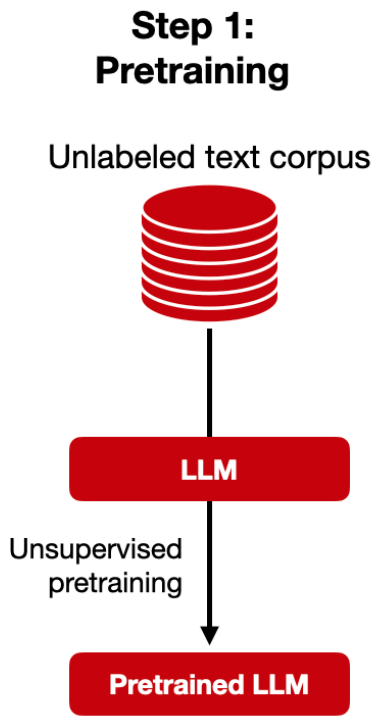
Masked Transformer:

- Randomly replace tokes with a mask:

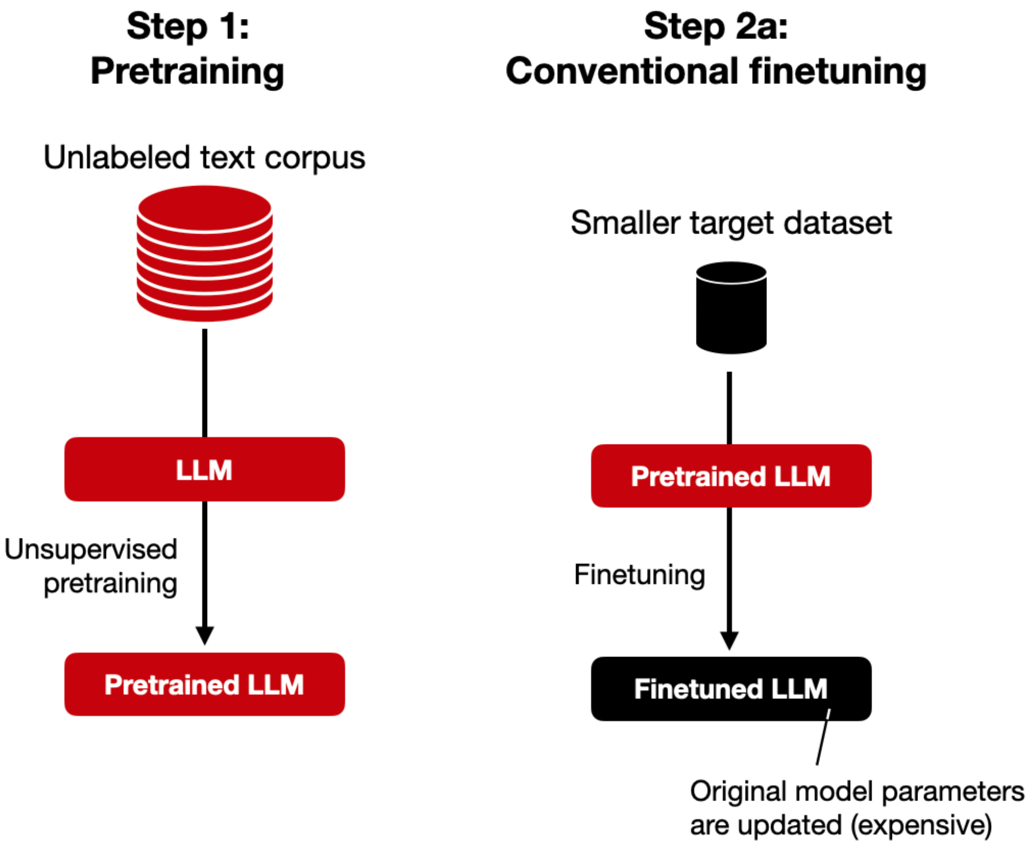
This is a sentence. → This is [MASK] sentence.

- Predict the masked tokens
- Bidirectional context
- BERT (Bidirectional Encoder Representations from Transformers)
- Commonly used for: Text Classification, Named Entity Recognition and Question Answering

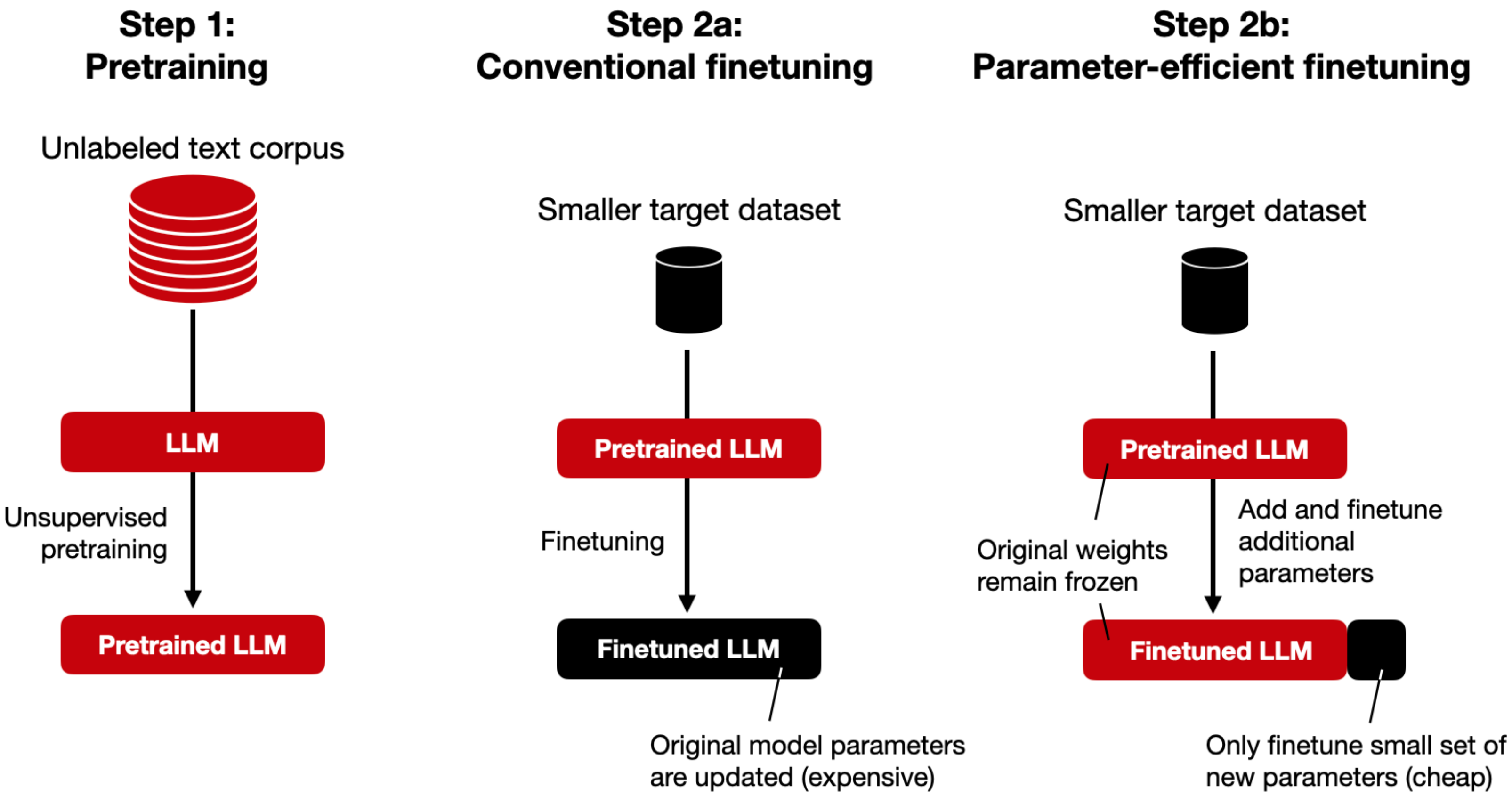
Finetuning


Step 1:
Pretraining
Unlabeled text corpus
LLM
Unsupervised
pretraining
Pretrained LLM

Finetuning


Step 1:
Pretraining
Unlabeled text corpus
LLM
Unsupervised
pretraining
Pretrained LLM
Step 2a:
Conventional finetuning
Smaller target dataset
Pretrained LLM
Finetuning
Finetuned LLM
Original model parameters
are updated (expensive)

Finetuning



Advantages:
- Faster training
- Fewer data required
- Easier distribution

Methods for Finetuning

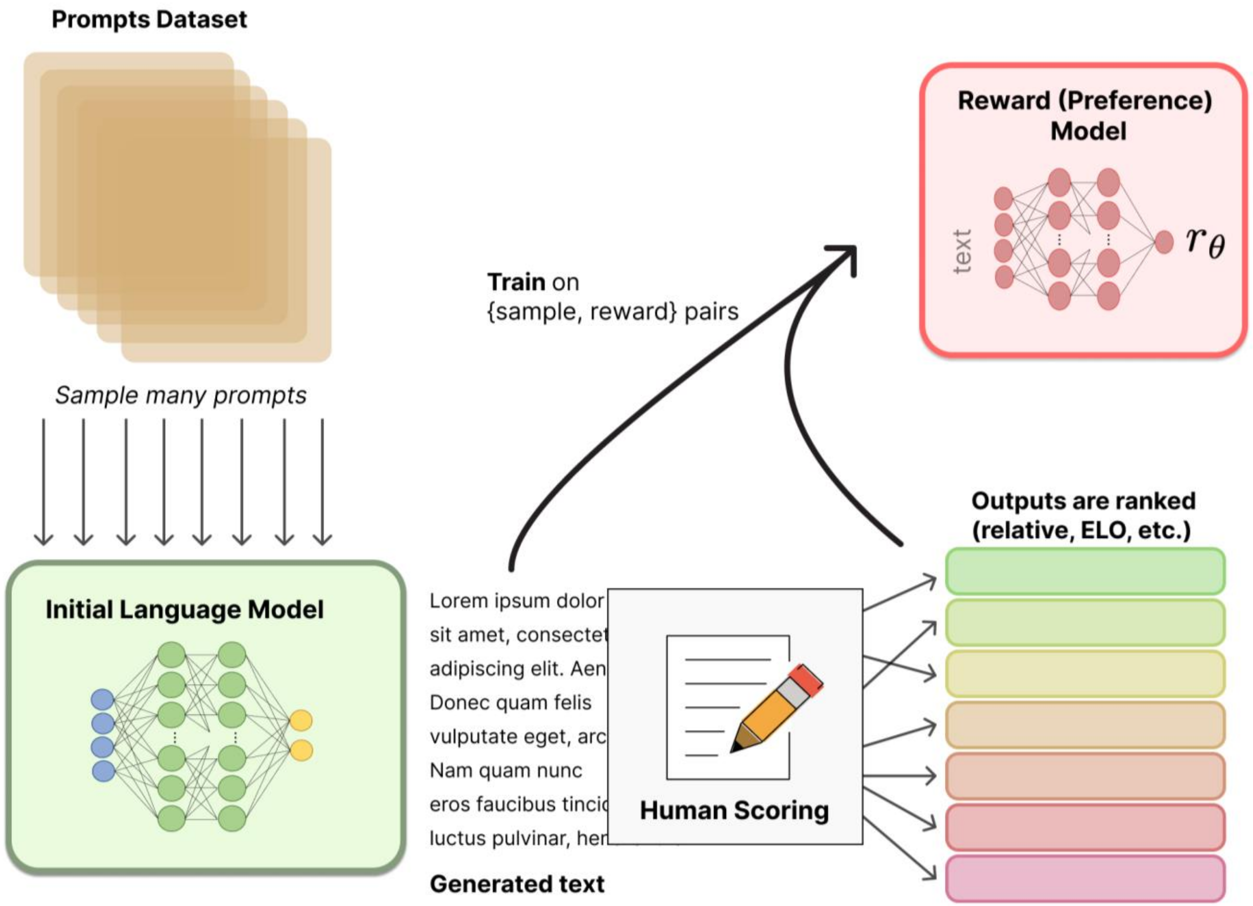
1. Supervised Finetuning
    - Used, when the answer is unique
    - Applications: teach classification, extraction and formatting
    - Examples:
        - Promt: What is the most famous song of Coldplay?
          Viva La Vida
        - Promt: A soft, flowing dress that falls in graceful folds in sky blue.
          {Category: dress, Color: blue}
        - Promt: I want to get a ride to Munich Airport tomorrow at 6:00am.
          book_ride(date=03.03.2024, time=0600, destination=MUC)
    - Problems: sometimes it is hard to acquire exact targets

Methods for Finetuning

2. Unsupervised Finetuning
   - Used, when we want the model the familiarize with special texts without answering any specific question
   - Applications: e.g. Learn documentation of a company

Methods for Finetuning

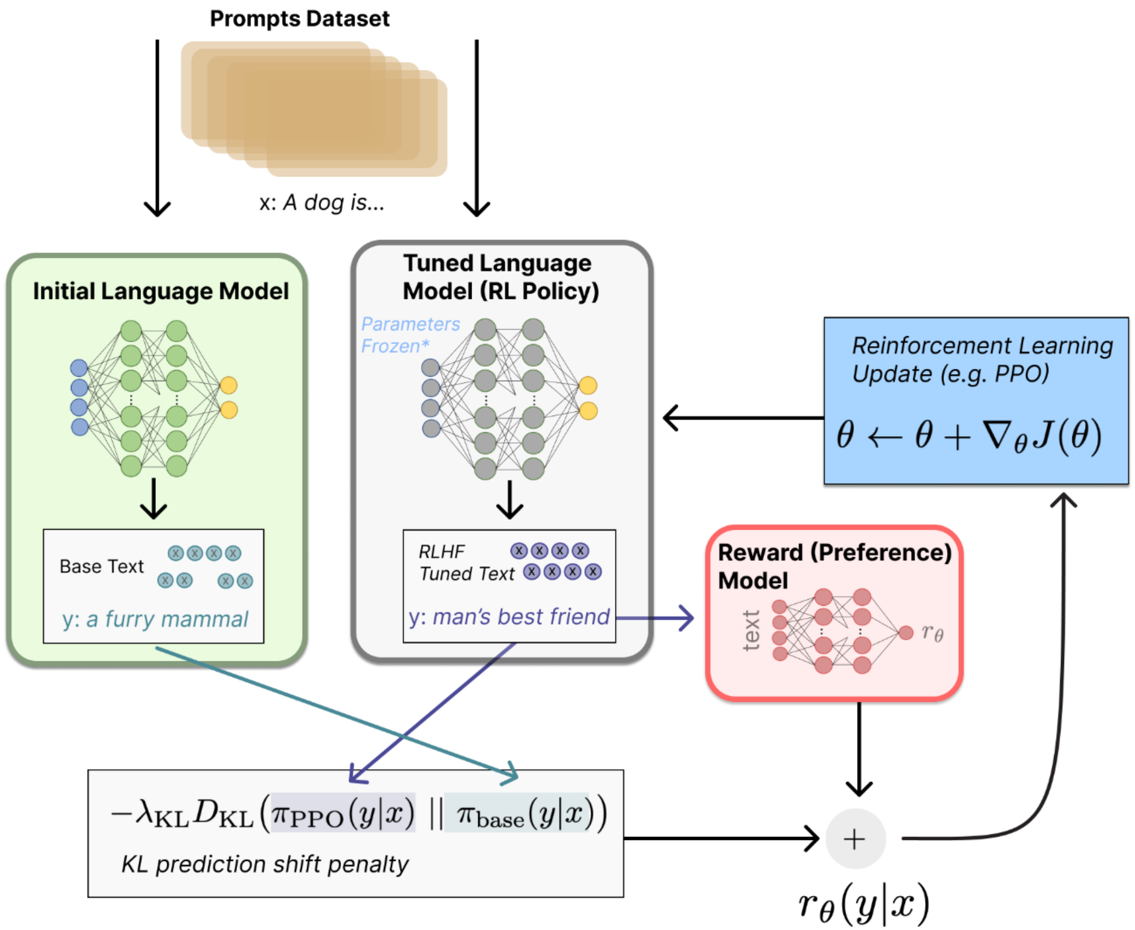
3. Reinforcement Learning from Human Feedback



Source: https://huggingface.co/blog/rlhf

Methods for Finetuning

3. Reinforcement Learning from Human Feedback



Source: https://huggingface.co/blog/rlhf

Exercise 2

Install the `requirements.txt`

We are using the packages:

- `torch`: machine learning library
- `transformers`: API to download pre-trained models from Huggingface
- `datasets`: API to download datasets from Huggingface

Hugging Face https://huggingface.co/models

Exercise 2

1. Download the *distilBERT* model from Hugging Face
2. Complete the sentence: `This is a great [MASK].`
3. It predicts something like:
    1. `This is a great day.`
    2. `This is a great person.`
    3. `This is a great house.`
4. Download the IMDb Film Review Dataset
5. Finetune the model on this dataset
6. Complete the sentence: `This is a great [MASK].`
7. It predicts something like:
    1. `This is a great movie.`
    2. `This is a great film.`
    3. `This is a great actor.`

Exercise 2

Complete the code in file `finetuning.py` to finetune the *distilBERT* model on a film review text corpus.

See PyCharm

Outlook: Research Directions

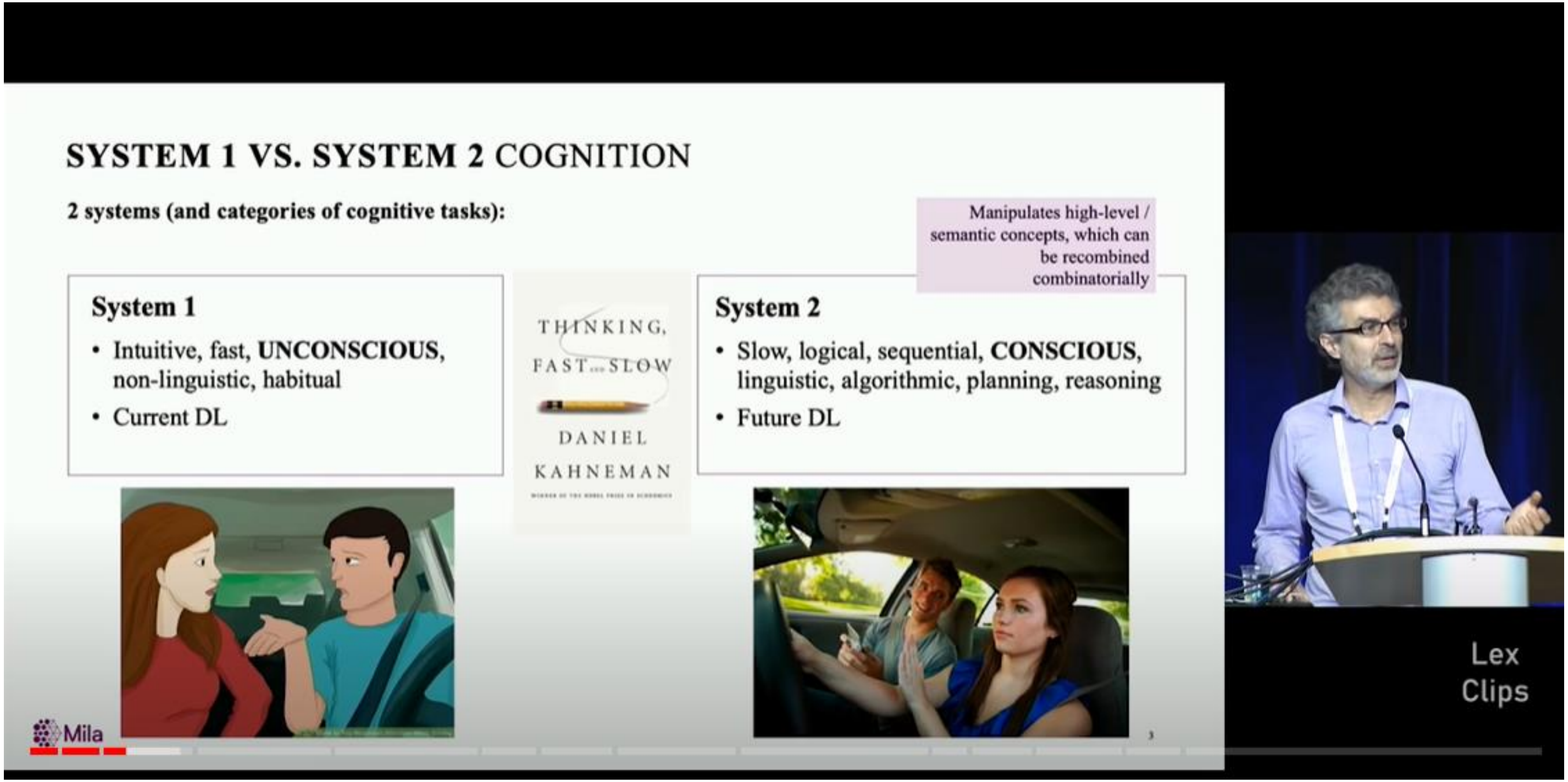
Outlook: Research Directions

- Deep Learning alone will probably not be enough to create really intelligent machines

(not just me saying that, but most of ML researchers, including Yoshua Bengio)

- Other concepts must be included:
  - Reasoning
  - Few- and Zero-Shot Learning (Meta-Learning)
  - Causality

Outlook: Research Directions



Reasoning

Abductive Reasoning:

Incomplete Observations → Best Explanation

Deductive Reasoning:

General Rule → Specific Conclusion

Inductive Reasoning:

Specific Observation → General Conclusion

I. Abduction	***II. Deduction***	***III. Induction***
Rule (first principle): All the beans in this bag are white.	*Rule (first principle)*: All the beans in this bag are white.	*Case (hypothesis)*: These beans are from this bag.
Result (conclusion): These beans are white.	*Case (hypothesis)*: These beans are from this bag.	*Result (conclusion)*: These beans are white.
Case (hypothesis): These beans are from this bag.	*Result (conclusion)*: These beans are white.	*Rule (generalized first principle or theory)*: All the beans in this bag are white.

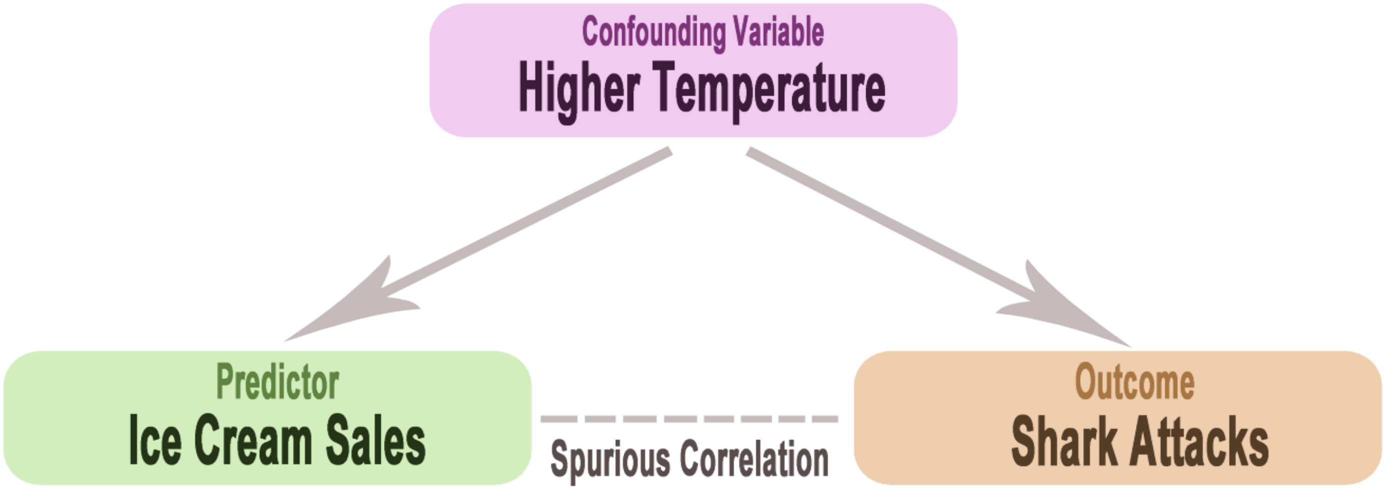
Few- and Zero-Shot Learning (Meta-Learning)

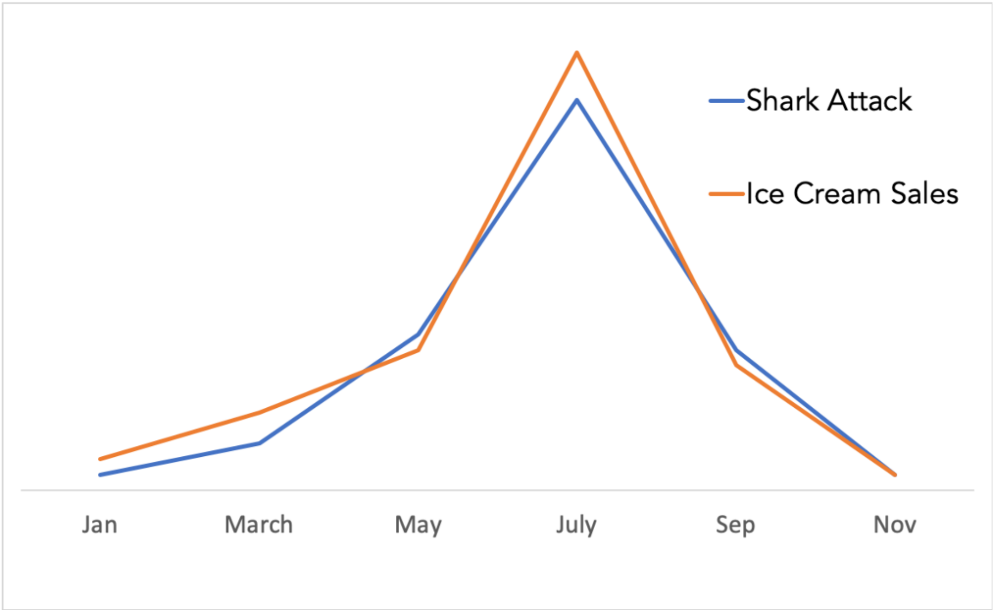
Training:

Inference:

Causality

Spurious Correlations in data (e.g. common cause)





Causality

Causal Calculus (do-Calculus):

Distinction between conditional probabilities: $P(cancer \mid smoking)$

and interventional probabilities: $P(cancer \mid do(smoking))$

Causal Graph:

- Represent causal relationships as a graph
- Three types of elements:
  - $A \rightarrow B$ (direct causation)
  - $A \leftarrow C \rightarrow B$ (common cause)
  - $A \rightarrow C \leftarrow B$ (common effect)
- Causal discovery: attempt of recovering causal graphs from observational data
- Allows for counterfactual thinking (what if…?)

Causality

Neural networks are very prone to learning spurious correlations

⇒ poor performance on out-of-distribution data

(A) **Cow: 0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98

(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97

(C) No Person: 0.97, **Mammal: 0.96**, Water: 0.94, Beach: 0.94, Two: 0.94