Machine Learning
(DS522I1D)

Module 1

Machine Learning

- Machine Learning is the science of programming computers so thev can learn from the data.

Artificial Intelligence vs Machine learning

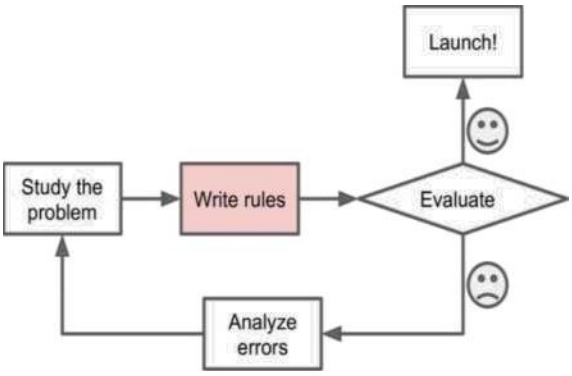
Artificial Intelligence	Machine learning
Artificial intelligence is a technology which enables a machine to simulate human behavior.	Machine learning is a subset of AI which allows a machine to automatically learn from past data without programming explicitly.
The goal of AI is to make a smart computer system like humans to solve complex problems.	The goal of ML is to allow machines to learn from data so that they can give accurate output.
In AI, we make intelligent systems to perform any task like a human.	In ML, we teach machines with data to perform a particular task and give an accurate result.

Examples of ML Applications

- Image classification: Analysing images of products automatically
- Medical diagnosis: Detecting tumors in brain scans
- Natural language processing: Automatically classitving news articles
- Text classification: Automatically flagging offensive comments
- Text summarization: Summarizing long documents
- Creating chatbot: Personal assistant
- Speech recognition: Making app react to voice comments
- Clustering: Segmenting clients based on their purchase
- Data visualization: Representing complex dataset
- Regression: Forecasting company's revenue
- Anomaly detection: Detecting credit card fraud
- Reinforcement learning: Building an intelligent boat for a game

Example of ML in Spam filter:

- Consider how to write a spam filter using traditional programming techniques.

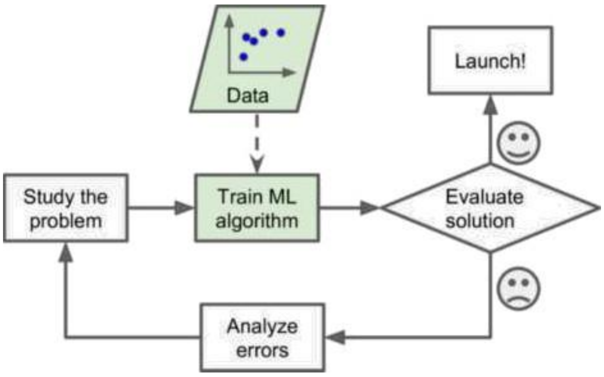


The traditional approach

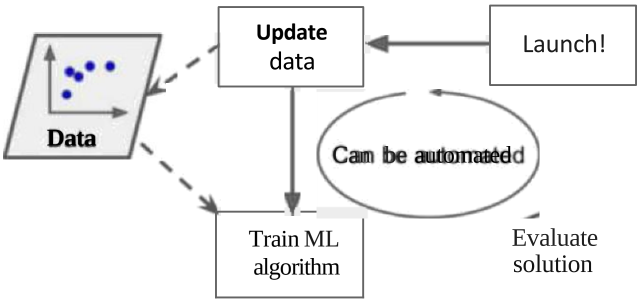
- First look at what spam typicallv looks like. Some words or phrases (such as “4U," "credit card,“ "free," and "amazing") tend to come up a lot in the subject.
- We have to write a detection algorithm for each of the patterns that are noticed, and the program would flag emails as spam if a number of these patterns are detected.
- Test the program, and repeat steps 1 and 2 until it is good enough.
- Since the problem is not small, the program will likely become a long list of complex rules, which is pretty hard to maintain.

» In contrast, a spam filter based on Machine Learning techniques automatically learns which words and phrases are good predictors of spam by detecting unusually frequent patterns of words in the spam.

- The program is much shorter, easier to maintain, and most likely more accurate.
- Moreover, if spammers notice that all their emails containing “4U" are blocked, they might start writing "For U” instead.
- A spam filter using traditional programming techniques would need to be updated to flag “For U” emails.
- If spammers keep working around your spam filter, you will need to keep writing new rules forever.
- In contrast, a spam filter based on Machine Learning techniques automatically notices that “For U” has become unusually frequent in spam flagged by users, and it starts flagging them without user intervention,



Machine Learning approach



Automatically adapting to change

Types of Machine Learning Systems

There are so many different types of Machine Learning systems. They are classified based on:

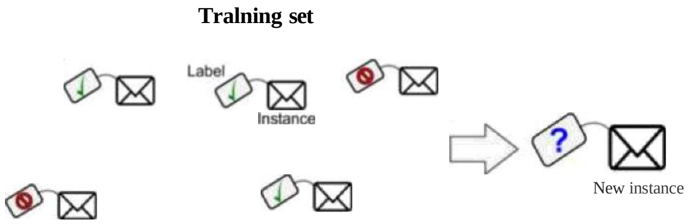
- Whether or not they are trained with human supervision (Supervised, Unsupervised, Semi supervised and Reinforcement Learning).

» Whether or not they can learn incrementally on the fly (Online learning and Batch learning).

- Whether they work by simply comparing new data points to known data points, or instead build a predictive model (Instance-based and Model-based learning).

Supervised Learning

- Supervised learning is a type of Machine learning in which the machine needs external supervision to learn.
- In supervised learning, the training data fed to the algorithm are labeled.
- In supervised learning, the training data consists of pairs of input features (also known as predictors or independent variables) and their corresponding output labels (also known as targets or dependent variables).
- The goal of supervised learning is to train a model that can accurately predict or classify new, unseen data based on the patterns it has learned from the labelled training data. The model learns from the examples in the training data, which are typically labelled by human experts or annotated with known outcomes.
- A typical supervised learning task is classification. The spam filter is a good example of this. It is trained with many example emails along with their class (spam or ham), and it must learn how to classify new emails.



A labelled training set for supervised learning

Example of Supervised learning algorithms:

Here are some of the most important supervised learning algorithms:

- k-Nearest Neighbor
- Linear Regression

» Logistic Regression

» Support Vector Machine

- Decision Tree
- Random Forest
- Neural networks

Unsupervised Learning

- It is a type of machine learning in which the machine does not need any external supervision to learn from the data.
- In unsupervised learning, the training data is unlabelled.

» The goal of unsupervised learning is to discover patterns, relationships, or structures in the data without any prior knowledge or guidance.

- Unsupervised learning is often used when the task of labelling data is expensive, time-consuming, or not feasible. It is also useful when the goal is to explore and discover hidden patterns or structures within the data without specific preconceived notions.

Some of the most important unsupervised learning algorithms are:

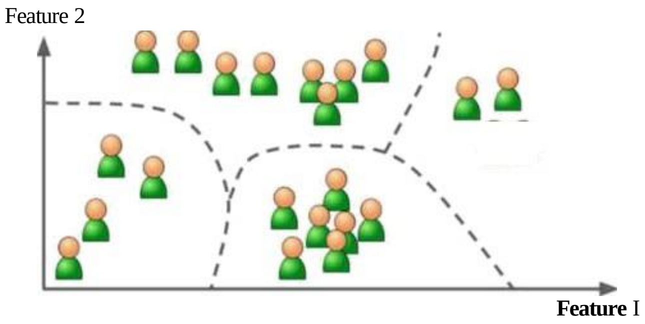
» Clustering

a K-Means

  a Hierarchical Cluster Analysis (HCA)

- Anomaly detection and noveltv detection
  a One-class SVM
  a Isolation Forest
- Visualization and dimensionality reduction
  a Principal Component Analysis (PCA)
  a Locally-Linear Embedding (LLE)

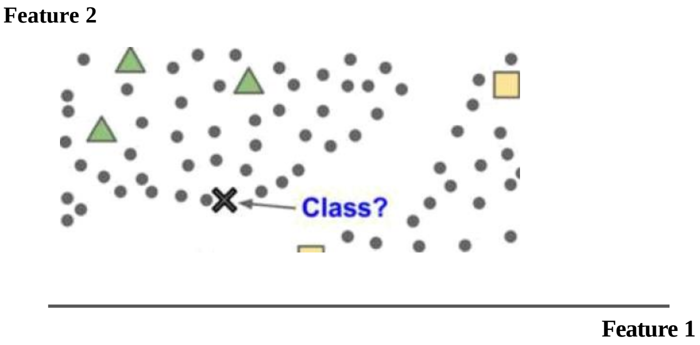
Clustering algorithms group data points into clusters based on their similarity or proximitv in the feature space. Examples of clustering tasks include customer segmentation, image segmentation, and document clustering.



Semisupervised Learning

- Semi-supervised learning is a tvpe of machine learning that combines a small amount of labelled data with a large amount of unlabelled data during training.
- Semi-supervised learning falls between unsupervised learning and supervised learning.
- The basic idea behind semi-supervised learning is that the availability of some labelled data can help improve the learning process for the model, even when a large amount of data is unlabelled.
- The model can leverage the information from the labelled data to learn the underlying patterns in the data and make predictions on the unlabelled data.

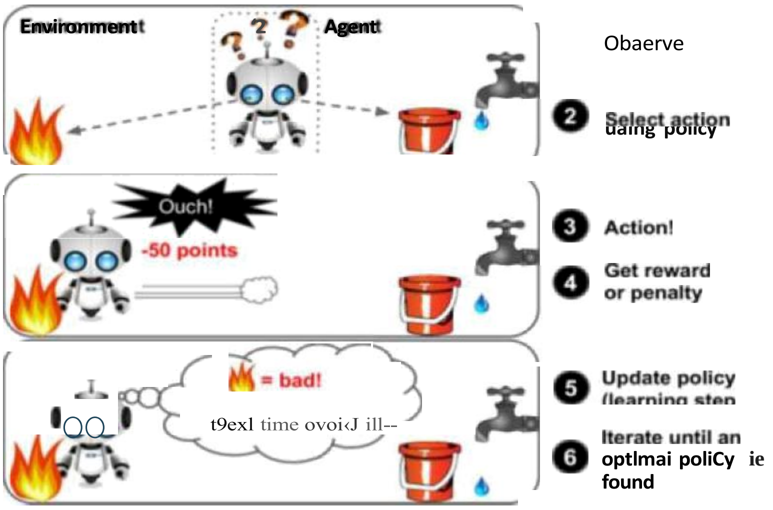
» Some photo-hosting services, such as Google Photos, are good examples of this. Once we upload all the photos to the service, it automatically recognizes persons.



Reinforcement Learning

- Reinforcement Learning is a tvpe of machine learning technique that enables the svstem (agent) to learn in an interactive environment by trial and error using feedback from its own actions and experiences.
- Reinforcement learning focuses on training agents to make decisions or take actions in an environment to maximize a cumulative reward signal.
- In reinforcement learning, an agent interacts with an environment, receives feedback in the form of rewards or penalties, and learns to take actions that optimize its behaviour over time.

- The agent is the learner or decision-maker that takes actions in the environment. The agent can observe the state of the environment and takes actions based on a policy, which is a mapping from states to actions.
- Reinforcement learning has been successfully applied to a wide range of applications, including game playing, robotics, recommendation systems, finance, and healthcare, among others.



Batch and Online Learning

Batch or Offline Learning

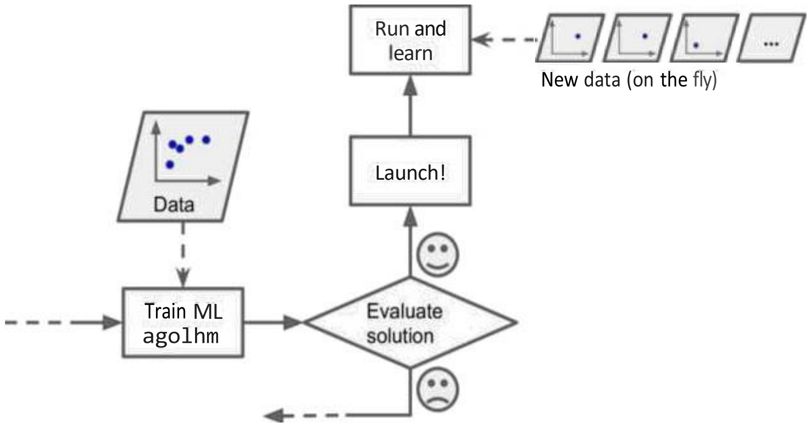
- In batch learning, the system is incapable of learning incrementally. It must be trained using all the available data.
- This will generally take a lot of time and computing resources, so it is typically done offline.
- First, the svstem is trained, and then it is launched into production and runs without learning anvmore. It just applies what it has learned.

» If we want a batch learning system to know about new data (such as a new type of spam), we need to train a new version of the system from scratch on the full dataset (not just the new data, but also the old data), then stop the old system and replace it with the new one.

Online Learning

» In online learning, we train the system incrementally by feeding it data instances sequentially, either individually or by small groups called mini-batches. Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives.

- Online learning is great for systems that receive data as a continuous flow and need to adapt to change rapidly.

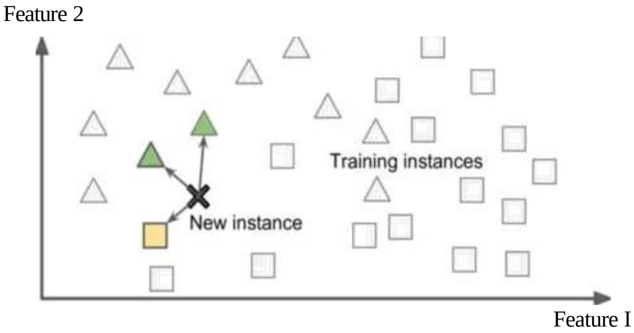


- An example of one of these systems might be one that predicts the weather or analyses stock prices.
- Once an online learning system has learned about new data instances, it does not need them anymore, so we can discard them. This can save a huge amount of space,
- Online learning algorithms can also be used to train systems on huge datasets that cannot fit in one machine's main memory.

Instance-Based and Model-Based Learning

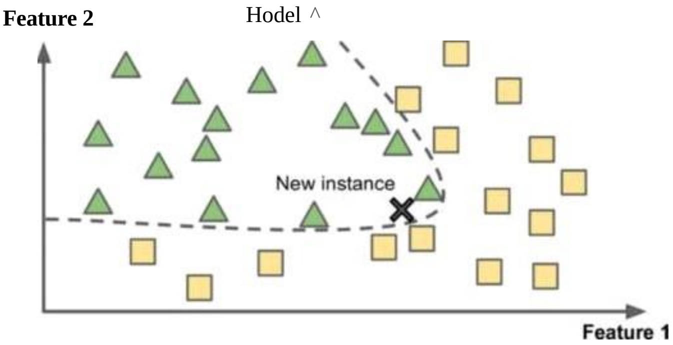
Instance-Based Learning

- Instance-based learning systems, also known as lazy learning systems, store the entire training dataset in memory and when a new instance is to be classified, it compares the new instance with the stored instances and returns the most similar one.
- These systems do not build a model using the training dataset.



Model-Based Learning

- Model-based learning systems are also known as eager learning systems, where the model learns the training data.
- These systems build a machine learning model using the entire training dataset, which is built by analysing the training data and identifying patterns and relationships. After that, the model can be used to make predictions on new data.



Main Challenges of Machine Learning

Insufficient Quantity of Training Data

- The most important task we need to do in the machine learning process is to train the data to achieve an accurate output.
- Less amount training data will produce inaccurate or too biased predictions.
- Machine-learning algorithm needs a lot of data to distinguish. For complex problems, it mav even require millions of data to be trained.

Poor-Quality Data

» Data plays a significant role in machine learning, and it must be of good quality as well. Noisy data, incomplete data, inaccurate data, and unclean data lead to less accuracy in classification and low-quality results.

- Hence, data quality can also be considered as a major common problem while processing machine learning algorithms.

Overfitting the Training Data

- Overfitting is a common problem in machine learning where a model becomes too specialized in capturing the training data and performs poorly on unseen data.
- It occurs when a model learns to memorize the training data rather than generalize from it. Overfitting can lead to poor performance and reduced predictive accuracy of the model.

» Overfitting is one of the most common issues faced by Machine Learning engineers and data scientists.

- Whenever a machine learning model is trained with a huge amount of data, it starts capturing noise and inaccurate data into the training data set. It negatively affects the performance of the model.
- The main reason behind overfitting is using non-linear methods used in machine learning algorithms as they build non-realistic data models.

Methods to reduce overfitting:

- Reduce model complexity by simplifying the model by selecting one with fewer parameters.

« Regularization

» Reduce the noise

- Reduce the number of attributes in training data.
- Constraining the model

Underfitting

- Underfitting is another common problem in machine learning where a model fails to capture the underlying patterns in the training data, resulting in poor performance on both the training and unseen data.
- Underfitting occurs when a model is too simplistic and lacks the capacity to capture the complexity of the data.

» Underfitting is just the opposite of overfitting. Whenever a machine learning model is trained with less amounts of data, and as a result, it provides incomplete and inaccurate data and destroys the accuracy of the machine learning model.

Methods to reduce Underfitting:

- Increase model complexitv

» Remove noise from the data

« Train with increased and better features

- Reduce the constraints
- Increase the number of iterations to get better results

Irrelevant features

- Although machine learning models are intended to give the best possible outcome, if we feed garbage data as input, then the result will also be garbage.

- Hence, we should use relevant features in our training sample. A machine learning model is said to be good if training data has a good set of features or less to no irrelevant features.

Non-representative Training Data

- The training data should be representative i.e., it should generalize well for the new cases (the cases that are going to occur) also.
- Finding representative training data is a serious challenge for every ML practitioner because using non-representative training data can lead to false predictions.

Testing and Validating

» The only way to know how well a model will generalize to new cases is to actually try it out on new cases. One way to do that is to put the model in production and monitor how well it performs. The steps involved here are as follows:

a Data Splitting: The first step is to split your dataset into two subsets: the training set and the test set. The training set is used to train the model, while the test set is used to evaluate its performance.

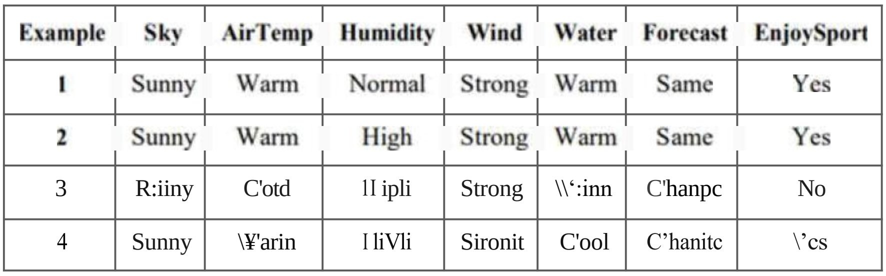
O Training: During the training phase, the model learns patterns and relationships in the training data. The goal is to minimize the error on the training set, i.e., the difference between the actual values and the predicted values by the model.

o Testing: Once the model is trained, it is evaluated using the test set. The test set contains data that the model has not seen during training. By applying the trained model to the test set, we can assess how well it generalizes to new, unseen data.

o Generalization Error: The difference between the model's performance on the training set and the test set is called the generalization error, also known as the out-of-sample error. This error indicates how well the model is expected to perform on new, unseen instances. A low generalization error suggests that the model has learned the underlying patterns in the data and can make accurate predictions on new data.

Concept Learning

- A task of acquiring potential hypothesis (solution) that best fits the given training examples is called concept learning,
- Consider the example task of learning the target concept "Days on which Aldo enjoys his favourite water sport”.



Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	R:iiny	C'otd	lI ipli	Strong	\\‘:inn	C'hanpc	No
4	Sunny	\¥'arin	I liVli	Sironit	C'ool	C’hanitc	\’cs

Table: Positive and negative training examples for the target concept ***EnjoySport.***

- The task is to learn to predict the value of EnjoySport for a random day, based on the values of its other attributes?
- Let each hypothesis be a vector of six constraints, specifying the values of the six attributes Sky, AirTemp, Humidity, Wind, Water, and Forecast.
- For each attribute, the hypothesis will either

a Indicate by a "?" that any value is acceptable for this attribute.

o Specify a single required value (e.g., Warm) for the attribute.

a Indicate bv a "@" that no value is acceptable.

- The hypothesis that Aldo enjoys his favourite sport only on cold days with high humidity is represented by the expression:

  (?, Cold, High, ?, ?, ?)
- The most general hypothesis-that every day is a positive example is represented by:

  (?, ?, ?, ?, ?, ?)
- The most specific possible hypothesis-that no day is a positive example is represented by:

  (a, a, a, a, a, e)

Find-S - Finding a Maximally Specific Hypothesis

- Find-S, also known as Find-S algorithm, is a concept learning algorithm used in machine learning and artificial intelligence to find a maximally specific hypothesis from a given set of training data.
- The Find-S algorithm is a simple approach that learns a hypothesis in the form of an instance of a target concept based on positive examples.
- The Find-S algorithm starts with an empty hypothesis that represents the most general description of the concept, and then iteratively refines the hypothesis by considering each positive example in the training data.
- The algorithm updates the hypothesis to be more specific with each positive example encountered, narrowing down the space of possible hypotheses until a maximallv specific hypothesis is obtained.
- Consider the Enjoysport dataset:

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	5 uiJnS'	\\'anal	I li¿ul\	S I rnny	\\'':irtn	S.cmv	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Find-S Algorithm:

```
1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x
     For each attribute constraint ai in h
     If the constraint ai i8 Satisfied by x
     Then do nothing
     Else replace ai in h by the next more general constraint that is satisfied by x
3. Output hypothesis h
```

» The first step of Find-S is to initialize h to the most specific hypothesis in
  H h - (&, 6, Ci, &, 8, &)

- Consider the first training example

  x1 = <Sunny, Warm, Normal, Strong, Warm, Same> (Positive)
- None of the "&" constraints in h are satisfied by this example, so each is replaced by the next more general constraint that fits the example

  h1 = <Sunny, Warm, Normal, Strong, Warm, Same>
- Consider the second training example

x2 = <Sunny, Warm, High, Strong, Warm, Same> (Positive)

- Compare h1 with x2
  h2 = <Sunny, Warm, ?, Strong, Warm, Same>
- Consider the third training example
  x3 = <Rainy, Cold, High, Strong, Warm, Changes (Negative)
- The Find-S algorithm simply ignores every negative
  example. h3 = < Sunny, Warm, ?, Strong, Warm, Same>
- Consider the fourth training example
  x4 = <Sunny, Warm, High, Strong, Cool, Change> (Positive)
- Compare h3 with x4
  h4 = < Sunny, Warm, ?, Strong, ?, ? >

The kev Property of the Find-S algorithm:

- Find-S is guaranteed to output the most specific hypothesis within H that is consistent with the positive training examples.
- Find-S algorithm's final hypothesis will also be consistent with the negative examples provided the correct target concept is contained in H, and provided the training examples are correct.

Version Space

- In concept learning, the version space represents the set of all hypotheses that are consistent with the observed training examples.
- Version Space is the subset of hypotheses H consistent with the training examples D.

$$VS_{H,D} \equiv \{h \in H \mid Consistent\,(h, D)\}$$

- A hypothesis h is consistent with the training examples if and only if h(x) = c(x).

List-Then-Elimination algorithm

- The List-Then-Elimination algorithm is a decision-making process that involves creating a list of options and then systematically eliminating them based on predetermined criteria or factors.
- This algorithm can be applied to a variety of situations, including problem-solving, decision-making, and choosing among multiple options.
- List-then-eliminate algorithm is used to obtain the version space.

Algorithm:

» Assign the list containing all hypothesis in H to version space c

- For each training example, (x, c(x))
  - a remove from version space any hypothesis h for which h(x) x c(x)
- Output the list of hypotheses in version space

Candidate-Elimination Learning Algorithm

- The Candidate-Elimination algorithm computes the version space containing all hypotheses from H that are consistent with an observed sequence of training examples.
- Uses the concept of version space.
- Considers both positive and negative values.
- Considers both specific and general hypothesis. For positive samples - change specific hypothesis and for negative samples — change general hypothesis.

- Consider the Enjovsport dataset:

I:xnmylc	hks	:tir“l”en›p	humidits	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

- Initializing the general and specific hypothesis:

$S_0 \langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing \rangle_+$

- The first training example is positive. So, convert specific hypothesis to general hypothesis.
- Compare Sowith training example to get Si.
- No update of G is needed. Gi will be same as Go.

Sunny, Warm, Normal, Strong, Warm, Same ⟩ +

So ⟨∅, ∅, ∅, ∅, ∅. ∅⟩

S1 ⟨*Sunny, Warm, Normal, Strong, Warm, Same*⟩

Gt, G1 ⟨?, ?, ?, ?, ?, ?⟩

- The second training example is again positive. So, convert specific hypothesis to general hypothesis.
- Compare Si with training example to get S2.

» No update of the G is needed. G2 will be same as Go.

h iHilly. \\'drill. Hieh. Strolle. \\'ann. SCtrlIt” *

S_1 ⟨*Sunny, Warm, Normal, Strong, Warm, Same*⟩

S2 *Sunny, Warm, ?, Strong, Warm, Same*

G_1, G_2 ⟨?, ?, ?, ?, ?, ?⟩

- The third training example is negative. So, convert general hypothesis to specific hypothesis.
- No update of the S is needed. S3 will be same as S2
- Compare S3 with training example to get G3.Change when there is change in attribute.

⟨Rainy, Cold, High, Strong, Warm, Change ⟩ –

S_2, S_3 ⟨*Sunny, Warm, ?, Strong, Warm, Same*⟩

G_3 ⟨*Sunny, ?, ?, ?, ?, ?*⟩ ⟨*?, Warm, ?, ?, ?, ?*⟩ ⟨*?, ?, ?, ?, ?, Same*⟩

Gz ⟨?, ?, ?, ?, ?, ?⟩

- The fourth training example is again positive. So, convert specific hypothesis to general hypothesis.
- Compare Ss with training example to get S‹.

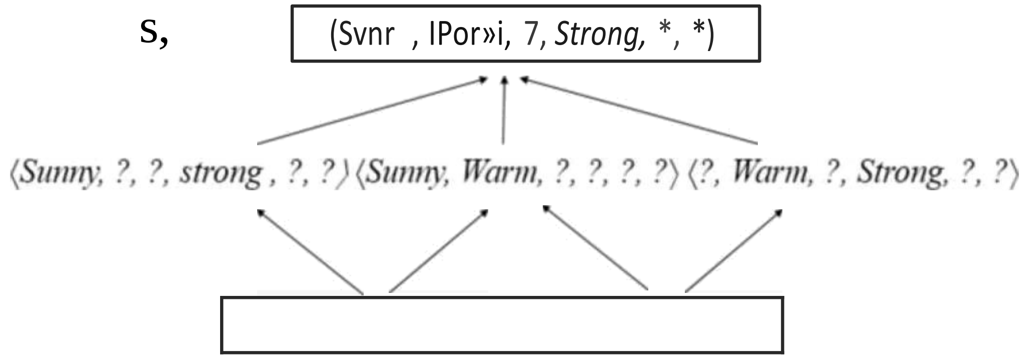
« Finally, compare E3 with S4 remove the cases that fails to cover the new positive example.

+z (Sunny, Wami, ?, Strong, Warm, Smnej

S« (Sung warm, ?, Strong ?, ?)

G_3 ⟨Sunny, ?, ?, ?, ?, ?⟩ ⟨?, Warm, ?, ?, ?, ?⟩ ⟨?, ?, ?, ?, ?, Same⟩

- After processing these four examples, the boundary sets S and G‹ delimit the version space of all hypotheses consistent with the set of incrementally observed training examples.


S,
(Svnr , IPor»i, 7, Strong, *, *)
⟨Sunny, ?, ?, strong , ?, ?⟩ ⟨Sunny, Warm, ?, ?, ?, ?⟩ ⟨?, Warm, ?, Strong, ?, ?⟩

Algorithm:

```
Initialize G to the set of maximally geaeml hypotheses in H
Initialize S to the set of maximally specific hypotheses in H
For each training example d, do
  • If d is a positive example
      • Remove from G any hypothesis inconsistent with d
      • For each hypothesis s in S that is not consistent with d
          • Remove s from S
          • Add to S all minimal generalizations h of s such that
              • h is consistent with d, and some member of G is mon general than h
          • Remove from S any hypothesis that is more gcneml than another hypothesis in S

  • If d is a negative example
      • Remove fiom S any hypothesis inconsistent with d
      • For tath hypothesis g in G that is not consistent with d
          • Remove g from G
          • Add to G all minimal specializations h of g such that
              • h is consistent with d, and some member of S is more specific than h
          • Remove from G any hypothesis that is lese general than another hypothesis in G
```