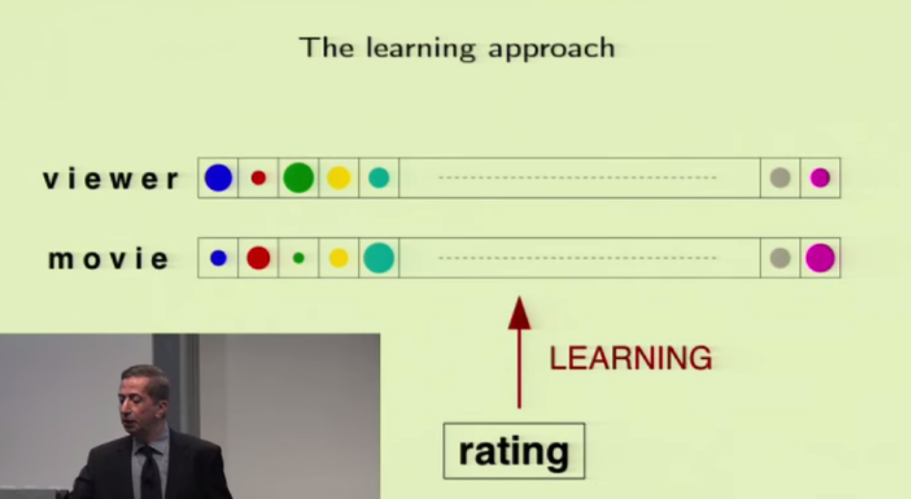
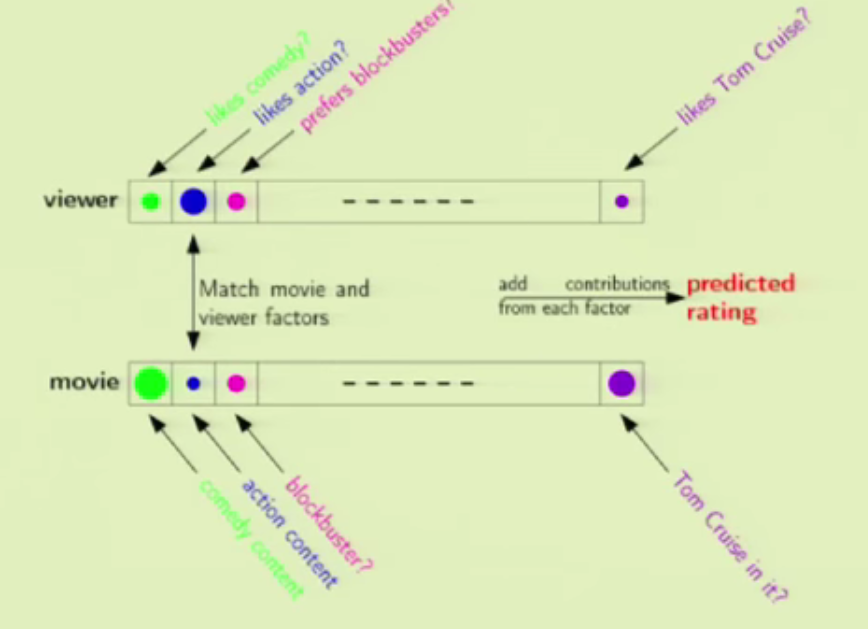
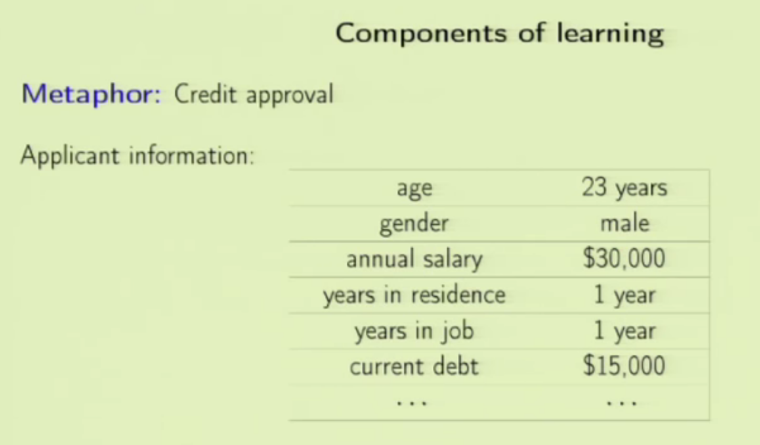
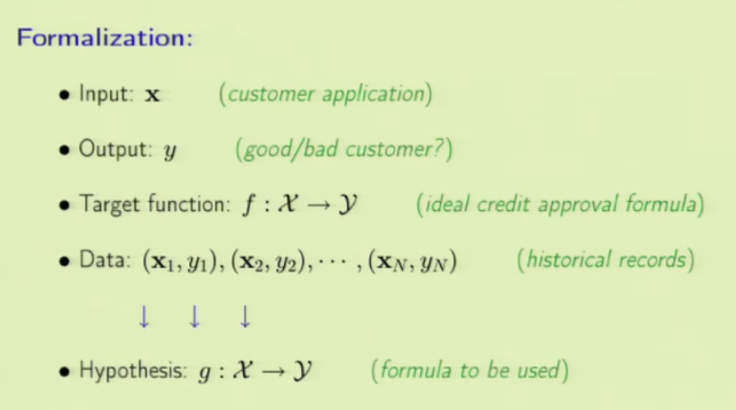
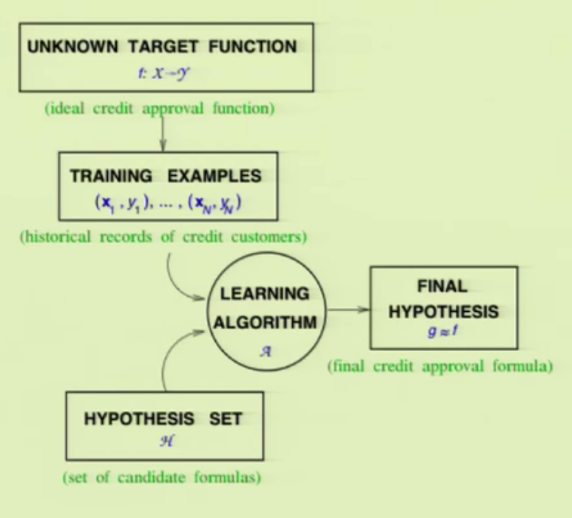
Logo of course, it is a technical figure, wait to know what it is.



The model on right uses machine learning, on the left uses straight forward way which isn’t efficient.

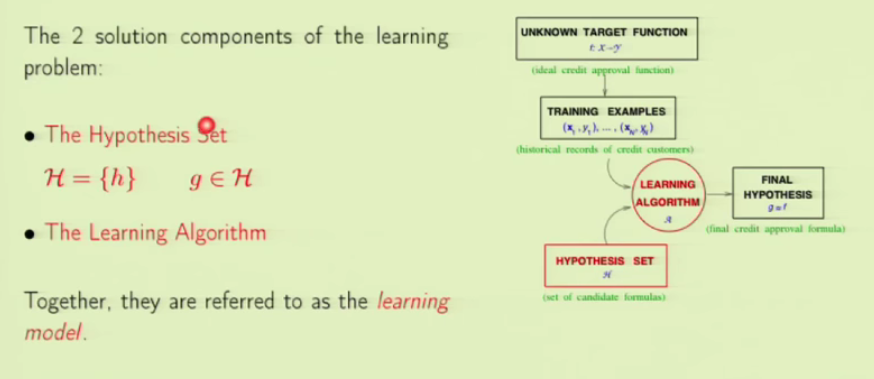
Approve or reject loan request based on historic data.



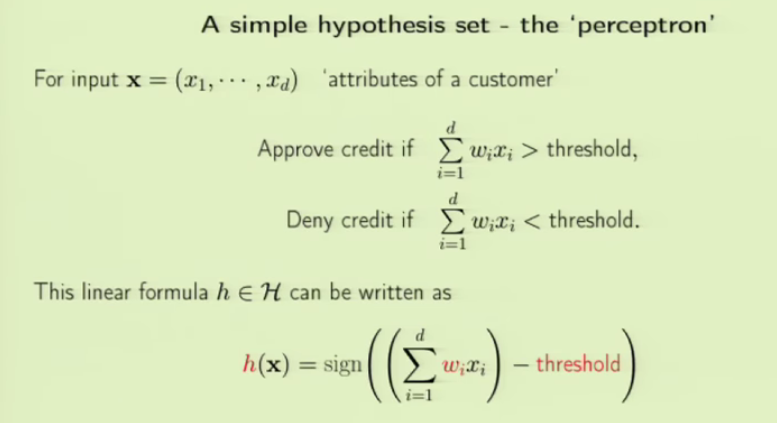
From historical data, we use our approval function to accept or deny loan. We make it better using the historical data so that we will get the usable function g which is our predictor.

From unknown to final, it looks like a normal flow. Why have the hypothesis set?

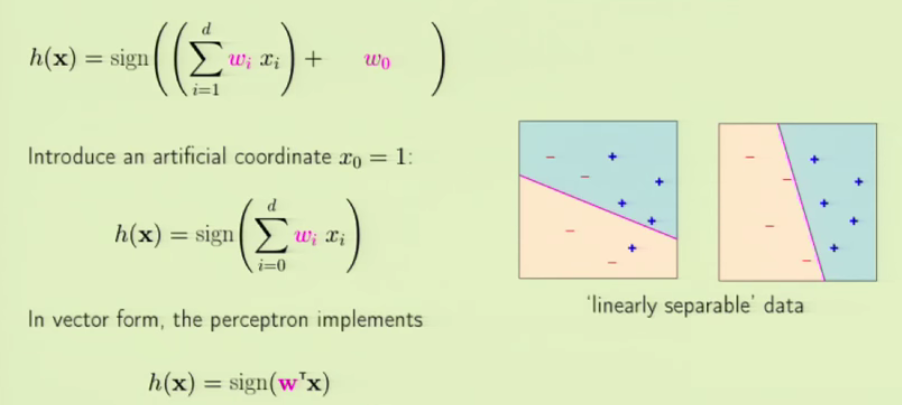
There is no downside with HT and there is an upside. Trying different modes will only make the learning better.



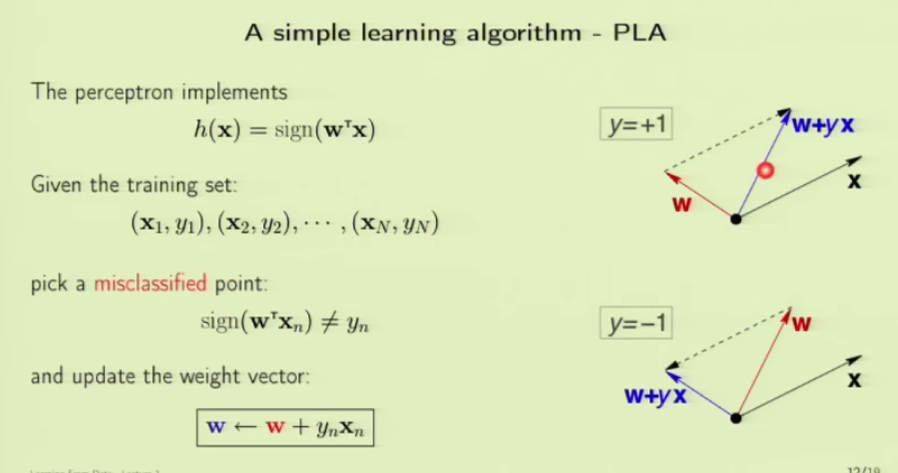
In the above component diagram, what we have control over is learning algo and hypothesis set. So they are the solution components. Learning model consists of both hypotheses set and learning algorithm. As in, perceptron model will be our hypothesis, perceptron learning algo is our learning algo. Neural network + back propagation, support vector machine + quadratic program, every model has these two components.

Example model: Perceptron, a simple model

Depending on the type of attribute or feature, a weight is assigned(like for the feature outstanding loan, negative weight is given meaning not desirable for further loan sanction). We calculate the credit score and compare it with threshold score. As we see in the hypothesis, what determines the output is the threshold we chose and the weights assigned to each feature.



When we work on linear data, the model we used will draw a separator between the classes of output. As per our choice of weights and threshold, the separating line changes as seen in the picture. The notation is changed, notice it. WE used w0 instead of threshold. WE took w0 as negative of threshold. W0 = -(threshold). WE do that because we introduce an artificial coordinate x0 = 1. By doing so, our hypothesis function gets simplified.

Once we have the hypothesis, we try to implement it with the learning algorithm using the historical data. What we do is, we feed in the x value to our H, and see what it predicts for a assigned weight W. As per the outcome, we try to adjust the weights W and then predict again. This way we keep doing it, until the model H predicts well. If model predicts -1 instead of 1, we multiply W with actual output y and add it to weight W to form new adjusted W. This is repeated for some iterations.