Some terms

Collaborative filtering - collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users
Assumes that one person has same opinion as the other person

Inductive learning Assumption - after seeing many training examples result in high accuracy on unobserved examples
Hypothesis approximates well

average hypothesis only depends on H, does not always lie in H
$H_2 \subset H_{10}$

Target function y = f(x)
Unbiased learner cannot generalise $\rightarrow$ inductive bias

h is +ve when h(x) = 1 $\geq_g$ more general than or equal to
$h_j \geq_g h_k \equiv \forall x \in X(h_k(x) = 1) \rightarrow (h_j(x) = 1)$
Not a total order has there may have the same specificity but different hypothesis (maximally specific h)

Consistency: h is consistent iff h(x) = f(x)
$h4(\langle 0, 1\rangle) = 1 \neq f(\langle 0, 1\rangle) = 0.$

Proposition 1
h is consistent with iff every +ve training instance satisfies h and every -ve training instance does not satisfy h.
Proposition 2
If h belongs to H, then it is consistent

Naive Bayes

Bayes Rule

$P(y|x) = \frac{P(x|y) \times P(y)}{P(x)}$
$P(X_1 = 0, X_2 = -1) = P(X_1 = 0 \mid C_1) \times P(X_2 = -1 \mid C_1) \times P(C_1) + P(X_1 = 0 \mid C_2) \times P(X_2 = -1 \mid C_2) \times P(C_2)$

a priori - equally probable

Naive - Assuming that features are conditionally independent of each other

Insensitive to the number of training examples

Calculate which is more likely
1. $P(Outcome|x) = \text{P}(\text{X}_1|Outcome) * P(X_2|Outcome) * P(Outcome)$
2.$P(Not\,Outcome|x) = \text{P}(\text{X}_1|Not\,Outcome) * P(X_2|Not\,Outcome) * P(Not\,Outcome)$

Calculate probability outcome occurs
#1/(#1+#2)

Perceptron classifier

$\text{x}_0\theta_0 + x_1\theta_1 + x_2\theta_2$
of the previous weight vector
first instance will compute based on initial vector
For -1, first point checks == -1, != 1 for the points after
Update weight = - (x $\times$ learning_rate) for the misclassified instance

For +1, first point checks != -1, == 1 for the points after
Update weight = + (x $\times$ learning_rate) for the misclassified instance

points are linearly separable if there are no longer any misclassified points
not misclassified: sign(x) function having -1 when x is ≤ 0

Supervised learning

Identifying objects
Output would be a continuous value
Given many emails, you want to **determine** if they are Spam or Non-Spam emails.

Given historical weather records, **predict** if tomorrow's weather will be sunny or rainy.

Unsupervised learning

No labels (no output) Kmeans
Input: Given a set of data points P, number of clusters k
1. Randomly pick k points from P as centers $c_j = 1..k$
2. Iterate (until max-iterations or c_j no longer changes up to some threshold)
- Assign each point to nearest center: $y_i = \arg\min \parallel p_i - c_j \parallel^2$
c_j = Re-estimate each center as mean of points assigned to it

Given a set of news articles from many different news websites, find out what are the main topics covered.

From the user usage patterns on a website, figure out what different groups of users exist.

RL

Planning problems
choices that maximise the outcome

Find S

1.Initialise h to most specific h
2. If the positive instance differs, then replace with ?

Version space

List all hypothesis, remove any hypothesis that is inconsistent with any training example

Candidate Elimination

prefers positive examples over negatives as there can only be one maximally specific hypothesis in each iteration ? don't care
no value
Set $S_0 = \langle \emptyset\emptyset\emptyset\emptyset\emptyset\emptyset \rangle$

$\mathrm{G}_0, G_1 = \langle ?????? \rangle$

For the negative example,
Create h from the specific hypothesis when it differs based on previous instance
Remove those that are inconsistent from the previous general hypotheses
Keep S to the previous instance S

For the positive example,
replace with ? when it differs

Cost functions

Logistic regression
Is binary classification
Need meta strategies for multi class and it chooses the model that has the highest confidence in predictions
Only gives values 0 and 1 so it is not a good result

$\mathrm{J}(\theta) = \frac{1}{m}\sum_{j=1}^{m} ln(1 + exp^{-y^{j\theta^{T}x^{j}}})$

Linear regression

J$(\theta) = \frac{1}{m}\sum\limits_{j=1}^{m}(\theta^T x^j - y^j)^2$

Linear algebra

trace(AB) = trace(BA) = trace(I_{n+1}) = $n + 1$

$Idempotency$
$H^2 = H$
$(I - H)^2 = I - 2H + H = I - H$

Pseudoinverse

$(\mathrm{X}^T X)^{-1} X^T$
$\theta = X^+ y$
$X = \begin{pmatrix} 1 & .. & .. \\ 1 & .. & .. \\ 1 & .. & .. \end{pmatrix}$

Maximizing likelihood

Minimize cross entropy = - $\frac{1}{m}\sum\limits_{j=1}^{m} ln\ g(y^j \theta^T x^j)$

Ridge Regression

$\frac{1}{m}\sum\limits_{i=1}^{m}(y_i - w^T x_i)^2 + \lambda \parallel w \parallel^2$
$-2X^\top (Y - X * \theta) + 2 * \alpha * \theta$

Noise

Not deterministic only deterministic for f(x) = 0
y - f(x)
P(y — x)
Noisy target = E[y—x] + (y - f(x))

Stochastic noise vs Deterministic

To reduce stochastic noise, the only way is to re-measure y
To reduce deterministic noise, change H

Underfitting and Overfitting

Why do we regularize? We fit the data too much, we are fitting to the noise, balance fitting to the observed data. If we fit the data too well, the performance can be poor called overfitting. Overfitting can happen even if the data is not noisy.

Overfit - high level of noise, high complexity of f(x)
Underfit - high bias and low variance

Reduce bias by increasing complexity of hypotheses
simple representation of more complex reality

Non-linear transformation

Transform this non-linear equation to a linear one: $y = ax^b$?

Under the condition $y > 0$ and $x > 0$
$ln(y) = ln(ax^b)$
$ln(y) = ln(a) + ln(x^b)$
$ln(y) = ln(a) + bln(x)$
let $y' = ln(y)$, $x' = ln(x)$, $a' = ln(a)$, the equation can be written as $y' = a' + bx'$