Nature-inspired computing

1. Describe the main components of an evolutionary program: population representation, generation, selection, combination, replacement, and stopping criteria?

Represent the population with a list of solutions, start with a randomly generated or systematically built population. Compare solutions to each other using a fitness function to evaluate them. Select the best ones to combine them into new solutions (crossover), mutate some to get a new random solution that can expand the search space.

Stopping criteria: n generations, when no change in top x for n iterations, when no change in population for n iterations, resources (time), target fitness.

2. Describe when to use genetic algorithms?

GAs are good, when there is a clear way to evaluate fitness of solutions (and we don't know the original function - if we would, you don't need GA for it), when we have a big space to search and when we can find a good representation of genes (agents). For example we can use them with TSP where a fitness function is the distance of the traversed path.

3. Describe the strengths and weaknesses of evolutionary programs.

Strengths: robust, adaptable and general, requires only fitness function and representation of genes

Weaknesses: can get stuck in local extreme, can take a long time to converge to solution, time complexity rises fast with bigger population

4. Describe the main characteristics of genetic algorithms (GA) and genetic programming (GP).

GA is based on evolution. => I think the same answer as in 1. Applies here

GP instead of representing solutions in list/objects, represent them with tree structures. Crossover: exchange subtree, mutation: random change in trees. Variable length encoding, more flexible, often grow in complexity

5. Describe terms from evolutionary computation such as population variability, fitness function, coevolution.

Fitness function: is a function that takes a solution as input and evaluates it, to see how "good" the solution is.

Population variability: we need to have a population that encompasses as big a solution space as possible to find a solution close to the optimal as possible (eg. 2ˆ30 solution space, population of 10 will probably not find a very good solution)

Coevolution: basically crossover => two agents affect evolution by combining traits.

Was mentioned more in context of solving related problems together

6. Describe different gene representations in GA, operations on them, and their strengths and weaknesses: bit and numeric vectors, strings, permutations, trees.

bit/numeric: good for problems that can be represented with numbers, cannot represent very complex problems, eg. good for knapsack problem

Strings:

Permutations: good for problems where we are looking for a solution of a sequence of numbers (TSP), then we can use GA to "learn" the best permutations

Trees: good for problems where we want to find the formula for the solution (as formulas can be nicely represented with trees)

7. What are linear crossover, Gray coding $of binary numbers, adaptive crossover, gaussian mutation, Lamarckian mutation, and elitism? What are their advantages compared to baselines?

Linear crossover: takes a linear combination of the two individuals, have a "probability" for each bit in each agent and take each bit with probability p from agent 1 and with probability (1-p) from agent 2

Gray coding: Encode binary numbers in such a way that incrementing a number by 1 takes only 1 bit change (Sth like this: Order binary representations of numbers in such a way that the next number is only one bit changed: 0 - 1 - 11 - 10 - 110 - ...)

Adaptive crossover: Use bit templates for crossover (1-first parent, 0-second parent). Learn which templates work best

Gaussian mutation: Mutate by adding a Gaussian error to the mutation

Lamarckian mutation: search for locally best mutation

Elitism: choose n of the best solutions in population and keep them for the next population

8. Describe the following evolutionary models: proportional and rank proportional roulette wheel, tournaments, single tournament, and stochastic universal sampling?

Tournaments: have agents "battle" each other, by assigning them probabilities according to their fitness values. Best solution => best probability of winning.

Proportional: Assign each agent a probability according to their fitness value. Use randomly generated numbers to select agents.

Rank proportional: Assign each agent probability according to their rank of fitness value.

Single tournament: randomly split population into small groups and apply crossover to two best agents from each group, their offspring replace the two worst agents from the group

Stochastic: F = sum(all fitness values), N = size of population we want. Make a F/N interval. Assign part of the interval to each agent according to fitness values. Use RNG to generate numbers, if generated number is within an interval of some agent => choose the agent

9. How to prevent niche specialization in GA?

We punish agents that are too similar to others => depending on the type of problem (min/max) decrease/increase the fitness value

10. Explain hypotheses on why GAs work?

When we have a big enough population and the right parameters, we can search a pretty big solution space.

11. What are the typical parameters of GAs?

Probability of crossover, probability of mutation, population size, max number of iterations, max number of iterations with the same best solution, selection method, termination criteria.

12. Where to use GAs and where not?

YES: where there are many local extrema, fitness function easily defined, robustness, don't need specialized methods

NO: huge solution spaces with large solutions (eg. list of list of list)

13. Why are GAs suitable for multiobjective optimization, and what is Pareto optimal solution?

Use fitness functions with different objectives and try to improve them.

Pareto: we cannot improve conflicting criteria without getting worse on others

https://en.wikipedia.org/wiki/Multi-objective_optimization

14. Explain the main problems of genetic programming.

Needs huge populations(thousands), it's slow, problems involving physical environments: making trees that are really executable, execution can change the environment which changes fitness function, calculating fitness function with simulation takes a lot of time.

Machine learning (ML)

Try to estimate f(X) so we can get the most accurate Y to the actual result.

$$Y = f(X)$$

15. Describe the two main goals of ML, prediction and inference, and explain why they are sometimes in contradiction.

Prediction: if we can make a good estimate, then we can make accurate predictions for the response (Y) based on X

Inference: we are interested in the type of relationship between Y and X, model interpretability is essential for inference.

If we want good accuracy (prediction), we might need a much more complicated model which will have lower interpretability and vice versa. But it can also happen that some complicated model gives us bad results (overfitting) and thus lower accuracy.

16. What parametric and non-parametric ML methods exist?

Parametric methods: Logistic regression, Naive bayes, simple neural networks

Non-parametric methods: kNN, decision trees, SVM

17. Describe the main characteristics of supervised, unsupervised, and semi-supervised ML methods?

Supervised learning: both X and Y are observed

Unsupervised: only X are observed, we need to use X to guess what Y would have been and build a model from there

Semi-supervised: only a small sample of labelled instances are observed but a large set of unlabeled instances

18. What is the difference between regression and classification? Give examples of problems for each type.

Regression: Y is continuous/numerical (predict the value of a share on the stock market, predict the temperature).

Classification: Y is categorical (predict if an event will happen, eg. is this email spam or not, will it be cloudy, rainy or sunny)

19. What are association rules, and how they differ from decision rules?

Association rules are rules that tell us how some "event" is associated with another (how some X is associated with some Y).

A decision rule is a simple IF-THEN statement consisting of a condition and a prediction.

20. What are outliers in ML?

A data object that does not comply with the general behavior of the data. It can be noise or an exception.

21. Contrast two different views on ML: as optimization and as search.

Usually the goal of classification is to minimize the test error. Therefore, many learning algorithms solve optimization problems.

Optimization: objective is to minimize test error (optimize cost function)

Search: find parameters that describe our f(X) = y best

22. Describe different properties of ML models: bias, variance, generalization, hypothesis language.

Bias refers to the error that is introduced by modeling a real life problem by a much simpler problem. The more flexible/-complex a method is, the **less** bias it will have

Variance refers to how much your estimate for f would change if you had a different training data set. The more flexible the method is, the **more** variance it has.

Generalization describes how well our method works on new unseen data (aka test data).

Hypothesis language describes the hypotheses which machine learning system outputs

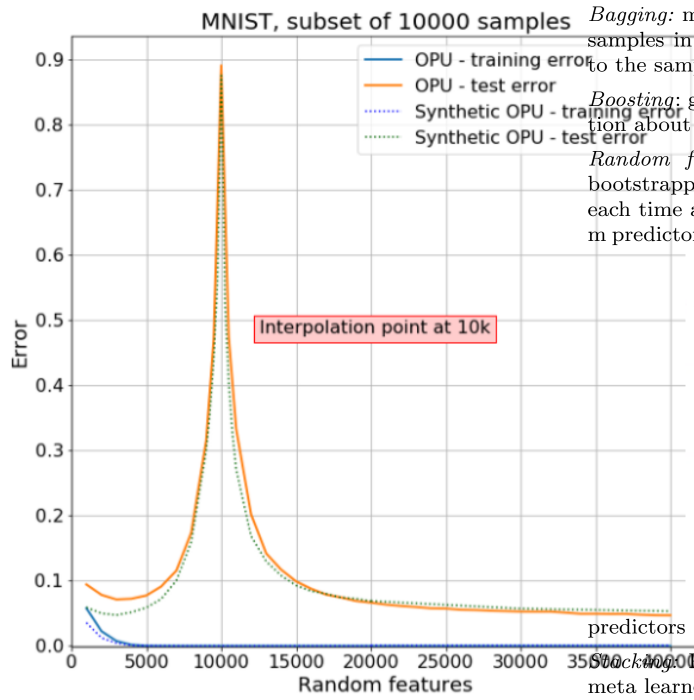
23. What is the bias-variance trade-off in ML?

If we have too much bias, we won't have a lot of variance giving us a very inflexible method that doesn't predict well. If we have too much variance, the model could overfit to the training data and will not work well with new unseen data. In both cases the error of prediction will be high, so we want to find that sweet spot where we minimize the error rate, but don't overfit.

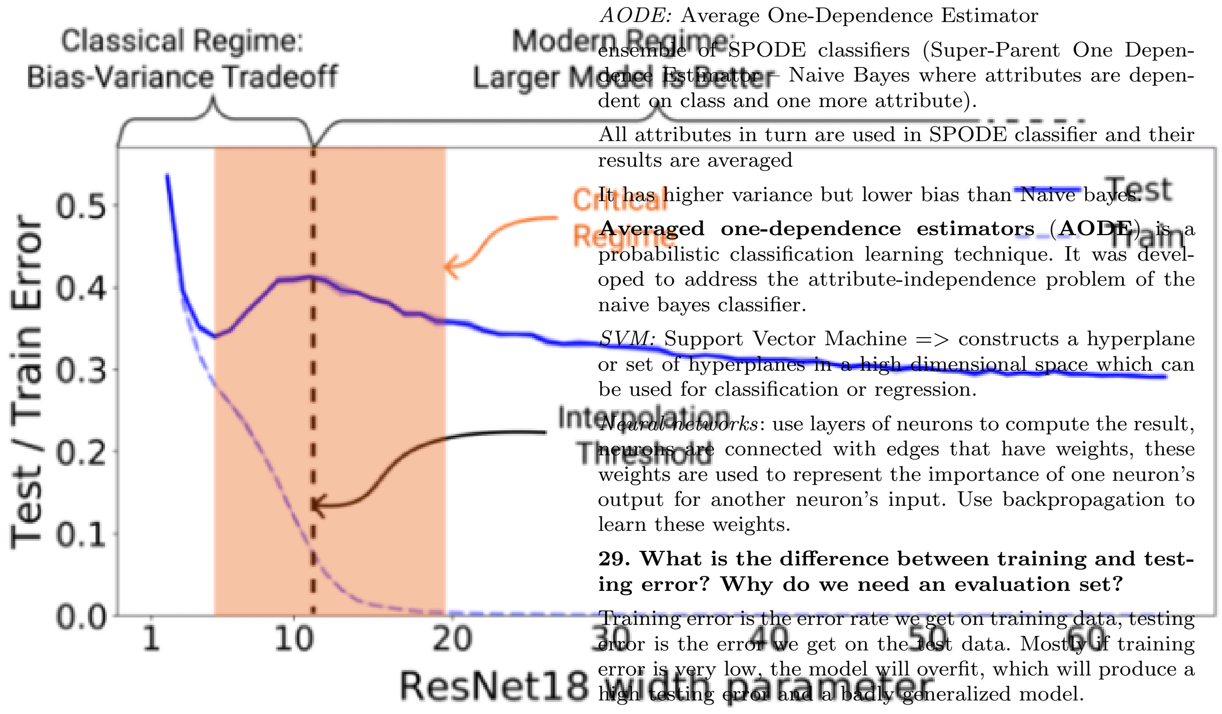
24. Describe the double descent concerning bias-variance trade-off.

For every model there is a spot in how much data we use that will have a very bad error rate. (eg. a model can predict well on the test and train set for 5000 samples and predict very poorly for 7500, but predict very well for 10000 again)

This is observed only in neural networks (and random forests(?)). Other models observe the "classic" overfitting phenomenon.



When model complexity keeps increasing, the testing error first starts decreasing due to the adaptation of model parameters to the data features. After the sweet spot that is proposed by the classical wisdom, testing error starts rising and generalization keeps worsening. However, after the complexity exceeds the interpolation threshold, the mystery happens. As long as we keep increasing the model complexity, test error keep decreasing and after certain complexity, the testing error start to be smaller than the sweet pot that we get within the <u>under-parameterization regime</u>



25. Describe bias-variance trade-off in relation to kNN classifier.

Variance generally decreases with increasing k, bias increases with increasing k

26. Describe methods that can speed-up the kNN algorithm: k-d trees, R-trees, RKD-tree, locally sensitive hashing, and hierarchical k-means.

- *k-d* trees are a generalization of BST, where each node holds a vector instead of a single value. Before building a tree we must normalize values to the interval [0,1], and we split each node on dimension so that we maximize variance in that dimension, and we use the median of that dimension as a splitting value. Leaves usually hold multiple values.
- *R-trees* are similar to k-d trees but are generalization of B-trees.
- *RKD-trees* are multiple trees where we split on random dimensions from a set of dimensions with highest variance. If the probability of not finding nearest neighbor in the single tree is p then with m trees is pˆm
- *Local sensitive hashing*: we have multiple hash tables with multiple hash functions, near instances are also near when hashed (hashing with random hyperplanes)
- *Hierarchical k-means*: recursively run k-means clustering, until clusters are small enough

27. What are the Bayes error rate and Bayes optimal classifier?

Bayes error rate refers to the lowest possible error rate that could be achieved if somehow we knew exactly what the "true" probability distribution of the data looked like.

Bayes optimal classifier for new x0 returns the maximally probable prediction value P(Y=y|X=x0)

28. Describe properties of the following models: kNN, decision rules, bagging, boosting, random forests, stacking, AODE, MARS, SVM, neural networks.

kNN: represent the data in a 2D/3D. . . space and compute distances between different data samples, use these distances to find the k nearest neighbors to our input x0 and classify x0 as the majority class of these k instances.

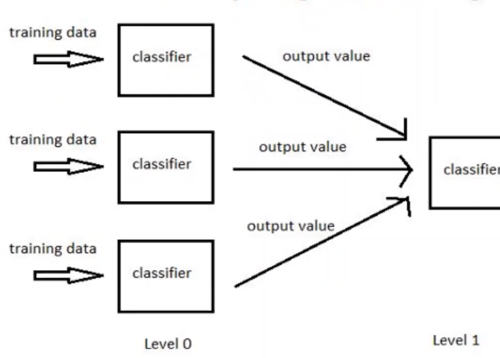
Decision rules: is a function which maps an observation to an appropriate action.

Bagging: make different bags for each classifier and put data samples in them, classify new data sample by comparing it to the samples in the bags

Boosting: grows tree sequentially => each tree uses information about errors of previous trees, weak learners ensemble

Random forests: build an number of decision trees on bootstrapped training sample, but when building these trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates form the full set of p predictors

Concept Diagram of Stacking



Stacking: Predictions of base learners are used as input for meta learner (shitty neural networks).

Method to combine heterogeneous predictors.

Predictions of base learners are used as input for meta learner.

MARS: Multivariate Adaptive Regression Splines

Generalization of stepwise Linear regression.

Not tree-based. Adds one variable at the time (sees which 1 is the best).

It is a non-parametric regression technique and can be seen as an extension of linear models that automatically models nonlinearities and interactions between variables.

AODE: Average One-Dependence Estimator

ensemble of SPODE classifiers (Super-Parent One Dependence Estimator - Naive Bayes where attributes are dependent on class and one more attribute).

All attributes in turn are used in SPODE classifier and their results are averaged

It has higher variance but lower bias than Naive bayes.

Averaged one-dependence estimators (AODE) is a probabilistic classification learning technique. It was developed to address the attribute-independence problem of the naive bayes classifier.

SVM: Support Vector Machine => constructs a hyperplane or set of hyperplanes in a high-dimensional space which can be used for classification or regression.

Neural networks: use layers of neurons to compute the result, neurons are connected with edges that have weights, these weights are used to represent the importance of one neuron's output for another neuron's input. Use backpropagation to learn these weights.

29. What is the difference between training and testing error? Why do we need an evaluation set?

Training error is the error rate we get on training data, testing error is the error we get on the test data. Mostly if training error is very low, the model will overfit, which will produce a high testing error and a badly generalized model.

We need the evaluation set to test our model on previously unseen data and see if we overfitted it.

30. Describe the properties and purpose of evaluation with cross-validation. Describe different biases of ML models stemming from data: reporting bias, automation bias, selection bias, group attribution bias, implicit bias.

Cross-validation: when we don't have enough data to split (or we don't want to split), we make k splits and build a model for each subset and test it on remaining data. Every instance is used for testing once and we get a general idea of model accuracy on that data.

Reporting bias: frequency of data is not real world frequency (people review only if they have extreme opinions . . .)

Automation bias: model is actually not better than human performance (but you love ML and you want to use it . . .)

Selection bias: data sets are not representatively selected (interview only friends and family, even selecting complete strangers we have some bias in selection)

Group attribution bias: is a tendency to generalize what is true of individuals to an entire group to which they belong. (you went to FRI and generalize that all are good students . . .)

Implicit bias: occurs when assumptions are made based on one's own mental models and

personal experiences that do not necessarily apply more generally. (i think, so it must be true)

31. What is the no-free-lunch theorem?

~~Nothing is free, if we want an algorithm to work faster we need to either change it in some way or get more computational power (upgrade computer)~~ ˜ don't know about this tho ...

No universal algorithm is the best algorithm. (we cannot say SVM is better than RF, we cannot mathematically prove that)

There cannot be a single best algorithm for every ML situation.

32. Describe three types of feature selection methods: filter, wrapper, and embedded methods. What are the main differences between them?

Filter methods: independent of learning algorithm, select the most discriminative features through a criterion based on the character of data (information gain, ReliefF)

Wrapper: use the intended learning algorithm to evaluate the features (eg. progressively add features to SVM while performance increases)

Embedded: select features in the process of learning (ridge, lasso)

https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

33. Describe the difference between impurity based and context-sensitive attribute evaluation.

Impurity based: assume <u>conditional independence between the attributes</u> (information gain, Gini index, MDL, distance measure, MSE, MAE (mean absolute error))

Context sensitive measures: contrary (Relief, Contextual Merit). Random forest or boosting based attribute evaluation.

34. Describe the main ideas of information gain and ReliefF evaluation measure.

Information gain: measure (im)purity (entropy) of labels before and after the split

IG(A) = H(T) - H(T|A).
H . . . Information entropy

H(T|A) . . . conditional entropy

Assumes attributes are independent.

ReliefF: criterion: evaluate attribute according to its power of separation between near instances. Increases/decreases worth of feature(s) when comparing the (dis)similarity between random nearby examples (based on certain attribute). Nearest k hits.

35. Explain how regularization can be used as a feature selection method?

(Regularization is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting.)

Example → Lasso (L1) regression . . . attributes (parameters in linear regression) will be set to 0 if they are useless . . .

36. Describe ridge regression (L2) and lasso (L1) and the difference between them?

https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c

The key difference between them is the penalty term.

Lasso → L1 type regularization, which means that it does not square the size of the attribute parameter. It only sums up the sizes and adds it to the error estimation. It will automatically converge these parameters to zero, if they don't contribute to the prediction.

In other words, if the parameter does not contribute to the prediction, it will be set to 0.

Ridge → L2 regularization, sum of square of parameters is added to error estimation (e.g. to RMSE). This is called penalty and is weighted (in L1 also) with the lambda parameter. We don't want our parameters to be huge because that leads to overfitting to train data.

The "dummy parameters" will be close to 0, but not equal to 0. This makes L2 reg. "useless" for feature extraction. We use L1 for that.

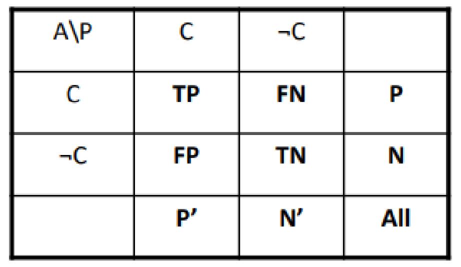
37. What are the advantages and disadvantages of the wrapper method for feature selection?

Forward selection, effective for a given learning model.

High computational load, attention to data overfitting. Evaluating prediction models needs to be a separate evaluation set.

38. Describe the confusion matrix and evaluation measures based on it?

The confusion matrix represents how data was classified by our classifier, compared to observed data.



A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

TP: true positive, values that the model predicted as positive and are observed to be positive

FP: false positive, values that the model predicted as positive and are observed to be negative

FN: false negative, values that the model predicted as negative and are observed to be positive

TN: true negative, values that the model predicted as negative and are observed to be negative

39. Describe ROC curves, sensitivity, specificity, precision, recall, F-measure, classification accuracy, mean squared error.

Classification accuracy: (TP + TN)/(TP + TN + FP + FN) => how accurate is the model

Precision: TP / (TP + FP) => what % of tuples that the classifier labeled as positive are actually positive

Recall: TP / (TP + FN) => what % of positive tuples did the classifier label as positive

sensitivity : TP/P => true positive recognition rate

Specificity: TN/N => true negative recognition rate

ROC curve: shows both y=TP and x=FP rate simultaneously, to summarize overall performance we also use area under the ROC curve (AUC), the larger AUC is, better the classifier

F-measure: => harmonic mean of precision and recall

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

where the one Yi is observed Yi and the second is predicted

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

40. What are the ideas of unsupervised and semi-supervised feature selection?

Semi-supervised: Typically a small sample of labelled and a large sample of unlabeled data is available. Use the label information of labeled data and data distribution or local structure of both labeled and unlabeled data to evaluate feature relevance

Unsupervised: criterion: preserve similarity between instances.

eigenvalues of L (<u>laplacian matrix</u>) measure the separability of the components of the graph and the eigenvectors are the corresponding soft cluster indicators

Or

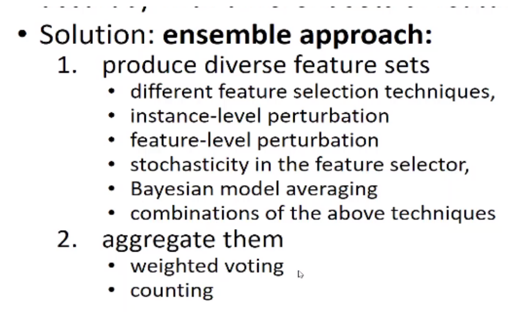
With <u>clustering</u>

41. How can we increase the stability of feature selection?

We can use an ensemble approach to:

1. Produce diverse feature sets

2. Then aggregate them



- Solution: **ensemble approach:**
  1. produce diverse feature sets
     - different feature selection techniques,
     - instance-level perturbation
     - feature-level perturbation
     - stochasticity in the feature selector,
     - Bayesian model averaging
     - combinations of the above techniques
  2. aggregate them
     - weighted voting
     - counting

42. Describe the main ideas of multi-view, multi-label, and multitask learning.

Multi-view: information from different sources, some measurements are irrelevant, noisy or conflicting. Different views typically provide complementary information.

Approaches:

- Baseline: concatenate all views
- Construct <u>tensor space</u> from views
- Relief like approach (different views contribute to the distances between objects)
- Multi-view clustering & feature selection

Multi-Label: Each instance may have more than one label

Approaches:

- transform to single label case
- Treat multiple labels directly
- Relief like approach (comparing sets of instance labels)

Multitask: learn several related tasks simultaneously with the same model. They share knowledge representation. Prevents overfitting.

43. What do online learning and online feature selection mean?

Online feature selection: in data stream scenario, instances arrive sequentially, potentially the learned concept changes, new features may appear

Online learning: same as above but for learning

44. Explain the main ideas of ensemble methods in ML, why and when they work?

Learn a large number of basic (simple) classifiers and merge the predictions. We need different weak classifiers (in the sense that they produce correct predictions on different instances), the law of large numbers does the rest

45. Explain the main differences between bagging and random forests?

Bootstrap **agg**regat**ing** (**Bagging**) is a procedure, where we take a training set D and create new subsets D_i by subsampling from D uniformly and with replacement (every instance has the same chance of being chosen and can be chosen multiple times). That way we will have about 1 - 1/e (63.2%) of unique instances in each subset D_i.

Averaging reduces variance. -> "Given a set of n independent observations Z1, ..., Zn, each with variance σ^2 , the variance of the mean Z of the observations is given by $\sigma^{2/n}$

RF expands on this idea by constructing a multitude (set aka množica) of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. But instead of using D_i to construct our trees we also use bagging to select a subset of m features:

$$m \approx \sqrt{p} \text{ or } m \approx 1 + \log_2 p$$

!!!not sure if we use bagging to select instances at the beginning and then just subsample the features for each tree or we also use bagging of features for each tree separately!!!

Old answer:

Random forests de-correlate the trees. In RF only a subset of features are selected at random out of the total and the best split feature from the subset is used to split each node in a tree. Bagging all features are considered for splitting a node.

46. What is the out-of-bag error estimation?

OOB error is the mean prediction error on each training sample x_i, using only the trees that did not have x_i in their bootstrap sample.

Since bootstrapping involves random selection of subsets of observations to build a training data set, then the remaining (36.8%) non-selected part could be the testing data.

47. How can one evaluate attributes with random forests or produce a similarity matrix?

Evaluation of attribute A is the difference between:

- Strength of the forest
- Strength of the forest when values of A are randomly shuffled

When two instances end in the same leaf of the tree we increase their similarity score, average over all trees gives <u>similarity measure</u> => similarity matrix

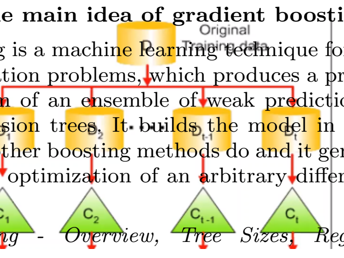
48. Describe the main parameters of random forests and boosting?

Boosting: B=number of trees, ƛ=the shrinkage parameter, a small positive number(small ƛrequires large B to work well), d=number of splits in each tree

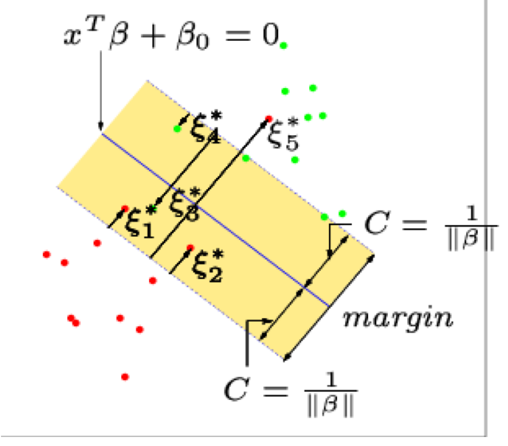
RF: B=number of trees, m = number of features to subsample for every split

49. Describe the main idea of gradient boosting?

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stagewise fashion like other boosting methods do and it generalizes them by allowing optimization of an arbitrary differentiable loss function.



Gradient Boosting - Overview, Tree Sizes, Regularization



50. Describe the notion of margin in kernel methods.

Suppose we have two class data, that can be separated with a straight line. We would like all the points to be as far from the line as possible (and on the correct side).

C is the minimum distance between each point and the separating line. **C= 1/ |b|** where b's are the parameter of the model. Margin is the area around the separating line that has width of 2C. We do not want points inside the margin. This is why we tune C such that the sum of all errors * 1/C will be smaller than some con7stant.

51. What is the purpose of different kernels (linear, polynomial, RBF) in SVM?

Linear: trivial

Polynomial: we allow SVM to produce a non-linear decision boundary.

Radial basis function (RBF):

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

Euclidean distance divided by a free parameter sigmaˆ2

Since the value of the RBF kernel decreases with distance and ranges between zero (in the limit) and one (when x = x'), it has a ready interpretation as a <u>similarity measure</u>

52. Describe how to use SVM for more than two classes?

One versus All: build k different models (k=number of classes) and classify an example to the class that gives highest probability.
One versus One: fit (k 2) models (every possible pair) and classify to the class that wins most pairwise competitions.

Choose 1v1 if k is small enough.

53. Describe different activation functions in neural networks (NNs).

Activation functions are mathematical equations that determine the output of a neural network.

Step functions: <u>f(x) = 1 if x > 0 else 0</u>

ReLU (Rectified Linear Unit): <u>f(x) = max(0, x)</u>

Sigmoid function → $$S(x) = \frac{1}{1+e^{-x}}$$

Softplus: f(x) = ln(1+eˆx) … approximation of ReLU

54. Describe the main idea of backpropagation learning for NNs.

- Initialize the weights to small random numbers, associated with biases.
- Propagate the inputs forward (using activation functions)
- Backpropagate the error (by updating the weights and biases)

What is backpropagation really doing? | Deep learning, chapter 3

55. Describe the role of criterion (loss) function in NN?

$$C = -\sum_j t_j \log y_j$$

target value

$$\frac{\partial C}{\partial z_i} = \sum_j \frac{\partial C}{\partial y_j}\frac{\partial y_j}{\partial z_i} = y_i - t_i$$

To see how much we missed in classifying an input. We use this to backpropagate and improve the network.

If we have a scalar output, we use criterion function to see where we made mistakes. We frequently use cross entropy as cost function C.

56. Describe the strengths and weaknesses of NN?

Weaknesses: long training time, require a number of parameters determined empirically, poor interpretability, overfitting is a usual, gradient based BP we have no guarantee of reaching the global optimum.

Strengths: high tolerance to noisy data, ability to classify untrained patterns, well suited for continuous valued inputs and outputs, algorithms are inherently parallel, successful on an array of real-world data. Can closely approximate any function.

57. Describe a few techniques for overfitting prevention in NNs.

Weight decay: over time if weights haven't been updated in a while, slowly decrement them and set them to 0.

Weight sharing: not all connections have unique weights, they are shared among connections.

Early stopping: stop before we reach a too high classification accuracy. Need a separate evaluation set.

Model averaging: train multiple models and average the weights to use on the final model. "Ensembling" (not a good option -> even 1 NN takes a lot of time to train)

Drop out: randomly (with some probability) drop a node, when training.

Generative pre-training: …

Bayesian fitting: not useable, too slow, complex

58. What are deep neural networks? What are their main strengths and weaknesses?

Deep neural networks are NNs with more than one hidden layer. They perform nonlinear regression (from a statistical point of view).

STRENGTHS: very powerful, high tolerance to noisy data, ability to classify untrained patterns, well suited for continuous valued inputs and outputs, algorithms are inherently parallel, successful on an array of real-world data

WEAKNESSES: long training time, poor interpretability, overfitting is as usual, requires a lot of data ...

59. What are the recurrent networks?

Is a NN where neurons are also connected backwards (backwards connections between neurons). One's output is the input back to it's parent(s).

Used in the text/signal/image processing. Learning is harder, unreliable gradients, they disappear faster. They are getting "dropped".

60. Describe the convolutional neural networks (CNN).

A class of deep NN, most commonly applied to analyzing visual imagery and language.

CNN were inspired by biological processes in that the connectivity between neurons resembles the organization of the animal visual cortex.

Idea: many copies of small detectors used all over the image.

It uses pooling and convolution. They learn filters/detectors and combinations to recognize some items (dots, edges, for example).

Not fully connected -> lighter model

61. Describe different components of CNNs.

Convolutional layer: The convolutional layer consists of a set of filters. (Each filter covers a spatially small portion of the input data). The network will learn filters that activate when they see some specific type of feature at some spatial position in the input.

Convolving the filter == dot product between filter and the input.

On this layer we have local connectivity and shared weights.

Pooling layer: Progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Speeds up learning and reduces overfitting.

Pooling partitions the input image into a set of non overlapping rectangles and, for each such sub region, outputs the maximum (minimum or average) value of the features in that region.

Problem: after several layers we lose the information about the exact location of the recognized pattern. E.g. nose on the forehead.

62. What are the advantages and disadvantages of CNNs?

Advantages: automatically detects important features without human supervision, lighter model (not fully connected layers -> less weights), free translation of variance, fewer parameters take less space -> can be computed in a memory of a GPU (or across CPUs).

Disadvantages: high computational cost, need a lot of training data!

63. What is 1d and 2d convolution?

1d is convolution over 1 dimension and is used for convolution on words, characters, lemmas,..

2d is convolution over 2 dimensions and is used for convolution on images/ text classification.

64. Describe the main idea and components of autoencoders?

Autoencoders are designed to reproduce their input (especially for images). They compress input into a latent-space of usually smaller dimension. Then they reconstruct the input from the latent space (even without the noise).

Encoder: compress input into a latent space of usually smaller dimension. h = f(x)

Decoder: reconstruct input from the latent space. r = g(f(x)) with r as close to x as possible

65. What is denoising an autoencoder?

Get a clean image as input, apply some noise to it and train the autoencoder to reproduce the clean image.

66. Describe the main idea and components of the generative adversarial networks?

Two neural networks contest with each other in a game (in the form of a zero sum game). Use one neural network to generate data for the second neural network to use as input and have the first NN try to "fool" the second one into misclassifying the input.

Generator: generate fake samples, tries to fool the Discriminator

Discriminator: tries to distinguish between real and fake samples

Training means improving G and D.

67. Describe different inference methods for predictive methods.

15.

68. Describe different techniques for the explanation of predictions.

((this one might be wrong, not sure))

Domain level: try to explain the "true causes and effects". Usually unreachable except for artificial problems with known relations (if we can test it with result functions).

Not applicable especially in medicine, business...

Model-based: Make the prediction process of a particular model transparent. Better models enable better explanation at the domain level.

Instance-level: explain predictions for each instance separately (model-based). Nomograms

(For titanic, we would have a separate nomogram for each person. We average them at the end)

Model-level: the overall picture of a problem the model conveys (model-based). Averaged instance-level models.

Model agnostic: Can be applied to any model. change one input to our black box and see if the output changes significantly. This means that that input is important. (perturbation-based explanation).

Won't work well for images. For that we use:

Model-specific explanation technique

Method EXPLAIN: Hide one attribute at a time.

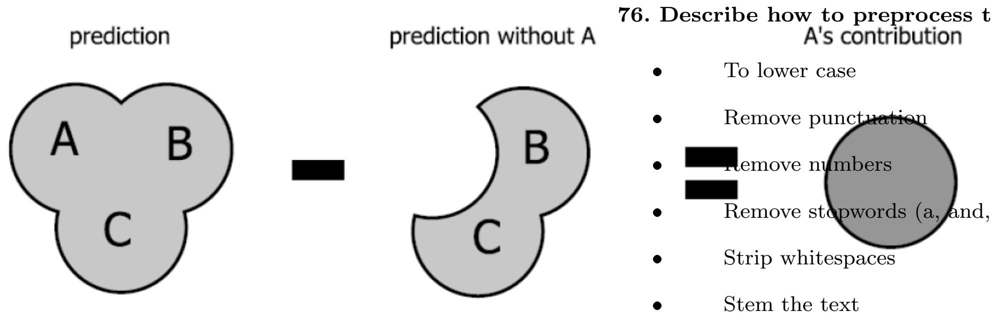
Weakness: there might be 2 attributes needed to be absent at the same time to see the importance of the 3rd.

69. What is the role of clustering in interpretability?

Clustering is useful in supervised tasks to get insight into the relation between predicted values Y and basic groups in the data.

70. Describe the main idea of perturbation-based explanation methods?

Importance of a feature or a group of features in a specific model can be estimated by simulating lack of knowledge about the values of the feature or randomly shuffling them to test its importance



71. Explain the difference between instance-based and model-based explanations?

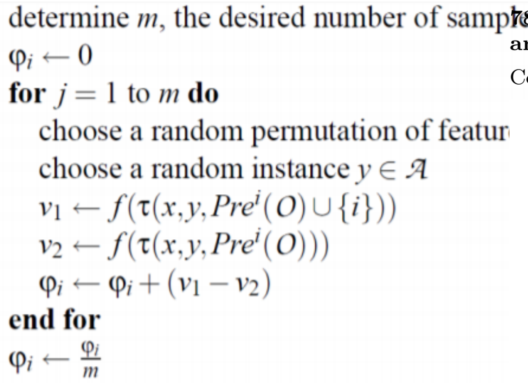
Model based tries to paint the whole picture, while instance based only explains the instances separately

Model based: Make the prediction process transparent of a particular model. Explanation is independent of the accuracy of a model. ← this is what knowledge extractors are interested in (the overall picture of a problem the model conveys).

Instance based: Explain predictions for each instance separately (presentation format: impact of each feature on the prediction value). ← this is what practitioners applying models are interested in.

72. Explain the main idea of the IME, LIME, and SHAP explanation technique?

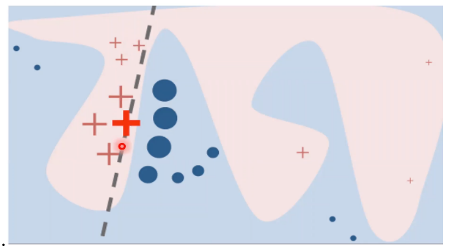
IME: Interactions-based Method for Explanation, the feature gets some credit for standalone contributions and for contributions in interactions



```
determine $m$, the desired number of samp
$\varphi_i \leftarrow 0$
for $j = 1$ to $m$ do
    choose a random permutation of featur
    choose a random instance $y \in \mathcal{A}$
    $v_1 \leftarrow f(\tau(x, y, Pre^i(\mathcal{O}) \cup \{i\}))$
    $v_2 \leftarrow f(\tau(x, y, Pre^i(\mathcal{O})))$
    $\varphi_i \leftarrow \varphi_i + (v_1 - v_2)$
end for
$\varphi_i \leftarrow \frac{\varphi_i}{m}$
```

- Alternative formulation of shapley value
- "hide" any subset of attributes at a time (2 a subsets!)
- the feature gets some credit for standalone contributions and for contributions in interactions

LIME: Local Interpretable Model-agnostic Explanations, perturbations in the locality of an explained instance.



- Faster than IME, works for many features (text and images)
- No guarantees that the explanations are faithful and stable.
- Neighborhood based: curse of dimensionality
- may not detect interactions due to simple interpretable local model (linear)

SHAP: SHapley Additive exPlanation, unification of several explanation methods, including IME and LIME

(faster than IME but still uses linear model with all its strengths and weaknesses)

Natural language processing (NLP)

73. What is the Turing test?

The turing test is a test where a human is communicating with two other agents over a computer, one of them another human, the other an AI. The test tests, if the AI is smart enough to fool the human communicating with it, that it is also human.

74. What is the micro-world approach to NLP?

Create a "world" out of data to analyze. Most text data cannot be directly processed, so we have to create our own world, where we can process data.

75. Describe the stages of linguistic analysis?

Prosody: the patterns of stress and intonation in a language

Phonology: systems of sounds and relationships among the speech sounds that constitute the fundamental components of a language

Morphology: the amissible arrangement of sounds in words: how to form words, prefixes and suffixes

Syntax: the arrangement of words and phrases to create well-formed sentences in a language

Semantics: the meaning of a word, phrase, sentence or text

Pragmatics: language in use and the context in which it is used, including such matters as deixis, taking turns in conversation, text organization, presupposition and implicature

Knowing the world: knowledge of the physical world, humans, society, intentions in communications

76. Describe how to preprocess text in text mining.

- To lower case
- Remove punctuation
- Remove numbers
- Remove stopwords (a, and, the, of,...)
- Strip whitespaces
- Stem the text

77. Describe lemmatization, stemming, POS tagging, dependency parsing, and named entity recognition.

Lemmatization: the process of grouping together the different inflected forms of a word so they can be analyzed as a single item

Stemming: reduce the words to their root "state" (it is getting out of use.)

https://blog.bitext.com/what-is-the-difference-between-stemming-and-lemmatization/

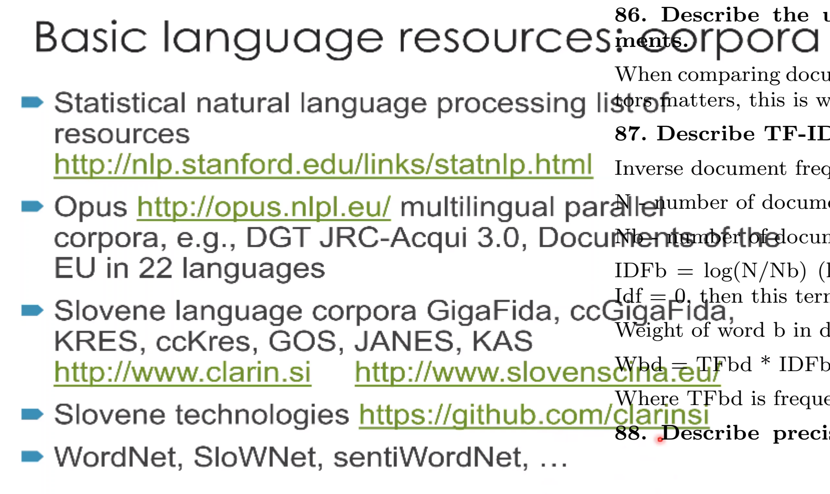
POS tagging: assigning the correct part of speech (noun, verb, subject, object,...) to words

Named entity recognition: seeks to locate and classify named entities mentioned in unstructured text into predefined categories such as person names, organizations, locations, medical codes,...

Dependency parsing: find connections (dependencies) between words

78. Describe the basic language resources for English and Slovene (or your language).

Corpora, wiki, SSKJ, FRAN



Basic language resources: corpora

- Statistical natural language processing list of resources http://nlp.stanford.edu/links/statnlp.html
- Opus http://opus.nlpl.eu/ multilingual parallel corpora, e.g., DGT JRC-Acqui 3.0, Documents of the EU in 22 languages
- Slovene language corpora GigaFida, ccGigaFida, KRES, ccKres, GOS, JANES, KAS http://www.clarin.si http://www.slovenscina.eu/
- Slovene technologies https://github.com/clarinsi
- WordNet, SloWNet, sentiWordNet, …

79. Describe the structure of WordNets.

WordNet is a database composed of synsets (cognitive synonyms):

1. Synonyms
2. Hypernyms
3. Hyponyms
4. Meronyms
5. holonyms
6. Etc.

https://wordnet.princeton.edu/ (maybe this will help)

WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations.

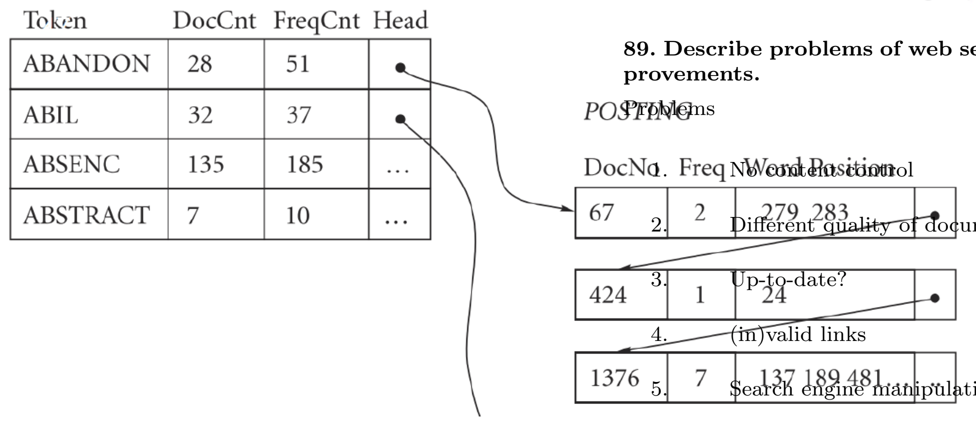
80. Describe approaches to document retrieval.

Historically people *used keywords,* but *today full text search* is used (by the help of organized databases, indexing and good searching algorithms)

81. Describe the inverted file index.

Is a data structure that maps words to documents.

Inverted file index means that we have a database where for every word we store in how many documents it appeared and the overall number of appearances. Then it has a pointer to the document where we can find the location of the word in the document.



Token	DocCnt	FreqCnt	Head
ABANDON	28	51	•
ABIL	32	37	•
ABSENC	135	185	…
ABSTRACT	7	10	…

POSTING

DocNo	Freq	Word Position	
67	2	279 283	•
424	1	24	•
1376	7	137 189 481 …	…

82. Compare search with logical operators and ranking based search.

Search with *logical operators* is outdated. It returns a lot of results, we need to write large queries, synonyms are a problem, there is no partial matching and no weighting.

Ranking based search is used nowadays for web search (Yahoo, Google, Bing, ...). Less frequent terms are more informative. It uses vector based representation of documents and queries. For ranking based search we can explain what bag-of-words approach is or dense embeddings.

83. Describe one-hot-encoding and bag-of-words representation.

One-hot-encoding is the vector representation that consists of only 1 bit set to 1 and all other bits to 0. It assures that machine learning does not assume that higher numbers are more important.

Bag of words representation is commonly used in NLP, where a text or a document is represented as the bag of its words and how many times every word appears.

84. Describe how to use term-document and term-term matrix?

Term-document matrix is the matrix where every line is one term and the columns are the documents. Every cell of the matrix shows how many times some term appeared in a document. This matrix is used for **comparison of terms**.

Document-term matrix is the other way around, (basically transposed TDM) and it is used for **comparison of documents**.

Term-term matrix is a matrix where every line is one term and every column is one term. If two terms appear together more often they have a higher score in the matrix.

85. What is word embedding? Which embeddings are sparse and which are dense?

Word Embeddings are dense representations of the individual words in a text, taking into account the context and other surrounding words that that individual word occurs with.

Sparse embeddings: SVD

Dense embeddings are the ones that have less dimension, less space, they capture synonyms better, and reduce noise. We use LSA (latent semantic analysis) for truncating the matrices with eigenvalues.

86. Describe the use of cosine similarity on documents.

When comparing documents only the angle between their vectors matters, this is why cosine similarity is used.

87. Describe TF-IDF weighting.

Inverse document frequency (idf) is equal to:

N - number of documents in collection

Nb - number of documents with word b

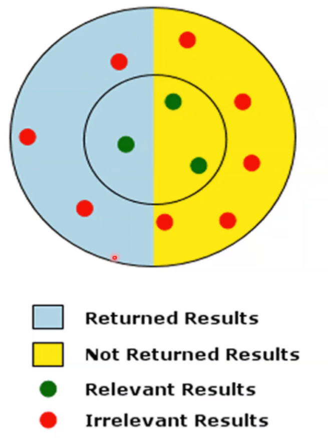
IDFb = log(N/Nb) (lower value == more distinct term. If Idf = 0, then this term is present in every document)

Weight of word b in document d would be equal to :

Wbd = TFbd * IDFbd

Where TFbd is frequency of the term b in document d

88. Describe precision, recall, and F1 measures in document retrieval.

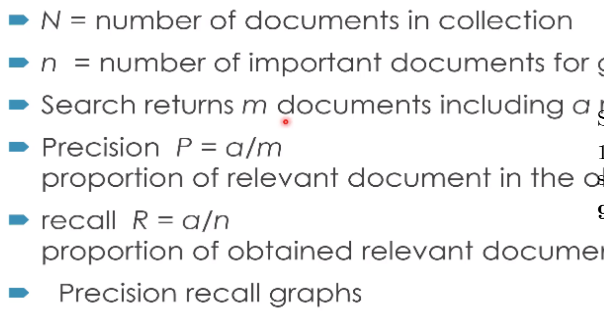


Precision: proportion of relevant documents in the obtained ones

Recall: proportion of obtained relevant documents. How many of the relevant documents we succeeded retrieving.

F1 is just weighted harmonic mean (where beta = 1), weighted precision and recall

(2 * P * R / P + R)



- N = number of documents in collection
- n = number of important documents for given query
- Search returns m documents including a relevant ones
- Precision $P = a/m$ proportion of relevant document in the obtained ones
- recall $R = a/n$ proportion of obtained relevant documents
- Precision recall graphs

89. Describe problems of web search and possible improvements.

Problems

1. No content control
2. Different quality of documents
3. Up-to-date?
4. (in)valid links
5. Search engine manipulation (link farms)

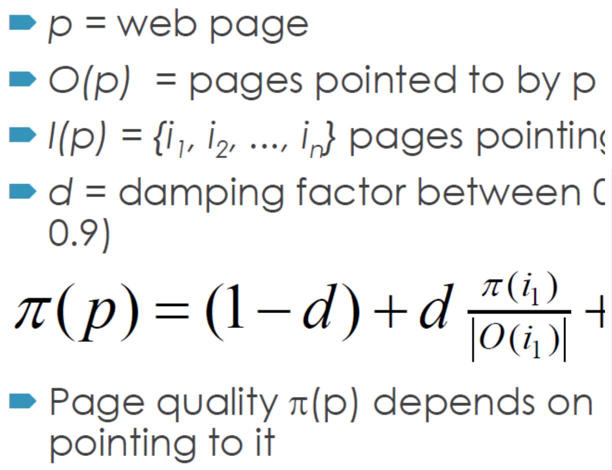
Improvements:

1. Use dictionary, thesaurus (a book that lists words in groups of synonyms and related concepts), synonyms
2. Query expansion with relevance information (user feedback, personalization, trusted document sources)
3. Semantic search
4. Specific types of queries require specific approaches
5. Trustful sources - Wikipedia
6. Hubs with relevant links
7. Graph theory and analysis
8. Additional information: titles, meta-information, URL
9. Ranking of documents based on links

90. Describe the idea of the PageRank algorithm and its possible uses.

Page rank algorithm determines the rank of a page based on the quality and number of pages pointing to it

Possible uses: was used by google to order search results



- p = web page
- $O(p)$ = pages pointed to by p
- $I(p) = \{i_1, i_2, ..., i_n\}$ pages pointin
- d = damping factor between 0.9)

$$\pi(p) = (1-d) + d\,\frac{\pi(i_1)}{|O(i_1)|} +$$

- Page quality π(p) depends on pointing to it

91. Describe the main ideas and implementation of LSA, word2vec, ELMo, and BERT.

LSA: uses term-context matrix, the idea being the words with similar context should be closer. It reduces the dimensionality of the matrix with SVD and uses k most important dimensions to represent the embedding of the words. (basically PCA)

Word2vec: instead of counting how many times a word appears near another word. It trains a classifier to answer that question (for example NN). Then it uses classifiers learned weights as the word embeddings. It doesn't take context into an account. Solution: ELMo and BERT.

ELMo: looks at the entire sentence before assigning each word in it an embedding. ELMo predicts the next word in a sequence of words - a task called Language Modeling (LM). first layers capture morphological and syntactic properties, deeper layers encode semantical properties.

BERT: predicts masked words in a sentence. also predicts order of sentences: is sentence A followed by sentence B or not … train a classifier built on the top layer foreach task that you fine tune for , e.g ., Q&A, NER, inference. achieves state of the art results for many tasks.

Used form: MLM (masked language model) - delete some words in the sentence and try to predict them. Needs context of both sides of the word.

Dominates text classification field.

92. Which are the desired properties of word embeddings?

They shall preserve relations from the original space. We need dense vector embeddings.

- Matrix based transformations to reduce dimensionality (SVD or LSA - latent semantic analysis)
- Neural embeddings (word2vec, Glove)
- Contextual neural embeddings (ELMo, BERT)

SVM, deep NN -> both require numerical input

1-hot-encoding and a bag of words do not preserve semantic similarity.

93. Compare different types of word embeddings.

1. Frequency based Embedding (Count vector, TD-IDF, co-occurrence vector)
2. Prediction based Embedding (Continuous Bag of words, Skip – Gram model)

- Dense vector embeddings
- Neural embeddings
- Diachronic embeddings
- Contextual embeddings
- Cross-lingual embeddings

94. Describe a few relations expressed with modern word embeddings.

Diachronic embedding: comparing words and their neighbours throughout history.

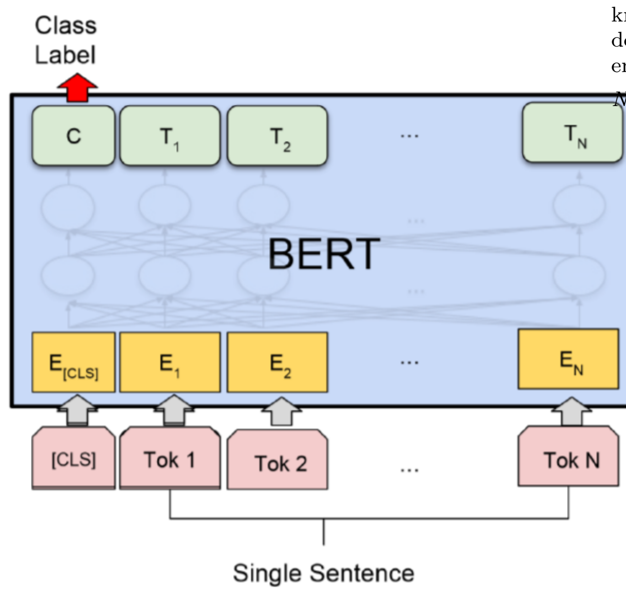
95. What sort of biases are reflected in word embeddings?

Cultural biases, usually negative biases

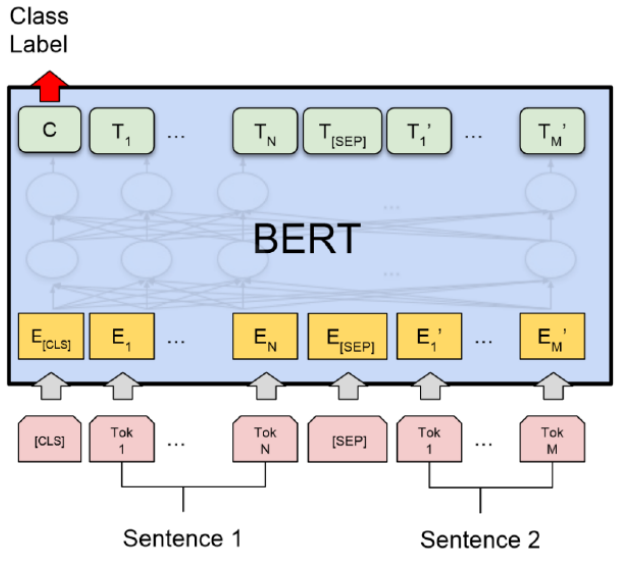
96. How to use BERT and multilingual BERT for text classification?

train a classifier built on the top layer for each task that you fine tune for , e.g ., Q&A, NER, inference.

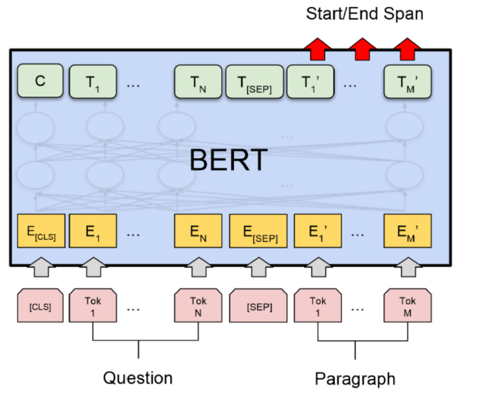
- Sentence classification (sentiment, grammar…):



- Two sentence classification:

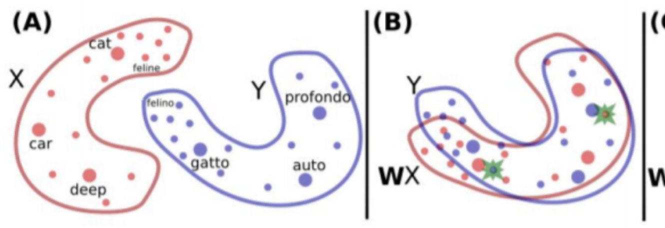


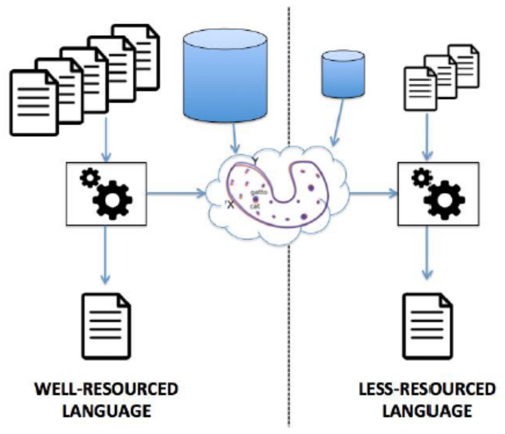
- Questions/answers:



97. Describe the idea and a few uses of cross-lingual embeddings?

Word clouds of different languages can be aligned.





- transfer between languages: models, resources
- embedded words enter neural networks
- replace them with cross lingual embeddings and easily switch languages

98. Describe a few semantic technologies and a few important NLP tasks.

Semantic technologies aka Text mining: to acquire new knowledge. Summarization, document relations, clustering of documents, related news, new topic detection, q&a, named entity recognition, inference, coreference resolution.

NLP tasks:

NLP applications

- document retrieval
- information extraction
- document classification
- document summarization
- sentiment analysis
- text mining
- machine translation,
- language generation

99. How to approach text summarization, sentiment classification, machine translation (MT), or question answering problems?

Text summarization: general, guided (describe in advance what sort of information do you want). One/multi document. Extractive and abstractive (mix 2 words like increase/decrease).

For short text we use abstractive summarization.

For longer texts we use extractive summarization.

Sentiment classification:

Binary, tenary, n-ary

We use lexicon of positive/negative words

Machine learning based.

MT:

With BERT, RNN, Encoder-Decoders, NMT(neural machine translation)

100. What are the language model and translation model in MT?

Language model: each target (English) sentence e is assigned a probability p(e). Estimation of probabilities for the whole sentences is not possible (why?), therefore we use language models, e.g., 3 gram models or neural language models.

Translation model: We have to assign a probability of p(f|e), which is a probability of a foreign language sentence f, given target sentence e. We search the e which maximizes p(e) * p(f|e). We take into account the position of a word and how many words are needed to translate a given word.

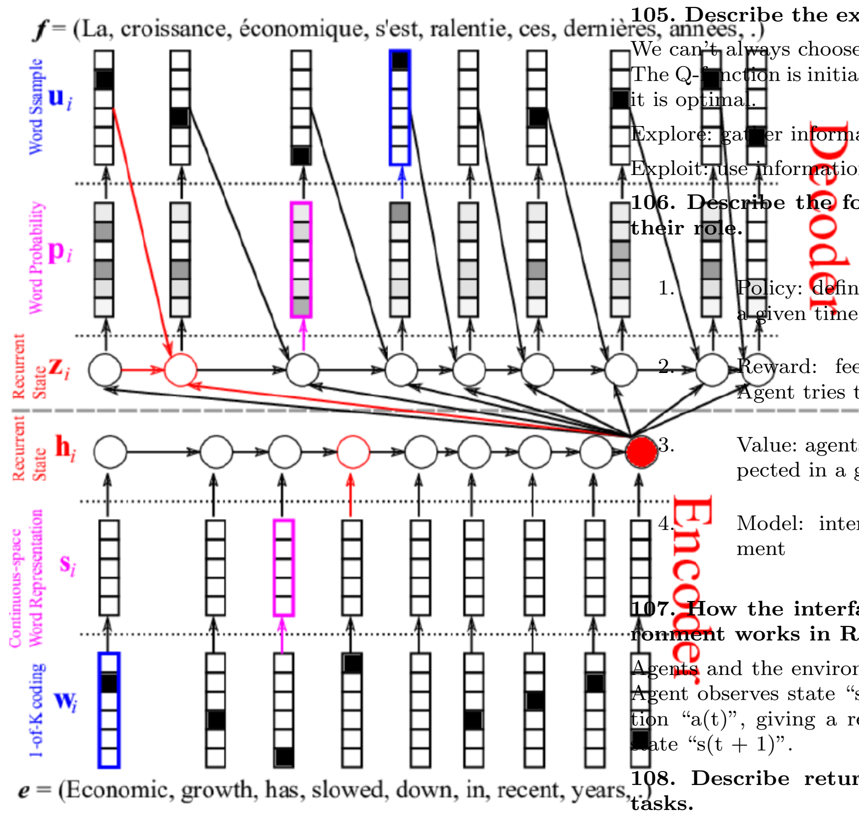
Noisy channel: given sentence e, we transmit it through noisy channel and get a corrupted sentence f. For reconstruction we need 1) how to speak original language (language model p(e)) and 2) how to transform f into e (translation model, p(f|e))

101. What is the encoder-decoder model in NLP?

Encoder: use word representation → word , 1 hot vector, dense embedding, recurrent network

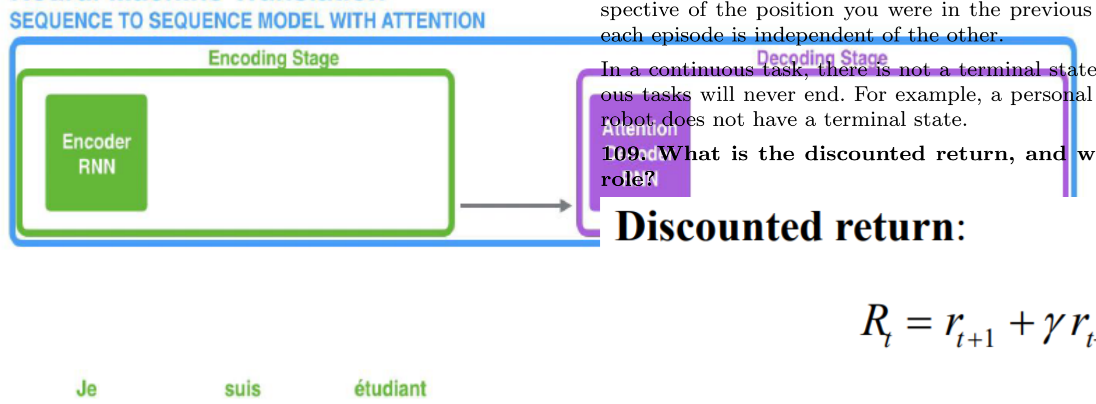
Decoder: computation of the next state of recurrent network, probability of the next word, selection of the next word

Encoder takes a sentence and transforms it into latent vector representation. Decoder takes that latent vector representation and transforms it back into a sentence. Both are language specific.



102. What is the attention mechanism in deep neural networks?

Usually for each word in a sentence a hidden state vector called context is output from an encoder and this vector is fed back into the input and not into the decoder until the end of sentence is detected, then decoder produces output one step at a time. This is problematic for long sentences, this is where the attention mechanism comes in which produces a special context vector for each decoder time step.



Reinforcement learning (RL)

https://www.youtube.com/watch?v=nyjbcRQ-uQ8&list=PLZbbT5o_s2xoWNVdDudn51XM8lOuZ_Njv&index=1&ab_channel=deeplizard

103. Describe when and why to apply RL.

We can use it when we are in an environment where we can afford to make mistakes. When we need to make decisions in *an uncertain environment.*

Why?: simple algorithms, works most of the time, no need to label the data (it takes a lot of time, money or it is just hard to - label regions of objects in 15 million images).

104. What are the differences between supervised learning and RL?

You don't get examples of correct answers, you have to try things in order to learn.

105. Describe the explore or exploit dilemma in RL?

We can't always choose the action with the highest Q-value. The Q-function is initially unreliable, we need to explore until it is optimal.

Explore: gather information from environment

Exploit: use information to make better decision

106. Describe the four main components of RL and their role.

1. Policy: defines agents choices and actions in a given time
2. Reward: feedback from the environment. Agent tries to maximize it
3. Value: agents expectation of what can be expected in a given state (it predicts rewards)
4. Model: internal representation of environment

107. How the interface between the agent and environment works in RL?

Agents and the environment interact at discrete time steps. Agent observes state "s(t)" at step "t" and produces an action "a(t)", giving a resulting reward "r(t + 1)" and next state "s(t + 1)".

108. Describe returns for episodic and continuing tasks.

Episodic: interaction breaks naturally into episodes (eg. plays of a game, trips through a maze)

Continuing: interaction does not have natural episodes.

IN OTHER WORDS ...

Episodic tasks are the tasks that have a terminal state (end). In RL, episodes are considered agent-environment interactions from initial to final states. For example, in a car racing video game, you start the game (initial state) and play the game until it is over (final state). This is called an episode. Once the game is over, you start the next episode by restarting the game, and you will begin from the initial state irrespective of the position you were in the previous game. So, each episode is independent of the other.

In a continuous task, there is not a terminal state. Continuous tasks will never end. For example, a personal assistance robot does not have a terminal state.

109. What is the discounted return, and what is its role?

Discounted return:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2$$

where $\gamma, 0 \leq \gamma \leq 1$, is the **discount rate.**

Lambda makes rewards (moves) that are further away less valuable.

110. What is the average reward model, and what are its advantages and disadvantages?

It's a model where the agent optimizes long-term average reward. The downside is that it does not know the difference between near and distant rewards

111. What is the role of Markov property in RL?

If a state summarizes all past sensations so as to retain all "essential" information it has the Markov property.

Used in MDP (Markov decision process) and Bellman Optimality Equation.

Markov property is that the next decision is solely dependent on the current state. All of the states before this one are meaningless for the next decision.

112. Describe the Markov decision problem (MDP).

If a task has the Markov property, it is basically a Markov Decision Process. If state and action sets are finite, it is a finite MDP. To define a finite MDP we need:

- State and action sets
- One step "dynamics" defined by transition probabilities
- Reward probabilities

113. What sort of learning simplifications does MDP allow in RL?

MDP can be solved by linear programming or by a dynamic programming method. MDP is a discrete, stochastic and controlled process. At any given time, the process is in a certain 's' state, and the user can select any 'a' action that is available in the 's' state. The process responds to this action at the next time unit by random moving to a new state s' and giving the user a corresponding reward.

114. Describe the State-value function and action-value functions?

State-value function: the value of a state is the expected return starting from that state, depends on the agents policy

Action-value function: the value of taking an action in a state under policy π is the expected return starting from that state, taking that action and thereafter following π.

$V^{\pi}(s)$ is the state-value function of MDP (Markov D starting from state s following policy π.

In the expression

$$V^{\pi}(s) = E_{\pi}\{G_t | s_t = s\}$$

G_t is the total DISCOUNTED reward from time step t, as opposed to R_t which is an immediate return. Here you are taking the expectation of ALL actions according to the policy π.

$Q^{\pi}(s, a)$ is the action-value function. It is the expected return starting from state s, following policy π, taking action a. It's focusing on the particular action at the particular state.

$$Q^{\pi}(s, a) = E_{\pi}\{G_t | s_t = s, a_t = a\}$$

The relationship between Q^{π} and V^{π} (the value of being in that state) is

$$V^{\pi}(s) = \sum_{a \in A} \pi(a|s) * Q^{\pi}(s, a)$$

115. Describe the Bellman equations and their role in RL?

Bellman eq. give us the ability to calculate all the expected rewards in all states. It is basically n equations with n variables. If we solve them we get an optimal reward for every state we are in. This is how we do RL ...

116. What is the role of the optimal value function and optimal action-value function?

For finite MDP's policies, they can be partially ordered:

$$\pi \geq \pi' \quad \text{if and only if} \quad V^{\pi}(s)$$

This means that there are always one or more policies that are better or equal to all the others. These are optimal policies. Optimal policies share the same state-value function and action-value function.

$$V^{*}(s) = \max_{\pi} V^{\pi}(s) \quad \text{for all} \quad s \in S$$

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \quad \text{for all} \quad s \in S \text{ and } a \in A(s)$$

Basically the optimal value function and the optimal action-value function return the expected return (reward) for following the optimal policy. This also means that they tell us what the optimal action in a state is.

117. How can we get the optimal policy from the optimal action-value function?

The value of a state under an optimal policy must equal the expected return for the best action from that state

$$Q^{*}(s,a) = E\left\{r_{t+1} + \gamma \max_{a'} Q^{*}(s_{t+1}, a') \middle| s_t = s, a_t = a\right\}$$

$$= \sum_{s'} \mathbf{P}^{a}_{ss'} \left[\mathbf{R}^{a}_{ss'} + \gamma \max_{a'} Q^{*}(s', a')\right]$$

Q is the unique solution of this system of nonlinear equations. Once we have Q* we can further calculate the optimal policy by taking the optimal action:

$$\pi^{*}(s) = \arg\max_{a \in A(s)} Q^{*}(s, a)$$

118. How to solve Bellman optimality equations?

Finding an optimal policy by solving the Bellman optimality equation requires the following:

- accurate knowledge of environment dynamics;
- enough space and time to do the computation;
- the Markov property must be true.

^can be done with dynamic programming

We usually have to settle for approximations → Monte Carlo, Value Iteration, Q-learning

119. When and how dynamic programming is used in RL?

We need a complete model of the environment and rewards (state space, action space, transition model).

Idea: start with any policy, then iteratively improve it (calculate V(policy), then improve policy based on that V(policy))

120. Describe policy-value iteration, value iteration, and policy iteration approaches to RL?

Policy iteration: policy_0 → V(policy_0) → policy_1 → V(policy_1) → . . .

V(policy_i) doesn't need to converge, just move policy towards best one

Value iteration:

$$V_{k+1}(s) = \max_{a} \sum_{s'} P^{a}_{ss'} \left[r^{a}_{ss'} + \gamma V_k(s')\right]$$

- use Bellman optimality equation as an update
- Converges to V*

121. Describe the convergence criterion for value iteration.

If the maximum difference between two successive value functions is less than ε, then the value of the greedy policy, (the policy obtained by choosing, in every state, the action that maximizes the estimated discounted reward, using the current estimate of the value function) differs from the value function of the optimal policy by no more than $2\varepsilon\lambda/(1 - \lambda)$ at any state. This is an effective stopping criterion for the algorithm

122. Describe the Monte Carlo approach to RL and when it is used.

We use Monte Carlo methods as an approximation for the optimal policy. We don't need full knowledge of the environment. We only need experience or simulate experience. This method can only be used for episodic tasks. The way it works is by simulating a few paths and then averages all the returns. So to estimate V(s) we average all observed returns in state s.

123. Describe the ε-greedy policy.

- with probability 1 − ε perform the optimal/greedy action
- with probability ε perform a random action
- will keep exploring the environment
- slowly move it towards greedy policy: $\varepsilon \to 0$

We use it in Q-learning as an "explore" method, because we can't always choose the action with the highest Q value. (The Q-function is initially unreliable, we need to explore until optimal!)

124. Describe learning with time differences (TD) in RL?

Previous states receive a portion of the difference to successors. difficult for analysis

For λ=0

$$V(s_t) = V(s_t) + c(V(s_{t+1}) - V(s_t))$$

- c is a parameter, slowly decreasing during learning ensuring convergence
- For lambda > 0, more than just immediate successors are taken into account (speed)

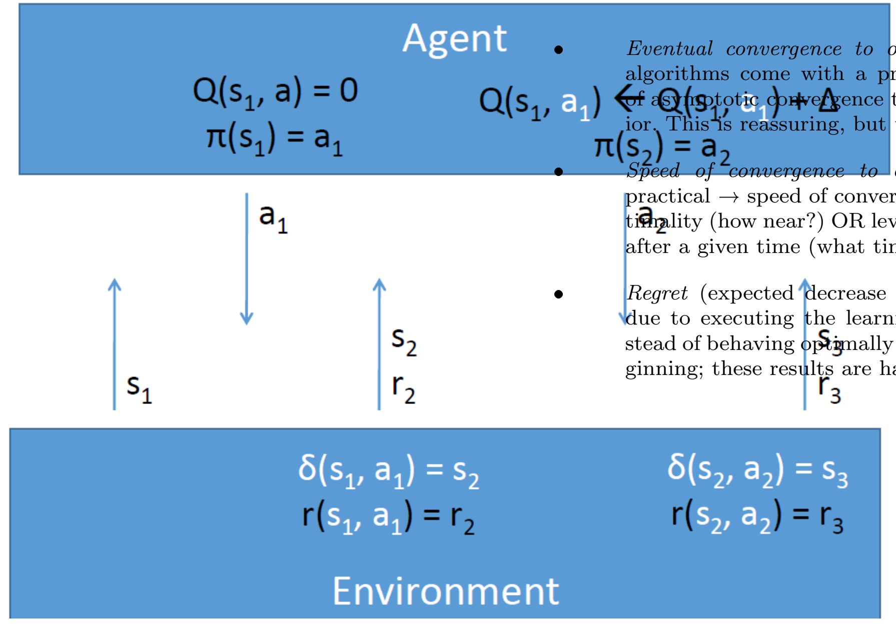
125. Describe the Q-learning.

Works with Q function instead of V function.

Q(s, a) estimates the **discounted cumulative reward** (start in s, take action a, follow the current policy thereafter)

Suppose we have the optimal Q function → optimal policy is argmax_b Q(s, b)

Q-Learning: The Procedure



Pseudo code:

```
Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
    Initialize $s$
    Repeat (for each step of episode):
        Choose $a$ from $s$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
        Take action $a$, observe $r$, $s'$
        $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$
        $s \leftarrow s'$;
    until $s$ is terminal
```

126. What are the updates in Q-learning? How to assure exploration?

assure exploration → epsilon - greedy!

Q-Learning: Updates

- The basic update equation

$$Q(s,a) \longleftarrow r(s,a) + \max_b Q(s',b)$$

- With a discount factor to give later rewa

$$Q(s,a) \longleftarrow r(s,a) + \gamma \max_b Q(s',b)$$

- With a learning rate for non-determinis

$$Q(s,a) \longleftarrow [1-\alpha]Q(s,a) + \alpha[r(s,a)$$

127. How to use function approximation in RL?

Used when in complex environments (Q is too complex), we describe a state with a feature vector. We can then calculate Q as any regression model by using the state feature vectors as its parameters. (<-- e.g.)

??

128. How to measure and compare the learning performance of RL learners?

- *Eventual convergence to optimality (*Many algorithms come with a provable guarantee of asymptotic convergence to optimal behavior. This is reassuring, but useless*)*
- *Speed of convergence to optimality* (more practical → speed of convergence to near optimality (how near?) OR level of performance after a given time (what time?))
- *Regret* (expected decrease in reward gained due to executing the learning algorithm instead of behaving optimally from the very beginning; these results are hard to obtain)