EXAM QUESTIONS

Nine questions will be drawn at random from the questions below for the exam.

Learning and generalization

1. What is "Machine Learning"? Give examples of learning machines.

Machine Learning is a discipline dedicated to the design and study of artificial learning systems, particularly systems that learn from examples. Learning machines include linear models, artificial neural networks, and decisions trees.

2. What is supervised learning? Name special cases of supervised learning depending on whether the inputs/outputs are categorical, ordinal, or continuous.

Supervised learning refers to learning in the presence of a teacher. When trying to learn to classify objects, the teaching signal is the class label. In this class, data objects are represented as vectors "**x**" of variables or "features". We seek to predict an attribute "y" of these data objects, that is another variable. A continuous variable is a real number. Both categorical and ordinal variables takes values from a finite set of choices. For categorical inputs the list is not ordered (e.g. the country of origin) while for ordinal inputs it is ordered (e.g. three clinical stages in the advancement of a disease.) Regardless of the type of inputs, if the output is continuous, the problem is a regression problem; if the output is categorical, the problem is a classification problem. "Ordinal regression" problems have ordinal outputs.

3. What is unsupervised learning? Give examples of unsupervised learning tasks.

In unsupervised learning problems, no teaching signal is available. Dimensionality reduction by principal component analysis and clustering are examples of unsupervised learning.

4. What is a loss function? What is a risk functional? Give examples.

A loss function is a function measuring the discrepancy between a predicted output f(**x**) and the desired outcome y: L(f(**x**), y). The risk is the average of L over many examples. Examples of loss functions include the square loss often used in regression $(y-f(\mathbf{x}))^2$ and the 0/1 loss used in classification, which is 1 in case of error and 0 otherwise.

5. What is the empirical risk? What is "empirical risk minimization"?

The empirical risk is the average loss over a finite number of given examples. Empirical risk minimization refers to finding the function f(**x**) in a family of functions that minimizes the empirical risk. Empirical risk minimization is a form of training/learning.

6. What is the expected risk?

The expected value of the loss, i.e. the average over an infinite number of examples.

7. What is "generalization"?

The capability the a predictive system f(**x**) has to make "good" predictions on examples that were not used for training.

8. What is "overfitting"?

Learning very well the training examples but making poor predictions on new test examples.

9. What are training/validation/test sets? What is "cross-validation"? Name one or two examples of cross-validation methods.

For the purpose of this class: The training data provided to you is the union of the training set and the validation set. The training data consist of input/output pairs. They are available for "training", i.e. determining the predictive model f(**x**). The test data consist of inputs only. The accuracy of the predictions made on test data by the predictive model will be assessed by someone who knows the "true" outputs, but does not give them to you. You are free to split training data into a subset

reserved for training (training set) and a subset reserved for evaluation (validation set). You may want to make several splits and average the results; this is called cross-validation. One cross-validation method called "**bootstrap**" consists in drawing with replacement several data splits in equal proportion. Another method called **k-fold** consits in dividing the training data into k subsets, training on (k-1) subsets and testing on the last one, then repeating the operation for all groups of (k-1) subsets and averaging the results. 3-fold and 10-fold cross-validation are popular methods. In the limit one can have as many folds as there are examples. The method is then called "**leave-one-out**".

10. What are hyper-parameters?

Predictive models have adjustable parameters subject to training. Some parameters are not "easy" to train with classical algorithms. They can be fixed during training. Then, their values are varied and an optimum may be selected by cross-validation. Such parameters are usually referred to as "hyper-parameters".

11. What are "latent" variables?

Learning systems have input variables (or "features"), output variables, and internal variables. Latent variables are internal variables. While input and output variables are observable from the outside may be provided for training, latent variables are not accessible, thus not provided for training. On must usually initialize them randomly and recompute their values in the process of learning.

12. What is "model selection"?

Machine learning usually consists in adjusting the parameters of a model. However, we may have a number of candidate models (e.g. linear models, kernel methods, tree classifiers, neural networks...) Model selection refers to choosing the model, which we believe will generalize best. Model selection encompasses also hyper-parameter selection and feature selection. Cross-validation is a commonly used method of model selection.

13. What do "multiple levels of inference" mean? Is it advantageous to have multiple levels of inference?

Inference refers to the ability of a learning system, namely going from the "particular" (the examples) to the "general" (the predictive model). In the best of all worlds, we would not need to worry about model selection. Inference would be performed in a single step: we input training examples into a big black box containing all models, hyper-parameters, and parameters; outcomes the best possible trained model. In practice, we often use 2 levels of inference: we split the training data into a training set and a validation set. The training set serves the trains at the lower level (adjust the parameters of each model); the validation set serves to train at the higher level (select the model.) Nothing prevents us for using more than 2 levels. However, the price to pay will be to get smaller data sets to train with at each level.

14. What is the likelihood?

Predictive models learn the mapping y=f(**x**) or more generally P(y|**x**). Conversely, generative models (which are probabilistic) learn the opposive, namely to predict the density of **x** given y P(**x**|y). In the maximum likelihood framework, we assume that the data was produced by a model. The model has some parameters. The goodness-of-fit of "likelihood" is the probability that the data was produced by the model, for a given choice of parameters.

Likelihood = Proba (data | model).

15. What means "maximum likelihood"?

The maximum likelihood method of inference chooses the set of parameters of the model that maximize the likelihood.

16. What is the Bayes formula?

Bayes formula: $P(A|B)\ P(B) = P(B|A)\ P(A)$
Applied to our problem, we can go from a predictive model to a generative model and vice et versa using:
$P(\mathbf{x}|y)\ P(y) = P(y|\mathbf{x})\ P(\mathbf{x})$

17. What is Bayesian learning?

In Bayesian learning, one assumes that the data was drawn from a double random process: first a function f is drawn according to a "prior" distribution P(f), then data pairs are drawn $D=\{\mathbf{x}_i, f(\mathbf{x}_i)\}$. In Bayesian learning, one seeks to estimate for a new example **x** the probability $P(y|\mathbf{x},D)$ by integrating over all possible choices of f, using P(f|D): $P(y|\mathbf{x},D) = \text{integral } P(y|\mathbf{x},f)\ dP(f|D)$.

18. What is a prior? What is a posterior?

P(f) is the prior on function space. Our revised opinion, after we have seen the data, is called the posterior P(f|D).

19. What is Maximum A Posteriori estimation (MAP)?

The Bayesian approach requires computing a difficult intergral. Instead, we can select only one function f that maximizes P(f|D). This is the Maximum A Posteriori (MAP) approach. We use Bayes' formula $P(f|D)P(D)=P(D|f)P(f)$ so that we can replace the maximization of P(f|D) by that of P(D|f)P(f), where P(D|f) is the likelihood and P(f) the prior.

20. What is "structural risk minimization"?

It is a means of controlling the complexity of a model by building a structure consisting of nested subsets of models. By doing that, we know that the complexity is increasing from subset to subset. It is then possible to penalize more complex models, without quantifying their complexity. Constraints on the structure are imposed and introduced into the risk functional via Lagrange multipliers.

21. What is "regularization"? What is a "ridge"?

Regularization is a means of solving "ill-posed" problems, such as inverting a matrix which is not invertible. The penalty term lambda $||\mathbf{w}||^2$ in ridge regression is called a regularizer. The positive coefficient lambda is called "ridge".

22. What is a Lagrangian?

In an optimization problem in which one seeks to minimize a functional under some constraints, a Lagrangian is the functional in which the constraints have been introduced by multiplying them by positive coefficients called "Lagrange multipliers". The net effect is to simplify the optimization problem by putting it in a canonical form.

23. What is the link between structural risk minimization and regularization?

We can choose a structure that penalizes models with large $||\mathbf{w}||^2$. Each element of the structure imposes a constraint $||\mathbf{w}||^2 < A$. Using a Lagrange multiplier lambda, we can obtain again the same penalized risk functional used for ridge regression, i.e. the regularizer lambda $||\mathbf{w}||^2$.

24. What is the correspondance between maximum likelihood and empirical risk minimization?

The likelihood of a model can be converted to a risk functional via the formula:
$R[f] = -\log P(D|f)$
For i.i.d. data, P(D|f) can be decomposed as the product of $P(\mathbf{x}_i,y_i|f)$. The risk being the sum of the losses for the example patterns, the loss function is then given by:
$L(f(\mathbf{x}_i), y_i) = -\log P(\mathbf{x}_i,y_i|f)$
Conversely, the risk may be interpreted as an "energy". A likelihood can be defined following the Boltzman distribution $P(D|f)= \exp -R[f]/T$, where T is a "temperature" parameter.
With this correspondance, minimizing the risk is equivalent to maximizing the likelihood.

25. What is the correspondance between MAP and regularized risk minimization?
In the MAP framework, we maximize the product of the likelihood and the prior P(Dlf)P(f). A regularized risk functional may be obtained by taking the negative log:
R[f] = - log P(Dlf) - log P(f)
where - log P(f) takes the role of the regularizer.
26. What is the correspondance between ridge regression, weight decay, Gaussian processes, and ARD priors?
Assume a linear model f(**x**)=**w**.**x**+b is used. Ridge regression means least-square regression with a 2-norm regularizer ll**w**ll^2. A stochastic gradient algorithm optimizing the corresponding regularized risk functional includes a "weight decay" term. Gaussian processes are Bayesian methods assuming the weights are picked using a prior P(f)=exp-allwll^2. Thefore the regularizer obtained by taking -logP(f) is the 2-norm regularizer ll**w**ll^2. In the case of the linear model, this prior is also called ARD (Automatic Relevance Determination). The method can be "kernelized" by introducing scaling factors.
27. Why does the 1-norm regularization yield "sparse" solutions?
The surfaces of equal regularization are hyper-diamonds. If the unregularized solution is close enough to an edge, the solution is pulled to the edge, corresponding to a number of weights being set to zero. For the 2-norm regularization, the surfaces of equal regularization are hyper-sheres. The weights do not get set preferentially to zero.
28. Why is the 1-norm regularization not suitable for the "kernel trick"?
To apply the kernel trick, it should be possible to express the cost function in terms of dot products of patterns. With the 2-norm regularizer, this is possible ll**w**ll$_2^2$ = **w** **w**^{T} = **alpha**^{T} XX^T **alpha**, where XX^T is the matrix of the dot products between all the pairs of patterns (that becomes the kernel matrix after applying the kernel trick). For the 1-norm regularizer, this is not possible.
29. What is a "link function"? Give examples.
For a discriminant function f, a link function is a function "linking" the functional margin z=y f(x) and the likelihood P(Dlf). Is is a means of converting the output of a discriminant function to posterior class probabilities. Link functions are usually S-shaped (sigmoids). The tanh and the logistic function 1/(1+e^{-z}) are often used. A piece-wise function, which is S-shaped, but is constant before -1 and after +1 may be used to implement "Bayesian" SVMs and get support vectors.

<u>Architectures and algorithms</u>
30. What is a "linear discriminant" classifier? Name examples.
A linear discriminant classifier if a function f(**x**) linear in its parameters. Examples include f(**x**)=**w.x**+b (weighted sum of the inputs), f(**x**)=**w**. **phi**(**x**) +b (the "Perceptron"), and f(**x**)=sum$_i$ alpha$_i$ k(**x**$_i$, **x**) + b (kernel method). The linear discriminant f(**x**)=**w.x**+b is also linear in the input components. It builds a decision surface f(**x**)=0, which is a hyperplane.
31. What is an artificial neuron? What is a McCullogh-Pitts neuron?
An artificial neuron is a very simplified model of brain neuron. For the McCullogh-Pitts artificial neuron, the inputs and outputs are binary (representing 2 states "active" or "inactive"), the synapse strength (connection between neuron via e.g. a neurotransmitter) is modeled by a weight, the "potential" of the neuron is modeled by a weighted sum of the inputs, and whether or not the neuron "fires" by the thresholded potential. Such an artificial neuron is a linear discriminant.
32. What is Hebb's rule?
This is the simplest way of training an artificial neuron: the synapse between two neurons is

reinforced if there is co-activity, or equivalently if for a given neuron its input is 1 and simultaneously its output is 1.

w_j <- $w_j + x_i$ y

For neurons with binary 0/1 states, the weight is updated only if positive activity takes place. But the rule can also be applied to "neurons" with +1/-1 states and for linear discriminant with continuous value inputs.

If there are the same numbers of examples in either class, a linear classifier f(**x**)=**w.x**+b trained with Hebb's rule classifies examples according to the nearest class centroid. It may classify the training examples with a few errors.

33. What is the "Perceptron"? What is the Perceptron algorithm?

The Perceptron is a linear discriminant invented by Rosenblatt in the sixties: f(**x**)=**w**. **phi**(**x**) +b. It may be trained with the Perceptron algorithm, which also applies to the simpler f(**x**)=**w.x**+b model. The Perceptron algorithm is like Hebb's rule, but updates are made only for misclassified examples. If training examples are "linearly separable" the Perceptron algorithm converges to a hyperplane separating the examples without error.

34. What is gradient descent?

Gradient descent is a method of optimization. Given a cost function (or risk functional) R[f] steps are made in the parameter space of f to decrease R[f] in the direction of the steepest local slope. The method converges to a local minimum of f. The error rate is the "natural" risk functional for classification problems. However, it cannot be optimized by gradient descent because of its discontinuities. It is often substituted by other risk functionals (the "Perceptron" risk, the mean-square-error, etc.). Those will be reviewed again in upcoming lectures.

35. What means "Batch gradient" descent? What means "stochastic gradient" or "on-line" gradient descent?

In "batch gradient" a weight update is made using information from all the training examples. Conversely, "on-line" or "stochastic" gradient makes a weight update for each example individually.

36. What is the "Adaline"? What is LMS?

The Adaline is a linear model f(**x**)=**w.x**+b proposed in the sixties by Widrow as an artificial neuron model. It was trained with an on-line gradient algorithm optimizing the square loss L(f(**x**), y) = (f(**x**) - y)^2.

37. What method(s) solve(s) exactly the least square problem for a linear model? How does this relate to LMS and Hebb's rule?

The normal equations, or "pseudo-inverse" method solve the least-square problem. So does the LMS algorithm. On can see that for the learning machine f(**x**)=**w.x**+b, the derivative of the loss L(f(**x**), y) = (f(**x**) - y)^2 with respect to a weight wj is -2 (1-yf(**x**)) y x_i. The weight update of LMS goes in the direction of the negative gradient of the loss for a single example. Hence, we have:

w_j <- w_j + eta (1-yf(**x**)) x_i y

where eta is the learning rate. We notice that LMS is similar to Hebb's rule, except that we have the factor (1-yf(**x**)). This factor decreases the update if the goal yf(**x**)=1 is nearly achived. Thus, similarly to the Percetron algorithm, it does not insist on learning examples already known.

38. What is a "kernel"? What is a dot product? Give examples of kernels that are valid dot products.

A kernel is similarity measure. The kernels we will be taking about in this class are dot products. We all know the "regular" dot product (or scalar product) in a Euclidean space **x.y**=sum$_j$ x_j y_j . More generally, a dot product on a vector space V is a positive symmetric bilinear form:

<.,.>: V X V -> R

(**x**, **x'**) -> <**x**, **x'**>
with
<a**x**, **x'**> = a <**x**, **x'**> and <**x**, a**x'**> = a <**x**, **x'**> (bilinearity)
and
<**x**, **x**> >= 0 with equality only for **x**=0 (positivity)
Kernels that can be expanded as k(**x**, **x'**) = **phi**(**x**) . **phi**(**x'**) are valid dot products.

39. What is an RBF? What is a "potential function"? What is a "Parzen window"?

These are all names for kernels. RBF stands for "radial basis function". A Gaussian kernel is an RBF. A potential function is also an RBF having the form of an electric potential. A Parzen window is also an RBF of any shape, used in particular for density estimation. You do not need to remember this nomenclature, this is just for your information.

40. What is a kernel classifier? Is a kernel classifier a linear discriminant classifier?

A kernel classifier is of the form f(**x**)=sum$_i$ alpha$_i$ k(**x**$_i$, **x**) + b . It is linear in its parameters, but usually not in its input components (except for the "linear kernel", that is k(**x**, **x'**)=**x.x'**. So it is a linear discriminant classifier according to our definition.

41. What is the "kernel trick"?

The kernel trick consists in noticing that there is an equivalence between the two types of linear discriminant:
f(**x**)=**w**. **phi**(**x**) +b (Perceptron) and f(**x**)=sum$_i$ alpha$_i$ k(**x**$_i$, **x**) + b (Kernel method), in the case where **w** = sum$_i$ alpha$_i$ **phi**(**x**$_i$) and if we define k(**x**$_i$, **x**)=**phi**(**x**$_i$).**phi**(**x**). Replacing one by the other does not seem to be an advantage (if N is the dimension of phi, the Preceptron dot product takes N operations, while the kernel machine would take Nm for m examples). However, we usually never compute the kernel as the dot product of the **phi** vectors, because we know a faster-to-compute formular that is a simple function of the **x** vectors (e.g. a function of **x**$_i$**.x** or of || **x**$_i$**-x** ||). Thus we may be replacing N operations by m.something_small. Furthermore, for some kernels, N may be infinite.

42. What is "ridge regression"? How can one train a ridge regression linear model?

Ridge regression is like least-square regression with an additional penalty term ||**w**||^2. To train a ridge regression linear model f(**x**)=**x** **w**^{T}, one can perform a matrix inversion of the regularized matrix (X^{T}X +lambda I) to get **w**^{T} = (X^{T}X +lambda I)$^{-1}$X^T**y** or perform gradient descent in the penalized risk functional: R[**w**] = || X**w**^{T} –**y** ||^2 + lambda ||**w**||^2. For the stochastic gradient, one derivates simply the contribution to the risk of one example: (**x**$_i$ **w**^{T} -y$_i$)^2 + lambda ||**w**||^2, yielding the learning rule Delta **w** = [(y$_i$ - **x**$_i$ **w**^{T}) **x**$_i$ - lambda **w**] or **w** <- (1-lambda) **w** + (y$_i$ - **x**$_i$ **w**^{T}) **x**$_i$.

43. What is "weight decay"? What is the connection to ridge regression?

Weight decay means decreasing the weights at every learning step according to (1-lambda)**w**. The weight decay in ridge regression results from derivaring the pernalty term lambda ||**w**||^2.

44. What is a "Gaussian process"? What is the connection to ridge regression and weight decay?

A Gaussian process is a generative model in which the weights of the target function are drawn according to a Gaussian distribution (for a linear model). The prior in function space is P(f) = exp -lambda ||**w**||^2. In Maximum A Posterori (MAP) framework one seeks to find the function f that maximizes P(fID), D being the data, or equivalently P(Dlf) P(f). By taking the negative log, one sees that -logP(Dlf) plays the role of the risk and -logP(f) the role of the penalty term lambda ||**w**||^2. Hence a Gaussian prior on the weights is equivalent to using the penalty lambda ||**w**||^2 in the risk minimization framework.

45. What method(s) solve(s) exactly the least square problem for a linear model? How does this relate to LMS and Hebb's rule?

The solutions are similar to those of ridge regression, but we let lambda go to zero. LMS stands for "least mean square". It is the rule obtained by derivating the square loss with repect to **w**: Delta **w** = $(y_i - \mathbf{x}_i \mathbf{w}^T)\ \mathbf{x}_i$. It is just like Hebb's rule Delta **w** = $y_i\ \mathbf{x}_i$ except that it learns less the examples for which the predictions are already good.

46. What is the "pseudo-inverse"? How is it linked to ridge regression?

The pseudo-inverse of X^T is $\lim_{\text{lambda->0+}} (X^TX + \text{lambda } I)^{-1} X^T$. It is involved in solving the least square regression problem. The solution for ridge regression is the same, except that lambda is now a given positive value.

47. What is kernel ridge regression? Give other examples of algorithms using the "kernel trick"?

We can use the same ridge regression algorithm for models linear in their parameters, but non linear in their input components ("Perceptrons" f(**x**)=**phi**(**x**) $\mathbf{w}^T$). Using the "kernel trick" i.e. the fact that **w** = $\text{Sum}_i\ \text{alpha}_i\ \mathbf{phi}_i(\mathbf{x})$, we can transform the problem of solving Phi **w**T=y into K **alpha** = **y**, where K is the matrix of dot products between the training examples in phi space. Kernel ridge regression amounts to solving that equation, after adding lambda to the diagonal of K. The kernel trick may be used with all the algorithms for which **w** = $\text{Sum}_i\ \text{alpha}_i\ \mathbf{phi}_i(\mathbf{x})$, in which the **phi** vectors appear only through their dot product and can therefore be replaced by a similarity measure k(**x, x'**) = **phi**(**x**).**phi**(**x'**)

48. What is Principal Component Analysis (PCA)? Which eigen value indicates the direction of largest variance? In what sense is the representation obtained from a projection onto the eigen directions corresponding the the largest eigen values optimal for data reconstruction?

PCA is a method of feature construction. The new features are linear combinations (weighted sums) of the old ones. They are obtained by rotating the input space into the axes of the principal components of XTX: X->XU, where the columns of U are eigenvectors. This transform has the following properties: (1) the eigen directions corresponding to the largest eigenvalues explain best the variance in the data; (2) If we limit ourselves to the n' eigen directions corresponding to the top eigenvalues and rotate back into the original axes: XU -> XUU^T, the reconstructed data XUU^T are closest to the original data X in the least square sense. So we cut down on the number of features with as small as possible information loss.

49. What is the connection between PCA and weight decay?

PCA cuts the dimensions corresponding to the smallest eigenvalues of X^TX. Weight decay pulls the weight to zero preferably in those directions, which are the directions of least resistance.

50. What is an irregular matrix? Name one way of regularizing a matrix.

An irregular matrix is a matrix which is not invertible. A square matrix of dimensions (n, n) is not invertible if its rank is smaller than n, or equavalently it has less than n non-zero eigenvalues. To regularize a matrix, one can add a small positive value to the diagonal, which has the effect of making all eigen values non-zero.

51. What is the decision function of a support vector classifier (SVC)?

There are two possible "dual" representations:

- The Perceptron representation f(**x**)=**w.phi**(**x**)
- The kernel method representation f(**x**)=$\text{sum}_k\ \text{alpha}_k\ y_k\ k(\mathbf{x},\mathbf{x}_k)$

52. What objectives are being pursued when training an SVC?

- Minimizing the number of training errors
- Having a margin between examples of wither class as large as possible

53. In what do those objectives differ from those of the Perceptron algorithm?

The Perceptron algorithm only attempts to minimize the number of training errors.

54. What are support vectors?

Support vectors are those examples that are closest to the decision boundary and entirey define the solution of the SVM optimization problem.

55. What are the main properties of the SVC solution?

- Unique solution
- A function only of "support vectors"
- Stable solution (does not change if any of the examples is removed, except a support vector; training error does not change under small changes of the weight vector.) Consequences: good leave-one-out error bounds.

56. Starting from a linear optimum margin classifier, how can one handle the non-linearly separable case (three answers)?

- Make the classifier non-linear using the **phi** functions.
- Use a negative margin classifier (not a unique solution anymore).
- Use a "soft-margin" classifier.

57. What are the two types of support vectors for soft margin classifiers?

- Marginal support vectors (on the margin).
- Non-marginal support vector (within the margin or misclassified).

58. What is the "kernel trick"?

In algorithms for linear models where an input vector **x** and training examples $\mathbf{x}_k$ appear only through their dot product $\mathbf{x}.\mathbf{x}_k$, the "trick" is to replace $\mathbf{x}.\mathbf{x}_k$ by another dot product $k(\mathbf{x}, \mathbf{x}_k)$ to make the model non-linear.

Another viewpoint is to start with a model linear in its parameter, but non-linear in its input **x**, $f(\mathbf{x})=\mathbf{w}.\mathbf{phi}(\mathbf{x})$. Then, if the learning algorithm yields a linear combination of the examples $w=\sum_k \alpha_k y_k \mathbf{phi}(\mathbf{x}_k)$, one obtains the dual representation:

$f(\mathbf{x})=\sum_k \alpha_k y_k \mathbf{phi}(\mathbf{x}_k).\mathbf{phi}(\mathbf{x}) = \sum_k \alpha_k y_k k(\mathbf{x}, \mathbf{x}_k)$,

where $k(\mathbf{x}, \mathbf{x}_k)=\mathbf{phi}(\mathbf{x}_k).\mathbf{phi}(\mathbf{x})$ is a valid dot product.

The "trick" is then to use any valid dot product $k(\mathbf{x}, \mathbf{x}_k)$ having a phi expansion (even an infinite expansion or an integral).

59. What is the loss function of an SVC?

For an example **x** of margin $z=y f(\mathbf{x})$: $\max(0, 1-z)$ or $\max(0, 1-z)^2$

60. Name other large margin loss functions. How does the plot of the loss vs. margin look like?

The logistic loss $\log(1+e^{-z})$, the Adaboost loss e^{-z}. They all increase sharply for $z<0$ (misclassified examples, but are non-zero in the $0<z<1$ margin region. They are also continuous smoothe functions allowing gradient descent (unlike the 0/1 loss).

61. What regularizers are used with SVC?

The 1-norm or the 2-norm of **w**.

62. Can those regularizers be used with the square loss as well? What are the names of the corresponding techniques?

Yes. 2-norm regularization -> kernel ridge regression. 1-norm regularization -> lasso.

63. How can one define a regression SVM?

One can define an epsilon-tube playing the role of a margin. Examples in the tybe do not incur a

loss. The loss increases linearly outside the tube.

64. Can one define an SVM for unsupervised learning?

Yes. In several different ways. The simplest one is the "one-class" SVM, with application to define the support of a distribution, novelty detection, and clustering.

Feature extraction

65. What are feature extraction, feature construction, and feature selection?

Feature extraction = feature construction + feature selection.

Feature construction means creating new features from the raw data (this includes normalizations, making products of features, using ad hoc algorithms like extracting edges in image processing).

Feature selection means reducing the number of features by removing irrelevant or redundant features.

2. **What are the three "ingredients" of a feature selection method?**

- Defining a **criterion of selection** (for individual features or feature subsets). The criterion may be a ranking coefficient that measures the degree of dependance of a feature with the target. It may be the performance of a learning machine.

- Choosing a **method of estimation** (for instance evaluating the criterion on training examples; in some cases cross-validation should be used).

- Choosing a **search strategy.** When the number of subsets of features to be assessed is too large to do an exhausive search, the space of feature subsets must be search in a more efficent way.

66. What are filters/wrappers/embedded methods?

- Filters use a criterion of selection that does not make use of the learning machine. An example of filter is the use of the Pearson correlation coefficient for feature ranking.

- Wrappers use the learning machine to evaluate the performance of alternative feature subsets. They use a search method to explore the space of possible subsets. The learning machine is considered as a "black box", i.e. no knowledge of the learning algorithm is necessary to apply the method.

- Embedded methods use feature selection strategies particular to given learning machines.

67. What is a "univariate method"?

A method making the assumption that variables are independent. Univariate feature selection methods assess the predictive power of individual features.

68. What is a "multivariate" method?

A method taking into account variable covariance. Multivariate feature selection methods assess the predictive power of feature subsets.

69. What is the Pearson correlation coefficient? How can it be used for feature selection?

The Pearson correlation coefficient between two vectors **x** and **y** is defined as: $C(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}).(\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}})/ (\mathbf{s}_{\mathbf{x}}\, \mathbf{s}_{\mathbf{y}})$

where $\boldsymbol{\mu}_{\mathbf{x}}$ is the mean of the coefficients of vector **x** and $\mathbf{s}_{\mathbf{x}}$ its variance. So, essentially, the Pearson correlation coefficient is a dot product (or scalar product) between **x** and **y**, after "standardization". The standardization operation consists in subtracting the mean and dividing by the standard deviation. The absolute value of the Pearson correlation coefficient is used to rank features. In this case, **x** is a column of the data matrix and **y** is the vector of target values. Important: elsewhere, we call x a line of the data matrix.

Feature construction

70. What is a sigma-pi unit?

A sigma-pi unit is a special kind of Perceptron in which the phi functions correspond to products of the original features. The unit is thus effectively computing a polynomial function of the inputs.

71. What is a bottleneck neural network? How does this relate to PCA?

A bottleneck neural network is a 2 layer network in which the input layer and output layer have same dimension n and the hidden layer has a number of outputs n'<n. A bottleneck network can be trained with the same examples at the input and the output. If the units are linear and if the square loss is used for training, a bottleneck network actually computes the the first n' principal components, which are the weights of the neurons of the first layer. The second layer reconstructs the inputs and the weights of the neurons are given by the transpose of the weight matrix of the first layer.

72. What becomes of the dot product between patterns when patterns are normalized with the L2 (Euclidean) norm?

The cosine between the two patterns.

73. What becomes of the dot product between feature and target when the features (and the target) are standardized?

The Pearson correlation coefficient.

74. When does it make sense to take the Log of the data matrix?

When the variance of the data increases with the magnitude of the features.

75. What is a sytematic error? What is an intrinsic error?

A systematic error is an error that can be explained and reduced by calibration or normalization. An intrinsic error corresponds the unexplained "random" noise.

76. How can one get rid of systematic errors?

By modeling the noise and trying to reverse the noise generating process, by calibration or normalization.

77. What is an ANOVA model?

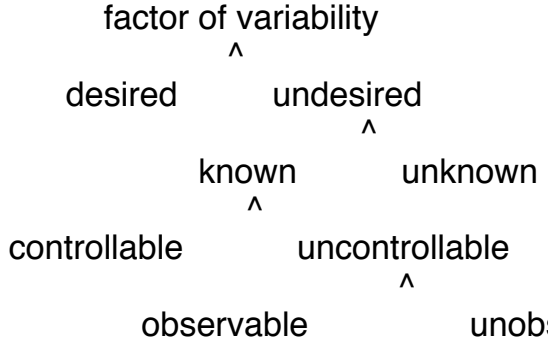
ANOVA stands for Analysis of Variance. An ANOVA model is a model of the effect on observations x of a systematic (or "controlled") factor of variability v taking a discrete number of values $\{v_1, v_2 \ldots v_j, \ldots\}$ and intrinsic variability e (random error, Normally distributed):

$$x_{ij} = m + v_j + e_{ij}$$

(i index of observation, j index of “treatment” of “class”)

The ANOVA model supposes additive noise and equal variance in the classes (so take the log if you see the variance increase with the variable magnitude). The ANOVA test compares the variance of the controlled factor v (variance explained by the model) to the intrinsic variance of e (residual variance or "intra-class" variance). If the first one is statistically significantly larger than the second one, factor v is found to contribute significantly to the noise.

78. Build a taxonomy of factors of variability in terms of whether they are desired, known, controllable or observable. Explain the various cases and give examples. When building a new instrument, in which direction should one go?



- The desired factor is our target (class labels) e.g. disease or normal
- The undesired factors are all the nuisance variable causing variance in the data that is not related to our target, e.g. differences in sample processing, temperature, patient gender, etc.
- The unknown factors are those which we have not considered yet (not recorded or controlled) the others are considered known.
- The uncontrollable factors are those on which we have no any handle (e.g. the weather, something happening inside the instrument to which we do not have access, some patient behavior that we cannot change).
- Controllable factors on the other hand let us choose values and lend themselves to experimental design.
- Unobservable factors are those uncontrolled factor that we cannot even record (something happening inside the instrument to which we do not have access, some patient behavior that we cannot monitor).
- Observable factors are all the remaining factors that we can record, even though we might not be able to control them (e.g. the weather).

When designing an instrument, we should try to go in the direction

unobservable -> observable
uncontrollable -> controllable
unknown -> known

so that we can more effectively reduce the undesired variance.

79. What is experimental design? What is a "confounding factor"? Give examples of experimental plans.

Experimental design is the science of planning experiments to most effectively study the effect on a set of given factors on a given outcome. A confounding factor in a factor (usually unknown) the value of which co-varies with another know factor under study. For example if we want to study the effect of age on weight but all our young people are male and all our old people are female, the gender is a confounding factor. Of course this situation is a bogus experimental design. Good planned experiments try to consider "all" possible combinations of assignments of variables to values. In a factorial plan, each variable is allowed to take only 2 values and for k variables this leads to 2^k experiments. To avoid the effect of possible unknown factors correlated with time, the order of the experiments can be randomized (randomized plan). To be able to study the variance of a given factot on the outcome, factors can be kept constant in some experiment blocks (block design).

80. Why is it important to record a lot of "meta data"? Why is it difficult to plan experiments with a lot of factors of variability? How should one proceed?

It is important to record a lot of "meta" data to be able to eventually explain some of the unexplained variance. However, all the factors recorded are not always controlled in the planned experiment because or the combinatorial explosion of the number of experiments to be ran when a lot of factors are considered simultaneously. One should proceed iteratively by ruling out hypotheses progressively.

81. What is a standard operating procedure (SOP)? What is calibration? What is this good for?

A standard operating procedure is a series of steps taken to generate the experimental data that is well documented and as reproducible as possible. SOP are used to reduce the unexplained variance. Calibration is a measurement made in a standard way, which allows normalizing the data (e.g. shifting or scaling it). For example, a standard solution may be periodically ran in place of the real solutions to be analyzed.

82. Is calibration always desirable?
Not always. The calibration measurement may also have variance. Calibration in some cases can result in an increase of variance. One may prefer instead to normalize with a local average the itself because the normalization factor would then be computed from more data.

83. What is a "match filter"? Give examples of learning algorithms using "match filters".
It is a vector of coefficients $\mathbf{t}_k$ or "template" that we use to compute a feature value f_k as the dot product between $\mathbf{t}_k$ and the input patterns $\mathbf{x}$: $f_k = \mathbf{t}_k \cdot \mathbf{x}$
Instead of a dot product, other similarity measures can be used. "Template matching" or "nearest neighbor" algorithms use match filters.

84. What is a "filter bank"? Give examples of classical transforms based on filter banks.
An ensemble of match filters is called a filter bank. Often the elements of a filter bank are chosen to be orthogonal. The cosine transform and the Fourier transform use orthogonal filters. So does PCA.

85. What is a convolution? Give examples of convolutional kernels. What is their effect on the signal?
A convolution is also a dot product operation aiming at producing new features. But this time, instead of using templates that are as different of one another as possible, we use a single template called "kernel" that we translate in all possible ways. For each position, we compute the dot product to obtain one feature in the new representation. A Gaussian kernel performs a local average and therefore smoothes the signal. A Mexican hat kernel enhances edges. Some kernels can be designed to extract end-points or lines.

86. What are the similarities and differences between methods based on filter banks and convolutional methods?
Both methods are based on dot products. Filter bank methods use templates that are as different as possible from one another. Convolutional methods use a single template in all possible positions.

87. If a convolution is performed in input space, to what transform does this correspond to in Fourier space and vice versa?
A convolution in input space corresponds to a match filtering in Fourier space (the match filter being the Fourier transform of the convolutional kernel) and vice versa.

88. What are low/high/band pass filters? Give examples of convolutional kernels and match filters in Fourier space implementing such filters.
A low pass filter removes high frequency components i.e. it smoothes the signal. Example: convolution with a Gaussian kernel. A high pass filter on the contrary removes low frequency components (e.g. the baseline). To achieve that effect, one can convolve with a wide Gaussian kernel and subtract the result from the original. A band pass filter lets components in a given frequency band go through. One can convolve with the difference of two Gaussian kernels of different width to achieve that effect. In Fourier space, the Fourier transform of a Gaussian being a Gaussian, one can just multiply with a Gaussian match filter.

89. What is the Fourier transform of: a rectangle, a triangle, a Gaussian, a sinc?
rectangle -> sinc
triangle -> sinc2
Gaussian -> Gaussian
sinc -> rectangle

90. Give examples of feature construction methods that are not simple normalizations and cannot be implemented by either match filters or convolutions.
- contour following algorithms

- connected component algorithms
- deskewing
- histograms

91. What is a convolutional neural network?

A multi-layer neural network implementing several successive convolutions. Each convolution is followed by a subsampling to progressively reduce the resolution of the input and extract higer and higher level features. The weights of the network are the coefficients of the convolutional kernels and they are obtained by training.

Filter methods

92. What purpose(s) may be pursued when selecting features?

- Removing useless features (pure noise or "distracters") to save computing time and data storage.
- Improving prediction performance (there is less risk of overfitting if we start from a lower dimensional space)
- Understanding the process that generated the data (reverse engineering).

93. How can one define feature "relevance"? What is easier to define, relevance or irrelevance?

Relevance might be defined by the existence of a dependence between a feature and the target values (or "desired outcome, e.g. the classification labels). Statistical independence is easy to define: $P(X,Y)=P(X)P(Y)$. So independence is easier to define than dependence. There are several ways of defining dependence and assessing it. One way if the measure the discrepancy between $P(X,Y)$ and $P(X)$ $P(Y)$ with the KL divergence. This criterion is called "Mutual Information". Features can sometimes be irrelevant by themselves but relevant "in the context of others". Therefore we need to introduce a notion of conditional relevance, e.g. conditional mutual information.

94. In what respect is it possible to assess feature relevance from observational data (i.e. without being able to control the values taken by the features and designing experiments)? What will the limitations be?

For "canned data" we might be able to observe the dependence between features and target, but we cannot be sure that a feature showing no dependence is actually irrelevant. Only designed experiments can allow us to explore the space of values of the features in a systematic way and rule out dependencies with confidence. Often observational data consits of a sub-optimal exploration of input space because some variable values were not explored or the value of some variables not recorded at all. One should beware of confounded factors: variables that have co-varied during the experiment. For instance all the disease patient samples were stored in certain conditions and all the healthy patient samples in a different condition. Storage is then a confounding factor.

95. Will causal relationships be determined from the feature selection process? Is the inference of causality necessary to build a good predictor?

Feature relevance was defined via the notion of statistical dependence/independence, a generalization of the notion of correlation. There is no implication about causality. Correlated events may be causally related in either direction or result from a common cause but not be directly causally related. For example, observing that a person has a rash and eats chocolate does not mean that the chocolate diet caused the rash. The person may have eaten chocolate as a compensation for the ugliness of the rash! Or there might be a common cause, for example anxiety resulting from the preparation of an exam.

Causality is more difficult to infer than variable dependence. Luckily, we do not need to infer causality to build good predictors. For example, protein levels in blood can be used for cancer

diagnosis. Some protein levels may be causing the disease (like the lack of a given tumor suppressor), others my be the consequence of the disease (like the presence of a given antibody). But both may equally well be used to diagnose the disease.

96. How can we define feature "usefulness"?

A feature is useful if, when added to a subset of other features, it results in a prediction performance improvement, or if when removed it results in a performance loss.

97. Are features useful to make predictions always relevant, and vice versa? Give examples.

No: useful features may be irrelevant and vice versa. For example, two useful features may be redundant, so the removal of one of them will not cause performance degradation. Note that we can remove either redundant feature, so usefulness is not an intrinsic feature characteristic, it depends on all other features. Conversely, irrelevant features may be useful. A simple constant input in a linear model adds a bias that may result in performance improvement, but a constant value is not "relevant" to the target. A more elaborate example is the case of a nuisance factor adding noise to two features, one of which "f1" being "relevant" and the other "f2" "irrelevant". The nuisance factor is a systematic error tan can be removed by subtracting f2 from f1 and resulting in improved performance, even though f2 has no relevance to the target.

98. In what respects is mutual information a good or a bad choice to assess feature relevance?

MI is a good choice because it does not make any assumption on the data distribution and is looking for dependence in an agnostic way. Therefore, it can unravel non-linear dependencies. It is a bad choice because it is very difficult to estimate from data, except in the case of 2-class binary classifications problems (both features are target are binary). For problems with multivariate or continuous features and/or targets, it is preferable to use ranking indexes based on simple statistics of the distribution (like mean and variance). Such ranking indexes include the Pearson correlation coefficient and Fisher's criterion.

99. What is the Pearson correlation coefficient? Why is it a measure of goodness of fit of the linear least-square model? Give the formula that relates R^2 and the F score (variance explained/residual variance).

The Pearson correlation coefficient is R=cov(X,Y)/sqrt(var(X) var(Y)). For the least-quare linear regression, 1-R^2 is equal to the ratio of the residual variance over the total variance (i.e. it is the normalized mean-square-error), thus R^2 is a measure of goodness-of-fit for the linear least-square model. It follows simply that 1+F=1/(1+R^2), because total variance = variance explained + residual variance.

100. Is correlation related to mutual information? Give examples in which uncorrelated signals may have a high mutual information? Can correlated signals have a low mutual information?

Correlation is related to mutual information, but in some cases uncorrelated variables can have a lot of mutual information (example of the sinusoid). On the other hand, correlated variables always have a lot of mutual information (there dependence is linear). In the case where X and Y are Gaussian distributed, there is a simple relation: MI = -(1/2) log(1-R2).

101. How are the S2N coefficient, the Pearson correlation coefficient and the Fisher score related?

They all essentially measure the same thing: the ratio of the "signal" (the difference between the mean values of the two classes), and the "noise" (the within class standard deviation). For unbalanced classes differences arise because some criteria give more importance to the more

abundant class, either for the calculation of the signal or that of the noise. The S2N coefficient is the best for unbalanced classes.

102. What is conditional relevance? Give examples of feature ranking method, which take into account the "context" of other features.

Conditional relevance is "relevance in the context of other features". We discussed in class the Relief criterion.

Statistics

Note: some answers were drawn from http://www.stats.gla.ac.uk/steps/glossary/

103. What is a random variable?

A random variable is a function that associates a unique numerical value with every outcome of an experiment. The value of the random variable will vary from trial to trial as the experiment is repeated. There are two types of random variable - discrete and continuous. A random variable has either an associated probability distribution (discrete random variable) or probability density function (continuous random variable). A ranking index R assessing the dependence between a feature and the target is a random variable.

104. What are the definitions and properties of: expected value, variance, standard deviation, coefficient of variance?

- The expected value E(X) (or population mean μ) of a random variable indicates its average or central value. For a constant a, E(aX)=aE(X). For two random variables X and Y, E(X+Y)=E(X)+E(Y). If X and Y are independent, E(XY)=E(X)E(Y).
- The variance of the random variable X indicates its spread and is defined to be: var(X)=E(X-E(X))^2=E(X^2)-E(X)^2. For two constants a and b, var(aX+b)=a^{2}var(X). For two independent random variables X and Y, var(X+Y)=var(X)+var(Y).
- The standard deviation (stdev(X)) is the square root of the variance.
- The coefficient of variance is the ratio stdev(X)/E(X).

105. What is an estimator?

An estimator is a quantity *calculated from the sample data*, which is used to give information about an *unknown quantity* in the population. For example, the sample mean is an estimator of the population mean. An estimator is a random variable.

Not all estimators are "equal", some are more powerful than others. Some are biased: for a given size of the sample data, their expected value is not the unknown quantity that we want to estimate. Some have a lot of variance.

106. What is a probability distribution?

The probability distribution of a discrete random variable is a list of probabilities associated with each of its possible values.

107. What is a cumulative density function (cdf)?

All random variables (discrete and continuous) have a cumulative distribution function. It is a function giving the probability that the random variable X is less than or equal to x, for every value x. Formally, the cumulative distribution function F(x) is defined to be: F(x)=Proba(X<=x), -Inf<x<+Inf. The cdf is obtained by integrating the pdf.

108. What is a probability density function (pdf)?

The derivarive of the cdf.

If you have doubts about the definitions of the Gaussian pdf and the central limit theorem, see http://www.stats.gla.ac.uk/steps/glossary/probability_distributions.html

109. What are the basic "ingredients" of a statistical test? What are possible outcomes?

1) A **null hypothesis H_0** that we want to test (and eventually one or several alternative hypotheses

H_1.)
2) A **test statistic T** that is a random variable such that if H_0 is true, the expected value of T is zero.
3) The **distribution of cdf of T** Proba(T<=t), if H_0 is true.
4) A **risk value** alpha and the corresponding threshold t_{alpha} such that alpha=Proba(T>t_{alpha}) . [This is for a one-sided test where the risk is blocked on one side; for a two-sided test the risk is equally spread on both sides of the cdf.]
5) A **realization of T**, t for a given population sample.
Then, if t>t_{alpha} we reject H_0 with risk alpha of being wrong. In the opposite case, the conclusion is less strong: we do not reject H_0. In hypothesis testing, we never "accept" H_0. [For a two-sided test, we reject if t>$t_{alpha/2}$ or t<$t_{alpha/2}$.]

110. What is a pvalue? What does a small pvalue indicate about the null hypothesis?
Given a test statistic T and a realization t, the pvalue is pval=Proba(T>t) [one-sided test]. Small pvalues shed doubt ont the null hypothesis.

Assessment methods

111. What is the definition of a probably approximatly irrelevant feature?
For a relevance index R, Proba(R>epsilon)<delta, for epsilon and delta positive constants.

112. If we want to test the statistical significance of the relevance of a feature, what kind of test can we perform? State the null hypothesis. What is the null distribution? What is the alternative distribution?
We can perform a hypothesis test with null hypothesis: "the feature is irrelevant". The null distribution is the distribution of irrelevant features for the given ranking index. The alternative distribution is the distribution of relevant features. Both are usually unknown, but the null distribution of random features is easier to model. We can for example use "random probes" to estimate it.

113. Give examples of test statistics used to test feature relevance. What is being used as ranking index?
The T statistic, the ANOVA statistic (F statistic), the Wilcoxon-Mann-Whitney statistic. The pvalue is the ranking index. For one-sided tests, it gives the same ranking as the test statistic. Some test statistics have positive and negative values; zero corresponds to irrelevant features, large absolute values correspond to relevant features; the sign indicates the direction of the correlation.

114. What is the false positive rate (FPR) for feature selection?
This is the fraction of all the irrelevant features that have been selected. It may be approximated by the fration of all the probes that have been selected. If the distribution of irrelevant features is known, it is also the pvalue.

115. In the case of multiple testing, does the FPR (or pvalue) estimate correctly the fraction of wrong decisions?
The FPR correctly estimates the type I errors (fraction of incorrect rejections of the null hypothesis, that is fraction of incorrect decisions that the feature is not irrelevant), if a single feature is being tested (assuming we could test it multiple times by drawing multiple samples of the same size.) It does **not** estimate the fraction of incorrect decisions that features are not irrelevant if multiple (different, independent) features are tested. In the case of multiple testing, the pvalue is larger. The Bonferroni correction consists in replacing pval by n pval, where n is the number of features tested.

116. What is the false discovery rate (FDR)?
This is the ratio of the number of irrelevant features selected over the total number of features

selected n_{sc}. It is bounded FDR <= FPR n/n_{sc}, where n is the number of features tested. Setting a threshold on the FDR rather that on the FPR amonts to correcting the pvalues and replacing them by n pval/n_{sc}.

117. What is the variance of the test error rate?

For an error rate E computed from m examples, the variance is: E(1-E)/m.

118. What is a good rule of thumb to compute the size of a test set necessary to obtain statistically significant results?

m=100/E.

119. What is a good test to assess the significance of the difference in performance of two classifiers?

The NcNemar paired test.

120. What is cross-validation? Give examples of cross-validation methods.

Cross-validation amounts to splitting the training data multiple times into training set and validation set. training is performed on the training set and test on the validation set. The validation results are then averaged. Note that this does not preclude of reserving a separate test set for the final testing. Examples of cross-validation techniques include k-fold cross-validation, bootstrap, leave-one-out.

121. Why is it wrong to rank the features with all the training set and then run cross validation on nested subsets defined by the ranking to determine the optimum number of features?

This is wrong because it biases the result: all the examples have been iplicitly be used for training since they were used for the feature ranking. It is better (but more computationally expensive) to remove one example, carry out the ranking and the training of all subsets, test on the withheld example, then average the results for each size of feature subset.

122. Is is best to carry out extensive experiments of feature/hyperparameter selection exploring a number of possibilities as large as possible?

No, this is dangerous and prone to overfitting. In this case, the validation set is being overfitted. If a large number of possibilities are investigated, they should be ranked in order of preference before running the experiments. Then, if two models have the same performance, within the error bar, the best ranked model is chosen.

Wrappers

123. What is the difference between filters and wrappers?

Fiters select features independently of the performances of the learning machine, using a "relevance" criterion. Wrappers select feature subsets on the basis of how well a learning machine performs.

124. Why do wrappers usually need a search method?

Training learning machines on all possible feature subsets is usually computationally infeasible. Wrappers use search strategies to efficiently explore the space of feature subsets.

125. Are all filters feature ranking methods?

No, feature ranking methods are a subset of the filter methods. However, some people call feature ranking methods filters. Filters include methods of selecting feature subsets on the basis of a relevance (or conditional relevance) criterion. Some people call filters methods that select features using one learning machine, but are then used for another one. For example, people use random forests (ensembles of decision tree) as a filter and then train an SVM with the selected features.

126. Are search methods only useful for warppers?

No, you can use a search method to generate feature subsets and then assess them with a "relevance" criterion rather than the score of a learning machine.

127. How is the assessment done for filter methods?
To be perfectly consistent with the definition of a filter, determining the optimum number of features should not use a classifier. Statistical tests and the proble methods are consistent in that respect. However, it is very common that people rank features with a "relevance" criterion, and determine the optimum number of features with the classifier performance. This is can be considered a hybrid between filters and wrappers or we can just call it a wrapper for which the search is guided by a relevance criterion.

128. How is the assessment done for wrapper methods?
Usually using cross-validation. However, there are "wrong" ways to do it. See question 11 in the assessment methods.

129. How many trainings does an exhaustive search in feature space require?
2^n, n being the number of features.

130. If we are selecting from data among N decision rules, what is the minimum number of examples we need to find the one that generated the data?
$\log_2 N$.

131. Combining the answers to the two previous questions, how should the number of examples scale with the number of features if we use exhaustive search in a wrapper setting?
We have to select among $N=2^n$ decision rules. The number of examples should therefore be of the order of n.

132. What is the number of subsets that are investigated in forward selection, backward elimination, and feature ranking?
For feature ranking and nested feature subset methods, N=n. For forward and backward selection N=n(n+1)/2.

133. How should the number of examples scale with the number of features for the methods of question 9?
The number of examples should be of the order of log n.

134. What are the differences, advantages and disadvantages of forward vs. backward selection?
Forward selection starts with an empty set of features and progressively add features. Backward elimination starts with all the features and progressively eliminate some. They both produce nested subsets of features. From the point of view of statistical complexity, they are equivalent. If the search is ended when a stopping criterion is met, forward selection may be computationally more efficient because the trainings are performed on smaller feature subsets. From the point of view of the quality of the solution, it depends: when we vary the number of features, the best feature subsets may not be nested. The forward nesting ensures some optimality of the small feature subsets, but does not guaranty finding the best larger ones. The backward nesting on the other hand ensures same optimality of the larger subsets, but may yield very poorly performing small subsets.

135. What is "floating search"?
A method of combining forward and backward selection. One step forward, then backward as long as we find better subsets than those of the same size obtained so far, or vice versa.

136. What is simulated annealing?
A method of optimization inspired physics of re-crystallization.
- Make a step in feature space, compute deltaE
- If deltaE<0, accept the change
- Otherwise, accept the change with probability exp(-deltaE/T)

- Progressively “cool down”.

137. What are "genetic algorithms"?

A method of optimization inspited by biology.

138. Why couldn't we use gradient descent instead of the search methods we talked about in class?

Because we are searching a discrete space. We will see when we talk about embedded methods how we can in fact use gradient descent.

Embedded methods

139. What is the difference between filters, wrappers, and embedded methods?

Fiters select features independently of the performances of the learning machine, using a "relevance" criterion. Wrappers and embedded methods both select feature subsets using a learning machine. Wrappers explore the space of all possible feature subsets using a search method; the learning machine is used to assess the subsets. Any off-the-shelf learning machine can be used for that purpose. Embedded methods perform feature selection in the process of learning. They return a feature subset and a trained learning machine.

140. What are the computational differences between nested subset methods for wrappers and embedded methods?

Assuming we split the data into one training set and one validation set:

- Wrappers perform n(n+1)/2 learning machine trainings.
- Embedded methods perform only n trainings.

This is due to the fact that the next feature to remove or add in embedded methods is chosen by estimating the change in cost function that will be incurred, not by retraining.

141. What is the statistical complexity difference between nested subset methods for wrappers and embedded methods?

It is approximatly the same, of the order of log(n). One can afford a number of features exponential in the number of examples.

142. What guides the search of embedded methods?

Changes in the cost function incurred by adding or removing inputs. These changes are estimated without retraining the learning machine, for instance using a Taylor series of the cost function (like in the case of OBD).

143. How can one move from a search in a dicrete space of feature subsets to a search in a continuous space?

By introducing feature scaling factors.

144. How does one perform feature selection using scaling factors?

One can perform gradient descent to optimize a performance bound. The resulting scaling factors indicate feature usefulness.

145. What is the idea behind the "zero norm" method?

We use $||w||_0$ =number_of_features as a regularizer. This is a shrinkage method. $||w||_0$ is used instead of $||w||_2^2$ or $||w||_1$. The idea is to make a tradeoff between the empirical risk (e.g. the number of training errors) and the number of features:

$R_{reg} = R_{emp} + \text{lambda } ||w||_0$

Information Theoreric Methods

146. What is the mutual information?

$MI(X,Y)= sum_{X,Y} p(x,y) \log p(x,y)/p(x)p(y)$

This is the KL divergence between p(x,y) and p(x)p(y). Independent random variables have zero MI.

147. What is the relation between mutual information and information gain?

Two names for the same thing: MI(X, Y)=H(Y)-H(Y|X).

148. What is the basic functioning of a binary tree classifier?

At the root node, all the training examples are in the same bin. The "disorder" before we see the training data is measured by H(Y). To split a node in two children, one selects the feature providing the largest information gain, that is the largest reduction in entropy: MI(X, Y)=H(Y)-H(Y|X). So, we need to discretize a candidate feature using a threshold and create two nodes containing the examples below and above the threshold. We then compute for the two candidate nodes their new entropy. H(Y|X) is the weighed average of these child node entropies, the weights being the proportions of examples that went into the children nodes. The procedure is iterated until all the nodes are "pure". The terminal nodes are labeled by the class of their members. To perform a classification of a new example, one lets the example go down the tree (going right or left according to the thresholds on the selected features), and classify it according to the label of the terminal node in which it ends up.

149. What is the connection between MI, channel capacity, and rate distorsion theory?

The channel capacity is defined as the maximum MI between the input and output of a noisy channel, over all input distributions. The rate distorsion function is the minimum MI between and signal and a representation of this signal ensuring data reproducibility with a minimum loss.

150. What is the information bottleneck method?

We consider a signal X, a new feature representation Phi(X) and an outcome Y. To maximize transmission (thus minimize error rate), we should maximize MI(X,Y | Phi)=MI(Phi,Y). Simultaneously, if we want a compact representation, we need to minimize the rate of distorsion measured by MI(X, Phi). This leads to the following optimization problem: min MI(X, Phi) - beta MI(Phi, Y)

151. How does MI relate to the Pearson correlation coefficient for Gaussian signals?

$MI = -(1/2) \log (1-R^2)$

152. Can you suggest methods of forward selection or backward elimination using MI?

- Forward selection 1. One method was described in class: we start with the feature having maximum MI with the target. Subsequent features are added by selecting the feature having maximal MI with the target, conditioned on each previously selected feature (not conditioned on all the previously selected features simultaneously). This works well for binary features, MI is hard to estimate otherwise.

- Backward elimination: One could use the Torkkola method described in class, starting with all scaling factors equal to one. Instead of adjusting the scaling factors by gradient descent, one can eliminate the feature with smallest gradient. By iterating, features are progressively eliminated.

- Forward selection 1: Same idea but starting with all the scaling factors at zero and adding the feature corresponding to the largest gradient. Iterate to progressively add features.

153. Can you suggest new embedded methods not described in class combining loss functions and models?

The answer to this question is not provided. Students having difficulties answering this question should come to the office hour.

Ensemble methods

154. What is usually referred to as a learning machine ensemble, mixture of experts, or committee machine?
$F(\mathbf{x})=\text{sum}_k \text{alpha}_k f_k(\mathbf{x})$, where $f_k(\mathbf{x})$ is an expert or a base learner (depending on the point of view)

155. Give examples of simple "base learners"
- Decision stumps: these are classifiers using a single variable and a threshold, which are equivalent to the root node of a tree classifier.
- Kernel basis functions.

156. What is the relation between ensemble methods and Bayesian learning?
In the Bayesian framework our model is an attempt to explain the data that can be "marginalized out": $P(y|\mathbf{x},D) = \text{sum}_f P(f|D) P(y|\mathbf{x},f,D)$. In this formula, $P(y|\mathbf{x},f,D)$ is our "base learner" and $P(f|D)$ is our weight. The "Bayesian" makes predictions according to $P(y|\mathbf{x},D)$, that is according to a vote of the base-learners with weight $P(f|D)$ measuring how confident they are of their prediction. This is also called the "weighted-majority" algorithm.

157. What is the relation between "stability" and ensemble methods?
Assume we have a training set D of size m. Let us consider an "ideal" committee built by voting among models of our model class trained with all possible training sets of size D. Loss functions can be split into two terms: a "bias" term (measuring the error or the ideal committe) and a "variance" term (measuring the discrepancy of the prediction of our model and that of the ideal committee). Both variance and bias must be reduced to ger good prediction error, but there is often a tradeoff. Committee work on reducing variance. Stable methods have low variance (predictors built with training sets of the same size make similar predictions). Noting that the "ideal committee" has zero variance, we see that it is very stable. Committee machines immitate the "ideal committee".

158. What makes a good "base learner"?
Looking at the bias-variance tradeoff (Question 4), we see that committees reduce variance, but not bias. In fact, the ideal committee has the same bias as the base learners. So to both reduce bias and variance, the base learners should have low bias.

159. What are MCMCs and how does one use them to train ensembles of learning machines?
MCMC stands for Markov Chain Monte Carlo. They are methods for "sampling" probability distributions. the can be used to train ensembles in the Bayesian sense by sampling the posterior distribution P(f|D). In this way, we get an empirical estimate of the weighted majority.

160. How does one use MCMCs for feature selection?
One can imagine a generative model in which subsets of the features are chosen with some probability distribution. One then can compute P(feature_set|D) by marginalizing out the models. Similarly, P(feature|D) can be computed by marginalizing out both the models and the other features.

161. What is "bagging"?
Bagging is a bootstrap method applied to learning ensemble of classifiers. Each base learner is train with a resampled training set. A resampled training set is obtained by sampling with replacement m examples in a training set of size m. All the base learners vote with the same weight.

162. How does one use the "out-of-bag" error estimate to create a feature ranking index and compute pvalues?
The out-of-bag error estimate is obtained by computing for each base learner the errors made on the examples not used for training (out-of-bag examples), then averaging the results for all base learners. The idea is the permute randomly the values of one feature in the out-of-bag examples

and compute the difference in error rate between the unperturbed and perturbed data. This difference may be used as a ranking index for features. If it is normalized by its standard error, it is approaximatly distributed with the Normal Law, which provides a means of computing a pvalue.

163. What are "random forests" (RF)?

Random Forests are ensembles of tree classifiers. each base learner is trained from a bootstrap sample. Additional variability is introduced by splitting nodes in the tree on the basis of on a random subset of the original features, not all the features. Typically, if there is a total of N features, one uses sqrt(N) features.

164. How does one exploit the built-in feature selection of tree classifiers to define a ranking index with RF?

At each node, the feature selected to split the data is the one providing the largest information gain, i.e. having largest mutual information with the target. An index is obtained for each feature by adding the information gains it did or would provide at each node split. An average index can be obtained by averaging the indices obtained from several trees in a forest.

165. What is "boosting"?

Boosting is a method of training an ensemble method by adding base learners in sequence. Each newly added base learner is trained with a higher proportion of the examples that were hard to learn up to then.

166. How does one use boosting and decision stumps to feature selection?

One can use decision stumps as base learners. A decision stump is a classifier built from one variable (like the root node of a tree classifier). Boosting with decision stumps is a form of forward selection of variables.

167. What are generic methods of combining sets of features selected with various methods?

- Averaging ranking indices to get a new global index
- Averaging ranks to get a new global index
- Intersecting feature subsets (soft intersections may be defined as setting a threshold on the number of times a feature happens in all subsets considered)
- Computing the "centroid" of feature subsets (the subset intersecting most with all others)
- Computing the "centroid" of feature rankings (the ranking closest to all other rankings)