Linear Models (Linear Regression, Logistic Regression, Softmax)

- Supervised learning
    - What, why, how
        - Machinelearning learns mathematical models from data
    - What is supervised learning and the model
        - D = (x,y). X = input/features. Y = target (classicication or regression)
        - Target function h(x) =~ f(x)
        - error in and out and Error functions
            - Least squares
            - Capture the problem we consider (FBI)...
    - Linear models:
        - Linear regression: real number. h(x) = w^t x
        - Linear classification: binary decision. h(x) = sign(w^t x)
        - Logistic regression: Probability of class. h(x) = o- (w^t x)
- Linear Regression,
    - What does it. Example, house price vs size
    - h(x) = w^t x
    - Error + typical not on w0
    - Minimize (why good idea, and how -v-)
- Optimization / gradient descent
    - Minimize Ein
        - Formel, Compute, Solve for diff = 0
        - Global or local minimizer we found?
    - Convexity
        - (what) What is convex
        - (why) If tangent is below the function (bowl) -> diff f(x) = 0 is a global minimizer
        - (why) why do we want this)
        - (how) tjek doubelt diff er >= 0 for alle x. f(x) = x^2 -> f''(x) = 2 >= 0.

          Affine Functions:

          $$f(x) = w^\intercal x + c, \quad H_f(x) = 0$$

          Quadratic Functions:

          $$f(x) = \frac{1}{2}x^\intercal Ax + b^\intercal x + c, \quad H_f(x) = A$$

        - Iff A is positive semidefinite

$$E_{in}(w) = \frac{1}{n} \sum_{(x,y)\in D} (w^\intercal x - y)^2 \quad = \frac{1}{n}(Xw - y)^\intercal (Xw - y)$$

$$= w^\intercal (X^\intercal X) w + (-2y^\intercal X)^\intercal w + y^\intercal y$$

(1/n died mistakenly, luckily it does not matter)

        -

Convexity of Linear Regression

Quadratic Functions: $f(x) = \frac{1}{2} x^\intercal A x + b^\intercal x + c$

Convex if A is positive semidefinite

$$E_{in}(w) = w^\intercal (X^\intercal X) w + (-2y^\intercal X)^\intercal w + y^\intercal y$$

$X^\intercal X$ positive semidefinite?

Symmetric: Clearly

$$w^\intercal X^\intercal X w = (Xw)^\intercal Xw = ||Xw||_2^2 \geq 0$$

Found Global Minimum!

        -
        - Convex = good (usually)

- *Logistic Regression*
    - Target function may be probaliststic
    - Soft threshold
    - Example: Probability of him paying loan back.
    - Can be classifier, return most likely

Likelihood of the data. NOT $P(\theta \mid D)$

$$P(D \mid \theta) = \prod_{(x,y)\in D} p(y \mid x) = \prod_{(x,y)\in D} \sigma(\theta^\intercal x)^y (1 - \sigma(\theta^\intercal x))^{1-y}$$

$$\mathrm{NLL}(D \mid \theta) = -\sum_{x,y\in D} (y \ln(\sigma(\theta^\intercal x)) + (1-y)\ln(1 - \sigma(\theta^\intercal x))$$

$$E_{\text{in}} = \frac{1}{|D|}\mathrm{NLL}$$

Neg. Log likelihood is convex
Exercise next week
(Named Cross Entropy)

    - ??

- Non-linear data
    - Preprocessing / transformation of features.
    - Ø og Ø^-1
- *Regularization?*

Learning Theory (VC Dimension, Bias Variance, Regularization and Validation)

- Supervised learning
    - What, why, how
        - Machinelearning learns mathematical models from data
    - What is supervised learning and the model
        - D = (x,y). X = input/features. Y = target (classicication or regression)
        - Target function h(x) =~ f(x)
        - error in and out and Error functions
            - Least squares
            - Capture the problem we consider (FBI)...
    - 1) Ensure low Ein and 2) have Eout close to Ein
    - Hypothesis generalizes if Ein and Eout close
    - Balance between model complexity and Ein -> Eout
    - Hypothesis space / model complexity
        - Many functions can fit some data, how to find correct one
    - We want to rid the error of the dataset
- *Hoeffding's Inequality?*
    - Ingen garantier, kun sandsynligheder
    - Coin flip example
    - P(|v-my| > epsilon) < 2e^(-2 * epsilon^2 * N)   v = sample mean, my = coin bias
    - Overfør til classification problem. 1-0 loss. **Generalization bound**
        - Fixed hypothesis
        - P(|Ein - Eout| > epsilon) < 2e^(-2 * epsilon^2 * N)
    - Udvidelse til P(|Ein - Eout| > epsilon) < 2 **M** e^(-2 * epsilon^2 * N)
        - More hypothesis -> better fit -> worse generalization bound
- VC-dimension bound
    - Upper bound Eout, (propability)
    - Eout <= Ein + Omega (complexity term, VC(H))
    - Dichotomy
        - |{h(x_1)...h(x_N) in {0,1}^N | h in H}| <= 2^N
        - RHS - længde || = skrevt H(x_1...x_N)
    - Growth function
        - mH(N) = max x_1 to x_n on dichotomy
    - Breakpoint
        - K is breakpoint if mH(N) < 2^K
    - VC(H) = max N such that mH(N) = 2^N
    - 2D example 3 points can always be split into all dichotomies, 4 cannot

- $$E_{\text{out}}(g) \le E_{\text{in}}(g) + \sqrt{\frac{8}{N}\ln\frac{4m_{\mathcal{H}}(2N)}{\delta}}$$

  Independent of learning algorithm, target function, input distribution
- $E_{out}(h) \le E_{in}(h) + \Omega(N, \mathcal{H}, \delta)$

- Regularization
  - We don't want to fit the noise
  - Tell the learning process to use smaller weights, unless it improves the result by a lot
  - L2
- *Validation*
  - *Cross-validation*

Support Vector Machines (Kernels)

- Supervised learning
  - What, why, how
    - Machinelearning learns mathematical models from data
  - What is supervised learning and the model
    - D = (x,y). X = input/features. Y = target (classicication or regression)
    - Target function h(x) =~ f(x)
  - $D = \{(x, y) \mid x \in \mathbb{R}^d, y \in \{-1, 1\}\}$
  - Antag linear sceperable, many dimension makes it more likely, kernels.
  - One of the best of the box
- Functional margin
  - Idea, make margin
  - $\hat{\gamma}^{(i)} = y^{(i)}(w^T x + b)$
  - Correct prediction: $y_i(w^\intercal x_i + b) > 0$
  - Support vectors
- Geometrisk margin
  - If ||w|| = 1 så er de ens
  - Is invarient to rescaling of the parameters
  - Define same y uden hat ^
  - $\gamma = \frac{\hat{\gamma}}{||w||}$
- Traning
  - Max y mens vi overholder prediction
  - Dual form lets us make efficient algorithm for solving the problem we have
  - Lagrange multipliers
  - SVM finds the maximally separating hyperplane and is defined by "few" support vectors
  - The dual optimization problem finds the support vectors (dual also convex)
  - New points are classified by their weighed sum of inner products with the support vectors
  -
- Kernels
  - Feature mapping
  - K(x,z) = Ø(x)^t Ø(z)
  - Fordi algo bruger kun Inner products kan vi
  - Work in higher dimension space for free

$$x = (x_1, x_2), z = (z_1, z_2)$$

Polynomial kernel of degree 2

$$\begin{aligned} K(x,z) &= (1 + x^\intercal z)^2 \\ &= (1 + x_1 z_1 + x_2 z_2)^2 \\ &= \underline{1 + x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 z_1 + 2x_2 z_2 + 2x_1 z_1 x_2 z_2} \end{aligned}$$

Feature Transform

$$\begin{aligned} (x_1, x_2) &\xrightarrow{\Phi} (1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2) \\ \langle x, z \rangle_\Phi &= \phi(x)^\intercal \phi(z) \\ &= \underline{1 + x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 z_1 + 2x_2 z_2 + 2x_1 z_1 x_2 z_2} \end{aligned}$$

- Slackness

$$\min_{w,b,\xi} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i$$

S. To $\quad \underline{y_i(w^\intercal x_i + b) \geq 1} - \xi_i \quad \forall i$

$$\xi_i \geq 0$$

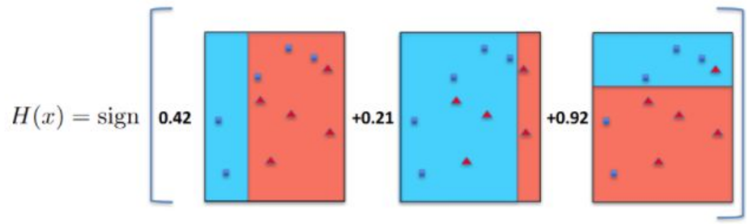
  - Little like regurlazion
  - Bedre imod outliers

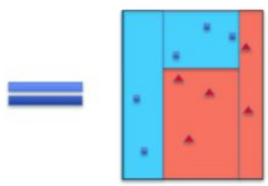
Neural Nets (Backpropagation, Deep Convolutional Nets)

- Neuron
    - Activation functions
    - NN CAN approximate any continuous function (weak and useless statement in practice)
    - Neural Networks are **universal function approximators**
    - Deeper than 3 levels does not help much
    - Overfitting? Add regularization and other things are better.
- Standard feed forward network
    - Layers, input
    - $\mathrm{nn}(X) = F_3(W_3 F_2(W_2 F_1(X W_1 + b_1) + b_2) + b_3)$
    - Non-linear, else could have been one matrix multiplication
    - Weights are learned with stochastic gradient descent
    - their gradients are derived with chain rule (and computed with backpropagation).
    - The derivative on each variable tells you the sensitivity of the whole expression on its value.
- Backpropagation
    - Example
    - Not convex
    - Gradian decent - chain rule
    - Late note: preprocessing matter, affects learning rate.
    - Early stopping
- Convolutional net
    - Convolutional pooling layers -> feature detector
        - Images
        - Edge detector
- *Other things*
    - recurrent networks
        - can remember, feedback
        - Good for time series, text
    - Dropout traning

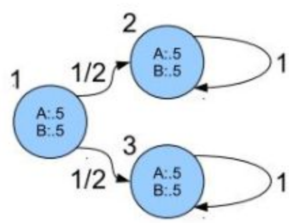
Decision Trees and Ensemble Methods (Bagging, Boosting)

- Supervised learning
    - What, why, how
        - Machinelearning learns mathematical models from data
    - What is supervised learning and the model
        - D = (x,y). X = input/features. Y = target (classicication or regression)
        - Target function h(x) =~ f(x)
        - error in and out and Error functions
            - Least squares
            - 1/0-loss
            - Capture the problem we consider (FBI)...
- Decision trees
    - Basic Construction (building blocks)
    - Good at handleing both classification and regression
    - In base form, a white-box. Easy to interpret
    - Can (sometimes) be plotted in 2d
- Construction
    - Many trees same function
    - Small trees should generalize better, because less rules
    - Select next attribute greedily. As in close to perfect split
    - Split that minimizes error on next level
    - Tend to overfit
    - Bias variance decomposition
- Bagging / Bootstap Aggregating (Minimize variance)
    - 1) Generate additional data from dataset
    - 2) train
    - 3) return majority vote or return mean prediction
    - Eout og bias variance decomposition. Bias should remain the same.
    - Any reducing in error should come from reducing variance.
    - Random forest. Loss white-box. Knowledge of the masses
- Boosting (Mainly reduce bias)
    - Boosting - Make weak learners to strong learners
    - Weak learner
        - someone that does slightly better than random
        - Small tree (2 layers maybe)
        - Have LOW variance, usually not overfitting
        - High bias because they underfit
    - Boosting: Make weak learner into strong, by combining (weak) base classifiers into powerful committee.

- Each classifier should be good at different parts of input space
- Goal: output weighted combinations classifiers that best achieves goal
- $$h(x) = \text{sign}\left(\sum_t \alpha_t h_t(x)\right)$$
- 


- They do better than one hypothesis both in practice and theory.

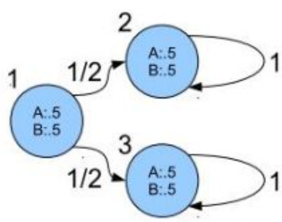
Hidden Markov Models - Decoding (Basic algorithms and applications, Viterbi and posterior decoding)

- Simpel models
  - Ovsercations are independent and identically distributed (I.I.D.)
    - Propability
  - 1st order Markov
    - Propability
- Hidden Markov model
  - The 3 tabels. Emmission, start, and transition
  - The joint probability
  - Example
    - 

  - Number procision
    - Log-space
    - scaling
- Decoding (what is)
  - Finding the underlying explanation
- Viterbi decoding
  - Most likely explanation. SYNTACTICALLY correct
  - Uses w(omega) table
  - Dynamic programming
- Posterior
  - Most likely state to be in in the n'th step
  - Alpha (forward algo) og beta (backward algo)

Hidden Markov Models - Training (Basic algorithms and applications, building models and selecting initial model parameters)

- Simpel models
  - Ovsercations are independent and identically distributed (I.I.D.)
    - Propability
  - 1st order Markov
    - Propability
- Hidden Markov model
  - The 3 tabels. Emmission, start, and transition
  - The joint probability
  - Example

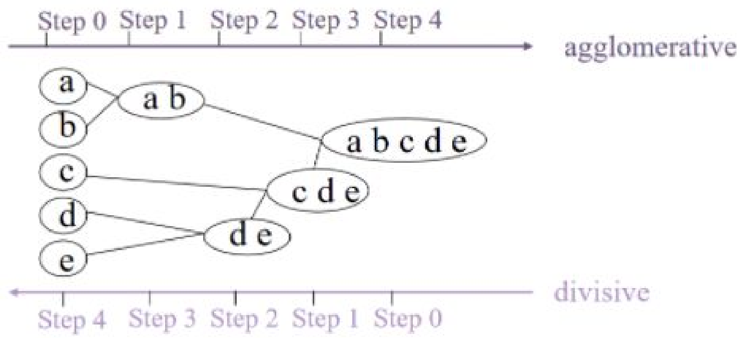


  - Number procision
    - Log-space
    - Scaling
- Building a model
  - Talk about DNA handin
  - Domain knowledge
  - We can define sequences, setting all output to 1
- Training
  - Counting and Psudo-counting
  - We have a lot of X (observations) and Z
  - Intuition: Parameters should reflect what we have seen
  - Emission and transition
- Viterbi training
  - Viterbi decoding = Most likely explanation. SYNTACTICALLY correct
  - Finds local maximum of p(X,Z|HMM)
  -
    1. Decide on some initial parameter θ^0
    2. Find the most likely sequence of states **Z*** explaining **X** using the the Viterbi Algorithm and the current parameters θ^i
    3. Update parameters to θ^{i+1} by "counting" (with pseudo counts) according to (**X**,**Z***).
    4. Repeat 2-3 until P(**X**,**Z*** | θ^i) is satisfactory (or the Viterbi sequence of states does not change).

Unsupervised Learning - Clustering

- General clustering
  - Group set of data
  - Unsupervised (no labels)
  - Trying all partitions to minimize some function is too expensive
  - Example, what is a good cluster and a bad one.
- K-means clustering
  - Partitioning algorithm, find k partitions and minimize function.
  - Imprice iteratively
  - Centroid of a cluster $\mu_C = \frac{1}{|C|} \sum_{x_i \in C} x_i$
  - Cluster compactness measure $TD(C_j) = \sqrt{\sum_{p \in C_j} dist(p, \mu_{C_j})^2}$
  - Clustering compactness measure (ideal clustering minimizes this)

    $$TD = \sqrt{\sum_{j=1}^{k} TD^2(C_j)}$$
  - Lloyd's algorithm
    - + Relatively efficient
    - - Specify k
  - Clusters forced to have convex shape
  - Result and runtime dependent on initial partition
  - Often tends at local optimum
  - K-medoid
    - Better against noise
    - Works in spaces where mean not defined, L1 norm (manhatten dis)
  - Selecting k: Silhouette coefficient. Increasing k will always make TD better
- Density-based clustering
  - Cluster are dense regions separated by non-dense regions
  - Example: Snake example?
  - Outliers
  - Core objects. MinPts and epsilon needs to be learned
  - Epsilon: Graph and take first valley - minPts have heuristic math formel.
- Hierarchical clustering
  - Clustering at different level
  - Single-link (min), complete (max), avg

- 
Step 0 Step 1 Step 2 Step 3 Step 4
agglomerative
a
b
c
d
e
a b
d e
c d e
a b c d e
divisive
Step 4 Step 3 Step 2 Step 1 Step 0

Unsupervised Learning - Outlier Detection and Dimensionality Reduction

- *Outlier*
    - A things that’s so different that it might be an error
    - Meaning
        - Might be fraud
        - Surveillance
        - Data cleaning
    - Be careful of removing outliers, might be valid data and if you don’t notice you might learn something incorrect.
- Unsupervised outlier detection
    - Similar to clustering
    - Perspektiver til **Density-based clustering**
        - Every object not in a cluster is an outlier
    - **Distance based**
        - DB(precentage, distance)
        - Global based
- Local outlier
    - Compare distance to neighbors based on own distance
    - Note angle in high dimension
- Dimensionality reduction
    - Why: Reduce data, but keep structure/information. Make the data cheaper/better/easier to handle.
    - Project data to lower space.
    - We want to keep variance
    - Number billeder. PCA (principal components) 784 dim -> 49 dim
    - Preprocessing: scale and zero mean