SSU

- MINIMALIZACE EMPIRICKÉHO RIZIKA
- ODHAD MAXIMÁLNÍ VĚROHODNOSTI
- EM ALGORITMUS
- HLUBOKÉ SÍTĚ A JEJICH UČENÍ
- STANDARDNÍ A HLUBOKÉ NEURONOVÉ SÍTĚ A JEJICH UČENÍ

- OBJECT FEATURES
  - $x \in X$
  - ARE OBSERVABLE
- STATE OF OBJECT
  - $y \in Y$
  - USUALLY HIDDEN
- PREDICTION STRATEGY
  - INFERENCE RULE
  - $h: X \to Y$
  - IF y IS
    - CATEGORIAL VARIABLE
      - CLASSIFICATION
    - REAL VALUED VARIABLE
      - REGRESSION
- TRAINING EXAMPLES
  - $\{(x, y) \mid x \in X, y \in Y\}$
- LOSS FUNCTION
  - $L: Y \times Y \to \mathbb{R}_+$
  - PENALISES WRONG ANSWERS (PREDICTIONS)
  - $L(y, h(x))$ IS LOSS FOR PREDICTING $y' = h(x)$ IF TRUE STATE IS y

- GOAL IS TO FIND OPTIMAL PREDICTION STRATEGY h
  - THAT MINIMALISES COST

- STATISTICAL MACHINE LEARNING
  - X, Y ARE RANDOM VARIABLES
    - THEY ARE JOINED BY UNKNOWN PROBABILITY DISTRIBUTION FUNCTION $p(x, y)$
    - WE CAN COLLECT EXAMPLES (x, y) DRAWN FROM $p(x, y)$
  - TYPICAL CONCEPTS
    - REGRESSION

      $$y = f(x) + \varepsilon$$

      - WHERE f IS UNKNOWN AND ε IS RANDOM ERROR
    - CLASSIFICATION

      $$p(x, y) = p(y) \cdot p(x \mid y)$$

      - WHERE $p(y)$ IS PRIOR CLASS PROBABILITY AND $p(x|y)$ IS CONDITIONAL FEATURE DISTRIBUTION
  - RISK OF INFERENCE RULE

$$R(h) = E\,l(Y, h(X)) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \cdot l(y, h(x))$$

  - HOW TO ESTIMATE IT, IF $p(x, y)$ IS NOT KNOWN
    - TESTING SET
      - COLLECT I. I. D. TEST SAMPLES $S^l = \{(x^i, y^i) \in X \times Y \mid i = 1 \dots l\}$ DRAWN FROM $p(x, y)$
      - ESTIMATE BY EMPIRICAL RISK

$$R(h) \approx R_{S^l}(h) = \frac{1}{l} \sum^{l} l(y^i, h(x^i))$$

- CHOOSING OPTIMAL INFERENCE RULE $h(x)$

  - IF $p(x,y)$ IS KNOWN

    - BEST POSSIBLE RISK IS

$$R^* = \inf_{h \in Y^X} R(h) = \inf_{h \in Y^X} \sum_{x \in X} \sum_{y \in Y} p(x,y)\, l(y, h(x)) = \sum_{x \in X} p(x) \inf_{y' \in Y} \sum_{y \in Y} p(y|x) l(y,y')$$

    - THE BEST INFERENCE RULE IS BAYES RULE

$$h^*(x) = \operatorname{argmin}_{y' \in Y} \sum_{y \in Y} p(y|x) \cdot l(y,y')$$

    - BUT $p(x,y)$ IS NOT KNOWN

- LEARNING TYPES

  - TRAINING DATA

    - $T^m = \{(x^i, y^i) \in X \times Y \mid i = 1, \dots, m\}$ ... SUPERVISED DATA

    - $T^m = \{x^i \in X \mid i = 1, \dots, m\}$ ... UNSUPERVISED DATA

    - $T^m = T_l^{m_1} \cup T_u^{m_2}$, WHERE $T_l^{m_1}$ ARE LABELED AND $T_u^{m_2}$ ARE UNLABELED ... SEMI-SUPERVISED LEARNING

  - PRIOR KNOWLEDGE ABOUT TASK

    - DISCRIMINATIVE LEARNING

      - OPTIMAL INFERENCE RULE h^* IS IN SOME CLASS OF RULES H

      - REPLACE TRUE RISK BY EMPIRICAL RISK

$$R_T(h) = \frac{1}{|T|} \sum_{(x,y) \in T} l(y, h(x))$$

      - MINIMALIZE WITH RESPECT TO $h \in H$

$$h_T^* = \operatorname{argmin}_{h \in H} R_T(h)$$

      - SVM, DNN, STRUCTURED OUTPUT SVM

- GENERATIVE LEARNING
  - TRUE $P(x, y)$ IS IN SOME PARAMETRISED FAMILY OF DISTRIBUTIONS
  - $P = P_{\theta^*} \in \mathcal{P}$ FOR EXAMPLE
  - SPLIT DEPENDENCE ON UNKNOWN $\theta \in \Theta$ AND RANDOM T
    - 1. $\theta_T^* = \underset{\theta \in \Theta}{\operatorname{ARGMAX}}\ P_\theta(T)$
      - MAXIMUM LIKELIHOOD ESTIMATOR
    - 2. SET $h_T^* = h_{\theta_T^*}$ WHERE h_θ DENOTES BAYES INFERENCE RULE FOR PROBABILITY DISTRIBUTION P_θ
  - MIXTURE MODELS, HIDEN MARKOV MODEL, MARKOV RANDOM FIELD

- EMPIRICAL RISK MINIMIZATION
  - WE WANT TO FIND PREDICTION STRATEGY WITH MINIMAL EXPECTED RISK

$$R(h) = \int_{\mathcal{Y}} \mathcal{L}(y, h(x))\, P(x, y)\, dx = \underset{(x,y)\sim P}{E}\left(\mathcal{L}(y, h(x))\right)$$

  - BUT WE DON'T KNOW $P(x, y)$
  - SO WE CAN MINIMALIZE EMPIRICAL RISK

$$R_{S^\ell}(h) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(y^i, h(x^i))$$

  - IS IT GOOD APPROXIMATION?

- LAW OF LARGE NUMBERS
  - ARITHMETIC MEAN OF RESULTS OF RANDOM TRIALS GETS CLOSER TO THE EXPECTED VALUE AS MORE TRIALS ARE PERFORMED
  - HOEFFDING INEQUALITY
    - LET $\{z^1, \dots, z^\ell\} \in [a, b]^\ell$ BE REALIZATIONS OF INDEPENDENT RANDOM VARIABLES WITH SAME EXPECTED VALUE μ. THEN FOR ANY $\varepsilon > 0$ IT HOLDS THAT:

$$P\left(\left|\frac{1}{\ell}\sum_{i=1}^{\ell} z^i - \mu\right| \geq \varepsilon\right) \leq 2e^{-\frac{2\ell\varepsilon^2}{(b-a)^2}}$$

"PROBABILITY OF RESULT BEING DIFFERENT THAN ε IS SMALLER THAN..."

- FOR WHICH ε IS μ IN INTERVAL $(\hat{\mu}_\ell - \varepsilon, \hat{\mu}_\ell + \varepsilon)$ WITH PROBABILITY AT LEAST η?

$$P(|\hat{\mu}_\ell - \mu| < \varepsilon) = 1 - P(|\hat{\mu}_\ell - \mu| \geq \varepsilon) \geq 1 - 2e^{-\frac{2\ell\varepsilon^2}{(b-a)^2}} = \eta$$

- FROM THIS

$$\varepsilon = |b-a| \sqrt{\frac{\log(2) - \log(1-\eta)}{2\ell}}$$

- MINIMAL NUMBER OF SAMPLES TO μ BE IN $(\hat{\mu}_\ell - \varepsilon, \hat{\mu}_\ell + \varepsilon)$ WITH PROBABILITY AT LEAST η

$$\ell = \frac{\log(2) - \log(1-\eta)}{2\varepsilon^2} (b-a)^2$$

- PROBABILITY OF BEING BAD TEST SET CAN BE BOUNDED BY
  - ℓ_{MIN}, ℓ_{MAX} IS MAXIMAL AND MINIMAL VALUE OF LOSS FUNCTION

$$P\left(|R_{S_\ell}(h) - R(h)| \geq \varepsilon\right) \leq 2e^{-\frac{2\ell\varepsilon^2}{(\ell_{MIN} - \ell_{MAX})^2}}$$

- LEARNING BY EMPIRICAL RISK MINIMALIZATION
  - ERM
  - EMPIRICAL RISK

$$R_{T_m}(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(y^i, h(x^i))$$

  - ERM LEARNING ALGORITHM RETURNS h_m

$$h_m \in \underset{h \in H}{\operatorname{ARGMIN}}\ R_{T_m}(h)$$

  - DEPENDING ON H AND ℓ WE GET INSTANCES OF
    - SVM
    - LINEAR REGRESSION
    - LOGISTIC REGRESSION
    - NN
    - ANN BASED

- BUT IF WE DON'T CONSTRAIN H
  - WE DON'T HAVE ANY GUARANTEE THAT EMPIRICAL RISK IS GOOD APROXIMATION OF TRUE RISK
    - AND REGARDLES OF THE NUMBER OF EXAMPLES
  - IF WE DON'T THROW AWAY MEMORIZING PREDICTING STRATEGY FOR EXAMPLE

- EXCESS ERROR
  - DEVIATION OF LEARNED PREDICTOR FROM THE BEST ONE

$$\underbrace{\left(R(h_m) - R^*\right)}_{\text{EXCESS ERROR}} = \underbrace{\left(R(h_m) - R(h_H)\right)}_{\substack{\text{ESTIMATION ERROR} \\ \text{"ERROR BY NOT PROPERLY SELECTED FROM } H\text{"}}} + \underbrace{\left(R(h_H) - R^*\right)}_{\substack{\text{APPROXIMATION ERROR} \\ \text{"ERROR BY BOUNDING SPACE OF } H\text{"}}}$$

(BEST ATTAINABLE RISK; LEARNED PREDICTOR; BEST IN H PREDICTOR)

- STATISTICALLY CONSISTENT LEARNING ALGORITHM
  - IF FOR ANY $P(x, y)$ AND $\varepsilon > 0$ HOLDS:

$$\lim_{m \to \infty} P\left(R(h_m) - R(h_H) \geq \varepsilon\right) = 0$$

  - "WE CAN MAKE ESTIMATION ERROR ARBITRARLY SMALL IF WE HAVE ENOUGHT EXAMPLES"

  - COMBINING WITH UNIFORM LAW OF LARGE NUMBERS
    - "PROBABILITY OF SEING "BAD TRAINING SET" FOR AT LEAST ONE HYPOTHESIS FROM H CAN BE MADE ARBITRARLY LOW IF WE HAVE ENOUGHT EXAMPLES"

- LINEAR CLASSIFIER

$$h(x; w, b) = \text{SIGN}\left(\langle w, \phi(x)\rangle + b\right) = \begin{cases} +1 & \text{IF } \langle w, \phi(x)\rangle + b \geq 0 \\ -1 & \text{IF } \langle w, \phi(x)\rangle + b < 0 \end{cases}$$

(ϕ: FEATURES)

- WITH MINIMAL EXPECTED RISK

$$R^{0/1}(h) = E_{(x,y)\sim P}\left(\ell^{0/1}(y, h(x))\right) \quad \text{WHERE } \ell^{0/1}(y, y') = [\![y \neq y']\!]$$

- ERM LEADS TO

$$(w^*, b^*) \in \underset{(w,b)\in(\mathbb{R}^n \times \mathbb{R})}{\text{ARGMIN}} \; R^{0/1}_{T^m}\left(h(\cdot; w, b)\right)$$

- WHERE EMPIRICAL RISK IS

$$R^{0/1}_{T^m}\left(h(\cdot; w, b)\right) = \frac{1}{m}\sum_{i=1}^{m} [\![y^i \neq h(x^i; w, b)]\!]$$

- VAPNIK - CHERVONENKIS DIMENSION

- VC
- LET $H \subseteq \{-1, +1\}^X$ AND $\{x^1, \dots, x^m\} \in X^m$ BE A SET OF m INPUT OBSERVATIONS
- THE SET $\{x^1, \dots, x^m\}$ IS SHATTERED BY H IF FOR ALL $y \in \{+1, -1\}^m$ THERE EXISTS $h \in H$ SUCH THAT $h(x^i) = y^i$, $i \in \{1, \dots, m\}$
- VC DIMENSION OF H IS CARDINALITY OF LARGEST SET OF POINTS FROM X WHICH CAN BE SHATTERED BY H.
- VC DIMENSION OF LINEAR CLASSIFIER OPERATING IN n-DIMENSIONAL SPACE IS $n+1$

- EXAMPLES ARE LINEARLY SEPARABLE IF THERE EXISTS $(w, b) \in \mathbb{R}^{n+1}$ SUCH THAT

$$y^i\left(\langle w, \phi(x^i)\rangle + b\right) > 0, \quad i \in \{1, \dots, m\}$$

- CAN BE SOLVED BY PERCEPTRON ALGORITHM

- SVM

  - PRIMAL PROBLEM

    - ERM OF LINEAR CLASSIFIER

$$(w', b') = \underset{\|w\| \le r,\ b \in \mathbb{R}}{\operatorname{ARGMIN}} \left(\frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y^i(\langle w, \phi(x^i) \rangle + b)\} \right)$$

    - IT CAN BE FORMULATED AS EQUIVALENT QUADRATIC PROGRAM

(λ: REGULARIZATION CONSTANT)

$$(w^*, b^*, \xi^*) = \underset{\substack{(w,b) \in \mathbb{R}^{n+1} \\ \xi \in \mathbb{R}^m}}{\operatorname{ARGMIN}} \left(\frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right)$$

s.t.

$$y^i(\langle w, \phi(x^i) \rangle + b) \ge 1 - \xi_i \qquad i \in \{1, \dots, m\}$$

$$\xi_i \ge 0$$

  - DUAL PROBLEM

    - CONVEX QUADRATIC PROGRAM

$$\alpha^* = \underset{\alpha \in \mathbb{R}^m}{\operatorname{ARGMAX}} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2\lambda} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^i y^j \langle \phi(x^i), \phi(x^j) \rangle \right)$$

s.t. $\sum_{i=1}^{m} \alpha_i y^i = 0$

$$0 \le \alpha_i \le \frac{1}{m} \qquad i \in \{1, \dots, m\}$$

    - THE SOLUTION OF PRIMAL SOLUTION (w^*, b^*) IS OBTAINED FROM $\alpha(\lambda)$ BY

$$w^* = \sum_{i=1}^{m} y^i \phi(x^i) \alpha_i^* \quad , \quad b^* = y^i - \langle w^*, \phi(x^i) \rangle \quad \forall i \in I_{SV}^{<}$$

WHERE $I_{SV}^{<} = \{ i \in \{1, \dots, m\} \mid 0 < \alpha_i^* < \frac{1}{m} \}$ ARE BOUNDARY SVs

    - TO REPRESENT THE CLASSIFIER WE NEED ONLY SUPPORT VECTORS

      - TRAINING EXAMPLES WITH INDICES $I_{SV} = \{ i \in \{1, \dots, m\} \mid \alpha_i^* > 0 \}$

- KERNEL SVM

- WE CAN USE KERNEL FUNCTION

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

- WE CAN TRAIN SVM ONLY USING KERNEL FUNCTIONS

$$\alpha^* = \underset{\alpha \in \mathbb{R}^m}{\operatorname{ARGMIN}} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^i y^j k(x^i, y^j) \right)$$

↑ INSTEAD OF $\langle \phi(x), \phi(x') \rangle$

$$\text{S.t.} \sum_{i=1}^{m} \alpha_i y^i = 0$$

$$0 \leq \alpha_i \leq \frac{1}{m} \quad i \in \{1, \dots, m\}$$

- EVALUATION OF PREDICTION RULE

$$h(x; \alpha^*, b^*) = \text{SIGN}\left(\langle w^*, \phi(x) \rangle + b^* \right) = \text{SIGN}\left(\sum_{i \in I_{SV}} y^i \alpha_i^* k(x^i, x) + b^* \right)$$

- EXAMPLE OF KERNEL

$$k(x, x') = \langle \phi(x), \phi(x') \rangle = (\langle x, x' \rangle + 1)^2$$

- POLYNOMIAL OF d DEGREE

$$k(x, x') = \langle x, x' \rangle^p$$

- INHOMOGENOUS POLYNOMIAL OF p DEGREE

$$k(x, x') = (\langle x, x' \rangle + 1)^p$$

- GAUSSIAN KERNEL

$$k(x, x') = \exp\left(-\sigma \|x - x'\|^2\right)$$

- HILBERT SPACE

  - COMPLETE VECTOR SPACE WITH DOT PRODUCT $\langle \cdot, \cdot \rangle$

    $H \times H \rightarrow \mathbb{R}$

  - SATISFIES FOLLOWING REQUIREMENTS

    - SYMMETRY

      $$\langle f, g \rangle = \langle g, f \rangle \quad \forall f, g \in H$$

    - LINEARITY

      $$\langle \alpha_1 f_1 + \alpha_2 f_2, g \rangle = \alpha_1 \langle f_1, g \rangle + \alpha_2 \langle f_2, g \rangle \quad \forall f_1, f_2, g \in H$$

      $$\alpha_1, \alpha_2 \in \mathbb{R}$$

    - POSITIVE DEFINITENESS

      $$\langle f, f \rangle \geq 0 \quad \text{WITH EQUALITY IFF } f = 0$$

- DOT PRODUCT DEFINES A NORM $\|f\|_H = \sqrt{\langle f, f \rangle}$

- POSITIVE DEFINITE KERNEL

  - LET X BE NON-EMPTY SET

  - FUNCTION $\lambda : X \times X \rightarrow \mathbb{R}$ IS POSITIVE DEFINITE KERNEL IF IT IS

    - SYMETRIC

    - AND FOR ANY FINITE SET OF INPUTS $x^1, \dots, x^m$ THE KERNEL MATRIX $K \in \mathbb{R}^{m \times m}$ WITH ELEMENTS $K_{i,j} := \lambda(x^i, x^j)$ IS POSITIVE SEMI-DEFINITE

      $$K = \begin{pmatrix} \lambda(x^1, x^1) & \lambda(x^1, x^2) & \dots & \lambda(x^1, x^m) \\ \lambda(x^2, x^1) & \lambda(x^2, x^2) & \dots & \lambda(x^2, x^m) \\ \vdots & \vdots & & \vdots \\ \lambda(x^m, x^1) & \lambda(x^m, x^2) & & \lambda(x^m, x^m) \end{pmatrix}$$

      - POSITIVE SEMI-DEFINITE MATRIX $K \in \mathbb{R}^{m,m}$

        - $\forall \alpha \in \mathbb{R}^m : \alpha^T K \alpha \geq 0$

- NEURAL NETWORKS

$$\hat{y} = f(s) = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

- WE CAN ADD BIAS TO w_1 AND ENHANCE EACH x BY $(1, x)$

- ACTIVATION FUNCTIONS

- SIGMOID

$$\sigma(s) = \frac{1}{1+e^{-s}} = \frac{e^s}{e^s+1}$$

- TANH

$$\tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} = 2\sigma(s) - 1$$

- RELU

$$f(s) = \max(0, s)$$

- LINEAR NEURON

- SINGLE NEURON WITH LINEAR ACTIVATION FUNCTION

- LINEAR REGRESSION

$$y = \langle x, w \rangle$$

- SQUARED ERROR

$$L(w) = \sum_{i=1}^{n} \underbrace{(y_i - \langle w, x_i \rangle)^2}_{\ell(y_i, \hat{y}_i)} = (y - Xw)^T (y - Xw)$$

- MINIMALIZATION OF $L(w)$

$$\frac{\partial L}{\partial w} = 0 \quad \dots \quad w^* = (X^T X)^{-1} X^T y$$

- LOGISTIC REGRESSION

  - NEURON USING SIGMOID ACTIVATION FUNCTION

$$\hat{y} = \sigma(\langle w, x \rangle)$$

  - y IS TARGET CLASS
  - $\hat{y}$ IS OUTPUT CLASS

  - LIKELIHOOD OF LOGISTIC REGRESION

$$P(y/w, x) = \prod_{i=1}^{m} \hat{y}_i^{y_i} (1-\hat{y}_i)^{1-y_i}$$

  - NEGATIVE LOG-LIKELIHOOD

$$L(w) = \sum_{i=1}^{m} \underbrace{-[y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i)]}_{\ell(y_i, \hat{y}_i)}$$

    - CROSS ENTROPY

  - $\frac{\partial L}{\partial w} = 0$ HAS NO ANALYTICAL SOLUTION

    - WE HAVE TO USE ANALYTICAL METHODS

- LINEAR LAYER

  - $Y = XW$

- SOFTMAX LAYER

  - MULTINOMIAL CLASSIFICATION
  - K MUTUALLY EXCLUSIVE CLASSES
  - $\sigma_2(s) \triangleq \frac{e^{s_2}}{\sum_{c=1}^{K} e^{s_c}}$
  - REPRESENTS PROBABILITY DISTRIBUTION
  - DESCRIBES CLASS MEMBERSHIP PROPERTIES $P(y = 2/s) = \sigma_2(s)$

- MULTINOMIAL LOGISTIC REGRESSION

  - LINEAR LAYER + SOFTMAX LAYER

  - $\hat{y}_k = \sigma_k(x^T W)$

  - CLASSIFIER $\hat{k}(x, W) = \underset{k}{\text{ARGMAX}}\ \hat{y}_k$

- LOSS FUNCTIONS

  - BINARY CLASSIFICATION

    - CROSS-ENTROPY

$$\left(-\sum_{i=1}^{m} [y_i \log \hat{y}_i + (1-y_i) \log (1-\hat{y}_i)]\right)$$

  - MULTINOMIAL CLASSIFICATION

    - MULTINOMIAL CROSS-ENTROPY

$$\left(-\sum_{i=1}^{m} \sum_{c=1}^{K} y_{ic} \log(\hat{y}_{ic})\right)$$

  - REGRESSION

    - SQUARED ERROR

$$\sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

  - MULTI-OUTPUT REGRESSION

    - SQUARED ERROR

$$\sum_{i=1}^{m} \sum_{c=1}^{K} (y_{ic} - \hat{y}_{ic})^2$$

- BACKPROPAGATION OVERVIEW

  - METHOD TO COMPUTE GRADIENT OF LOSS FUNCTION WITH RESPECT TO ITS PARAMETERS

    $$\nabla L(w)$$

  - GRADIENT IS USED BY OPTIMALIZATION METHODS

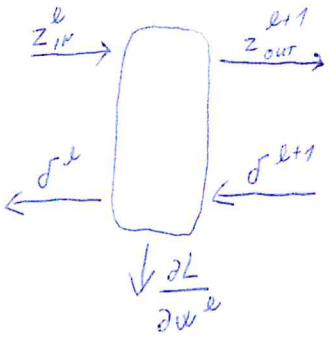
- EXAMPLE ON MULTINOMIAL LOGISTIC REGRESSION

  - LOSS IS MULTINOMIAL CROSS-ENTROPY

$$L(w) = -\sum_{i=1}^{m}\sum_{c=1}^{K} [\![y_i = c]\!] \log\left(\frac{\exp(\langle x_i, w_c\rangle)}{\sum_{z=1}^{K}\exp(\langle x_i, w_z\rangle)}\right)$$

- USAGE OF CHAIN RULE

- DIVIDE AND CONQUER APPROACH



- $\delta^{l} = \frac{\partial L}{\partial z^{l}}$ IS SENSITIVITY OF LOSS TO MODULE INPUT FOR LAYER l

  - THEN:

$$\delta_i^{l} = \frac{\partial L}{\partial z_i^{l}} = \sum_j \frac{\partial L}{\partial z_j^{l+1}} \cdot \frac{\partial z_j^{l+1}}{\partial z_i^{l}} = \sum_j \delta_j^{l+1} \frac{\partial z_j^{l+1}}{\partial z_i^{l}}$$

- WE CAN DO IT SIMILARLY FOR PARAMETERS

$$\frac{\partial L}{\partial w_i^{l}} = \sum_j \frac{\partial L}{\partial z_j^{l+1}} \cdot \frac{\partial z_j^{l+1}}{\partial w_i^{l}} = \sum_j \delta_j^{l+1} \frac{\partial z_j^{l+1}}{\partial w_i^{l}}$$

- FOR EACH LAYER, WE NEED TO SPECIFY

  - FORWARD PASS : $z^{l+1} = f(z^{l})$

  - BACKWARD PASS : $\frac{\partial z^{l+1}}{\partial z^{l}}$

  - PARAMETER PASS (OPTIONAL) : $\frac{\partial z^{l+1}}{\partial w^{l}}$

- EXAMPLE

  - LINEAR LAYER

    - FORWARD

$$z_j^{l+1} = \sum_{i=0}^{\tilde{m}} w_{ij} z_i^l \qquad j = 1 \dots K$$

    - BACKWARD

$$\frac{\partial z_j^{l+1}}{\partial z_i^l} = w_{ij} \qquad i = 0, \dots, m \qquad j = 1, \dots, K$$

    - PARAMETER

$$\frac{\partial z_j^{l+1}}{\partial w_{ir}} = [\![j = r]\!] z_i^l$$

  - SQUARED ERROR

    - FORWARD

$$z^{l+1} = \sum_{i=1}^{\tilde{m}} (y_i - z_i^l)^2$$

    - BACKWARD

$$\frac{\partial z^{l+1}}{\partial z_i^l} = -2 (y_i - m z_i^l) \qquad i \in \{1, \dots, m\}$$

- GRADIENT DESCENT

$$\theta^* = \underset{\theta}{\operatorname{ARGMIN}}\, L(\theta)$$

- "GOAL IS TO FIND PARAMETERS θ THAT WILL MINIMIZE THE LOSS"

$$\theta^{(t+1)} = \theta^{(t)} - \eta^{(t)} \nabla L(\theta^{(t)})$$

($\eta^{(t)}$: LEARNING RATE)

- UPDATING WEIGHTS

  - FULL LEARNING
    - AFTER ALL TRAINING EXAMPLES ARE USED
  - ONLINE LEARNING
    - AFTER EACH SAMPLE, BUT IT MAY HARM CONVERGENCE IN LATER STAGES OF TRAINING
  - BATCH LEARNING

- OTHER TYPES
  - MOMENTUM
    - SIMULATES INERTIA
  - ADAGRAD
    - ADAPTIVE GRADIENT METHOD
    - REDUCE LEARNING RATES FOR PARAMETERS HAVING HIGH VALUES OF GRADIENT
  - RMS PROP
    - SIMILAR TO ADAGRAD

- REGULARIZATION
  - DEALS WITH OVERFITTING
  - OTHER POSSIBILITIES
    - MORE DATA
    - SIMPLER MODEL

- L2
  - $L(w) = (y - Xw)^T (y - Xw) + \lambda w^T w$
    ↖ L2 REGULARIZATION
  - MINIMIZE THE WEIGHTS IN NETWORK

- L1
  - SUM ABSOLUTE VALUES OF WEIGHTS

- EARLY STOPING
  - USE VALIDATION SET
  - STOP IS VALIDATION LOSS STARTS TO GROW

- DROPOUT

- AUGMENTING DATASET

- DON'T INITIALIZE WEIGHTS TO ZERO
  - NO GRADIENT
- INITIALIZE BY SMALL NUMBERS
  - GAUSSIAN INITIALIZATION
  - WORKS OK FOR SHALLOW NETWORK
- PROBLEM OF VANISHING GRADIENT
- CONVOLUTIONAL NEURAL NETWORKS
  - USES A SMALL RECEPTIVE FIELD WHICH SHARES WEIGHTS
    - TRANSLATION EQUIVARIANCE
    - EACH RECEPTIVE FIELD USES MULTIPLE FILTERS
- STRIDE
  - HIGHER STRIDE PRODUCES SMALLER OUTPUT
- ZERO PADDING
  - CONVOLUTIONAL LAYER REDUCES SPATIAL SIZE OF OUTPUT
  - THIS IS FIXED BY ZERO PADDING
- NONLINEARITIES
  - SIGMOID
  - TANH
  - RELU
- MAX POOLING
  - REDUCES SPATIAL RESOLUTION
  - LESS PARAMETERS
  - IT HELPS WITH OVERFITTING
- LENET-5
- ALEX NET
- VGG NET
- GOOGLENET
- RES NET

- GENERATIVE LEARNING

  - MODEL JOINT PROBABILITY DISTRIBUTION $p(x, y)$

  - USE INFERENCE RULE

$$a(x) \in \underset{y \in Y}{\operatorname{ARGMAX}} \sum_{y' \in Y} p(y' \mid x)\, \ell(y', y)$$

  - LEARNING

    - IF DISTRIBUTION $p_\theta(x, y)$ IS KNOWN UP TO PARAMETERS $\theta \in \Theta$ ONLY

      - THAN ESTIMATE θ^* FROM TRAINING DATA

- GENERATIVE LEARNING MAKES STRONGER ASSUMPTIONS AND IS MORE DATA EFFICIENT WHEN ASSUMPTIONS ARE (NEARLY) CORRECT

- MLE

  - MAXIMUM LIKELIHOOD ESTIMATION

$$\theta^*(T^m) = \underset{\theta \in \Theta}{\operatorname{ARGMAX}} \frac{1}{m} \sum_{i=1}^{m} \log p_\theta(x^i) = \underset{\theta \in \Theta}{\operatorname{ARGMAX}} \frac{1}{m} \sum_{i=1}^{m} \log \sum_{y \in Y} p_\theta(x^i, y)$$

    - IF θ IS SINGLE PARAMETER OR VECTOR OF HOMOGENOUS PARAMETERS

      - MAXIMIZE LOG-LIKELIHOOD DIRECTLY

    - IF θ IS COLLECTION OF HETEROGENEOUS PARAMETERS

      - EXPECTATION MAXIMIZATION ALGORITHM (EM)

        - INTRODUCE AUXILIARY VARIABLES $\alpha_i(y) \geq 0$ s.t. $\sum_{y \in Y} \alpha_i(y) = 1$ FOR EACH SAMPLE

        - CONSTRUCT LOWER BOUND OF LOG-LIKELIHOOD $L(\theta, T^m) \geq L_B(\theta, \alpha, T^m)$

$$L(\theta, T^m) \geq L_B(\theta, \alpha, T^m)$$

$$L(\theta, T^m) = \frac{1}{m} \sum_{i=1}^{m} \log \sum_{y \in Y} p_\theta(x^i, y) = \frac{1}{m} \sum_{i=1}^{m} \log \sum_{y \in Y} \frac{\alpha_i(y)}{\alpha_i(y)} p_\theta(x^i, y) \geq$$

$$\geq L_B(\theta, \alpha, T^m) = \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in Y} \alpha_i(y) \log p_\theta(x^i, y) - \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in Y} \alpha_i(y) \log \alpha_i(y)$$

- MAXIMIZE LOWER BOUND BLOCK-COORDINATE ASCENT
  - INITIALIZE SOME $\theta^{(0)}$
  - E-STEP
    - FIX CURRENT $\theta^{(t)}$
    - MAXIMISE $L_B(\theta^{(t)}, \alpha, T^m)$ W.R.T. α-s

$$\alpha_i^{(t)}(y) = P_{\theta^{(t)}}(y | x^i)$$

  - M-STEP
    - FIX CURRENT $\alpha^{(t)}$ ABOVE
    - MAXIMIZE $L_B(\theta, \alpha^{(t)}, T^m)$ W.R.T. θ

$$\theta^{(t+1)} = \underset{\theta \in \Theta}{\text{ARGMAX}} \frac{1}{m} \sum_{i=1}^{m} \sum_{y \in Y} \alpha_i^{(t)}(y) \log P_\theta(x^i, y)$$

  - THE SEQUENCE OF LIKELIHOOD VALUES IS INCREASING AND SEQUENCE $\alpha^{(t)}$ IS CONVERGENT

- HIDDEN MARKOV MODELS
  - FOR
    - TEXT RECOGNITION
    - SPEECH RECOGNITION
    - ROBOT SELF LOCALISATION
  - $S = (s_1, s_2, \dots, s_m)$ IS SEQUENCE OF LENGTH m WITH ELEMENTS FROM FINITE SET K.
    - JOINT PROBABILITY DISTRIBUTION

$$P(s_1, s_2, \dots, s_m) = P(s_1) \cdot P(s_2 | s_1) \cdot P(s_3 | s_2, s_1) \dots P(s_m | s_1 \dots s_{m-1})$$

  - JOINT PROBABILITY DISTRIBUTION IS MARKOV MODEL IF

$$P(s) = P(s_1) \cdot P(s_2 | s_1) \cdot P(s_3 | s_2) \dots P(s_m | s_{m-1}) = P(s_1) \cdot \prod_{i=2}^{m} P(s_i | s_{i-1})$$

  - MOST PROBABLE SEQUENCE OF STATES CAN BE COMPUTED BY DYNAMIC PROGRAMMING
  - LEARNING MARKOV MODEL
    - WE ARE GIVEN TRAINING DATA
    - ESTIMATE PARAMETERS OF MARKOV MODEL BY MAXIMUM LIKELIHOOD ESTIMATE
      - $\alpha(s_{i-1} = l, s_i = k)$ IS FRACTION OF SEQUENCES IN T_m, FOR WHICH $s_{i-1} = l$ AND $s_i = k$
      - ESTIMATES FOR CONDITIONAL PROBABILITIES ARE GIVEN BY

$$P(s_i = k | s_{i-1} = l) = \frac{\alpha(s_{i-1} = l, s_i = k)}{\alpha(s_{i-1} = l, s_i = 1) + \dots + \alpha(s_{i-1} = l, s_i = |K|)}$$

- HIDDEN MARKOV MODELS

  - $S = (s_1, s_2, \dots, s_n)$

    - SET OF HIDDEN STATES

    - SEQUENCE OF CHARACTERS FOR EXAMPLE

    - "USED FOR LANGUAGE MODEL"

  - $X = (x_1, x_2, \dots, x_n)$

    - SEQUENCE OF FEATURES

    - SEQUENCE OF IMAGES OF CHARACTERS FOR EXAMPLE

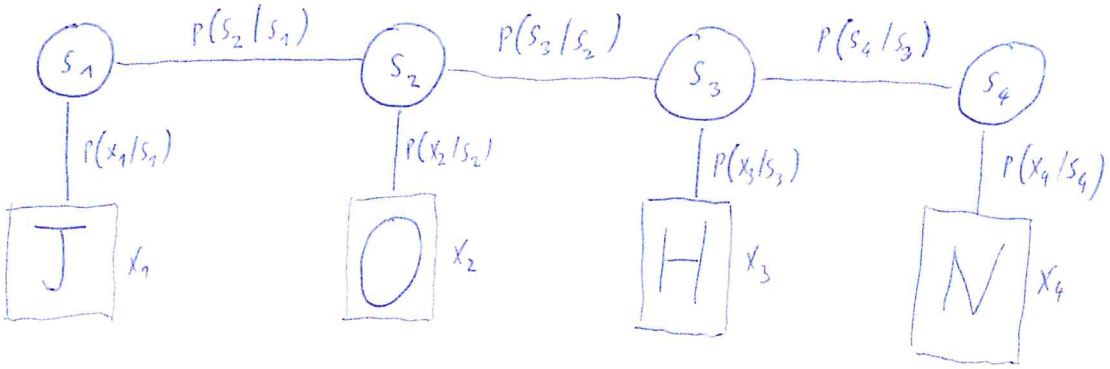
    - "USED FOR APPEARANCE MODEL OF CHARACTERS"

  - $P(s_i | s_{i-1})$

    - LANGUAGE MODEL

  - $P(x_i | s_i)$

    - APPEARANCE MODEL



- FIND MOST PROBABLE SEQUENCE OF HIDDEN STATES GIVEN SEQUENCE OF FEATURES

$$s^* \in \underset{s \in K^n}{\operatorname{ARGMAX}} \; P(s_1) \prod_{i=2}^{n} P(s_i | s_{i-1}) \prod_{i=1}^{n} P(x_i | s_i)$$

- LEARNING MODEL PARAMETERS FROM TRAINING DATA

  - SAME AS FOR MARKOV MODELS

- MARKOV RANDOM FIELDS

  - LET (V, E) BE UNDIRECTED GRAPH

  - LET $S = (S_i, i \in V)$ BE FIELD OF RANDOM VARIABLES INDEXED BY THE NODES OF THE GRAPH AND TAKING VALUES FROM FINITE SET K.

  - JOINT PROBABILITY DISTRIBUTION $p(s)$ IS GIBBS RANDOM FIELD ON GRAPH (V, E) IF IT FACTORISES OVER NODES AND EDGES

$$p(s) = \frac{1}{Z(w)} \exp\left[\sum_{i \in V} w_i(s_i) + \sum_{\{i,j\} \in E} w_{ij}(s_i, s_j)\right]$$

  - PROBABILITY DISTRIBUTION $p(s)$ IS MARKOV RANDOM FIELD W.R.T. GRAPH (V, E) IF

$$p(s_A, s_B | s_C) = p(s_A | s_C) \cdot p(s_B | s_C)$$

HOLDS FOR ANY SUBSET $A, B \subset V$ AND SEPARATING SET C

- MULTICLASS LINEAR CLASSIFIER

  - X
    - SET OF OBSERVATIONS

  - $Y = \{1, \dots y\}$
    - SET OF CLASS LABELS

  - $h(x; w) \in \underset{y \in Y}{\operatorname{ARGMAX}} \langle w_y, \phi(x) \rangle$

    - $\phi : X \to \mathbb{R}^d$ IS FEATURE MAP
    - $w = (w_1, \dots, w_y) \in \mathbb{R}^{d \cdot Y}$ ARE PARAMETERS

  - WE CAN REWRITE THE SCORING FUNCTION TO

$$\langle w_y, \phi(x) \rangle = \langle w, \phi(x, y) \rangle$$

  - WHERE:
    - $\phi : X \times Y \to \mathbb{R}^{d \cdot Y}$ IS
      - $\phi(x, y) = (0, \dots, \underbrace{\phi(x)}_{y\text{-TH SLOT}}, \dots, 0)$

- EXAMPLE

  - OCR FOR SEQUENCES

    - $X = (x_1 \dots x_L) \in I^L$

      - SEQUENCE OF IMAGES WITH CHARACTERS

    - $Y = (y_1, \dots, y_L) \in A^L$

      - SEQUENCE OF CHARACTERS FROM $A = \{A \dots Z\}$

    - $p(x_i | y_i)$

      - APPEARANCE MODEL FOR CHARACTERS

    - $p(y_i | y_{i-1})$

      - LANGUAGE MODEL

$$y \in \underset{y \in A^L}{\operatorname{ARGMAX}} \left(p(y_1) \prod_{i=2}^{L} p(y_i | y_{i-1}) \prod_{i=1}^{L} p(x_i | y_i) \right)$$

- MAP

$$\hat{y} \in \underset{y \in A^L}{\operatorname{ARGMAX}} \left(\log p(y_1) + \sum_{i=2}^{L} \log p(y_i | y_{i-1}) + \sum_{i=1}^{L} \log p(x_i | y_i) \right)$$

- LETS ASSUME FOLLOWING PARAMETRIZATION

$$\log p(y_1) = \langle w, \phi(y_1) \rangle$$

$$\log p(y_i | y_{i-1}) = \langle w, \phi(y_{i-1}, y_i) \rangle$$

$$\log p(x_i | y_i) = \langle w, \phi(x_i, y_i) \rangle$$

- MAP THAN BECOMES LINEAR CLASSIFIER

$$\hat{y} = \underset{(y_1 \dots y_L) \in A^L}{\operatorname{ARGMAX}} \left\langle w, \phi(y_1) + \sum_{i=2}^{L} \phi(y_{i-1}, y_i) + \sum_{i=1}^{L} \phi(x_i, y_i) \right\rangle$$

$$\phi(x, y)$$

- TRAIN BY ERM

- CORRECTLY CLASSIFIED EXAMPLE

$$\langle \phi(x^i, y^i), w \rangle > \langle \phi(x^i, y), w \rangle \quad \forall y \in Y \setminus \{y^i\}$$

- PERCEPTRON ALGORITHM

  - WE WANT TO FIND w SO THAT

    $$\langle w, a^i \rangle > 0 \quad \forall i \in \{1, 2, \dots k\}$$

  - 1. $w \leftarrow 0$
  - 2. FIND VIOLATING $\langle w, a^i \rangle \leq 0$
  - 3. IF THERE IS NO VIOLATING
    - RETURN w

    OTHERWISE

    $w \leftarrow w + a^i$

    GOTO 2

- STRUCTURED (AVG)PERCEPTRON
  - LEARNING $h(x; w) \in \operatorname{ARGMAX}_{y \in Y} \langle w, \phi(x, y) \rangle$
  - FROM EXAMPLES $T_m = \{(x^i, y^i) \in (X \times Y) \mid i = 1, \dots, m\}$
  - IT LEADS TO SOLVING

    $$\langle \phi(x^i, y^i), w \rangle - \langle \phi(x^i, y), w \rangle > 0 \quad \forall i \in \{1, \dots m\}, y \in Y \setminus \{y^i\}$$

    - 1. $w \leftarrow 0$
    - 2. FIND MISSCLASSIFIED EXAMPLE $(x^i, y^i) \in T^m$ SUCH THAT

      $$y^i \neq \hat{y}^i \in \operatorname*{ARGMAX}_{y \in Y} \langle w, \phi(x^i, y) \rangle$$ "PREDICTION PROBLEM"

    - 3. IF THERE IS NO MISCLASSIFIED EXAMPLE RETURN w

      OTHERWISE

      $w \leftarrow w + \phi(x^i, y^i) - \phi(x^i, \hat{y}^i)$

      GO TO 2.

- STRUCTURED OUTPUT SVM

- APPROXIMATES EMPIRICAL RISK MINIMALIZATION BY CONVEX PROBLEM

$$w^* \in \underset{w \in W_r}{\operatorname{ARGMIN}}\, R^{\Psi}(w) \quad \text{WHERE} \quad R_{\Psi m}(w) = \frac{1}{m} \sum_{i=1}^{m} \Psi(x^i, y^i, w)$$

- WHERE

- $W_r \subseteq \mathbb{R}^n$ - CONVEX FEASIBLE SET, E.G. $W_r = \{ w \in \mathbb{R}^n \mid \|w\| \leq r \}$

- $\Psi : X \times Y \times \mathbb{R}^n \to \mathbb{R}$ - CONVEX PROXY APPROXIMATING THE TRUE LOSS $\ell(y^i, y)$

$$\langle w, \phi(x^i, y^i) \rangle \geq \langle w, \phi(x^i, y) \rangle + \ell(y^i, y) \quad \forall y \in Y \setminus \{y^i\}$$

- MARGIN RESCALING LOSS

$$\Psi(x^i, y^i, w) = \max \left\{ 0, \max_{y \in Y \setminus \{y^i\}} \{ \ell(y^i, y) + \langle w, \phi(x^i, y) \rangle - \langle w, \phi(x^i, y^i) \rangle \} \right\}$$

- SO-SVM

- CONVEX CONSTRAINED OPTIMIZATION PROBLEM

$$w^* \in \underset{w \in W_r}{\operatorname{ARGMIN}}\, R^{\Psi}(w)$$

$$R^{\Psi}(w) = \frac{1}{m} \sum_{i=1}^{m} \max_{y \in Y} \{ \ell_i(y) + \langle w, \phi_i(y) \rangle \}$$

- $W_r = \{ w \in \mathbb{R}^n \mid \|w\| \leq r \}$ IS CONVEX SET

- $\ell_i(y) = \ell(y^i, y)$, $\phi_i(y) = \phi(x^i, y) - \phi(x^i, y^i)$

- SO-SVM UNCONSTRAINED PROBLEM

$$w^* \in \underset{w \in \mathbb{R}^n}{\operatorname{ARGMIN}} \left(\frac{\lambda}{2} \|w\|^2 + R^{\Psi}(w) \right)$$

- WITH SLACK VARIABLES

$$w^* = \underset{w \in \mathbb{R}^n,\, \xi \in \mathbb{R}^m}{\operatorname{ARGMIN}} \left(\frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right)$$

$$\text{S.t. } \xi_i \geq \ell_i(y) + \langle w, \phi_i(y) \rangle \quad \forall i \in \{1, \dots, m\}\ \forall y \in$$

- CUTTING PLANE ALGORITHM

- 1. $w_1 \in W$, $t \leftarrow 1$

- 2. COMPUTE NEW CUTTING PLANE AND OBJECTIVE VALUE

$$a_t = \frac{1}{m} \sum_{i=1}^{m} \phi_i(\hat{y}^i)$$

$$b_t = \frac{1}{m} \sum_{i=1}^{m} \ell_i(\hat{y}^i)$$

$$R^{\Psi}(w_t) = b_t + \langle w_t, a_t \rangle$$

WHERE $\hat{y}^i$ IS SOLUTION OF LOSS AUGMENTED PREDICTION PROBLEM:

$$\hat{y}^i = \underset{y \in Y}{\operatorname{ARGMAX}} \left(\ell_i(y) + \langle w, \phi_i(y) \rangle \right) = \underset{y \in Y}{\operatorname{ARGMAX}} \left(\ell_i(y_i, y) + \langle w, \phi(x^i, y) \rangle \right)$$

- 3. SOLVE REDUCED PROBLEM

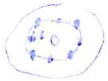
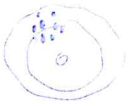
$$w_{t+1} = \underset{w \in W}{\operatorname{ARGMIN}} R_t^{\Psi}(w), \quad \text{WHERE} \quad R_t^{\Psi}(w) = \max_{i=1,\dots,t} \left(\ell_i + \langle w, a_i \rangle \right)$$

- 4. IF $\min_{i=1,\dots,t+1} R(w_t) - R^{\Psi}(w_{t+1}) \leq \varepsilon$

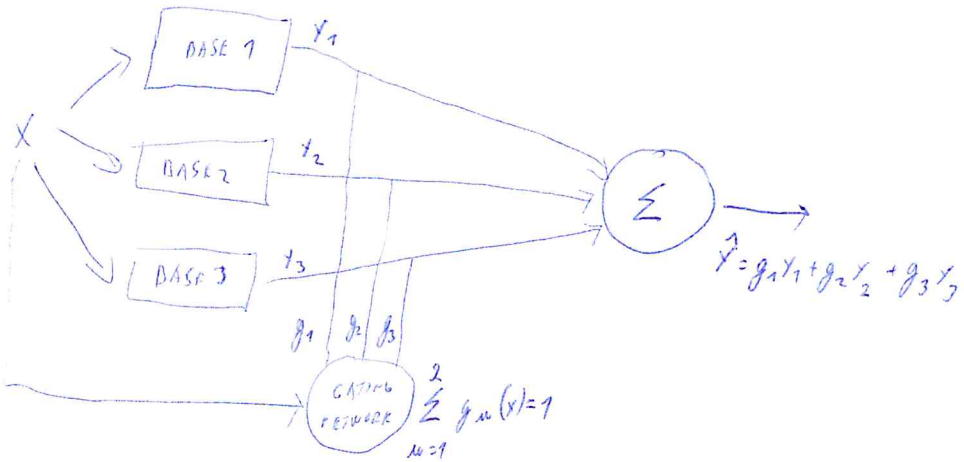
- EXIT

ELSE

$t \leftarrow t+1$ AND GO TO 2

- ENSEMBLING
  - "WISDOM OF CROWD"
  - AVERAGING OR TAKING MAJORITY VOTE
  - CANCELLING EFFECT OF NOISE OF INDIVIDUAL OPINIONS
  - $g_m(x) = \mathbb{E}_{T_m}(h_m(x))$
    - PREDICTOR AVERAGED OVER MULTIPLE DATASETS
  - ERROR = BIAS^2 + VARIANCE + NOISE
  - APPROACHES — "OFF CENTER" — "SPREAD"
    - BAGGING
      - BOOTSTRAP AGGREGATING
      - SAMPLE DIFFERENT TRAINING SETS FROM ORIGINAL TRAINING DATA
      - TRAIN PREDICTORS ON THESE SETS AND AVERAGE THEM
        - HIGH VARIANCE LOW BIAS  "IN THE CENTER, BUT INACCURATE"
    - BOOSTING
      - SEQUENTIALLY TRAIN LOW VARIANCE HIGH BIAS PREDICTORS
        
        - "ACCURATE BUT OFF-CENTER"
      - SUBSEQUENT PREDICTORS LEARN FROM MISTAKES OF PREVIOUS ONE
      - INCREASE WEIGHTS FOR MISLABELED SAMPLES

- COMBINATION OF BASE AND META LEARNERS



- DECISION / REGRESSION TREES

  - TRAINING SET

$$T^m = \{(x_i, y_i) \mid i = 1 \dots m\} \quad x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$$

  - INPUT SPACE SPLIT INTO REGIONS DEFINED IN LEAVES

$$R_r, \; r \in \{1, \dots, M\}$$

  - WE CAN MODEL REGION RESPONSES BY CONSTANTS c_r, $r \in \{1, \dots, M\}$

  - PREDICTION

$$h(x) = \sum_{r=1}^{M} c_r \, [\![x \in R_r]\!]$$

  - FOR SUM OF SQUARES LOSS FUNCTION $\sum_{i=1}^{m} (y_i - h(x_i))^2$

    - WE SET RESPONSES TO BE AVERAGES OVER REGIONS

$$\hat{c}_r = \frac{1}{|S_r|} \sum_{x_i \in R_r} y_i$$

$$S_r = \{(x_i, y_i) : (x_i, y_i) \in T^m \wedge x_i \in R_r\}$$

- BUT NP-COMPLETE LEARNING

- WE NEED HEURISTICS

  - GREEDY APPROACH

  - "CHOOSE RECURSIVELY THE MOST IMPORTANT ATTRIBUTE"

    - IMPORTANT IS THE ATTRIBUTE THAT REDUCE LOSS THE MOST

    - THEN SPLIT ATTRIBUTE j IN SPLIT POINT s INTO TWO PARTS (R IS CURRENT REGION)

      - ORDINAL ("CONTINUOUS") ATTRIBUTE:

$$R_L(j, s) = \{x \mid x \in R \wedge x_j \leq s\} \qquad R_R(j, s) = \{x \mid x \in R \wedge x_j > s\}$$

      - NOMINAL ("DISCRETE") ATTRIBUTE:

$$R_L(j, s) = \{x \mid x \in R \wedge x_j = s\} \qquad R_R(j, s) = \{x \mid x \in R \wedge x_j \neq s\}$$

- HOW TO CHOOSE ATTRIBUTE j AND SPLIT POINT s

$$\underset{(j,s)}{\operatorname{ARGMIN}} \left(\min_{c_L} \left(\sum_{x_i \in R_L(j,s)} (y_i - c_L)^2 \right) + \min_{c_R} \left(\sum_{x_i \in R_R(j,s)} (y_i - c_R)^2 \right) \right)$$

- INNER OPTIMALIZATION IN REGION

  - BY AVERAGING DESIRED OUTPUT

$$\hat{c}_L = \frac{1}{|S_L(j,s)|} \sum_{x_i \in R_L(j,s)} y_i$$

$$\hat{c}_R = \frac{1}{|S_R(j,s)|} \sum_{x_i \in R_R(j,s)} y_i$$

- BIAS AND VARIANCE

  - SMALL CHANGE IN DATA MAY LEED TO BIG DIFFERENCE IN FINAL TREES

  - GROWN DECISION TREES HAS USUALLY

    - LOW BIAS

    - HIGH VARIANCE

    - WHICH LEADS TO OVERFITTING

    - IDEA IS TO AVERAGE MULTIPLE MODELS

- BOOTSTRAPING

  - WAY HOW TO MAKE SEVERAL DATASETS FROM SINGLE DATASET

    - NEW DATASETS HAS THE SAME SIZE AS ORIGINAL

    - SAMPLING FROM T^m WITH REPLACEMENT

    - NEW DATASETS T_i^m HAS $\approx 63,2\%$ OF UNIQUE SAMPLES OF ORIGINAL DATASET

- BAGGING
  - BOOTSTRAP AGGREGATING
  - USE BOOTSTRAPING TO MAKE SEVERAL NEW DATASETS
  - TRAIN MODELS ON DATASETS
  - AVERAGE MODELS
  - IN CASE OF DECISION TREES
    - RANDOM FOREST
      - GROWS TREES TO MAXIMA DEPTH TO REDUCE BIAS
      - TREES ARE DECORELATED BY
        - TRAINING EACH OF DIFFERENT BOOTSTRAP DATASET
        - RANDOMIZATION OF SPLIT ATRIBUTE SELECTION
          - SELECT RANDOM SET OF ATTRIBUTES CONSIDERED FOR SPLITTING
      - AVERAGING FOR REGRESSION
      - MAJORITY VOTE FOR CLASSIFICATION
    - OUT OF BAG ERROR
      - HOW TO TEST ERRORS
      - MEASURE ERRORS ONLY OF TREES, WHICH WERE NOT TRAINED ON SAMPLE (x_i, y_i)
      - "K-FOLD CROSS-VALIDATION"

- BOOSTING

  - SEQUENTIALLY TRAIN WEAK LEARNERS

  - LOW VARIANCE HIGH BIAS

  - SUBSEQUENT PREDICTORS FIX MISTAKES OF PREVIOUS ONES REDUCING BIAS

  - ADA-BOOST

    - BINARY CLASSIFIER

      - 1. INITIALIZE WEIGHTS OF TRAINING SAMPLES

$$w_i = \frac{1}{m} \qquad i = 1, 2, \dots, m$$

      - 2. FOR $k = 1$ TO K:

        - a) FIT CLASSIFIER $f_k(x; \theta_k)$ TO TRAINING DATA USING LOSS WEIGHTED BY w_i:

$$\theta_k = \underset{\theta}{\operatorname{ARGMIN}} \sum_{i=1}^{m} w_i \, [\![y_i \neq f_k(x_i; \theta)]\!]$$

        - b) COMPUTE WEIGHTED ERROR RATE

$$\varepsilon_k = \frac{\sum_{i=1}^{m} w_i \, [\![y_i \neq f_k(x_i; \theta_k)]\!]}{\sum_{i=1}^{m} w_i}$$

        - c) COMPUTE WEIGHT

$$\alpha_k = \log\left(\frac{1 - \varepsilon_k}{\varepsilon_k}\right)$$

        - d) SET $w_i \leftarrow w_i \cdot \exp\left(\alpha_k \cdot [\![y_i \neq f_k(x_i; \theta_k)]\!]\right)$ FOR $i = 1, 2, \dots m$

      - 3. RETURN $h_m(x) = \operatorname{sign}\left[\sum_{k=1}^{K} \alpha_k f_k(x; \theta_k)\right]$

- FORWARD STAGEWISE ADDITIVE MODELING

- FSAM

- 1. INITIALIZE $f_0(x) = 0$

- 2. FOR $k = 1$ TO K:

- a) FIND

$$(\beta_k, \theta_k) = \underset{\beta, \theta}{\operatorname{ARGMIN}} \sum_{i=1}^{m} L(y_i, f_{k-1}(x_i) + \beta b(x_i; \theta))$$

WHERE $b(x_i, \theta_k)$ IS BASIS FUNCTION AND β_k IS CORRESPONDING COEFFICIENT

- b) SET $f_k(x) = f_{k-1}(x) + \beta_k b(x; \theta_k)$

- 3. RETURN $h_m(x) = f_K(x)$

- FSAM UPDATE IS SIMILAR TO GRADIENT DESCENT

$$f_k(x) = f_{k-1}(x) + \beta_k b(x; \theta_k)$$

↖ STEP SIZE ↖ NEGATIVE OF GRADIENT

- GRADIENT BOOSTING MACHINE

- 1. INITIALIZE $f_0(x) = 0$ OR $f_0(x) = \operatorname{ARGMIN}_{\gamma} \sum_{i=1}^{m} L(y_i, \gamma)$

- 2. FOR $k = 1$ TO K:

- a) COMPUTE

$$g_k = \left[\frac{\partial L(y_i, f_{k-1}(x_i))}{\partial f_{k-1}(x_i)} \right]_{i=1}^{m}$$

- b) FIT REGRESSION MODEL $b(\cdot; \theta)$ TO $-g_k$

$$\theta_k = \underset{\theta}{\operatorname{ARGMIN}} \sum_{i=1}^{m} [(-g_k)_i - b(x_i; \theta)]^2$$

- c) CHOOSE FIXED STEP SIZE $\beta_k = \beta \in [0, 1]$ OR USE LINE SEARCH

$$\beta_k = \underset{\beta > 0}{\operatorname{ARGMIN}} \sum_{i=1}^{m} L(y_i, f_{k-1}(x_i) + \beta b(x_i; \theta_k))$$

- d) SET $f_k(x) = f_{k-1}(x) + \beta_k b(x; \theta_k)$

- 3. RETURN $h_m(x) = f_K(x)$

- GRADIENT BOOSTED TREES

  - ALL WEAK LEARNERS ARE DECISION OR REGRESSION TREES
  - LIMIT SIZE OF TREE
  - META PARAMETERS (K (NUMBER OF TREES), β (LEARNING RATE)) HAVE TO BE FOUND USING CROSS VALIDATION
  - MODEL IS BUILD SEQUENTIALLY
    - UNLIKE RANDOM FORESTS