Training data: $S_n = \left\{\overline{x}^{(i)}, y^{(i)}\right\}_{i=1}^{n}$

Training error: $E_n(\overline{\theta}) = \frac{1}{n}\sum_{i=1}^{n}[[y^{(i)} \neq h\left(\overline{x}^{(i)};\theta\right)]] = \frac{1}{n}\sum_{i=1}^{n}[[y^{(i)}\cdot h\left(\overline{x}^{(i)};\theta\right) \leq 0]]$.

Perceptron Algorithm:

On input $S_n = \left\{\overline{x}^{(i)}, y^{(i)}\right\}_{i=1}^{n}$

Initialize $k=0,\ \overline{\boldsymbol{\theta}}^{(\mathbf{0})} = \overline{\mathbf{0}},\ b^{(0)} = 0$

while there exists a misclassified point

  for $i = 1 \cdots n$

    if $y^{(i)} \neq h\left(\overline{x}^{(i)};\overline{\theta}^{(k)}\right)$

      $\overline{\theta}^{(k+1)} = \overline{\theta}^{(k)} + y^{(i)}\overline{x}^{(i)}$

      $b^{(k+1)} = b^{(k)} + y^{(i)}$

      $k++$

If data are linearly separable, perceptron converges.

Empirical Risk: $R_n(\overline{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\text{Loss}\left(h\left(\overline{x}^{(i)};\overline{\theta}\right), y^{(i)}\right)$

1. 0-1: $\text{Loss}_{0-1}\left(h\left(\overline{x}^{(i)};\overline{\theta}\right), y^{(i)}\right) = [[y^{(i)}\left(\overline{\theta}\cdot\overline{x}^{(i)}\right) \leq 0]]$

2. Hinge: $\text{Loss}_h(z) = \max\{1-z, 0\}$

Convex function: $\lambda \in [0,1], \forall x, x'\ f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda)f(x')$

Gradient: $\nabla_{\overline{\theta}}R_n(\overline{\theta}) = \left[\frac{\partial R_n(\overline{\theta})}{\partial\theta_1}, \ldots, \frac{\partial R_n(\overline{\theta})}{\partial\theta_d}\right]^{\mathrm{T}}$

GD: $\overline{\theta}^{(k+1)} = \overline{\theta}^{(k)} - \eta\nabla_{\overline{\theta}}R_n\left(\overline{\theta}\right)\big|_{\overline{\theta}=\overline{\theta}^k}$

SGD: look at one mis-classified example each time

Initialize $k=0,\ \overline{\boldsymbol{\theta}}^{(\mathbf{0})} = \overline{\mathbf{0}}$

while convergence criteria is not met

  randomly shuffle points

  for $i = 1 \cdots n$

    if $y^{(i)}\left(\overline{\theta}\cdot\overline{x}^{(i)}\right) < 1$

      $\overline{\theta}^{(k+1)} = \overline{\theta}^{(k)} - \eta\nabla_{\overline{\theta}}\text{Loss}_h\left(y^{(i)}\left(\overline{\theta}\cdot\overline{x}^{(i)}\right)\right)\big|_{\overline{\theta}=\overline{\theta}^k} = \theta^{(k)} + \eta y^{(i)}\overline{x}^{(i)}$

      $k++$

Logistic loss: $\text{Loss}_{\log}(z) = \log_2(1+\exp(-z))$

$\nabla_{\overline{\theta}}\text{Loss}_{\log}\left(y^{(i)}\left(\overline{\theta}\cdot\overline{x}^{(i)}\right)\right) = \frac{1}{\ln 2}\cdot\frac{-y^{(i)}\bar{x}^{(i)}}{1+\exp(y^{(i)}(\overline{\theta}\cdot\overline{x}^{(i)}))}$

Regression: $y \in \mathbb{R}$, regression function $f: \mathbb{R}^d \to \mathbb{R}$, where $f \in \mathcal{F}$

Empirical risk for linear reg: $R_n(\overline{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\text{Loss}\left(y^{(i)} - \left(\overline{\theta}\cdot\overline{x}^{(i)}\right)\right)$

Least square loss function: $R_n(\overline{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(y^{(i)} - \left(\overline{\theta}\cdot\overline{x}^{(i)}\right)\right)^2}{2}$

$\nabla_{\overline{\theta}}R_n(\overline{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \overline{\theta}\cdot\overline{x}^{(i)})\cdot(-\overline{x}^{(i)})$

SGD for linear regression: $\overline{\theta}^{(k+1)} = \overline{\theta}^{(k)} + \eta_k\left(y^{(i)} - \overline{\theta}^{(k)}\cdot\overline{x}^{(i)}\right)\overline{x}^{(i)}$

Find closed-form: $\nabla_{\overline{\theta}}R_n\left(\overline{\theta}\right)\big|_{\overline{\theta}=\overline{\theta}^*} = 0 = -\frac{1}{n}\sum_{i=1}^{n}\overline{x}^{(i)}y^{(i)} + \frac{1}{n}\sum_{i=1}^{n}\overline{x}^{(i)}\left(\overline{x}^{(i)}\right)^T\bar{\theta}^* =: -\bar{b} + A\bar{\theta}^*$

Define $X = \left[\overline{x}^{(1)}, \ldots, \overline{x}^{(n)}\right]^T$, $\overline{y} = \left[y^{(1)}, \ldots, y^{(n)}\right]^T$

$\bar{b} = \frac{1}{n}X^T\bar{y}$, $A = \frac{1}{n}X^TX$. Hence $\boxed{\bar{\theta}^* = \left(X^TX\right)^{-1}X^T\overline{y}}$

Regularization: $J_{n,\lambda}(\overline{\theta}) = \lambda Z(\overline{\theta}) + R_n(\overline{\theta})$

Ridge regression: $\boxed{J_{n,\lambda}(\overline{\theta}) = \lambda\frac{\|\overline{\theta}\|^2}{2} + \frac{1}{n}\sum_{i=1}^{n}\frac{\left(y^{(i)} - \left(\overline{\theta}\cdot\overline{x}^{(i)}\right)\right)^2}{2}}$

By setting the gradient 0, $\boxed{\overline{\theta}^* = \left(\lambda I + X^TX\right)^{-1}X^TY}$

SVM maximum margin separator. $\gamma^{(i)}(\overline{\theta}, b) = \frac{(\overline{\theta}\cdot\overline{x}^{(i)}+b)y^{(i)}}{||\overline{\theta}||}$

$\max_{\overline{\theta},b}\min_i\gamma^{(i)}(\overline{\theta}, b) \Rightarrow \max_{\overline{\theta}}\frac{1}{||\overline{\theta}||}$ s.t. $y^{(i)}\left(\overline{\theta}\cdot\overline{x}^{(i)} + b\right) \geq 1,\ \forall i$

Lagrange Multipliers:

$\min_{\overline{\theta}} f(\overline{x};\overline{\theta})$ subject to $h_i(\overline{x};\overline{\theta}) \leq 0, \forall i = 1, \ldots, n$ is equivalent to

$$\min_{\overline{\theta}}\max_{\overline{\alpha}}\quad f(\overline{x};\overline{\theta}) + \sum_{i=1}^{n}\alpha_i h_i(\overline{x};\overline{\theta}) \text{ s.t. } \alpha_i \geq 0, \forall i = 1, \ldots, n$$

For $h_i < 0$, $\alpha_i = 0$, for $h_i = 0$, $\alpha_i > 0$, for $h_i > 0$, $\alpha_i = \infty$.

SVM Primal Formulation

$$\begin{array}{ll} \min_{\overline{\theta}} & \frac{1}{2}\|\overline{\theta}\|^2 \\ \text{subject to} & y^{(i)}\left(\overline{\theta}\cdot\overline{x}^{(i)}\right) \geq 1, \forall i = 1, \ldots, n \end{array}$$

After applying Lagrange multiplier,

$$\boxed{\begin{array}{ll} \min_{\overline{\theta}}\max_{\overline{\alpha}} & \frac{1}{2}\|\overline{\theta}\|^2 + \sum_{i=1}^{n}\alpha_i\left(1 - y^{(i)}\left(\overline{\theta}\cdot\overline{x}^{(i)}\right)\right) \\ \text{subject to} & \alpha_i \geq 0, \forall i = 1, \ldots, n \end{array}}$$

Dual Formulation

$$\boxed{\begin{array}{ll} \max_{\overline{\alpha}}\min_{\overline{\theta}} & \frac{1}{2}\|\overline{\theta}\|^2 + \sum_{i=1}^{n}\alpha_i\left(1 - y^{(i)}\left(\overline{\theta}\cdot\overline{x}^{(i)}\right)\right) \\ \text{subject to} & \alpha_i \geq 0, \forall i = 1, \ldots, n \end{array}}$$

Set the gradient w.r.t. $\bar{\theta}$ to be zero, get $\overline{\theta}^* = \sum_{i=1}^{n}\alpha_i y^{(i)}\overline{x}^{(i)}$

$$\boxed{\max_{\overline{\alpha},\alpha_i\geq 0}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y^{(i)}y^{(j)}\overline{x}^{(i)}\cdot\overline{x}^{(j)}}$$

$y^{(i)}\overline{\theta}^*\cdot\overline{x}^{(i)} = 1$, if $\alpha_i > 0$ is support vector

$y^{(i)}\overline{\theta}^*\cdot\overline{\overline{x}}^{(i)} > 1$, if $\alpha_i = 0$ not support vector

Feature mapping $\bar{x} \in \mathbb{R}^d \Rightarrow \phi(\bar{x}) \in \mathbb{R}^p$

Soft Margin SVMs for non-linearly separable data

$$\boxed{\begin{array}{ll} \min_{\bar{\theta},\xi} & \frac{1}{2}\|\overline{\theta}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} & y^{(i)}(\overline{\theta}\cdot\overline{x} + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ \forall i = 1, \ldots, n \end{array}}$$

Its Lagrangian

$L(\overline{\theta}, \overline{\alpha}, b, \overline{\gamma}, \overline{\xi}) =$

$$\frac{1}{2}\|\overline{\theta}\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\left(1 - \xi_i - y^{(i)}\left(\overline{\theta}\cdot\overline{x}^{(i)} + b\right)\right) - \sum_{i=1}^{n}\gamma_i\xi_i$$

The final equation is the same as hard margin.

Kernel has an associated feature mapping

$$K\left(\overline{x}^{(i)}, \overline{x}^{(j)}\right) = \phi\left(\overline{x}^{(i)}\right)\cdot\phi\left(\overline{x}^{(j)}\right)$$

Kernalized Dual SVM

$\max_{\overline{\alpha}}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y^{(i)}y^{(j)}K\left(\overline{x}^{(i)}, \overline{x}^{(j)}\right)$, subject to $\alpha_i \geq 0\quad \forall i = 1, .., n$

Classify new example: $h\left(\overline{x}^{(j)}\right) = \text{sign}(\sum_{i=1}^{n}\alpha_i y^{(i)}\underbrace{\overline{x}^{(i)}\cdot\overline{x}^{(j)}}_{K(\overline{x}^{(i)},\overline{x}^{(j)})})$

Kernel Algebra: 1. $K(\overline{x}, \overline{z}) = K_1(\overline{x}, \overline{z}) + K_2(\overline{x}, \overline{z})$; 2. $K(\overline{x}, \overline{z}) = \alpha K_1(\overline{x}, \overline{z})$ $(\alpha > 0)$; 3. $K(\overline{x}, \overline{z}) = K_1(\overline{x}, \overline{z})K_2(\overline{x}, \overline{z})$

Feature Selection

Shannon Entropy and Information Gain

$$H(X) = -\sum_{i=1}^{k}\Pr\left(X = x_i\right)\log_2\Pr\left(X = x_i\right)$$

$$H(Y|X = x_i) = -\sum_{j=1}^{k}\Pr\left(Y = y_j|X = x_i\right)\log_2\Pr\left(Y = y_j|X = x_i\right)$$

$$H(Y|X) = \sum_{i=1}^{m}\Pr(X = x_i)\cdot H(Y|X = x_i)$$

$$IG(X, Y) = H(Y) - H(Y|X)$$

Build Tree by selecting the best spilt each time

```
BuildTree(DS)
   if (y(i) == y) for all examples in DS
          return y
   elseif (x(i) == x) for all examples in DS
          return majority label
   else
      xs = argminx H(y|x)
      for each value v of xs
          DSv = {examples in DS where xs = v}
          BuildTree(DSv)
```

Bootstrap Sampling: Pick n samples uniformly at random from the original training dataset. About $1-(1-1/n)^n \to 63\%$ are selected.
Note that Bagging reduces variance (estimation error), but bias (structural error) may still remain. Also, independence of classifiers is a strong assumption.
To further decorrelate the decision trees learnt, use **Random Forests:** 1. bagging. 2. random feature subset.
for $b=1,\ldots,B$
  draw bootstrap sample $S_n(b)$ of size n from S_n
  [grow decision tree $DT^{(b)}$]
output ensemble $\{DT^{(1)},\ldots,DT^{(B)}\}$
subprocedure for growing $DT^{(b)}$: until stopping criteria are met, recursively repeat following steps for each node of tree:
1. select k features at random from d features
2. pick best feature to split on (using IG)
3. split node into children.

Boosting: General strategy for combining weak classifiers into a strong classifer
AdaBoost
In each round, make sure $\sum_{i=1}^{n} \widetilde{w}_m(i) = 1$
set $\widetilde{W}_0(i) = \frac{1}{n}$ for $i=1\ldots n$
for $m=1$ to M do:
  find $h\left(\overline{x};\overline{\theta}^{*(m)}\right)$, a weak clf that approximately minimizes the weighted training error ϵ_m:

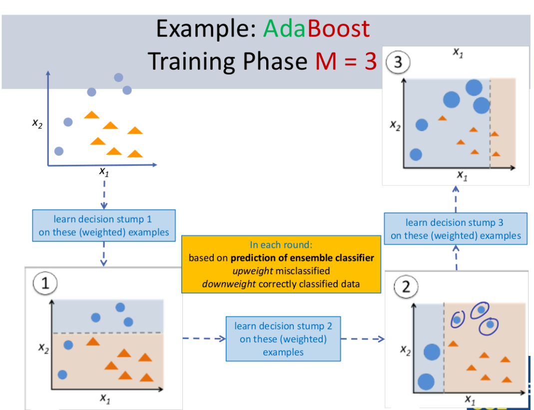
$$\boxed{\epsilon_m = \sum_{i=1}^{n} \widetilde{W}_{m-1}(i)\left[\left[y^{(i)} \neq h\left(\overline{x}^{(i)};\overline{\theta}^{(m)}\right)\right]\right]}$$

  given $\overline{\theta}^{*(m)}$, compute $\hat{\epsilon}_m$ and α_m that minimizes wrighted training loss: $\boxed{\hat{\alpha}_m = \frac{1}{2}\ln\left(\frac{1-\hat{\epsilon}_m}{\hat{\epsilon}_m}\right)}$
  update weights on all training examples:
  for $i=1$ to n do:

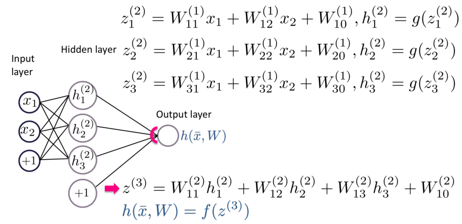
$$\boxed{\widetilde{W}_m(i) = \overbrace{\widetilde{W}_{m-1}(i)\exp\left\{-y^{(i)}\hat{\alpha}_m h\left(\overline{x}^{(i)};\overline{\theta}^{*}_{m}\right)\right\}}^{\exp\left(-y^{(i)}h_m(\overline{x}^{(i)})\right)} / \underbrace{Z_m}_{\text{normalize}}}$$

  end for
end for
output final classifier $h_M(\overline{x}) = \sum_{m=1}^{M}\hat{\alpha}_m h\left(\overline{x};\overline{\theta}^{*(m)}\right)$
Decision stumps (DT with depth 1) as weak clf:

$$h(\overline{x};\overline{\theta}) = \text{sign}\left(\theta_1\left(x_k-\theta_0\right)\right),\text{ where } \overline{\theta} = (\overbrace{k}^{\text{coordinate}}, \overbrace{\theta_0}^{\text{position}}, \overbrace{\theta_1}^{\text{direction}})$$



Neural Network
SGD-single layer
(0)Initialize parameters to small random values
(1)Select a point at random
(2)Update the parameters based on that point and the gradient:
$\overline{\theta}^{(k+1)} = \overline{\theta}^{(k)} - \eta_k \nabla_{\overline{\theta}}\text{Loss}\left(y^{(i)}h\left(\overline{x}^{(i)};\overline{\theta}\right)\right)$ where $z = h(\overline{x}^{(i)};\overline{\theta})$
$\frac{\partial Loss(yz)}{\partial w_i} = \frac{\partial Loss(yz)}{\partial z}\frac{\partial z}{\partial w_j}$
assume $Loss = max(1, 1-yz)$, then $\frac{\partial Loss(yz)}{\partial w_i} = -yx_j$
Finally, $w_j^{(k+1)} = w_j^{(k)} + \eta_k x_j y$



$w_{ki}^{(j)}$ is the weight of layer j, unit k, input i
$z_i(k) = \sum_{n=1}^{d} w_{in}x_n$
$h_i^k = g(z_i^{(k)})$
Activation Function
1. rectified linear $f(z) = max(0, z)$
2. threshold $f(z) = sign(z)$
3. sigmoid $f(x) = \frac{1}{1+e^{-x}}$
4. tanh $f(z) = tanh(z)$

two-layer
use back-propagation(GD+chain rule)
$v_j^{(k+1)} = v_j^{(k)} + \eta_k y h_j[[(1-yz) > 0]]$
$\frac{\partial\,\text{Loss}(yz)}{\partial w_{ji}} = \frac{\partial\,\text{Los}\,s(yz)}{\partial z}\frac{\partial z}{\partial h_j}\frac{\partial h_j}{\partial z_j}\frac{\partial z_j}{\partial w_{ji}}$
$w_{ji}^{(k+1)} = w_{ji}^{(k)} + \eta_k y[[(1-yz) > 0]]v_j\,[[z_j > 0]]\,x_i$
backprop
1. For each training instance, make a prediction $h(\bar{x}^i, \bar{\theta})$
2. Measure the $Loss(y^{(n)}h(\bar{x}^{(n)}, \bar{\theta}))$
3. go through each layer in reverse to measure the error contribution of each connection(bkwd propagate)
4. tweak weight to reduce error(SGD update)

Some Math
Gradient: consider $f, g: \mathbb{R}^n \to \mathbb{R}$ and $k \in \mathbb{R}$
1. $\nabla kf = k\nabla f$
2. $\nabla(f \pm g) = \nabla f \pm \nabla g$
3. Product Rule: $\nabla(fg) = f\nabla g + (\nabla f)g$
4. If A is symmetric and $q(\vec{x}) = \vec{x}^T A\vec{x}$, then $\nabla q(\vec{x}) = 2A\vec{x}$.
5. Hessian: $\left[\nabla^2 f\right]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$
6. Gradient of L2-norm: $\nabla_{\overline{\theta}}\|\overline{\theta}\|^2 = 2\overline{\theta}$

Chain Rule: consider a function $g: \mathbb{R}^n \to \mathbb{R}$ and $f: \mathbb{R} \to \mathbb{R}$, then $\nabla f \circ g(\vec{x}) = f'(g(\vec{x}))\nabla g(\vec{x})$
Eigenvalues/vecotrs: 1. Solve $\det(A-\lambda I) = 0$ for λ. 2. For each λ, solve $(A-\lambda I)\vec{v} = \vec{0}$ for $\vec{v}$
Positive (semi-)definite
1. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be positive definite if $\forall \overline{z} \in \mathbb{R}^n - \{\overline{0}\}, \overline{z}^T A\overline{z} > 0$
2. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be positive semi-definite if $\forall \overline{z} \in \mathbb{R}^n, \overline{z}^T A\overline{z} \geq 0$
For a positive (semi-)definite matrix, all its eigenvalues are positive (non-negative).