PERCEPTRON

$k \leftarrow 0,\ \bar{\theta}^{(k)} \leftarrow \bar{0}$
while at least one point is misclassified **do**
  for $i = 1, ..., n$ **do**
    if $y^{(i)}(\bar{\theta}^{(k)} \cdot \bar{x}^{(i)}) \le 0$ **then**
      $\bar{\theta}^{(k+1)} \leftarrow \bar{\theta}^{(k)} + y^{(i)}\bar{x}^{(i)}$
      $k{+}{+}$
    end if
  end for
end while
return $\bar{\theta}^{(k)}$

To incorporate offset: set $\bar{x}^{(i)} = [1, \bar{x}^{(i)}]^T$
Then offset is θ_0

GRADIENT DESCENT

$\bar{\theta}^{(0)} \leftarrow \bar{0},\ k \leftarrow 0$
while convergence criteria not met **do**
  $\bar{\theta}^{(k+1)} \leftarrow \bar{\theta}^{(k)} - \eta \nabla_{\bar{\theta}} R_n(\bar{\theta})_{|\bar{\theta}=\bar{\theta}^{(k)}}$
  $k{+}{+}$
end while
return $\bar{\theta}^{(k)}$

$R_n(\bar{\theta})$ is sum of the loss over every point, averaged.

$$R_n(\bar{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \max\{1 - y^{(i)}(\bar{\theta} \cdot \bar{x}^{(i)}), 0\}$$

$$loss_{log}(z) = ln(1 + e^{-z})$$

STOCHASTIC GRADIENT DESCENT

$\bar{\theta}^{(0)} \leftarrow \bar{0},\ k \leftarrow 0$
while convergence criteria not met **do**
  randomly shuffle points
  for $t = 1...n$ **do**
    $\bar{\theta}^{(k+1)} \leftarrow \bar{\theta}^{(k)} - \eta \nabla_{\bar{\theta}} \text{loss}(y^{(t)}, h(\bar{x}^{(t)}; \bar{\theta}))_{|\bar{\theta}=\bar{\theta}^{(k)}}$
    $k{+}{+}$
  end for
end while
return $\bar{\theta}^{(k)}$

Note: argument of loss function $z = y^{(i)}(\bar{\theta} \cdot \bar{x}^{(i)})$ in classification and $z = y^{(i)} - \bar{\theta} \cdot \bar{x}^{(i)}$ in regression

HARD MARGIN SVM

$$\min_{\bar{\theta}, b} \frac{1}{2}\|\bar{\theta}\|^2 \quad \text{subject to} \quad y^{(i)}(\bar{\theta} \cdot \bar{x}^{(i)} + b) \ge 1,\ \forall i = 1, \ldots, n$$

SOFT MARGIN SVM

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

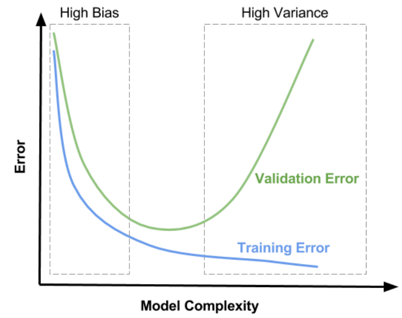
$$\text{subject to} \quad y^{(i)}(\bar{\theta} \cdot \bar{x}^{(i)} + b) \ge 1 - \xi_i,\ \xi_i \ge 0,\ \forall i = 1, \ldots, n$$

HARD MARGIN DUAL FORMULATION

$$\max_{\bar{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_j \alpha_i y^{(i)} y^{(j)} \bar{x}^{(i)} \cdot \bar{x}^{(j)}$$

Kernel trick: $\bar{x}^{(i)} \cdot \bar{x}^{(j)} = K(\bar{x}^{(i)}, \bar{x}^{(j)})$
Example: if $K(\bar{x}, \bar{z}) = 6x_1 z_1$, $\phi(x) = \sqrt{6}x_1$

BIAS-VARIANCE TRADEOFF



EVALUATING CLASSIFIERS

	Predicted +	Predicted -
Actually +	TP	FN
Actually -	FP	TN

Classification Error = FP + FN
Accuracy = (TP + TN) / N
FP Rate = FP / (TN + FP)
TP Rate/Sensitivity/Recall = TP / (TP + FN)
Precision = TP / (TP + FP)
Specificity = TN / (TN + FP)

LAGRANGIAN

$min_{\bar{w}} f(\bar{w}) \quad s.t \quad h_i(\bar{w}) \le 0,\ g_i(\bar{w}) \le 0$ for all i
Then (for dual variables $\alpha_i \ge 0,\ \beta_i \ge 0$):

$$L(\bar{w}, \bar{\alpha}, \bar{\beta}) = f(\bar{w}) + \sum_{i=1}^{n} \alpha_i h_i(\bar{w}) + \sum_{i=1}^{n} \beta_i h_i(\bar{w})$$

Set the gradient of L with respect to $\bar{w}$ to 0 to solve for $\bar{w}$ *

REGRESSION

Least squares loss: $R_n(\bar{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \frac{(y^{(i)} - (\bar{\theta} \cdot \bar{x}^{(i)}))^2}{2}$

Closed form: $\bar{\theta}^* = (X^T X)^{-1} X^T \bar{y}$

REGULARIZATION

Idea: want a model that fits pretty well but isn't too complex

$J_{n,\lambda}(\bar{\theta}) = \lambda Z(\bar{\theta}) + R_n(\bar{\theta})$

Lambda is a hyperparameter that quantifies the penalty for too much complexity

ex: ridge regression (L2 regularization with squared loss):

$$J_{n,\lambda}(\bar{\theta}) = \lambda\frac{||\bar{\theta}||^2}{2} + \frac{1}{n}\sum_{i=1}^{n}\frac{(y^{(i)} - (\bar{\theta}\cdot\bar{x}^{(i)}))^2}{2}$$

Ridge Regression closed form:

$$\bar{\theta}^* = (\lambda' I + X^T X)^{-1} X^T \bar{y}$$

Where $\lambda' = n\lambda$

ENTROPY

Intuitively: like the uncertainty of *Y*

Entropy: $H(Y) = -\sum_{k=1}^{K} p(Y = y_k) log_2 p(Y = y_k)$

Entropy of *Y* conditioned on X = x:

$H(Y \mid X = x) = -\sum_{k=1}^{K} p(Y = y_k \mid X = x) log_2 p(Y = y_k \mid X = x)$

Conditional entropy:

$$H(Y \mid X) = \sum_{x} p(X = x) H(Y \mid X = x)$$

Information gain:

$IG(Y, X) = H(Y) - H(Y \mid X)$

Can greedily split on feature *X* which minimizes $H(Y \mid X)$ aka maximizes $IG(Y, X)$

DECISION TREE ALGORITHM

BuildTree(dataset *DS*)

if ($y^{(i)} == y$) for all examples in *DS*

  return y

elif ($\bar{x}^{(i)} == \bar{x}$) for all examples in *DS*

  return majority label

else

  $x_s, t_s = argmin_{x,t} H(y \mid [[x > t]])$

  DS_g = {examples in *DS* where $x_s \geq t_s$}

  BuildTree(DS_g)

  DS_l = {examples in *DS* where $x_s < t_s$}

  BuildTree(DS_l)

To prevent overfitting: set max depth or prune

BAGGING

1. Sample $\tilde{n}$ points *B* times with replacement
2. Build *B* decision trees using each of the *B* bootstrap replicates
3. Aggregate their prediction

Assume each decision tree has < 50% mis-classification rate and classifiers are independent

As $B \to \infty$, misclassification rate → 0. Reduce variance without increasing bias.

RANDOM FORESTS

1. Bootstrap sampling
2. At each node, best split is chosen from random subset of *m* < *d* features

ADABOOST

Stumps (decision trees with depth 1):

$h(\bar{x}, \bar{\theta}_m) = sign(\theta_{1,m}(x_d - \theta_{0,m}))$

where $\bar{\theta}_m = [d, \theta_{0,m}, \theta_{1,m}]$

d = split dimension, $\theta_{0,m}$ = split value,

$\theta_{1,m}$ = split direction (+ or -)

Give higher weight to previously misclassified points.

Algorithm:

1. Initialize the observation weights $\tilde{w}_0^{(i)} = \frac{1}{n}$, for all $i \in [1 \ldots n]$
2. For $m = 1$ to M:
   (a) Find: $\bar{\theta}_m = \arg\min_{\bar{\theta}} \sum_{i=1}^{n} \tilde{w}_{m-1}^{(i)} [\![y^{(i)} \neq h(\bar{x}^{(i)}; \bar{\theta})]\!]$
   (b) Given $\bar{\theta}_m$, compute: $\hat{\epsilon}_m = \sum_{i=1}^{n} \tilde{w}_{m-1}^{(i)} [\![y^{(i)} \neq h(\bar{x}^{(i)}; \bar{\theta}_m)]\!]$.
   (c) Compute $\alpha_m = \frac{1}{2}\ln\left(\frac{1-\hat{\epsilon}_m}{\hat{\epsilon}_m}\right)$.
   (d) Update un-normalized weights for all $i \in [1 \ldots n]$:
   $$w_m^{(i)} \leftarrow \tilde{w}_{m-1}^{(i)} \cdot \exp\left[-y^{(i)}\alpha_m h(\bar{x}^{(i)}; \bar{\theta}_m)\right],$$
   (e) Normalize weights to sum to 1:
   $$\tilde{w}_m^{(i)} \leftarrow \frac{w_m^{(i)}}{\sum_i w_m^{(i)} := Z_m}$$
3. Output the final classifier: $h_M(\bar{x}) = \sum_{m=1}^{M} \alpha_m h(\bar{x}; \bar{\theta}_m)$

$[[y^{(i)} \neq h(\bar{x}^{(i)}, \bar{\theta})]]$ evaluates to 1 if the expression is true (we mispredicted) and 0 if it's false

$||x||_1 = \sum_{i=1}^{n} |x_i|$

$\lim_{x \to 0} x log(x) = 0$