Intro into Machine Learning

Decision Trees. Bagging. Ensembles. RandomForest. Stacked generalization

Third Machine Learning in High Energy Physics Summer School,
MLHEP 2017, July 17—23

Alexey Artemov[1,2]

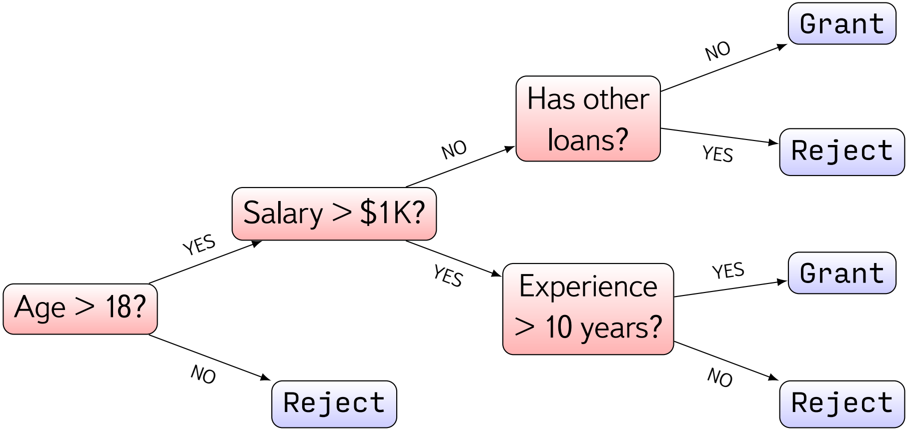
[1] Yandex LLC [2] National Research University Higher School of Economics

Lecture overview

- Decision Trees
- Bagging and Random Forests
- Learning Theory continued
- Stacked generalization

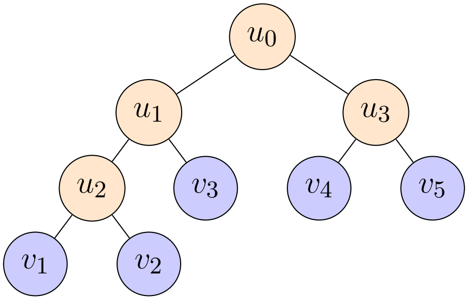
Decision trees

Decision making at a bank



Decision tree formalism

- Decision tree is a binary tree V
- Internal nodes $u \in V$: predicates $\beta_u : \mathbb{X} \to \{0, 1\}$
- Leafs $v \in V$: predictions x
- Algorithm $h(\mathbf{x})$ starts at $u = u_0$
  - Compute $b = \beta_u(\mathbf{x})$
  - If $b = 0$, $u \leftarrow \mathrm{LeftChild}(u)$
  - If $b = 1$, $u \leftarrow \mathrm{RightChild}(u)$
  - If u is a leaf, return b
- In practice: $\beta_u(\mathbf{x}; j, t) = [\mathbf{x}_j < t]$



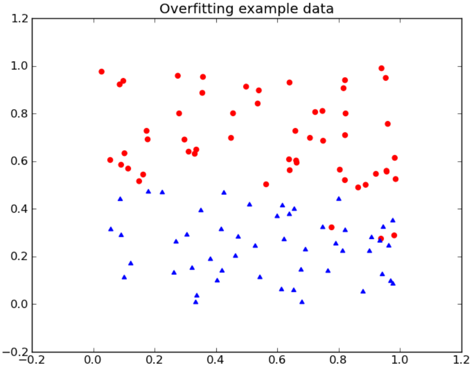
Greedy tree learning for binary classification

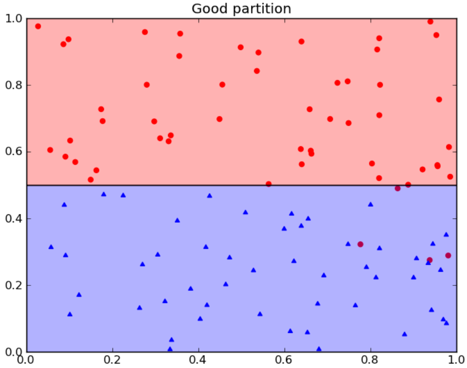
- Input: training set $X^\ell = \big\{(\mathbf{x}_i, y_i)\big\}_{i=1}^{\ell}$
  1. Greedily split X^ℓ into R_1 and R_2:

     $$R_1(j,t) = \{\mathbf{x} \in X^\ell | \mathbf{x}_j < t\}, \qquad R_2(j,t) = \{\mathbf{x} \in X^\ell | \mathbf{x}_j > t\}$$

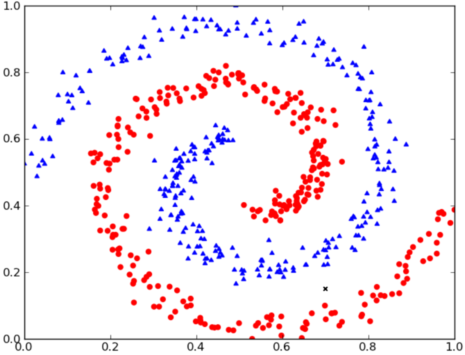
     optimizing a given loss: $Q(X^\ell, j, t) \to \min\limits_{(j,t)}$
  2. Create internal node u corresponding to the predicate $[\mathbf{x}_j < t]$
  3. If a stopping criterion is satisfied for u,
     declare it a leaf, setting some $c_u \in \mathbb{Y}$ as leaf prediction
  4. If not, repeat 1–2 for $R_1(j,t)$ and $R_2(j,t)$
- Output: a decision tree V

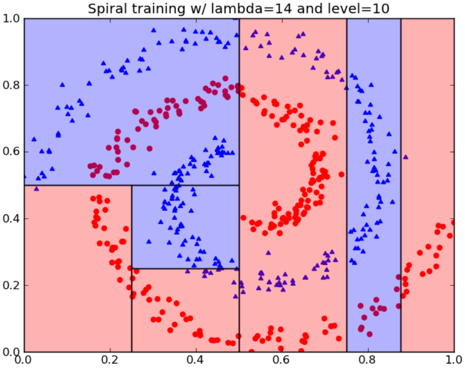
Greedy tree learning for binary classification



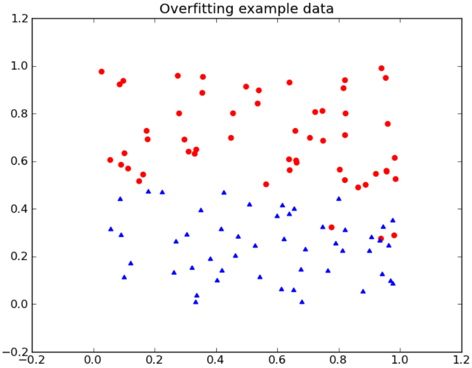


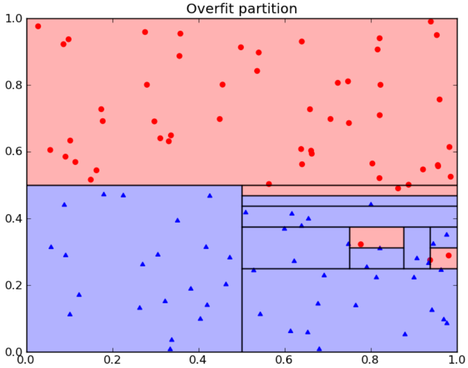
Greedy tree learning for binary classification





With decision trees, overfitting is extra-easy!





Design choices for learning a decision tree classifier

- Type of predicate in internal nodes
- The loss function $Q(X^\ell, j, t)$
- The stopping criterion
- Hacks: missing values, pruning, etc.

- CART, C4.5, ID3

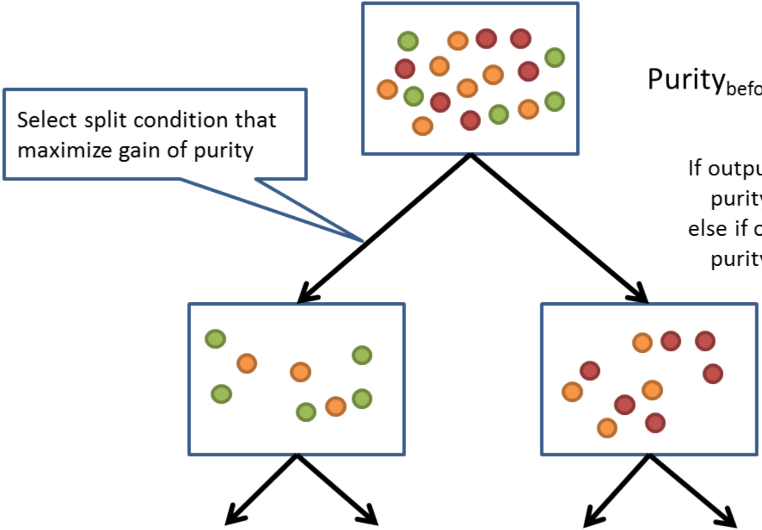
The loss function $Q(X^\ell, j, t)$

- R_m: the subset of X^ℓ at step m
- With the current split, let $R_l \subseteq R_m$ go left and $R_l \subseteq R_m$ go right
- Choose predicate to optimize

$$Q(R_m, j, t) = H(R_m) - \frac{|R_l|}{|R_m|} H(R_l) - \frac{|R_r|}{|R_m|} H(R_r) \to \max$$

- $H(R)$: impurity criterion
- Generally $H(R) = \min\limits_{c \in \mathbb{Y}} \frac{1}{|R|} \sum\limits_{(\mathbf{x}_i, y_i) \in R} L(y_i, c)$

The idea: maximize purity



$\text{Purity}_{\text{before-split}}$

If output is numeric
    purity = variance
else if output is categorical
    purity = entropy or gini

$\text{Purity}_{\text{after-split}}$

Picture credit: https://dzone.com/refcardz/machine-learning-predictive

Examples of information criteria

- Regression:
  - $H(R) = \min_{c \in \mathbb{Y}} |R|^{-1} \sum_{(\mathbf{x}_i, y_i) \in R} (y_i - c)^2$
  - Sum of squared residuals minimized by $c = |R|^{-1} \sum_{(\mathbf{x}_j, y_j) \in R} y_j$
  - Impurity $\equiv$ variance of the target
- Classification:
  - Let $p_k = |R|^{-1} \sum_{(\mathbf{x}_i, y_i) \in R} [y_i = k]$ (share of y_i's equal to k)
  - Miss rate: $H(R) = \min_{c \in \mathbb{Y}} |R|^{-1} \sum_{(\mathbf{x}_i, y_i) \in R} [y_i \neq c]$

    Minimizing miss rate $1 - p_{k_*}$,

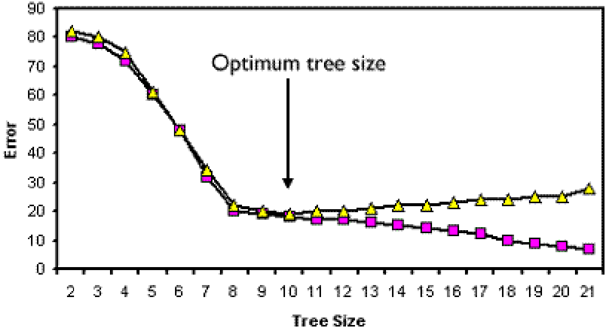
    Gini index $\sum_{k=1}^{K} p_k (1 - p_k)$,

    Cross-entropy $- \sum_{k=1}^{K} p_k \log p_k$

Stopping rules for decision tree learning

- Significantly impacts learning performance
- Multiple choices available:
  - Maximum tree depth
  - Minimum number of objects in leaf
  - Maximum number of leafs in tree
  - Stop if all objects fall into same leaf
  - Constrain quality improvement
    (stop when improvement gains drop below s%)
- Typically selected via exhaustive search and cross-validation

Decision tree pruning

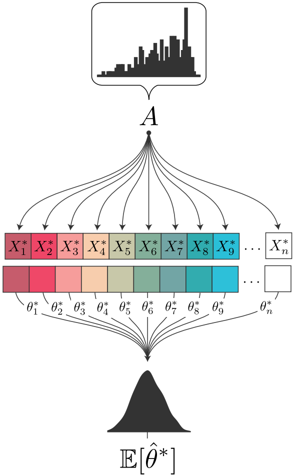


- Learn a large tree (effectively overfit the training set)
- Detect overfitting via K-fold cross-validation
- Optimize structure by removing least important nodes

Bagging and Random Forests

The bootstrapping procedure

- Input: a sample $X^\ell = \{(x_i, y_i)\}_{i=1}^{\ell}$
- Bootstrapping: generate new samples X_1^m of (x_i, y_i) drawn from X^ℓ uniformly at random with replacement (replicated (x_i, y_i) possible!)
- Ensemble learning idea:
  1. Generate N bootstrapped samples $X_1^m, \ldots, X_N^m$
  2. Learn N hypotheses $h_1, \ldots, h_N$
  3. Average predictions to obtain $h(x) = \frac{1}{N} \sum_{i=1}^{N} h_i(x)$
  4. Profit!



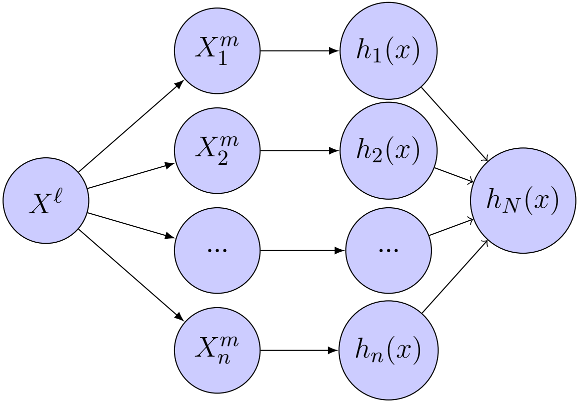
Picture credit: http://www.drbunsen.org/bootstrap-in-picture

Bagging: bootstrap aggregation (+ demo)

- Input: a sample $X^\ell = \{(x_i, y_i)\}_{i=1}^{\ell}$
- Weak learners via bootstrapping $\tilde{\mu}(X^\ell) = \mu(\tilde{X}^\ell)$
- Ensemble average

$$h_N(x) = \frac{1}{N}\sum_{i=1}^{N} h_i(x) =$$

$$= \frac{1}{N}\sum_{i=1}^{N} \tilde{\mu}(X^\ell)(x)$$



The Random Forest algorithm

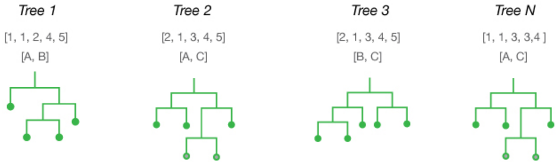
- Bagging over (weak) decision trees
- Reduce error via averaging over instances and features
- Input: a sample $X^\ell = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$, where $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{Y}$
- The algorithm iterates for $i = 1, \ldots, N$:
  1. Pick p random features out of d
  2. Bootstrap a sample $X_i^m = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ where $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{Y}$
  3. Learn a decision tree $h_i(\mathbf{x})$ using bootstrapped X_i^m
  4. Stop when leafs in h_i contain less that n_{min} instances

$\mathbf{x}_i \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$

$X^\ell = \{(\mathbf{x}_i, y_i)\}_{i=1}^{5}$

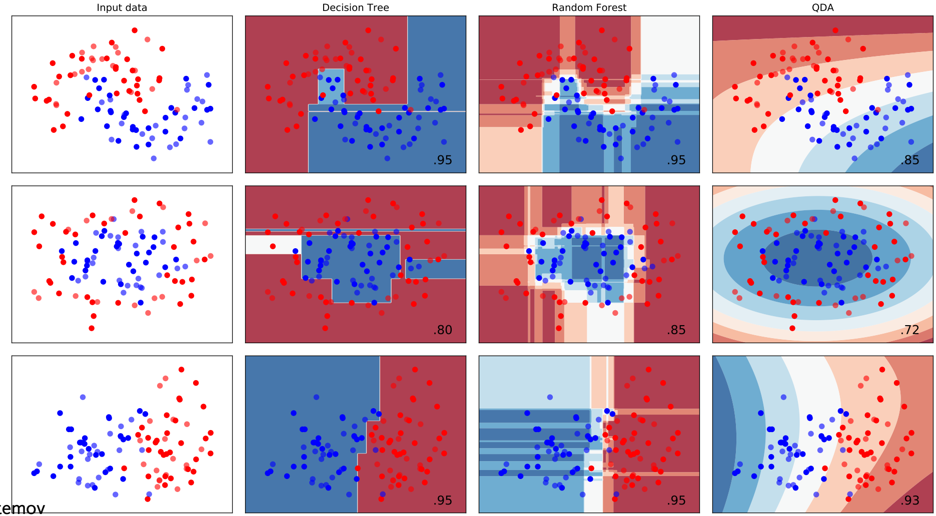
Bootstrap X_i^m, $i \in \{1, 2, 3, 4\}$

Learn Tree$_i(\mathbf{x})$ using X_i^m



Picture credit: http://www.thefactmachine.com/random-forests

Random Forest: synthetic examples



Learning Theory (continued)

Expected risk formalism

- Input: the training set $X^\ell = \{(x_i, y_i)\}_{i=1}^{\ell}$
- Suppose $(x_i, y_i) \in \mathbb{X} \times \mathbb{Y}$ are generated from a distribution $p(x, y)$
- Consider the MSE loss $Q(y, h) = \big(y - h(x)\big)^2$
- Expected risk: average (over $p(x, y)$) squared loss when using h

$$R(h) = \mathbb{E}_{x,y}\Big[\big(y - h(x)\big)^2\Big] = \int_{\mathbb{X}}\int_{\mathbb{Y}} p(x, y)\big(y - a(x)\big)^2 dxdy.$$

- A statement: the expected risk is minimized by

$$h_*(x) = \mathbb{E}[y \,|\, x] = \int_{\mathbb{Y}} yp(y \,|\, x)dy = \arg\min_h R(h)$$

Bias-variance decomposition

- Input: the training set $X^\ell = \{(x_i, y_i)\}_{i=1}^\ell$
- Learning method $\mu : (\mathbb{X} \times \mathbb{Y})^\ell \to \mathbb{H}$
- Can evaluate quality using average (over possible samples X^ℓ)

$$Q(\mu) = \mathbb{E}_{X^\ell}\left[\mathbb{E}_{x,y}\left[\left(y - \mu(X^\ell)(x)\right)^2\right]\right] =$$

$$= \int_{(\mathbb{X}\times\mathbb{Y})^\ell} \int_{\mathbb{X}\times\mathbb{Y}} \left(y - \mu(X^\ell)(x)\right)^2 p(x, y) \prod_{i=1}^{\ell} p(x_i, y_i) dx dy dx_i dy_i$$

Bias-variance decomposition

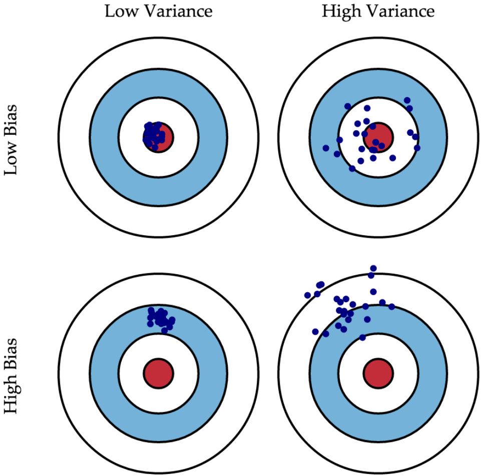
- We arrive at the famous **bias-variance**(-noise) **decomposition**

$$Q(\mu) = \underbrace{\mathbb{E}_{x,y}\left[\left(y - \mathbb{E}[y \,|\, x]\right)^2\right]}_{\text{noise}} +$$

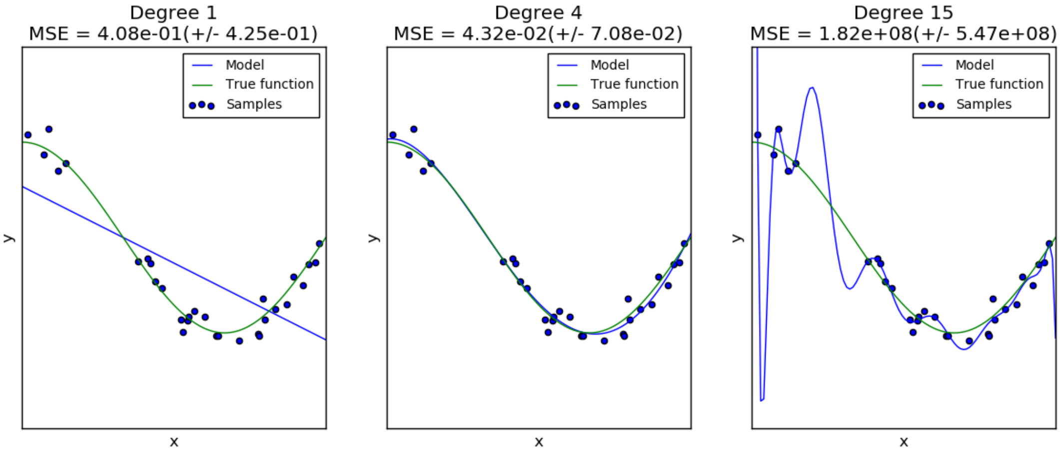
$$+ \underbrace{\mathbb{E}_{x}\left[\left(\mathbb{E}_{X^\ell}\left[\mu(X^\ell)\right] - \mathbb{E}[y \,|\, x]\right)^2\right]}_{\text{bias}} + \underbrace{\mathbb{E}_{x}\left[\mathbb{E}_{X^\ell}\left[\left(\mu(X^\ell) - \mathbb{E}_{X^\ell}\left[\mu(X^\ell)\right]\right)^2\right]\right]}_{\text{variance}}$$

- Noise term: an error of an ideal learner (nobody can do better!)
- Bias term: learner's approximation of the ideal algorithm
  - The more complex the learning algorithm, the lower the bias
- Variance term: sensitivity to sample replacement
  - Simple algorithms have lower variance

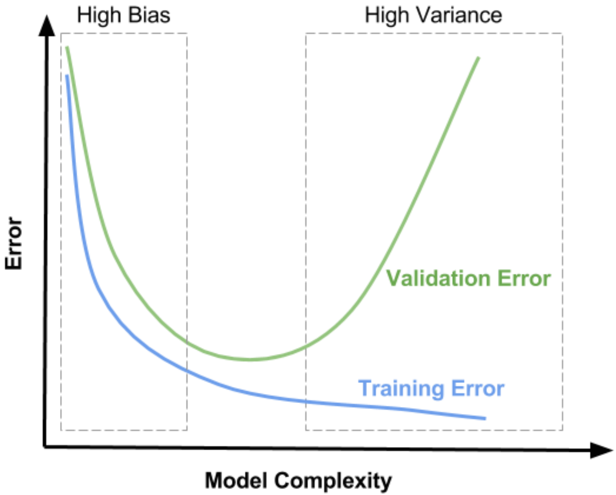
Bias-variance decomposition



Polynomial fits of different degrees



Bias-variance tradeoff



Picture credit: http://www.learnopencv.com/bias-variance-tradeoff-in-machine-learning

An application: statistical analysis of Bagging

- Bias: not made any worse by bagging multiple hypotheses

$$\mathbb{E}_{x,y}\left[\left(\mathbb{E}_{X^\ell}\left[\frac{1}{N}\sum_{n=1}^{N}\tilde{\mu}(X^\ell)(x)\right] - \mathbb{E}[y\,|\,x]\right)^2\right] =$$

$$= \mathbb{E}_{x,y}\left[\left(\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_X^\ell[\tilde{\mu}(X^\ell)(x)] - \mathbb{E}[y\,|\,x]\right)^2\right] =$$

$$= \mathbb{E}_{x,y}\left[\left(\mathbb{E}_{X^\ell}\left[\tilde{\mu}(X^\ell)(x)\right] - \mathbb{E}[y\,|\,x]\right)^2\right]$$

An application: statistical analysis of Bagging

- Variance: N times lower for uncorrelated hypotheses, yet not as much an improvement for highly correlated

$$\mathbb{E}_{x,y}\Big[\mathbb{E}_{X^\ell}\Big[\Big(\frac{1}{N}\sum_{n=1}^{N}\tilde{\mu}(X^\ell)(x) - \mathbb{E}_{X^\ell}\Big[\frac{1}{N}\sum_{n=1}^{N}\tilde{\mu}(X^\ell)(x)\Big]\Big)^2\Big]\Big] =$$

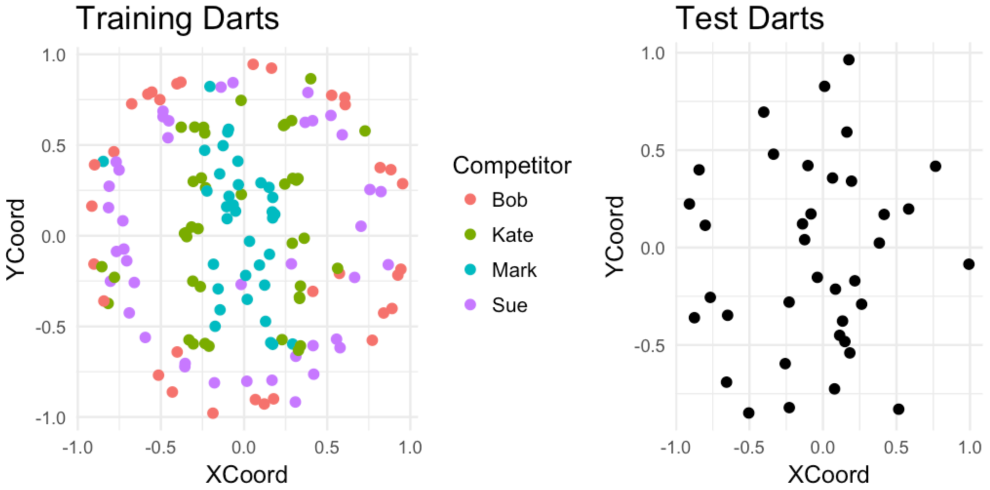
$$= \frac{1}{N}\mathbb{E}_{x,y}\Big[\mathbb{E}_{X^\ell}\Big[\Big(\tilde{\mu}(X^\ell)(x) - \mathbb{E}_{X^\ell}\big[\tilde{\mu}(X)(x)\big]\Big)^2\Big]\Big] +$$

$$+ \frac{N(N-1)}{N^2}\mathbb{E}_{x,y}\Big[\mathbb{E}_{X^\ell}\Big[\Big(\tilde{\mu}(X^\ell)(x) - \mathbb{E}_{X^\ell}\big[\tilde{\mu}(X^\ell)(x)\big]\Big) \times$$

$$\times \Big(\tilde{\mu}(X^\ell)(x) - \mathbb{E}_{X^\ell}\big[\tilde{\mu}(X^\ell)(x)\big]\Big)\Big]\Big]$$

Stacked generalization

Stacking motivation: the game of Darts



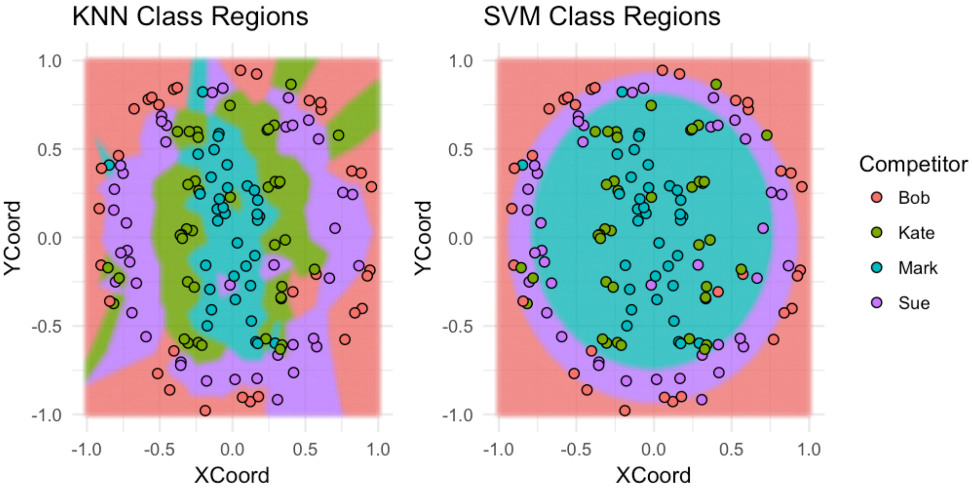
Picture credit: http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice

Base model training

- Select k nearest neighbours as base model 1
- Fit base model 1 in the most fancy way possible
  (grid search for optimal k using K-fold cross-validation, etc.)
- k-NN accuracy on Test Darts: 70%

- Select Support Vector Machine as base model 2
- Fit base model 2 in the most fancy way possible
  (different penalizations, grid search for optimal kernel width using K-fold cross-validation, etc.)
- SVM accuracy on Test Darts: 78%

Results for base models



Picture credit: http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice

Stacking base models

1. Partition `train` into 5 folds
2. Create `train_meta`/`test_meta`: same `row/fold Ids` as in `train/test`, empty `M1/M2`
3. For each $\mathrm{Fold}_i \in \{\text{Fold}_1, \ldots, \text{Fold}_5\}$
   3.1 Combine the other 4 folds for training $\to \mathrm{Fold}_{-i}$
   3.2 Fit each base model to Fold_{-i}, predict on Fold_i, save prdections to `M1/M2` in `train_meta`
4. Fit each base model to `train`, predict on `test`, save predictions to `M1/M2` in `test_meta`
5. Fit stacking model `S` to `train_meta`, using `M1/M2` as features
6. Use the stacked model S to make final predictions on `test_meta`

Results for base models

Picture credit: http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice

Conclusion

- Decision trees: intuitive and interpretable, yet prone to overfitting
- Bootstrapping: a general statistical technique for computing sample functionals (and their variance)
- Bagging: meta-learner over arbitrary weak algorithms via bootstrap aggregation
- The Random Forest algorithm: Bagging over decision trees
- Stacked generalization: blend output of weak learners (weak signals) with raw features

Bonus track

Minimum of the expected risk: the proof

- Transform the loss

$$\begin{aligned} Q(y, h(x)) &= (y - h(x))^2 = (y - \mathbb{E}(y \,|\, x) + \mathbb{E}(y \,|\, x) - h(x))^2 = \\ &= (y - \mathbb{E}(y \,|\, x))^2 + 2(y - \mathbb{E}(y \,|\, x))(\mathbb{E}(y \,|\, x) - h(x)) + \\ &+ (\mathbb{E}(y \,|\, x) - h(x))^2. \end{aligned}$$

- Write the expected risk

$$\begin{aligned} R(h) &= \mathbb{E}_{x,y} Q(y, h(x)) = \\ &= \mathbb{E}_{x,y}(y - \mathbb{E}(y \,|\, x))^2 + \mathbb{E}_{x,y}(\mathbb{E}(y \,|\, x) - h(x))^2 + \\ &+ 2\mathbb{E}_{x,y}(y - \mathbb{E}(y \,|\, x))(\mathbb{E}(y \,|\, x) - h(x)). \end{aligned}$$

Minimum of the expected risk: the proof

- Consider $\mathbb{E}_{x,y}(y - \mathbb{E}(y \,|\, x))(\mathbb{E}(y \,|\, x) - h(x))$ which is essentially some

$$\mathbb{E}_x \mathbb{E}_y [f(x, y) \,|\, x] = \int_{\mathbb{X}} \left(\int_{\mathbb{Y}} f(x, y) p(y \,|\, x) dy \right) p(x) dx$$

meaning that (as $\big(\mathbb{E}(y \,|\, x) - h(x)\big)$ is independent of y):

$$\begin{aligned}
&\mathbb{E}_x \mathbb{E}_y \Big[\big(y - \mathbb{E}(y \,|\, x)\big) \big(\mathbb{E}(y \,|\, x) - h(x)\big) \,|\, x \Big] = \\
&= \mathbb{E}_x \Big(\big(\mathbb{E}(y \,|\, x) - h(x)\big) \mathbb{E}_y \Big[\big(y - \mathbb{E}(y \,|\, x)\big) \,|\, x \Big] \Big) = \\
&= \mathbb{E}_x \Big(\big(\mathbb{E}(y \,|\, x) - h(x)\big) \big(\mathbb{E}(y \,|\, x) - \mathbb{E}(y \,|\, x)\big) \Big) = \\
&= 0
\end{aligned}$$

Minimum of the expected risk: the proof

- We obtain that the expected risk has the form

$$R(h) = \mathbb{E}_{x,y}(y - \mathbb{E}(y \,|\, x))^2 + \mathbb{E}_{x,y}(\mathbb{E}(y \,|\, x) - h(x))^2.$$

- Both summands are nonnegative meaning that the sum is minimized when

$$h_*(x) = \mathbb{E}(y \,|\, x) = \int_{\mathbb{Y}} yp(y \,|\, x)dy.$$

Bias-variance decomposition

- Input: the training set $X^\ell = \{(x_i, y_i)\}_{i=1}^\ell$
- Learning method $\mu : (\mathbb{X} \times \mathbb{Y})^\ell \to \mathbb{H}$
- Can evaluate quality using average (over possible samples X^ℓ)

$$Q(\mu) = \mathbb{E}_{X^\ell}\Big[\mathbb{E}_{x,y}\Big[\big(y - \mu(X^\ell)(x)\big)^2\Big]\Big] =$$

$$= \int_{(\mathbb{X}\times\mathbb{Y})^\ell} \int_{\mathbb{X}\times\mathbb{Y}} \big(y - \mu(X^\ell)(x)\big)^2 p(x, y) \prod_{i=1}^{\ell} p(x_i, y_i) dx dy dx_i dy_i$$

- For a fixed sample X^ℓ, we know the expected risk

$$\mathbb{E}_{x,y}\Big[\big(y - \mu(X^\ell)\big)^2\Big] = \mathbb{E}_{x,y}\Big[\big(y - \mathbb{E}[y \,|\, x]\big)^2\Big] + \mathbb{E}_{x,y}\Big[\big(\mathbb{E}[y \,|\, x] - \mu(X^\ell)\big)^2\Big]$$

Bias-variance decomposition

- Plugging the expression for fixed X^ℓ into $Q(\mu)$, we obtain

$$Q(\mu) = \mathbb{E}_{X^\ell}\Big[\underbrace{\mathbb{E}_{x,y}\Big[\big(y - \mathbb{E}[y \,|\, x]\big)^2\Big]}_{\text{independent of } X^\ell} + \mathbb{E}_{x,y}\Big[\big(\mathbb{E}[y \,|\, x] - \mu(X^\ell)\big)^2\Big]\Big]$$

- Transforming the second summand

$$\mathbb{E}_{x,y}\Big[\mathbb{E}_{X^\ell}\Big[\big(\mathbb{E}[y \,|\, x] - \mu(X^\ell)\big)^2\Big]\Big] =$$

$$= \mathbb{E}_{x,y}\Big[\mathbb{E}_{X^\ell}\Big[\big(\mathbb{E}[y \,|\, x] - \mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big] + \mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big] - \mu(X^\ell)\big)^2\Big]\Big] =$$

$$= \mathbb{E}_{x,y}\Big[\mathbb{E}_{X^\ell}\Big[\underbrace{\big(\mathbb{E}[y \,|\, x] - \mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big]\big)^2}_{\text{independent of } X^\ell}\Big]\Big] + \mathbb{E}_{x,y}\Big[\mathbb{E}_{X^\ell}\Big[\big(\mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big] - \mu(X^\ell)\big)$$

$$+ 2\mathbb{E}_{x,y}\Big[\mathbb{E}_{X^\ell}\Big[\big(\mathbb{E}[y \,|\, x] - \mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big]\big)\big(\mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big] - \mu(X^\ell)\big)\Big]\Big]$$

Bias-variance decomposition

- We prove that the last term is zero:

$$\begin{aligned}
&\mathbb{E}_{X^\ell}\Big[\big(\mathbb{E}[y \mid x] - \mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big]\big)\big(\mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big] - \mu(X^\ell)\big)\Big] = \\
&= \big(\mathbb{E}[y \mid x] - \mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big]\big)\mathbb{E}_X\Big[\mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big] - \mu(X^\ell)\Big] = \\
&= \big(\mathbb{E}[y \mid x] - \mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big]\big)\Big[\mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big] - \mathbb{E}_{X^\ell}\big[\mu(X^\ell)\big]\Big] = \\
&= 0.
\end{aligned}$$