Uplift Modeling Methods 1

ECML/PKDD'22 Uplift modeling tutorial & workshop

Szymon Jaroszewicz, Wouter Verbeke

Faculty of Mathematics and Information Science
WARSAW UNIVERSITY OF TECHNOLOGY

KU LEUVEN

September 18, 2022

Uplift modeling

- P^T probabilities in the treatment group
- P^C probabilities in the control group

Uplift models predict change in behaviour resulting from the action

$$P^T(Y \mid x) - P^C(Y \mid x)$$

Main difficulty of uplift modeling

The fundamental problem of causal inference

- Our knowledge is always incomplete
- For each training case we know either
  - what happened after the treatment, or
  - what happened if no treatment was given
- Never both!

- This makes designing uplift algorithms more challenging

P.W. Holland, *Statistics and Causal Inference*, 1986

Uplift modeling: basic methods

The two model approach

An obvious approach to uplift modeling:

1. Build a classifier M^T for $P^T(Y|\mathbf{X})$ on the treatment sample
2. Build a classifier M^C for $P^C(Y|\mathbf{X})$ on the control sample
3. The uplift model subtracts probabilities predicted by both classifiers

$$M^U(Y|\mathbf{X}) = M^T(Y|\mathbf{X}) - M^C(Y|\mathbf{X})$$

Also known as double model, T-learner

Two model approach

Advantages:

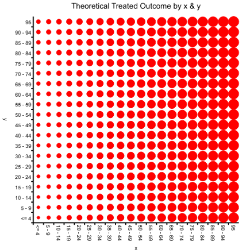
- Works with existing classification models
- Good probability predictions $\Rightarrow$ good uplift prediction

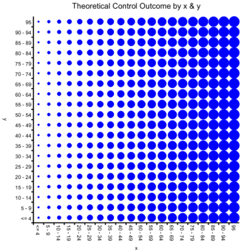
Disadvantages:

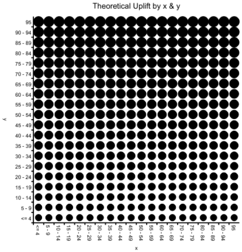
- Differences between class probabilities can follow a different pattern than the probabilities themselves
  - each classifier focuses on changes in class probabilities but ignores the weaker 'uplift signal'
  - algorithms designed to focus directly on uplift can give better results

Two model approach – failure example

- source: Radcliffe, Surry, *Real-World Uplift Modelling with Significance-Based Uplift Trees*, 2011
- Two variables, double decision tree







- qini measure 8.44% (double tree), 25.72% uplift tree

Two model approach, final remarks

- Designing algorithms which model uplift directly is the main research interest of uplift modeling
- However... in many cases the double model works surprisingly well

Decision trees for uplift modeling

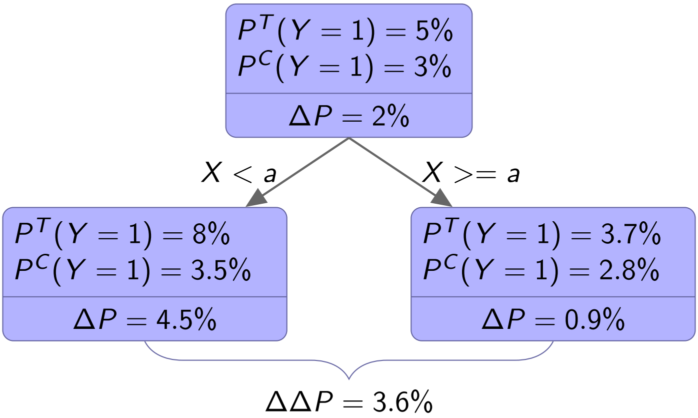
Decision trees for uplift modeling

Main idea

Modify splitting criteria to maximize differences between treated/control responses

Hansotia, Rukstales 2002

The $\Delta\Delta P$ criterion



Pick a test with highest $\Delta\Delta P$

Hansotia, Rukstales 2002

- It is not in line with 'modern' decision tree learning
  - splitting criterion directly maximizes the difference between probabilities (target criterion)
  - no pruning
- Rzepakowski, Jaroszewicz 2010, 2012
  - splitting criterion based on Information Theory, more in line with modern decision trees
  - pruning designed for uplift modeling
  - multiclass problems and multiway splits possible
  - if the control group is empty, the algorithm reduces to classical decision tree learning

Kullback-Leibler divergence

- Let $P = (p_1, \ldots, p_k)$, $Q = (q_1, \ldots, q_k)$ be two probability distributions
- The Kullback-Leibler divergence between them is defined as

$$KL\left(P : Q\right) = \sum_{i=1}^{k} p_i \log \frac{p_i}{q_i}$$

- Gibbs' inequality

$$KL(P : Q) \geq 0 \quad \text{with equality iff } P = Q$$

- Based on information theory
  - The KL-divergence can be interpreted as the number of extra bits per symbol if we build an optimal code based on a distribution Q instead of the true distribution P

KL divergence as a splitting criterion for uplift trees

- Measure difference between treatment and control groups using KL divergence

$$KL\left(P^T(Y) : P^C(Y)\right) = \sum_{y \in \mathrm{Dom}(Y)} P^T(y) \log \frac{P^T(y)}{P^C(y)}$$

KL divergence as a splitting criterion for uplift trees

- Measure difference between treatment and control groups using KL divergence

$$KL\left(P^T(Y) : P^C(Y)\right) = \sum_{y \in \mathrm{Dom}(Y)} P^T(y) \log \frac{P^T(y)}{P^C(y)}$$

- KL-divergence conditional on a test X

$$KL(P^T(Y) : P^C(Y) \mid X) =$$

$$\sum_{x \in \mathrm{Dom}(X)} \frac{N^T(X=x) + N^C(X=x)}{N^T + N^C} KL\left(P^T(Y|X=x) : P^C(Y|X=x)\right)$$

note the weighting factors
N^T and N^C denote counts in the treatment and control datasets

The KL_{gain}

How much larger does the difference between class distributions in T and C groups become after a split on X?

$$KL_{gain}(X) = KL\left(P^T(Y) : P^C(Y)|X\right) - KL\left(P^T(Y) : P^C(Y)\right)$$

Properties of the KL_{gain}

Properties:

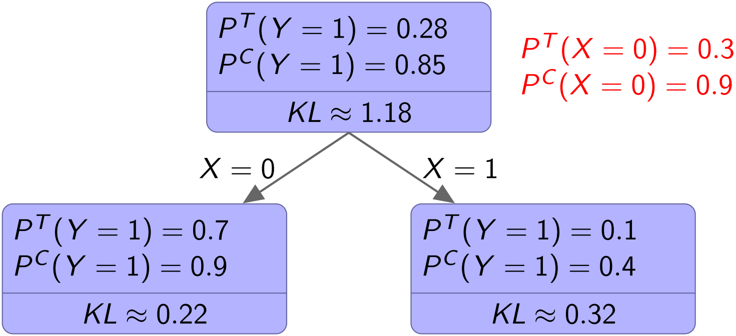
- If $Y \perp X$ then $KL_{gain}(X) = 0$
- If $P^T(Y|X) = P^C(Y|X)$ then

$$KL_{gain}(X) = \text{minimum}$$

- If the control group is empty, KL_{gain} reduces to entropy gain (Laplace correction is used on $P(Y)$)

Negative values of KL_{gain}

- Classification decision trees: $gain(X) \geq 0$
- $KL_{gain}(X)$ can be negative:



- Note the dependence of X on T/C group selection

Negative values of KL_{gain}

- Negative gain values are only possible when X depends on group selection
- This a variant of the Simpson's paradox

Theorem

If X is independent of the selection of the T and C groups then

$$KL_{gain}(X) \geq 0$$

- In practice we want X to be independent of the T/C group selection

The KL_{gain} ratio

- In standard decision trees, the gain is divided by test's entropy to punish tests with large number of outcomes
- In our case:

$$KL_{ratio}(X) = \frac{KL_{gain}(X)}{I(X)}$$

where

$$I(X) = H\left(\frac{N^T}{N}, \frac{N^C}{N}\right) KL(P^T(X) : P^C(X)) + \frac{N^T}{N} H(P^T(X)) + \frac{N^C}{N} H(P^C(X)) + \frac{1}{2}$$

- Tests with large numbers of outcomes are punished
- Tests for which $P^T(X)$ and $P^C(X)$ differ are punished
- This prevents splits correlated treatment indicator

Splitting criterion based on squared Euclidean distance

- Another splitting criterion based on Euclidean distance

$$E\left(P^T(Y) : P^C(Y)\right) = \sum_{y \in \mathrm{Dom}(Y)} \left(P^T(Y = y) - P^C(Y = y)\right)^2$$

- Better statistical properties (values are bounded)
- Symmetry
- Reduces to $Gini_{gain}$ when no control or treatment samples are present

Uplift trees – other approaches

- N. Radcliffe, Surry, *Real-World Uplift Modelling with Significance-Based Uplift Trees*, 2011
  - splitting criterion based on statistical tests
  - based on significance of a simple linear model
- L. Guelman et al., *Random Forests for Uplift Modeling: An Insurance Customer Retention Case*, 2012
  - splitting criterion based on statistical tests
  - extended to random forests

Trees for ITE estimation

- An approach from the ITE estimation community[1]
- Several splitting criteria based on MSE
  - equivalent to Euclidean distance based uplift trees[2]
  - propensity scores may be used to correct biased assignment
- Honesty
  - splits and leaf estimates on separate datasets
  - guanrantees covergence to true $P(Y|x)$
  - no need for propensity scores as $n \to \infty$
- Summary
  - nonrandomized trials allowed
  - nice asymptotic theory
  - data loss due to honesty

[1] S. Athey, G. Imbens. Recursive partitioning for heterogeneous causal effects, 2016

[2] Gutierrez and Jean-Yves Gérardy. Causal inference and uplift modelling: A review of the literature, 2017

Ensembles

- Bagging and Random Forests popular in uplift modeling (also ITE estimation)[1,2,3]
- Both methods work very well with uplift modeling
- Bagging often gives excellent results

- Boosting less common, but some methods exist[4]

[1] M. Soltys, S. Jaroszewicz, P. Rzepakowski. Ensemble methods for uplift modeling

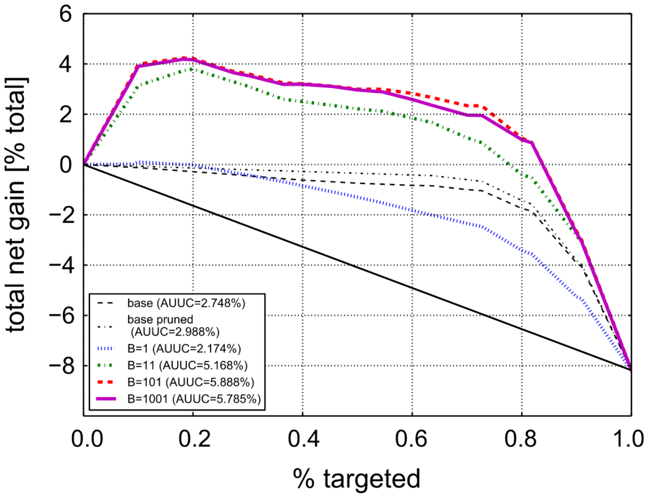
[2] L. Guelman et al., Random Forests for Uplift Modeling: An Insurance Customer Retention Case, 2012

[3] S. Wager, S. Athey. Estim. and Inference of Heterogeneous Treatment Effects using Random Forests, 2017

[4] M. Soltys, S. Jaroszewicz, Boosting algorithms for uplift modeling, 2018

Example: bagging a double tree

- Bone Marrow Transplant data (CGVT) from `R survival`



Why ensembles work so well?

- Building uplift models is usually more difficult than building classifiers
- Differences between treatment/control are smaller than within-class variablility
- So: uplift decision trees highly sensitive to small changes in training data
- This is turn results in highly diverse ensembles

Why ensembles work so well?

- Worst case for double model based approach:
  - high class variability
  - treatment and control class distributions *almost* identical
  - treatment and control models ignore the weak 'uplift signal'
- As a result: all M_i^T, M_i^C similar to each other and make highly correlated predictions
- Coviariance between two ensemble members

$$
\begin{aligned}
&\operatorname{cov}\left(M_i^T(\mathbf{x}) - M_i^C(\mathbf{x}), M_{i'}^T(\mathbf{x}) - M_{i'}^C(\mathbf{x})\right) \\
&\quad = \operatorname{cov}\left(M_i^T(\mathbf{x}), M_{i'}^T(\mathbf{x})\right) + \operatorname{cov}\left(M_i^C(\mathbf{x}), M_{i'}^C(\mathbf{x})\right) \\
&\quad\quad - \operatorname{cov}\left(M_i^T(\mathbf{x}), M_{i'}^C(\mathbf{x})\right) - \operatorname{cov}\left(M_i^C(\mathbf{x}), M_{i'}^T(\mathbf{x})\right) \\
&\quad \approx 0
\end{aligned}
$$

Linear models

Linear models for uplift modeling

- Still very important in practice
- Allow for theoretical understanding

The double linear model

- Idea: apply the two model approach (T-learner) to linear models

The double linear model

- Idea: apply the two model approach (T-learner) to linear models

- For regression:
  1. $\hat{\beta^T} = (X^{T\prime} X^T)^{-1} X^{T\prime} y^T$
  2. $\hat{\beta^C} = (X^{C\prime} X^C)^{-1} X^{C\prime} y^C$
  3. $\hat{\beta^U} = \hat{\beta^T} - \hat{\beta^C}$
- Get a single linear model of uplift/CATE

$$\hat{\tau}(x) = \hat{\beta^U} x$$

The double linear model

- Idea: apply the two model approach (T-learner) to linear models

- For regression:
  1. $\hat{\beta^T} = (X^{T'} X^T)^{-1} X^{T'} y^T$
  2. $\hat{\beta^C} = (X^{C'} X^C)^{-1} X^{C'} y^C$
  3. $\hat{\beta^U} = \hat{\beta^T} - \hat{\beta^C}$

- Get a single linear model of uplift/CATE

$$\hat{\tau}(x) = \hat{\beta^U} x$$

- For classification:
  - subtract probs predicted by two logistic models

Uplift modeling through class variable transformation

- Rediscovered many times (also analogous to double robust estimator)
- Allows for adapting an arbitrary classifier to uplift modeling

- Let $G \in \{T, C\}$ denote the group membership (treatment or control)
- Define an r.v.

$$Z = \begin{cases} 1 & \text{if } G = T \text{ and } Y = 1, \\ 1 & \text{if } G = C \text{ and } Y = 0, \\ 0 & \text{otherwise.} \end{cases}$$

- In plain English: flip the class in the control dataset

Uplift modeling through class variable transformation

- Now

$$
\begin{aligned}
&P(Z = 1|x) \\
&\quad = P^T(Y = 1|x)P(G = T|x) + P^C(Y = 0|x)P(G = C|x)
\end{aligned}
$$

- Assume that G is independent of x (randomization!):

$$
\begin{aligned}
&P(Z = 1|x) \\
&\quad = \; P^T(Y = 1|x)P(G = T) + \; P^C(Y = 0|x)P(G = C)
\end{aligned}
$$

Uplift modeling through class variable transformation

- Assume $P(G = T) = P(G = C) = \frac{1}{2}$ (otherwise reweight the datasets):

$$\begin{aligned} &2P(Z = 1|x) \\ &\quad = P^T(Y = 1|x) + P^C(Y = 0|x) \\ &\quad = P^T(Y = 1|x) + 1 - P^C(Y = 1|x) \end{aligned}$$

- Finally

$$P^T(Y = 1|x) - P^C(Y = 1|x) = 2P(Z = 1|x) - 1$$

Uplift modeling through class variable transformation

Conclusion

Modeling $P(Z = 1|X)$ is equivalent to modeling the difference between class probabilities in the treatment and control groups

The algorithm:

1. Flip the class in $\mathbf{D}^C$
2. Concatenate $\mathbf{D} = \mathbf{D}^T \cup \mathbf{D}^C$
3. Build any classifier on $\mathbf{D}$
4. The classifier is actually an uplift model

Advantages

- Any classifier can be turned into an uplift model
- A single model is built
  - coefficients are easier to interpret than for the double model
  - the model predicts uplift directly
    (will not focus on predicting classes themselves)
  - a single model is built on a large dataset
    (double model method subtracts two models built on small datasets)
- It seems such a model will almost always be better

Target variable transformation for regression

- Negate the sign of y in control and reweight
- Also rediscovered many times
- I.e. replace y with

$$\tilde{y}_i = \begin{cases} \frac{1}{p^T} y_i & \text{if treated} \\ -\frac{1}{p^C} y_i & \text{if control} \end{cases}$$

Target variable transformation for regression

- Negate the sign of y in control and reweight
- Also rediscovered many times
- I.e. replace y with

$$\tilde{y}_i = \begin{cases} \frac{1}{p^T} y_i & \text{if treated} \\ -\frac{1}{p^C} y_i & \text{if control} \end{cases}$$

- Linear models are easier to analyze
- Can we compare?

Comparison of uplift linear regression models

Theorem

Let $\hat{\beta}^U$ be the double regression estimator. If $\mathrm{Var} X_i = \Sigma$,

$$\sqrt{n}\left(\hat{\beta^U} - \beta^U\right) \xrightarrow{d} N\left(0, 2\left({\sigma^T}^2 + {\sigma^C}^2\right)\Sigma^{-1}\right)$$

Theorem

Let $\hat{\beta}^U$ be the transformed target regression. If $\mathrm{E} X_i = 0$ and $\mathrm{Var} X_i = \Sigma$ the

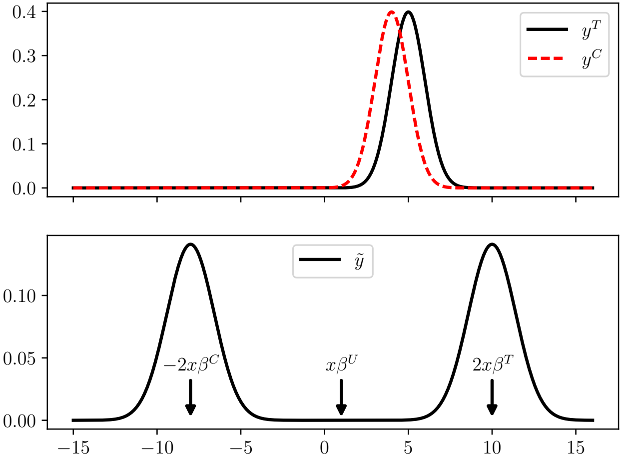
$$\sqrt{n}\left(\hat{\beta^U} - \beta^U\right) \xrightarrow{d} N\Bigg(0, 2\left({\sigma^T}^2 + {\sigma^C}^2\right)\Sigma^{-1} + bb' + \Sigma^{-1}\mathrm{Tr}(bb'\Sigma)\Bigg)$$

where $b = \beta^T + \beta^C$.

K. Rudaś, S. Jaroszewicz, *Linear regression for uplift modeling*, 2018

Intuition

distributions of treatment/control responses for fixed x



Corrected uplift regression

- Can we get a single uplift regression model without this problem?
- If we subtract some β^* from β^T, β^C uplift does not change

$$(\beta^T - \beta^*) - (\beta^C - \beta^*) = \beta^T - \beta^C = \beta^U$$

- If we pick $\beta^* = \frac{\beta^T + \beta^C}{2}$ we additionally get

$$b = (\beta^T - \beta^*) + (\beta^C - \beta^*) = 0$$

- How can we modify the original problem? We don't even know true β^T and β^C needed for β^*

Corrected uplift regression

1. Estimate β^*:

$$\hat{\beta}^* = (X'X)^{-1}X'y$$

2. Correct the original y

$$y^{corr} = y - X\hat{\beta}^*$$

3. Build transformed target uplift regression on corrected data

$$\hat{\beta}^U = (X'X)^{-1}X'\widetilde{y^{corr}}$$

K. Rudaś, S. Jaroszewicz, *Linear regression for uplift modeling*, 2018

Properties of the corrected uplift regression

Theorem

Let $\hat{\beta}^U$ be the corrected uplift regression estimator. Then

1. $\hat{\beta}^U$ is unbiased
2. If $\mathrm{E}X_i = 0$ and $\mathrm{Var}X_i = \Sigma$,

$$\sqrt{n}\left(\hat{\beta}^U - \beta^U\right) \xrightarrow{d} N\left(0, 2\left({\sigma^T}^2 + {\sigma^C}^2\right)\Sigma^{-1}\right)$$

- Asymptotic behavior identical to double regression (correction works)
- Experiments show it is also better for small n.
- Especially good for $\beta^T \approx \beta^C$.

Other approaches

- Regularization, shrinkage estimators
- Variable selection
- Uplift KNN
- Support Vector Machines
- Neural models
- Learning to rank

Software packages

- causalml from Uber
  - solid package
  - many methods
  - my recommendation
- EconML from Microsoft
  - focused on ITE estimation
- Several 'in developemnt' projects
  - tools4uplift and R package
  - scikit-uplift based on scikit-learn
  - uplift_sklearn based on scikit-learn