Cite as: Ioannis Tsamardinos, Vincenzo Lagani, Automated Machine Learning and Knowledge Discovery, ECCB 2018 Tutorial

Automated Machine Learning and Knowledge Discovery

IOANNIS TSAMARDINOS

PROFESSOR, CSD, UNIVERSITY OF CRETE

GNOSIS DATA ANALYSIS, CO-FOUNDER

VINCENZO LAGANI

ILIA STATE UNIVERSITY

GNOSIS DATA ANALYSIS, CO-FOUNDER

Outline

- **Part I (45')**
  - Introduction to the problem and the tutorial
  - Estimation of performance (single configuration)
- **Part II (45')**
  - Estimation of performance (multiple configurations)
  - Incorporating User Preferences
- **Part III (45')**
  - Feature Selection and Knowledge Discovery
  - Hyper-parameter search strategies
- **Part IV (45')**
  - Post-analysis interpretation and visualizations
  - AI-assisted Auto-ML (algorithm selection, pipeline synthesis, meta-learning, feature learning)
  - Putting all together – The Just Add Data Bio platform
  - Tools for Auto-ML

Outline

- **Part I (45')**
  - Introduction to the problem and the tutorial
  - Estimation of performance (single configuration)
- **Part II (45')**
  - Estimation of performance (multiple configurations)
  - Incorporating User Preferences
- **Part III (45')**
  - **Feature Selection and Knowledge Discovery**
  - **Hyper-parameter search strategies**
- **Part IV (45')**
  - Post-analysis interpretation and visualizations
  - AI-assisted Auto-ML (algorithm selection, pipeline synthesis, meta-learning, feature learning)
  - Putting all together – The Just Add Data Bio platform
  - Tools for Auto-ML

Feature Selection and Knowledge Discovery

Messages

- Feature Selection is ***arguably the main tool*** for knowledge discovery
- Causal models help understand the feature selection problem, in a non-parametric way
- **Causally-inspired algorithms:**
  - Provide theoretical guarantees
  - Applicable to any type of data for which a conditional independence test is available
  - Scale up to tens of millions of features **and** tens of millions of rows (**Big Volume Data**)
  - Competitive predictive performance against alternatives (e.g., Lasso)
  - Can find multiple, statistically-equivalent solutions
  - Solution(s) has intuitive causal interpretation
  - Robust, efficient implementations available

Why Feature Selection

Training data

predictors

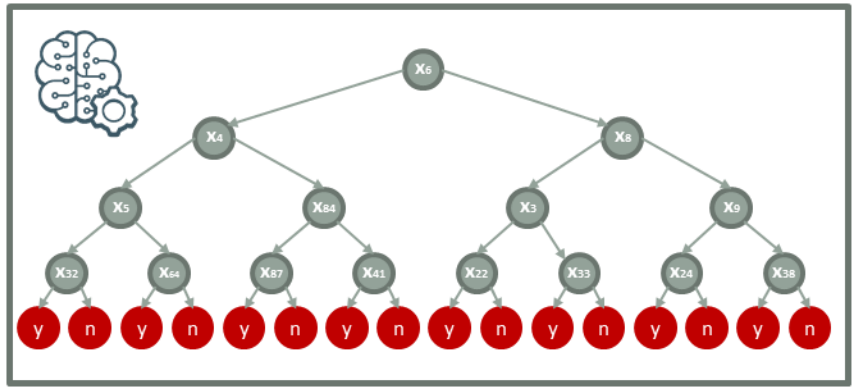
ID	x_1	x_2	x_3	x_4	....	x_m	target
1	26	0	0.3	0.06	...	2	yes
2	52	1	2.3	0.1	...	2	no
...	...	...	...	...	...	...	
n	34	0	5.8	0.04	...	3	no

instances

Learning Method

f

Model



Training data

predictors

ID	x_1	x_2	x_3	x_4	....	x_m	target
1	26	0	0.3	0.06	...	2	yes
2	52	1	2.3	0.1	...	2	no
...	...	...	...	...	...	...	
n	34	0	5.8	0.04	...	3	no

instances

Selection

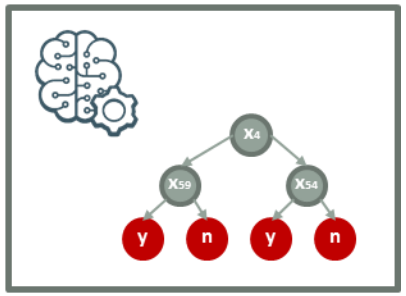
Feature selection

ID	x_3	x_4	x_m	target
1	0.3	0.06	2	yes
2	2.3	0.1	2	no
...	...	...	...	
n	5.8	0.04	3	no

Learning Method

f

Model



Why Feature Selection?

- Feature selection is *the* ***main tool for knowledge discovery*** *with data analysis*
  - Often it *the primary goal* of the analysis; the predictive model is only a side-product
  - Provides intuition to "what matters" for prediction
  - Connected to the *causal* mechanisms generating the data (Tsamardinos, Aliferis, AI&STATs 2003)
  - Dimensionality Reduction, e.g., PCA, is harder to interpret
- **Feature Selection = Knowledge Discovery**

- Also, may actually improve predictive performance
  - By removing irrelevant or redundant features, learning algorithms are facilitated
- Reduces the cost of storing, computing, measuring, processing the data

Defining Feature Selection (Oracle)

- $\mathrm{Ind}(T; \mathrm{X} \mid \mathbf{Z})$: X independent of T conditioned on (given) $\mathbf{Z}$
- Single Solution: Find a minimal-size feature subset $\boldsymbol{S} \subseteq \boldsymbol{F}$, s.t. $\mathrm{Ind}(T; \mathbf{F} \backslash \{\mathbf{S}\} \mid \mathbf{S})$
  - Equivalently: $\mathrm{P}(T \mid \boldsymbol{S}) = \mathrm{P}(T \mid \boldsymbol{F})$
  - Selected features $\boldsymbol{S}$ do not change the conditional probability of T
  - Selected features $\boldsymbol{S}$ carry all information to predict/diagnose T
  - There is no subset $\mathrm{S}' \subseteq \mathrm{S}$ s.t. $\mathrm{P}(T \mid \mathrm{S}') = \mathrm{P}(T \mid \mathrm{S})$
- *Definition requires knowledge of $P(T \mid \boldsymbol{F})$*
- S : a minimal-size set that renders all other features conditionally independent of T
  - S is called a **Markov Blanket of $\boldsymbol{T}$** [Markov Boundary in Pearl, Comput. Intel,1988]
- **NP-complete** problem even for linear regression [Welch, Biometrika, 69(2),1982]

Defining Feature Selection (no Oracle)

- Single Solution:
  - Maximize **performance** of model built with features $\boldsymbol{S}$ using **learner** $\boldsymbol{f}$, s.t. $|\boldsymbol{S}|$ is minimal

- No knowledge of conditional distribution of T, need to estimate from finite sample

Defining Feature Selection: Subtleties

- **"Maximizing performance"**: solution depends on performance metric
  - Example, $P(T+ \mid X+) = 0.6$, $P(T+ \mid X-) = 0.7$. and metric is accuracy: **accuracy is maximized without *X*!**
- **"using learner *f*"**: solution depends on learner
  - Example by [Kohavi & John, 1997]
  - $T = X + constant + \varepsilon$, $\varepsilon \sim N(0,1)$, $Y = 1$ always, and learner f is a linear classifier without a constant term
  - Optimal model fit with f as $T = X + constant \times Y$, Y participates in the solution!
- Sufficient conditions for finite-sample solution to converge to the asymptotic one
  - Learner should converge to (learn) $P(T \mid \mathbf{S})$ for a solution $\mathbf{S}$
  - Performance metric (loss) is **optimized** asymptotically **only** when $\mathrm{Ind}(T ; \mathbf{F} \backslash \{\mathbf{S}\} \mid \mathbf{S})$

Multiple Feature Selection (I)

- Important problem!
  - **Knowledge discovery**: Misleading to inform a biologist that only genes in *S* are "important" when other genes *S'* could replace them!
  - **Cost-aware feature selection**: when features have measurement cost, give options what to measure
- Multiple Solutions: Find ***all*** minimal-size subsets, s.t. $\boldsymbol{S \subseteq F}$, s.t. $\mathrm{Ind}(T ; \mathbf{F} \backslash \{\mathbf{S}\} | \mathbf{S})$
  - Not all minimal-size subsets have to have the same size!
- Much less studied problem

See KIAMB (Peña et al., 2007), TIE* (Statnikov et al., 2013), SES (Tsamardinos et al., 2012) & (Lagani et al., 2017) for current approaches.

Multiple Feature Selection (II)

- Related to **stability** of solutions!
- With finite sample, find all solutions that are "**statistically indistinguishable**"
  - Possible definitions of indistinguishable feature subsets S_1 and S_2
    - **Performance Equivalence**: performance metrics on predictions are the same (in a statistical sense) given the learning method
    - **Model Equivalence**: conditional distribution of predictions is the same given the learning method (independent of performance metric)
    - **Information Equivalence**: conditional distribution of predictions is the same (independently of performance metric and learner)

A Taxonomy of Feature Types (I)

- $\mathbf{X}$ provides no information for T in all contexts:
  - If $\mathbf{Ind}(\mathbf{X} ; \mathbf{T} \mid \mathbf{Z})$, $\forall \mathbf{Z} \subseteq \mathbf{F}$
  - Then X is ***irrelevant***

- $\mathbf{X}$ in all solutions (Markov Blankets)
  - If $\forall \boldsymbol{S} \subseteq \mathbf{F}$ s.t., $\boldsymbol{S}$ minimal, $\mathbf{Ind}(\mathbf{F} \backslash \mathbf{S} ; \mathbf{T} \mid \mathbf{S})$, then $X \subseteq \boldsymbol{S}$
  - Then X ***indispensable***

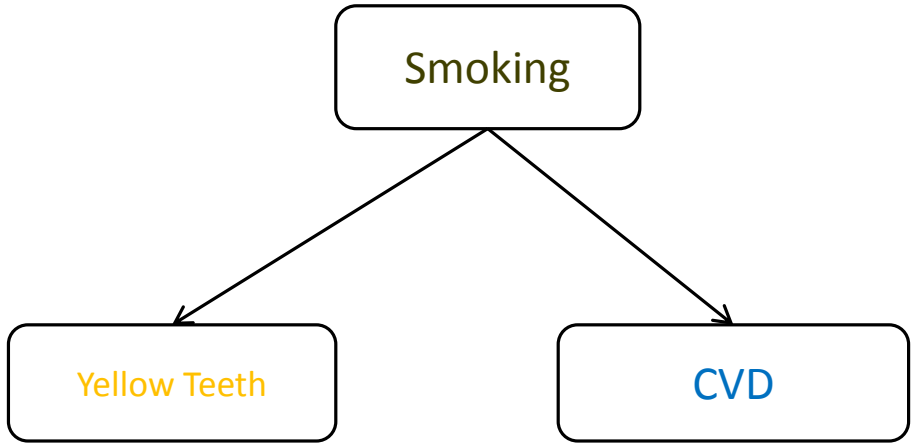
A Taxonomy of Feature Types (II)

- $\mathbf{X}$ not *indispensable*, but X in some solutions
  - The information provided by X is necessary for optimal prediction, yet, it can be substituted with other features
  - X ***replaceable***
  - **Replaceable features will not be stable!**
- $\mathbf{X}$ not irrelevant, or indispensable, or replaceable
  - $\mathbf{X}$ provides information for predicting T in some context (conditioned on some $\boldsymbol{Z} \subseteq \mathbf{F}$), but not required
  - Then X *is* ***redundant***
- Older classification to irrelevant, weakly relevant, strongly relevant [Kohavi & John, Artificial Intelligence, 97, 1-2,1997] coincides with irrelevant, redundant, indispensable when solution is unique

Causal models

Bayesian Networks (BNs)

Directed Acyclic Graph $\mathcal{G}$



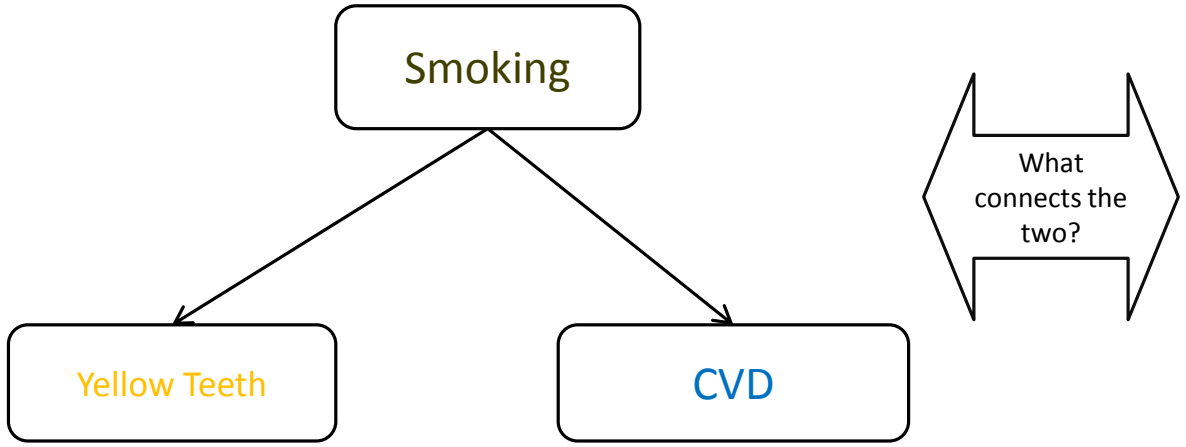
JPD$(\boldsymbol{V})$: $\mathcal{P}$

		CVD		
Yellow Teeth	Smoking	Y	N	
Y	Y	0.17	0.06	0.13
N	Y	0.06	0.02	0.08
Y	N	0.02	0.06	0.08
N	N	0.15	0.46	0.61
		0.4	0.6	1

(Causal) Markov Condition (MC):
Every variable is **independent** of its **non-descendants** given its **parents**
Causal interpretation: substitute "direct cause" for "parent" and "non-effect" for "non-descendant"

Bayesian Networks (BNs)

Directed Acyclic Graph $\mathcal{G}$

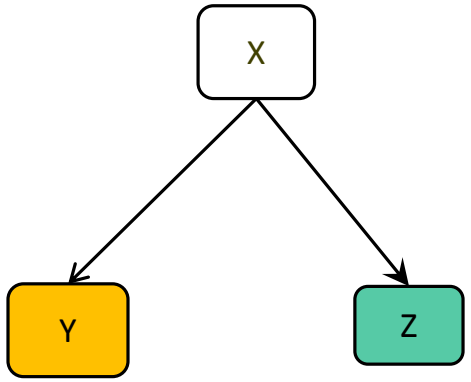


JPD$(\boldsymbol{V})$: $\mathcal{P}$

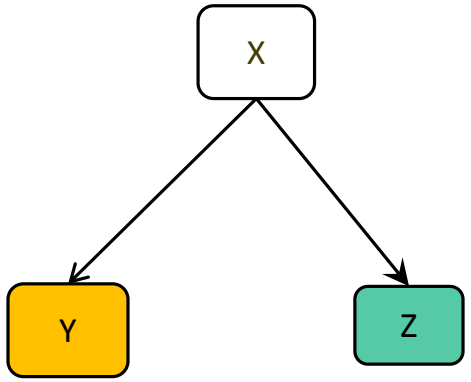
		CVD		
Yellow Teeth	Smoking	Y	N	
Y	Y	0.17	0.06	0.13
N	Y	0.06	0.02	0.08
Y	N	0.02	0.06	0.08
N	N	0.15	0.46	0.61
		0.4	0.6	1

(Causal) Markov Condition (MC):
Every variable is **independent** of its **non-descendants** given its **parents**
Causal interpretation: substitute "direct cause" for "parent" and "non-effect" for "non-descendant"

ASSUMPTIONS

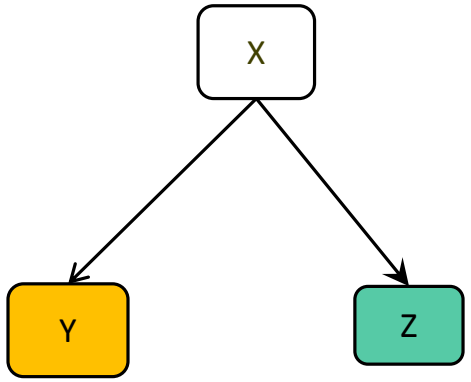


ASSUMPTIONS



Markov Condition:
Every variable is independent of
its non-descendants given its **parents**.

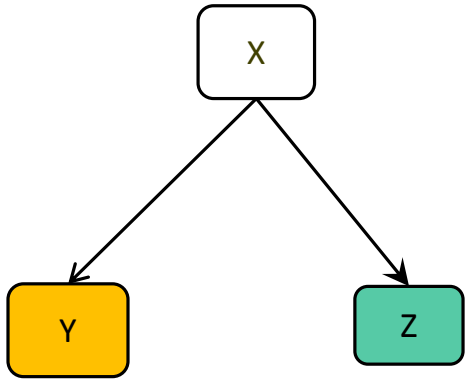
ASSUMPTIONS



$Ind(Y, Z \mid X)$

Markov Condition:
Every variable is independent of
its non-descendants given its **parents**.

ASSUMPTIONS

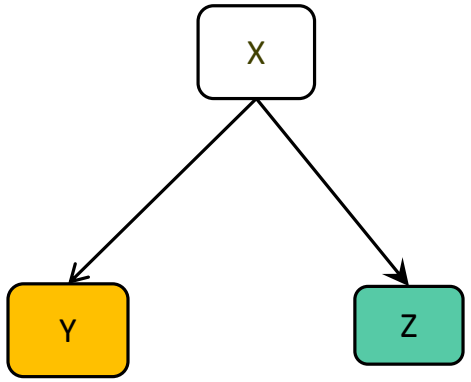


$Ind(Y, Z \mid X)$

Markov Condition:
Every variable is independent of **its non-descendants** given its **parents**.

Faithfulness Assumption:
Independences stem **only** from the network structure, **not the parameterization** of the distribution.

ASSUMPTIONS



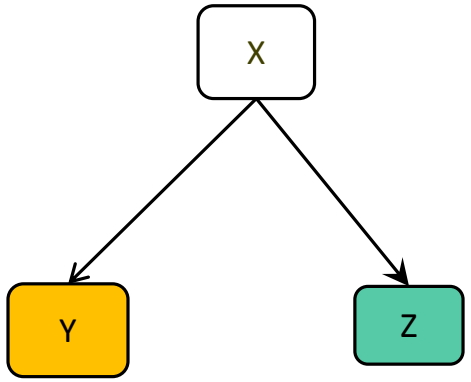
$Ind(Y, Z \mid X)$

$Dep(Y, Z \mid \emptyset)$
$Dep(X, Z \mid \emptyset)$
$Dep(X, Z \mid Y)$
$Dep(Y, X \mid \emptyset)$
$Dep(Y, X \mid Z)$

Markov Condition:
Every variable is independent of **its non-descendants** given its **parents**.

Faithfulness Assumption:
Independences stem **only** from the network structure, **not the parameterization** of the distribution.

ASSUMPTIONS



$Ind(Y, Z \mid X)$

$Dep(Y, Z \mid \emptyset)$
$Dep(X, Z \mid \emptyset)$
$Dep(X, Z \mid Y)$
$Dep(Y, X \mid \emptyset)$
$Dep(Y, X \mid Z)$

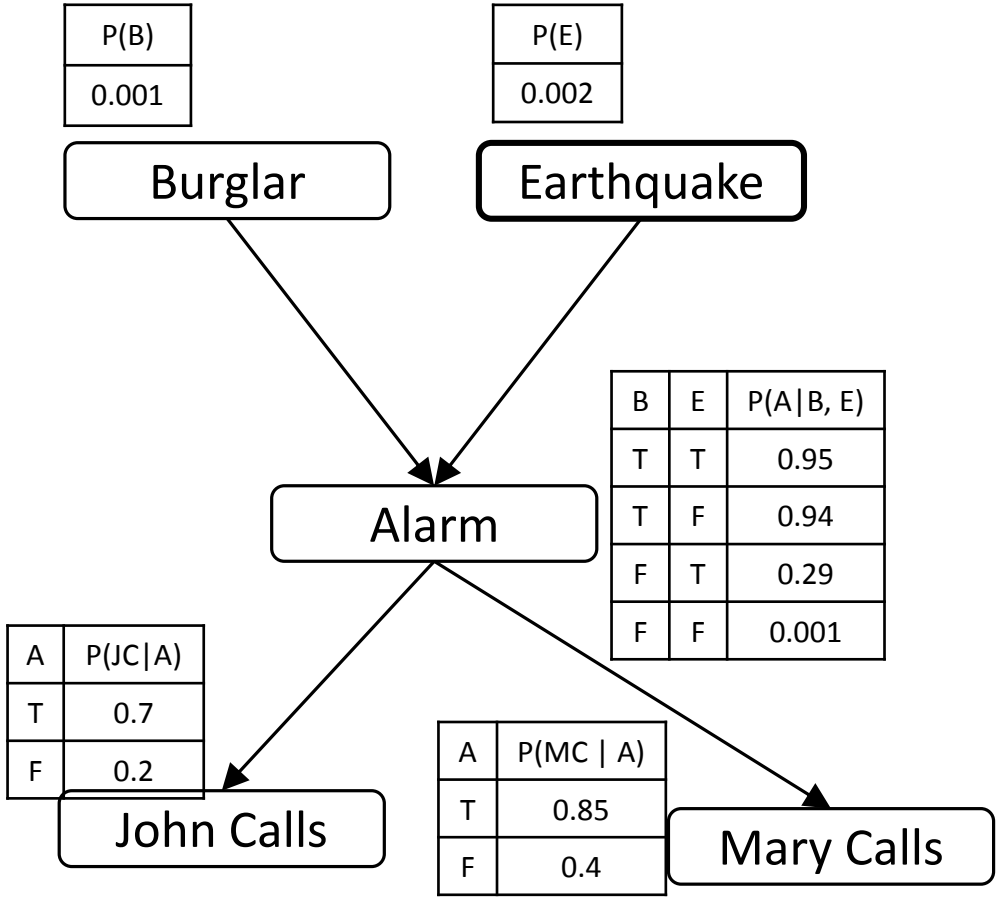
Markov Condition:
Every variable is independent of **its non-descendants** given its **parents**.

Faithfulness Assumption:
Independences stem **only** from the network structure, **not the parameterization** of the distribution.

Some independencies are determined explicitly by the MC, some are entailed using probability theory

All independencies in **J** can be identified in **G** using the graphical criterion of **d-separation.**

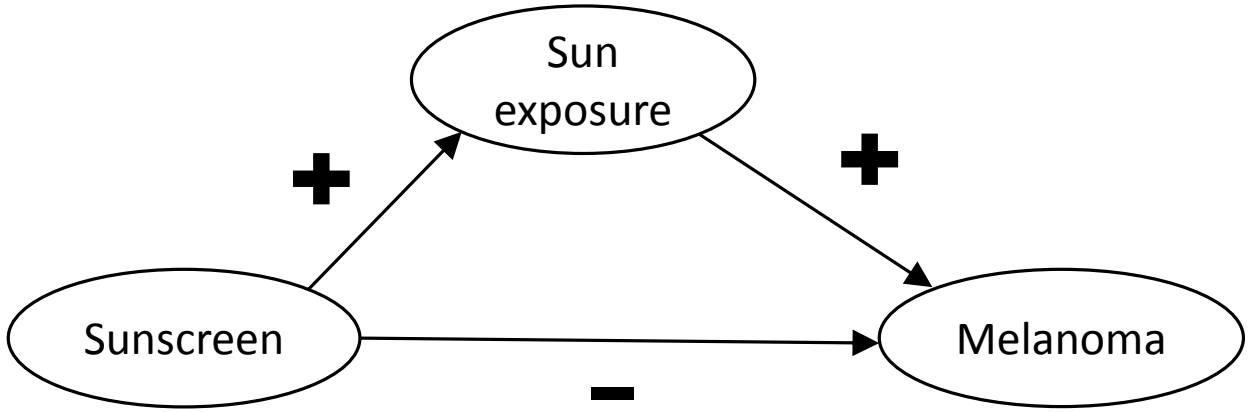
Example



Example by J.Pearl

- You have an installed alarm.
- Burglars set off the alarm.
- Earthquakes set of the alarm.
- When the alarm goes off, one of your neighbors (John or Mary) may call you.

A note on Faithfulness



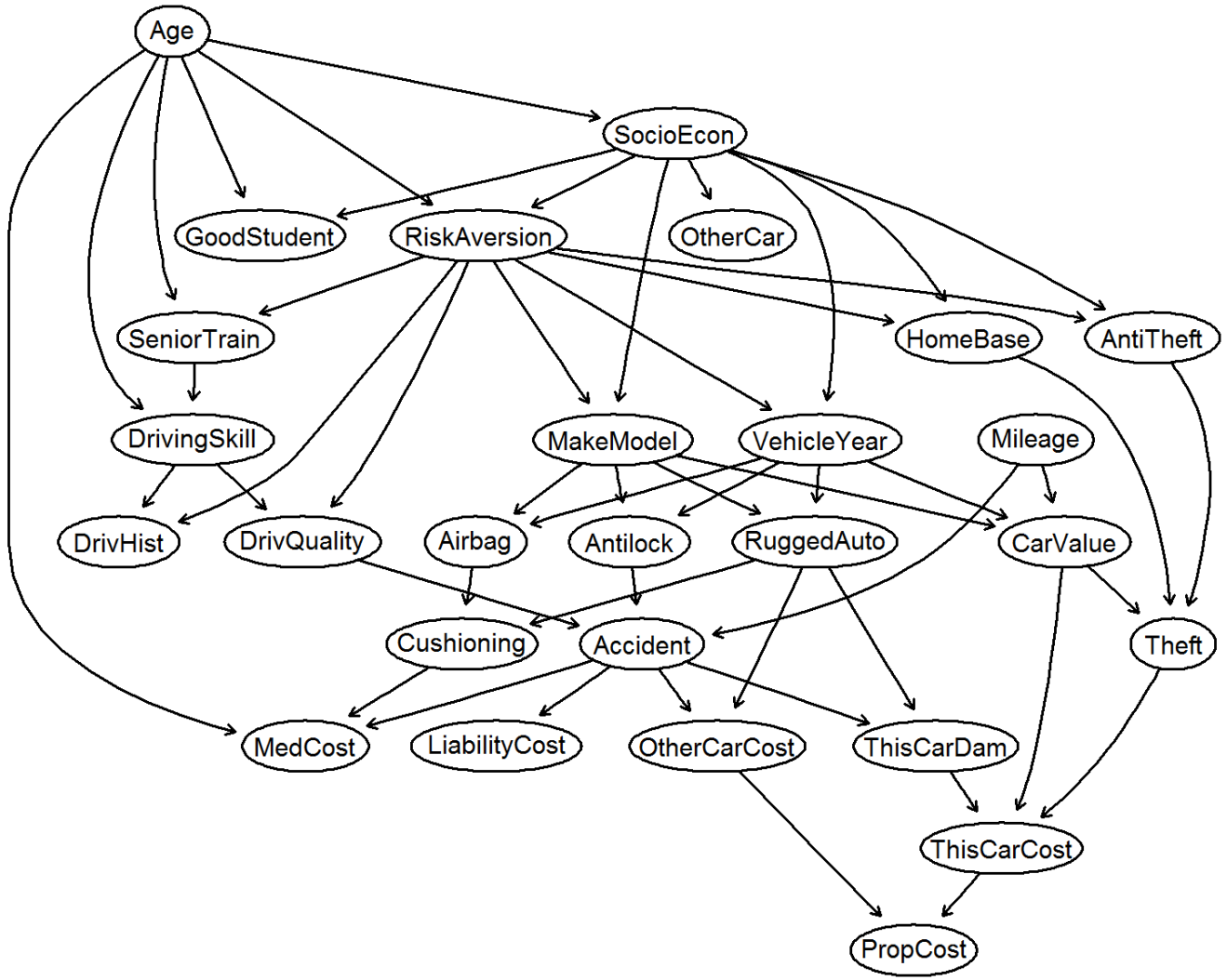
Sunscreen (directly) causally reduces your chances of melanoma
Sunscreen makes people stay longer in the sun, which increases the chances of melanoma

Faithfulness Violation:
The parameters are set so associations cancel each other out!

Causality and the feature selection

What is the Markov Blanket of `MakeModel`?

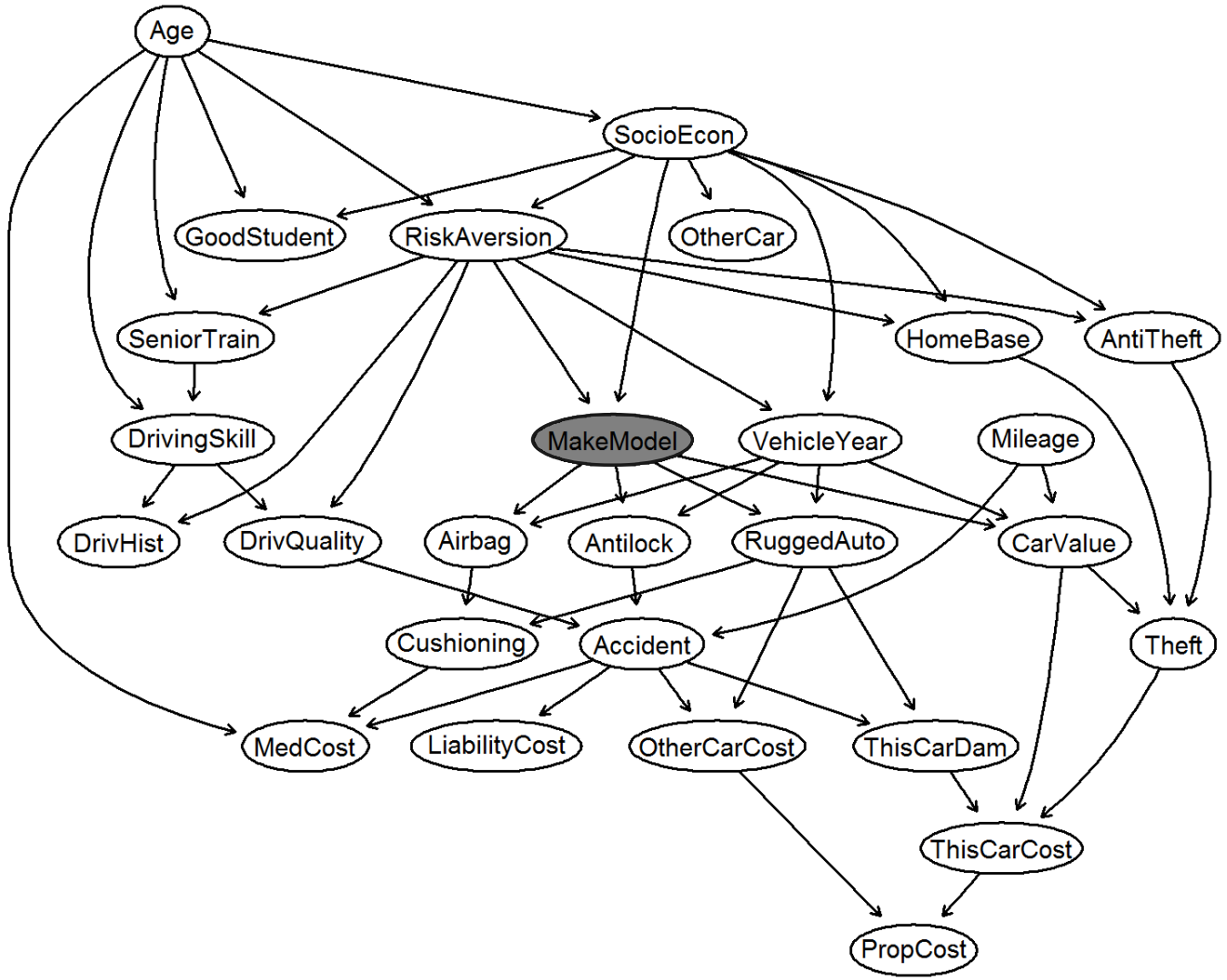
- Spouses = nodes with common children
- Markov Blanket
  - **Neighbors (parents and children)** of ***T***
  - **Spouses** of ***T***



:Target :Neighbors :Spouses

What is the Markov Blanket of `MakeModel`?

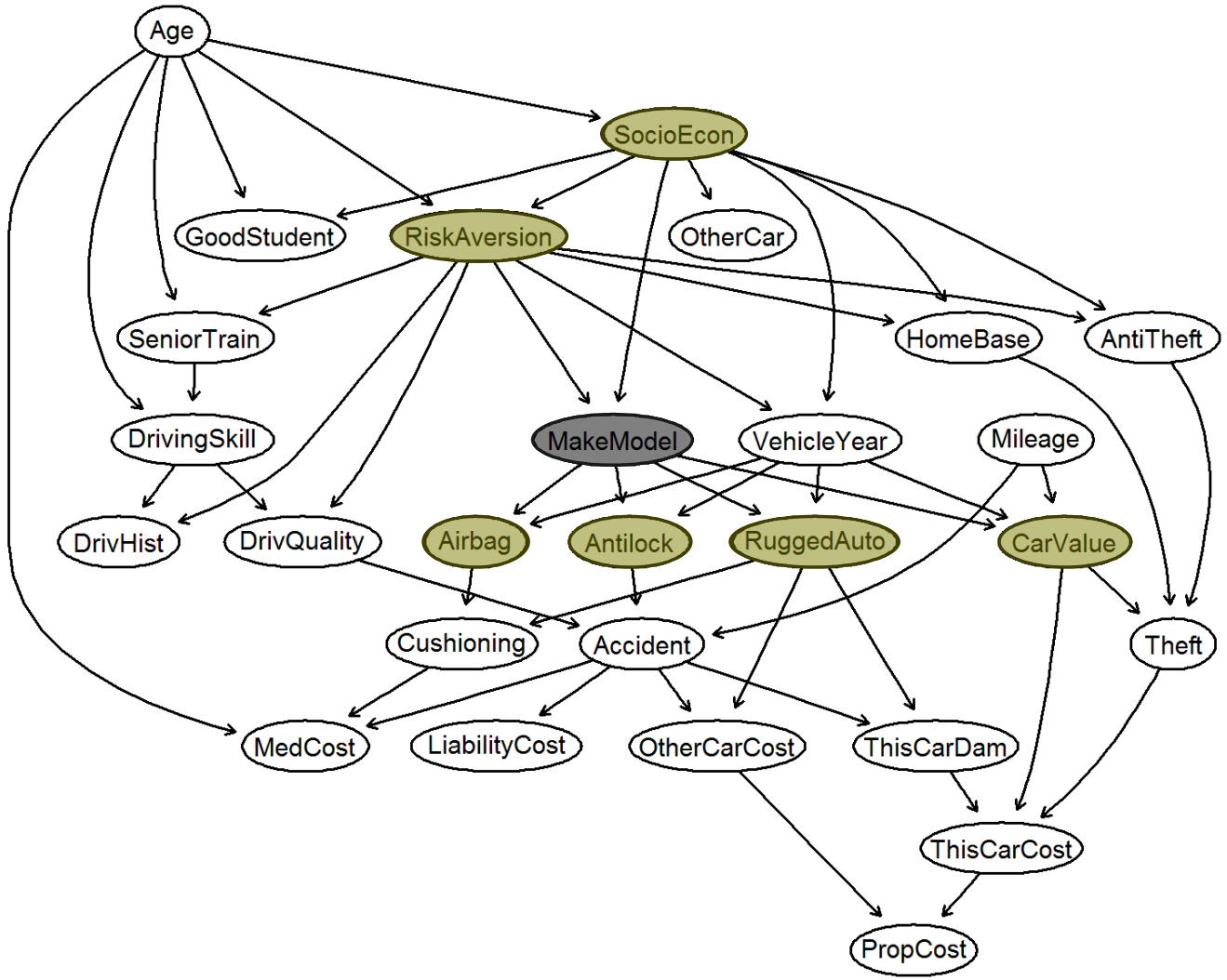
- Spouses = nodes with common children
- Markov Blanket
  - **Neighbors (parents and children)** of ***T***
  - **Spouses** of ***T***



:Target :Neighbors :Spouses

What is the Markov Blanket of `MakeModel`?

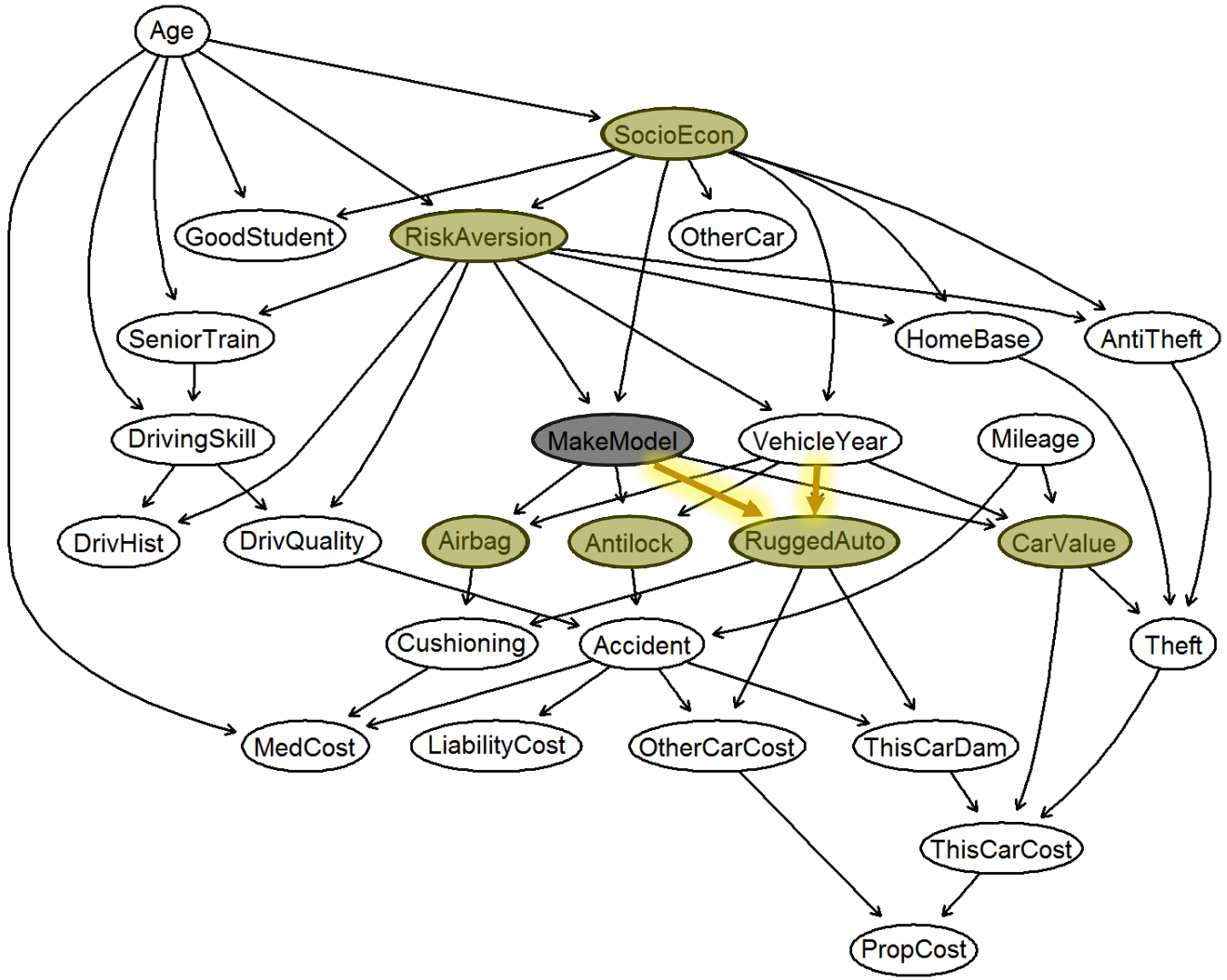
- Spouses = nodes with common children
- Markov Blanket
  - **Neighbors (parents and children)** of ***T***
  - **Spouses** of ***T***



:Target :Neighbors :Spouses

What is the Markov Blanket of `MakeModel`?

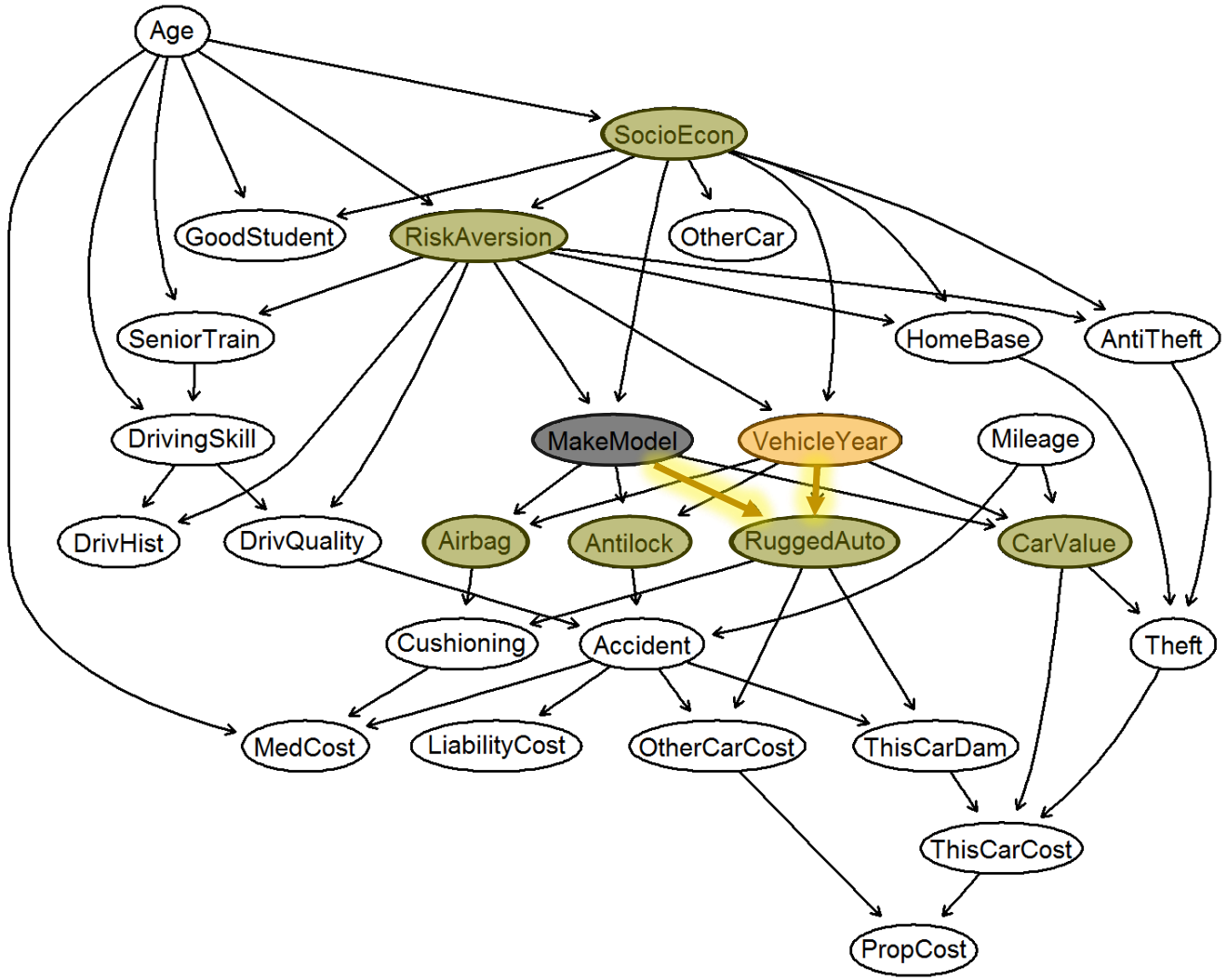
- Spouses = nodes with common children
- Markov Blanket
  - **Neighbors (parents and children)** of ***T***
  - **Spouses** of ***T***



:Target :Neighbors :Spouses

What is the Markov Blanket of `MakeModel`?

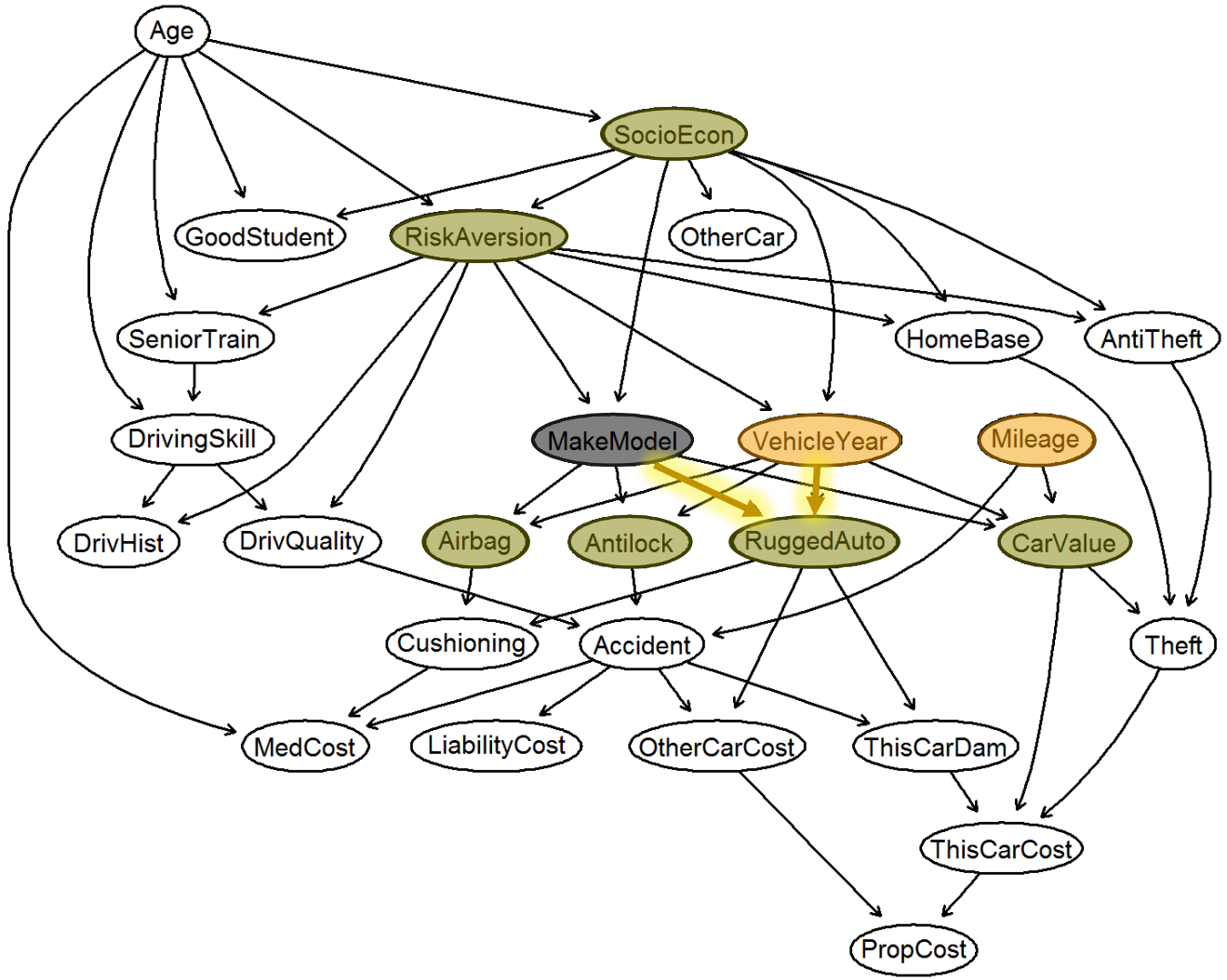
- Spouses = nodes with common children
- Markov Blanket
  - **Neighbors (parents and children)** of ***T***
  - **Spouses** of ***T***



:Target :Neighbors :Spouses

What is the Markov Blanket of `MakeModel`?

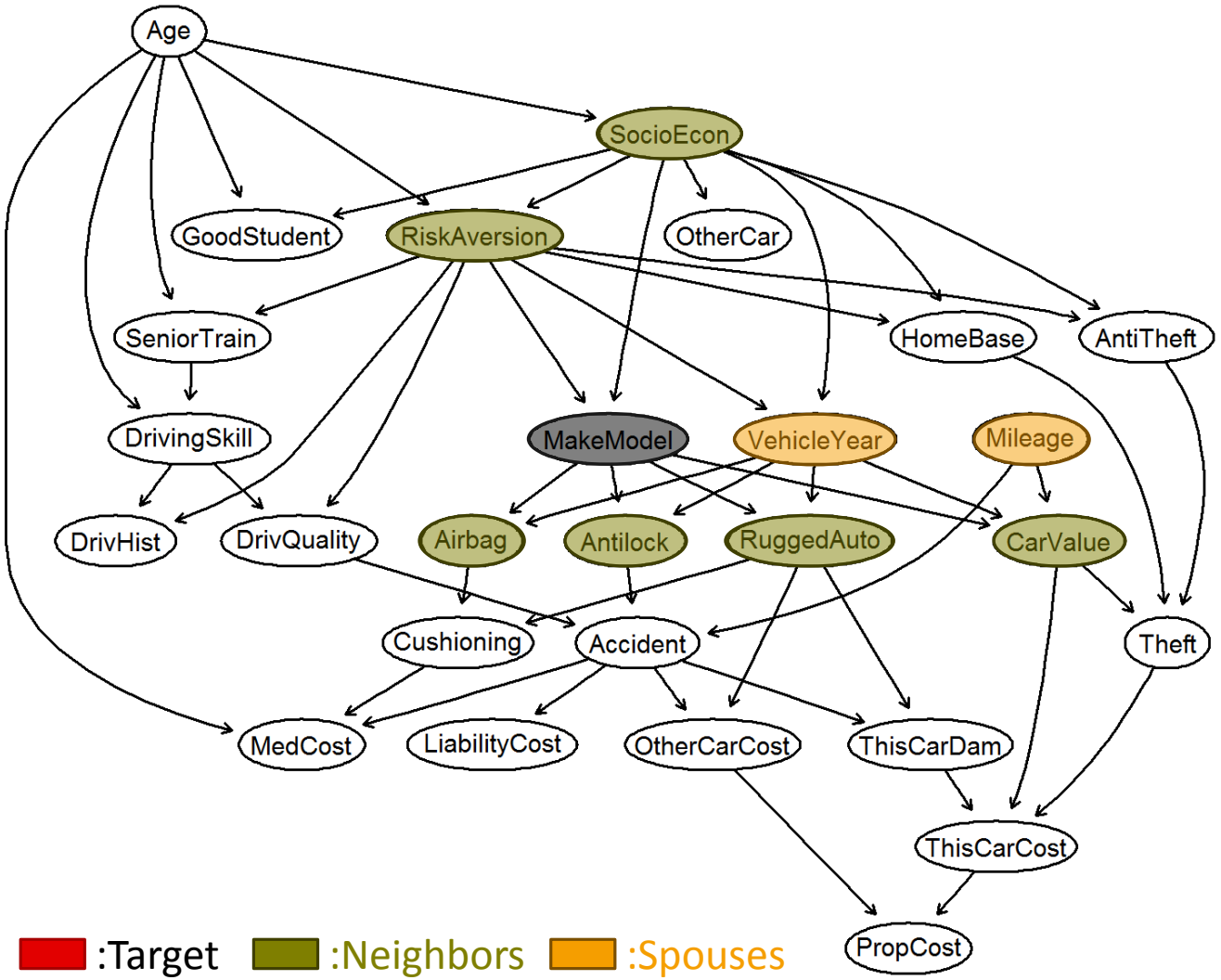
- Spouses = nodes with common children
- Markov Blanket
  - **Neighbors (parents and children)** of ***T***
  - **Spouses** of ***T***



:Target :Neighbors :Spouses

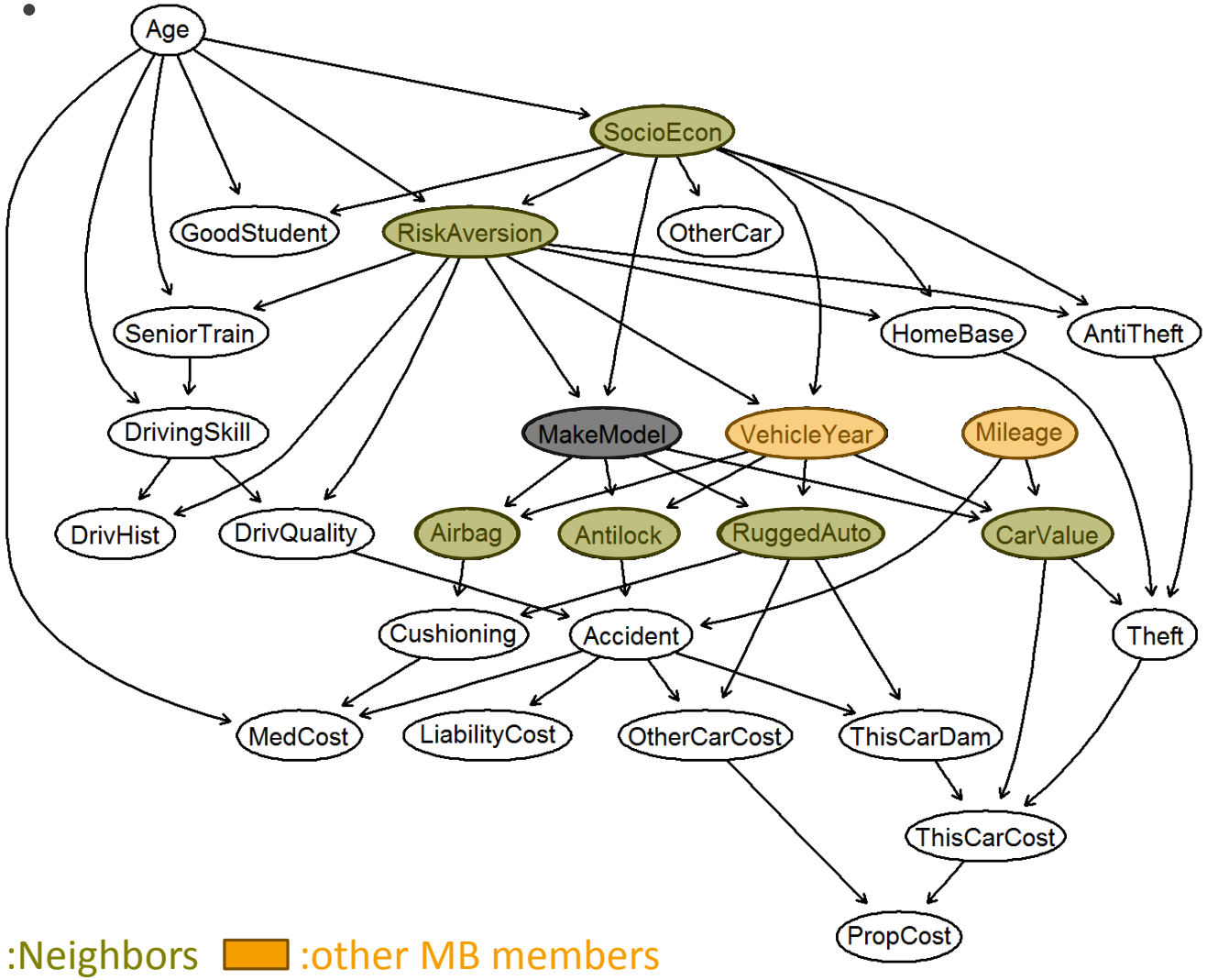
What is the Markov Blanket of `MakeModel`?

- **Markov Blanket:**
  - **Neighbors (parents and children)** of ***T***
  - **Spouses** of ***T***
- Theorem: The Markov Blanket of *T* is **unique** in Faithful distributions
- In distributions faithful to a Bayesian Network
  - **Markov Blanket** = ***indispensable***
  - Non-Markov Blanket features connected with a path to *T* are ***redundant***
  - Features not connected with a path to *T* are ***irrelevant***
  - There are no ***replaceable*** features



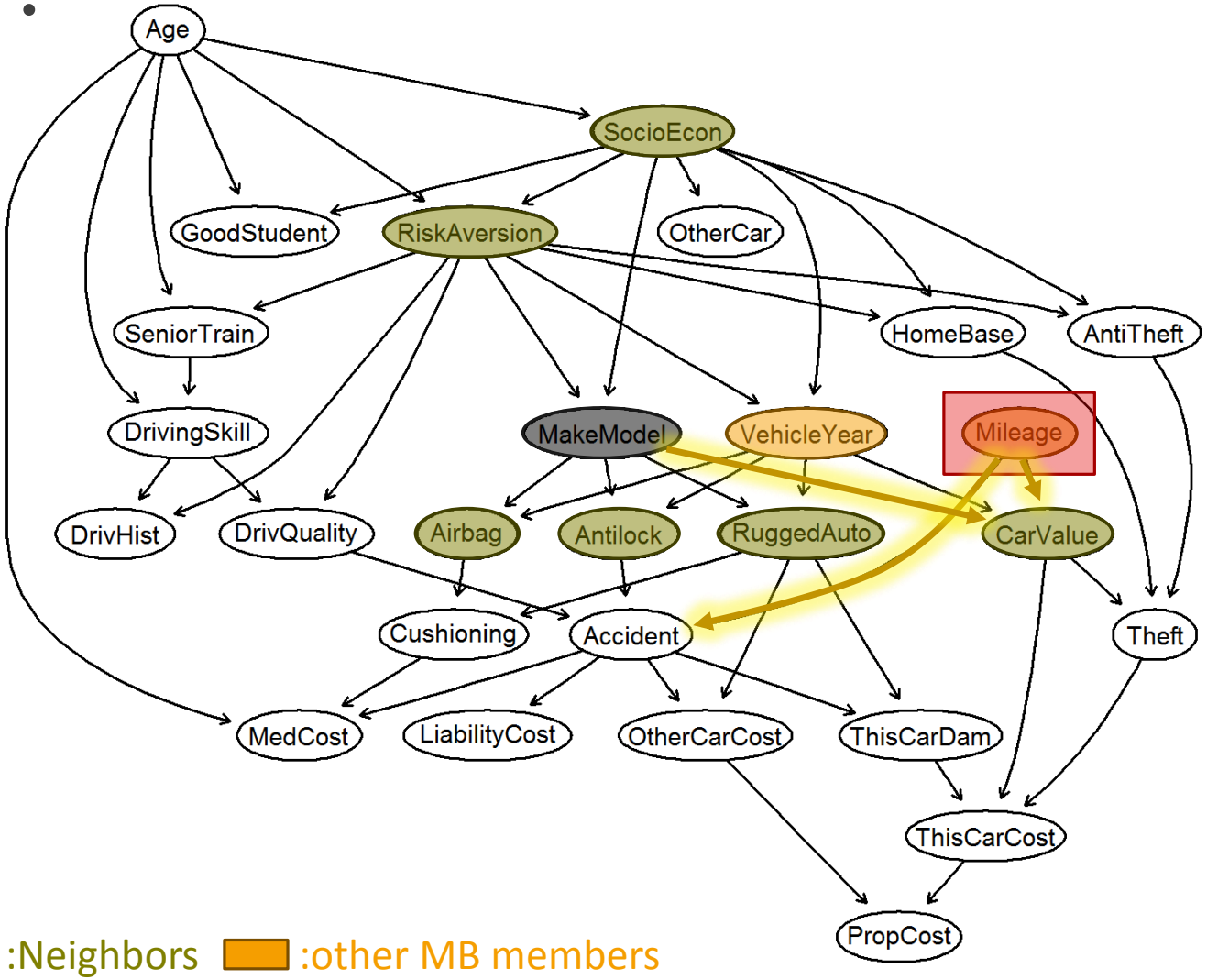
What is the Markov Blanket of `MakeModel` with latent variables?

- Nodes in boxes not measured
- **Collider path** = path where all intermediate, observed nodes (if any) are colliders
- **Markov Blanket**
  - All nodes connected to *T* with a collider path



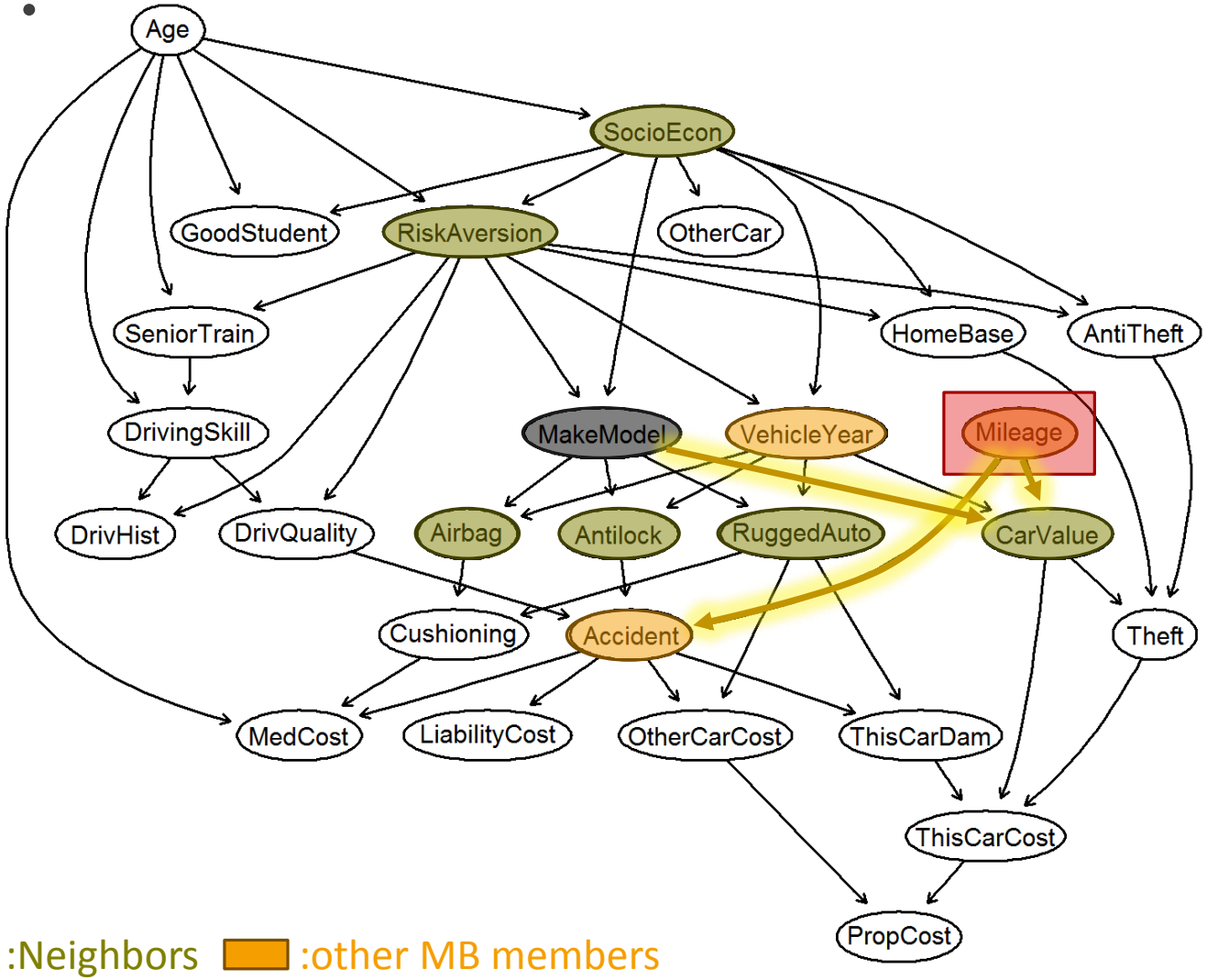
What is the Markov Blanket of `MakeModel` with latent variables?

- Nodes in boxes not measured
- **Collider path** = path where all intermediate, observed nodes (if any) are colliders
- **Markov Blanket**
  - All nodes connected to *T* with a collider path



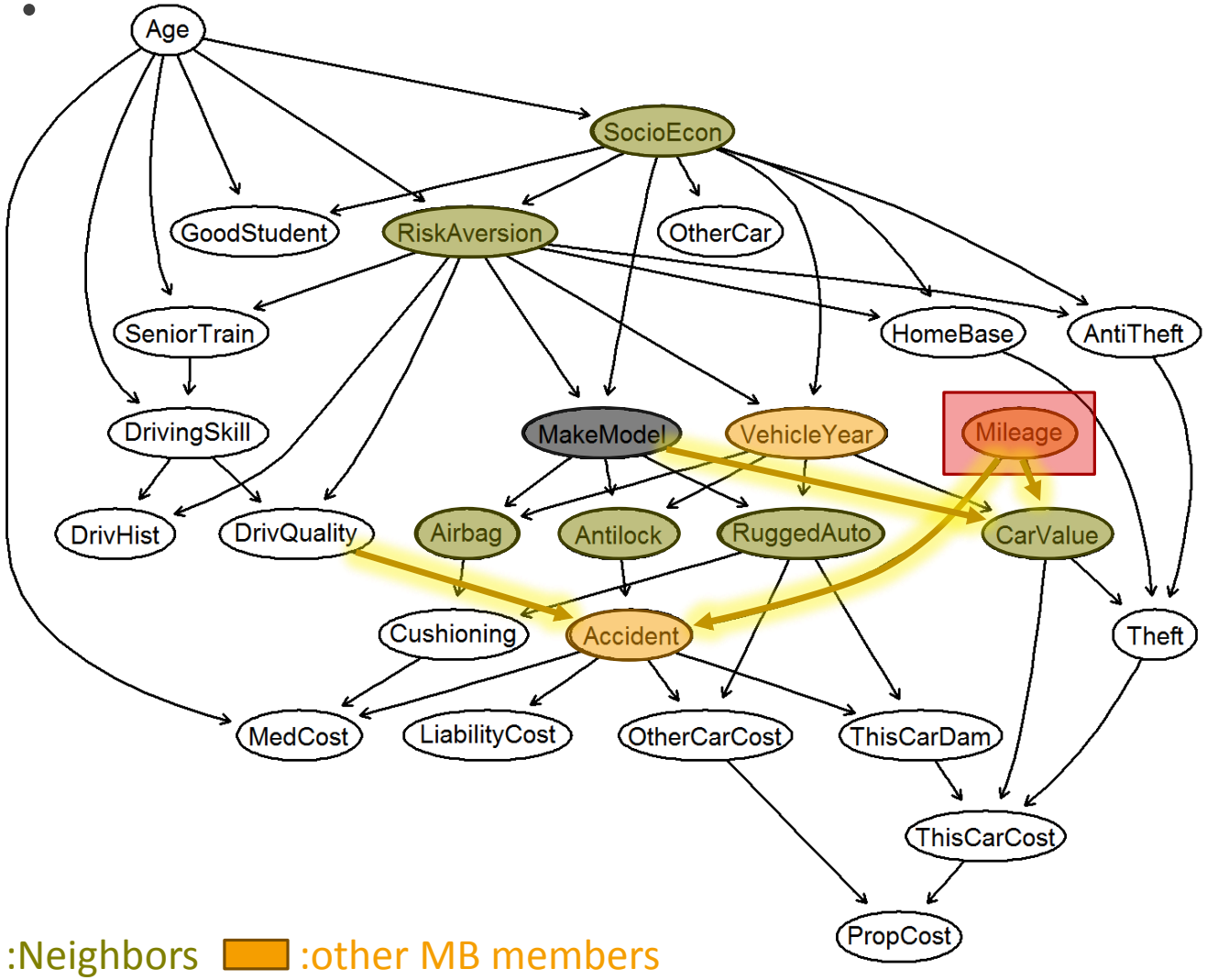
What is the Markov Blanket of `MakeModel` with latent variables?

- Nodes in boxes not measured
- **Collider path** = path where all intermediate, observed nodes (if any) are colliders
- **Markov Blanket**
  - All nodes connected to *T* with a collider path



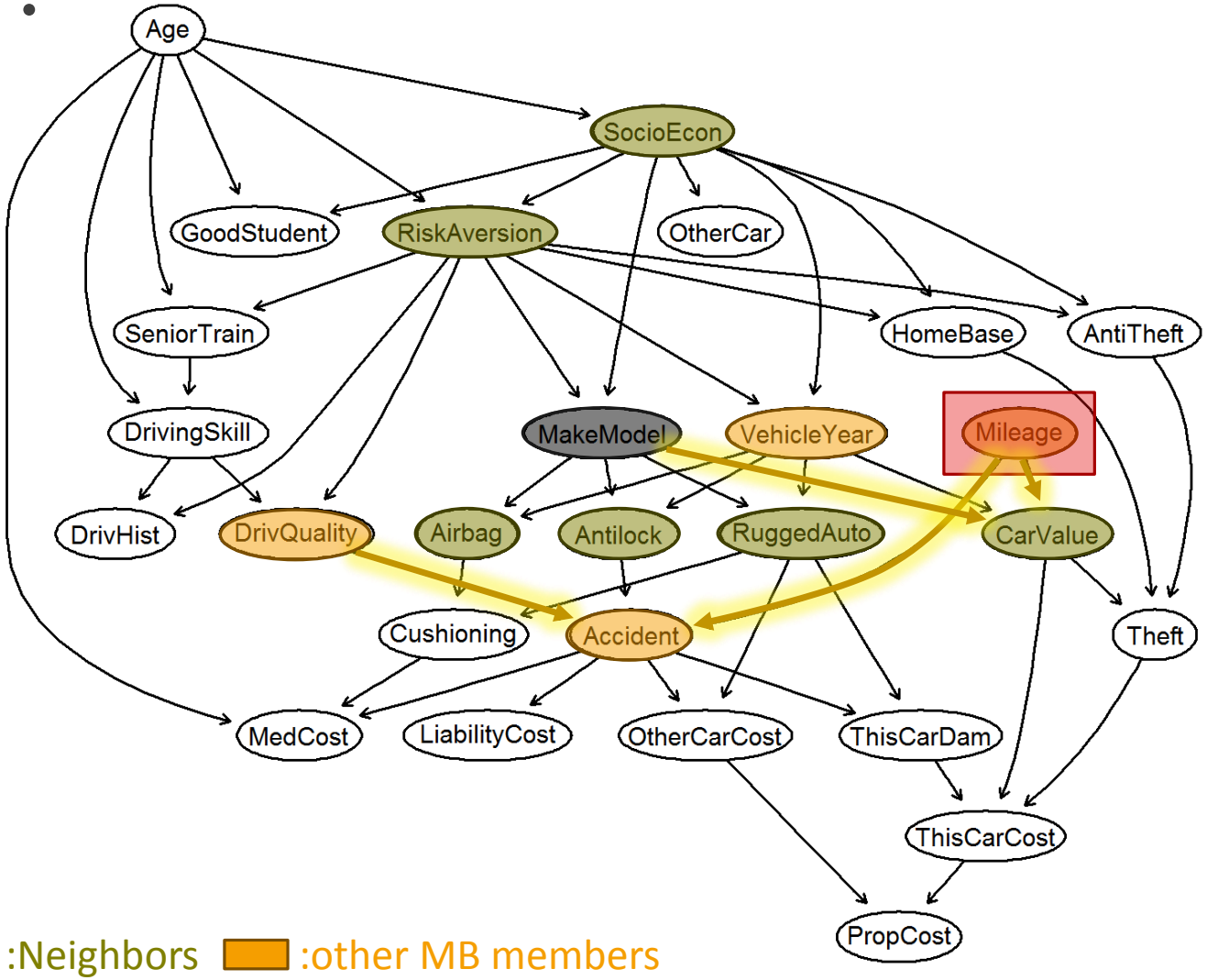
What is the Markov Blanket of `MakeModel` with latent variables?

- Nodes in boxes not measured
- **Collider path** = path where all intermediate, observed nodes (if any) are colliders
- **Markov Blanket**
  - All nodes connected to *T* with a collider path



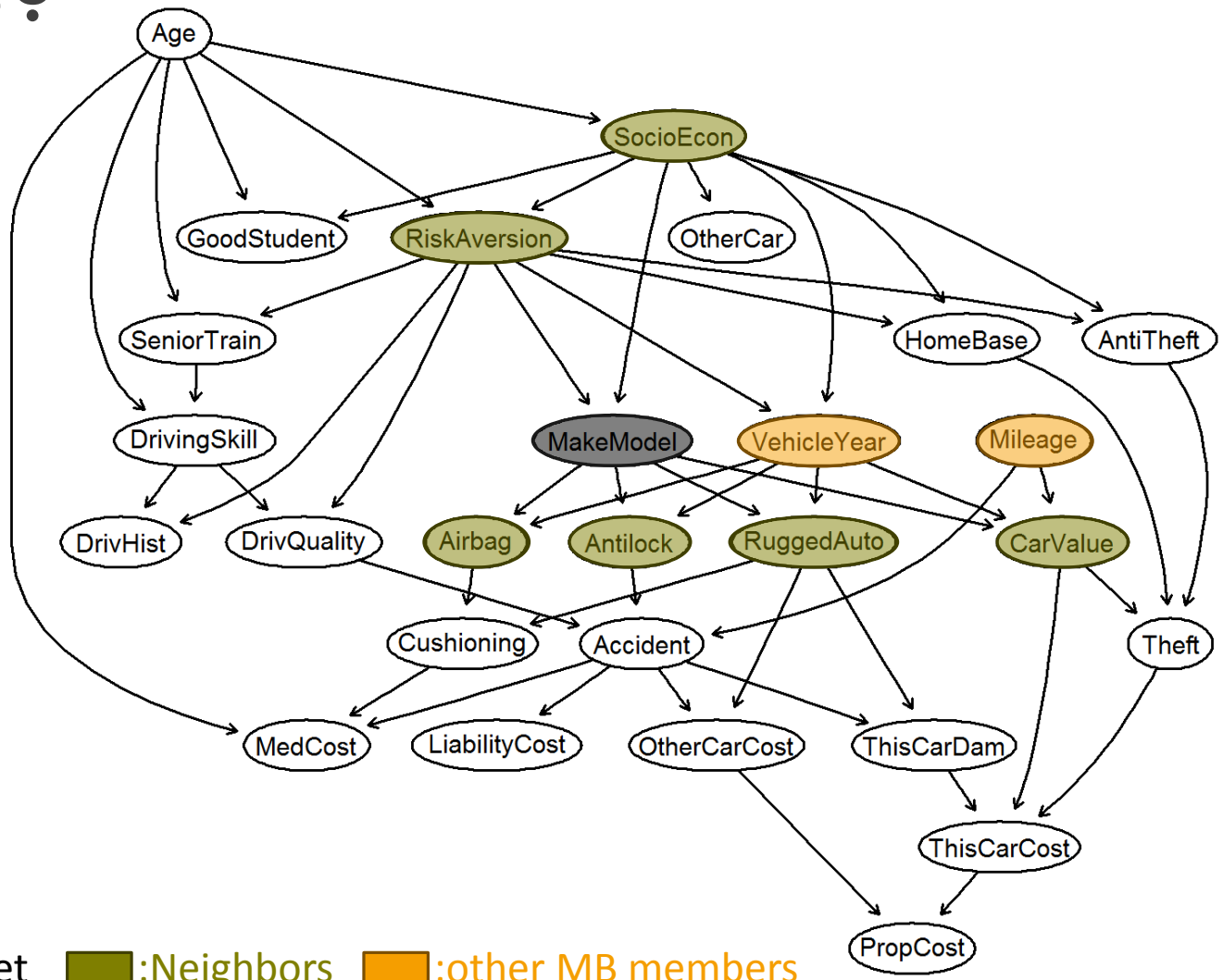
What is the Markov Blanket of `MakeModel` with latent variables?

- Nodes in boxes not measured
- **Collider path** = path where all intermediate, observed nodes (if any) are colliders
- **Markov Blanket**
  - All nodes connected to *T* with a collider path



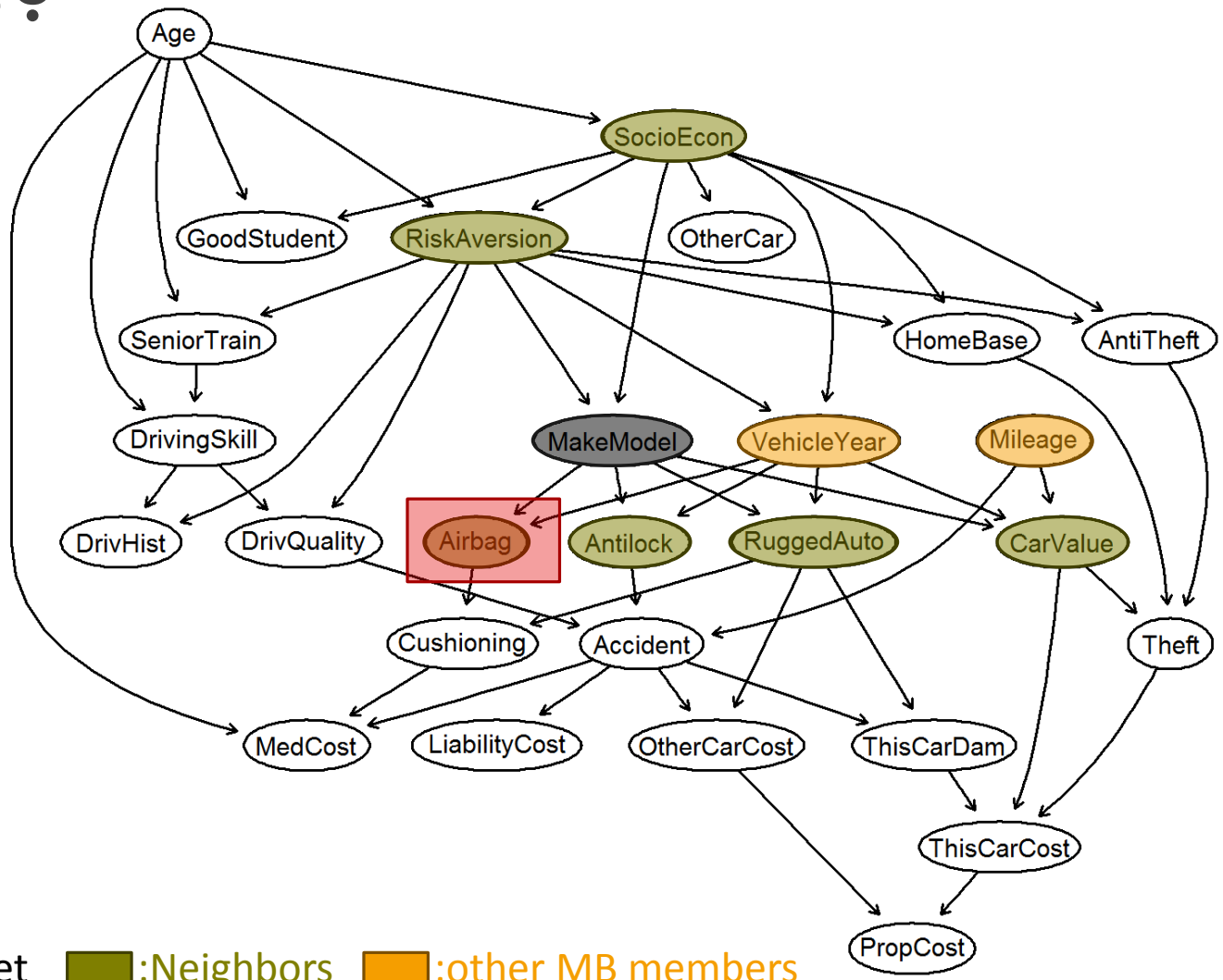
What is the Markov Blanket of `MakeModel` with latent variables?

- Nodes in boxes not measured
- **Subtleties**: paths are to be calculated on the marginal of the Bayesian Network (called a Maximal Ancestral Graph)
- Cushioning becomes a Child of *T* in the marginal network
  - Needs theory of marginal of Bayesian Networks: Maximal Ancestral Graphs
- **Markov Blanket**
  - All nodes connected to *T* with a collider path



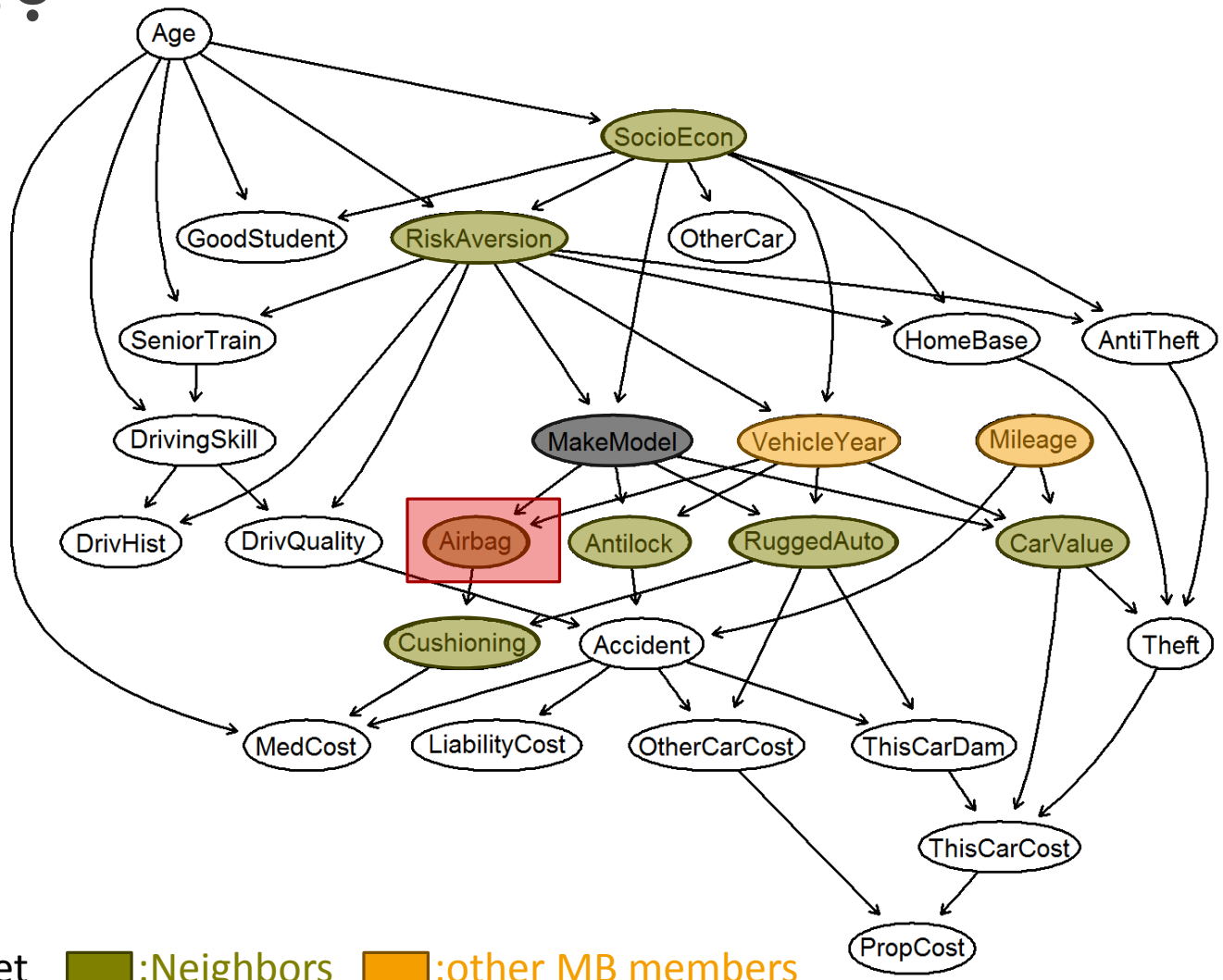
What is the Markov Blanket of `MakeModel` with latent variables?

- Nodes in boxes not measured
- **Subtleties**: paths are to be calculated on the marginal of the Bayesian Network (called a Maximal Ancestral Graph)
- Cushioning becomes a Child of *T* in the marginal network
  - Needs theory of marginal of Bayesian Networks: Maximal Ancestral Graphs
- **Markov Blanket**
  - All nodes connected to *T* with a collider path



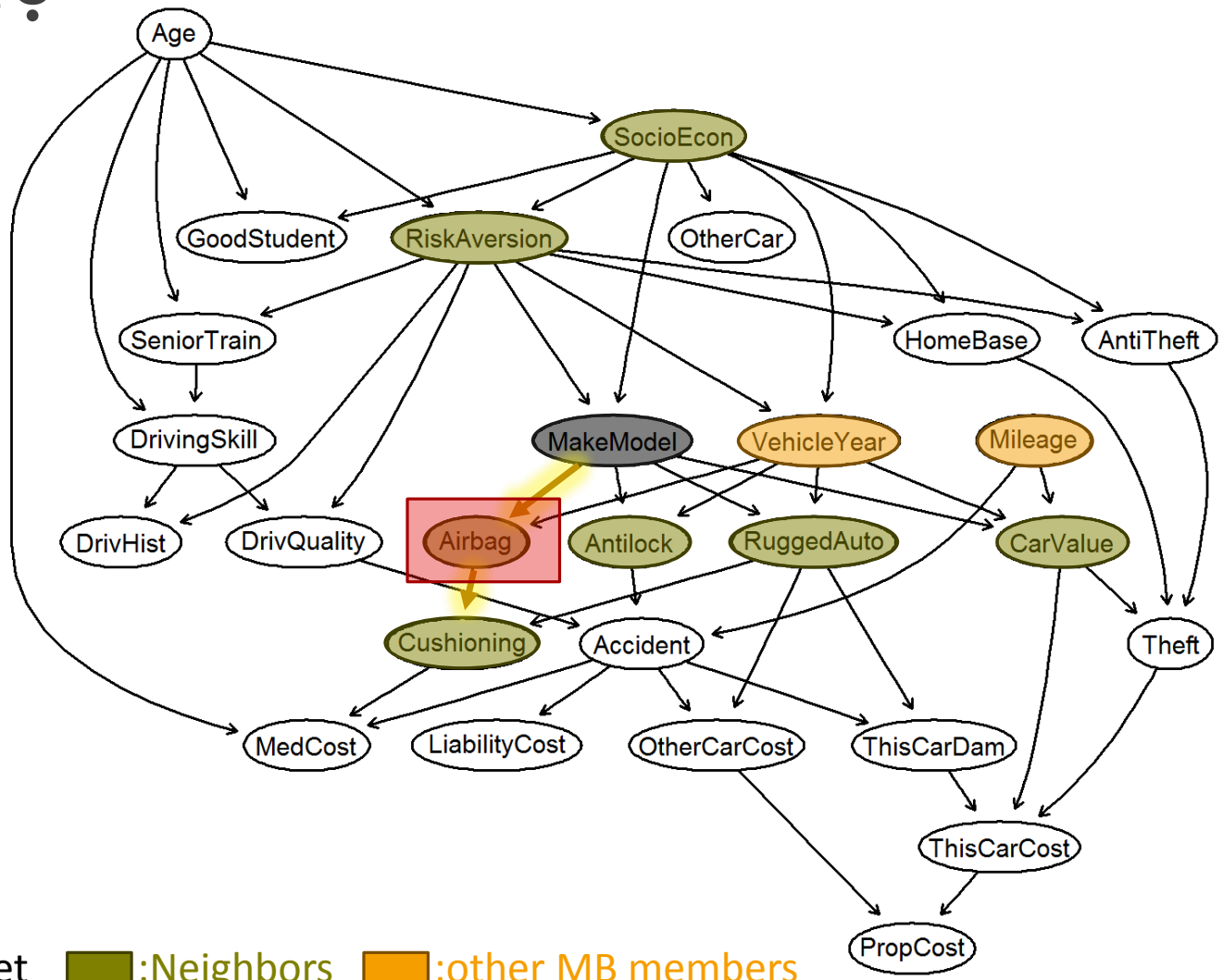
What is the Markov Blanket of `MakeModel` with latent variables?

- Nodes in boxes not measured
- **Subtleties**: paths are to be calculated on the marginal of the Bayesian Network (called a Maximal Ancestral Graph)
- Cushioning becomes a Child of *T* in the marginal network
  - Needs theory of marginal of Bayesian Networks: Maximal Ancestral Graphs
- **Markov Blanket**
  - All nodes connected to *T* with a collider path



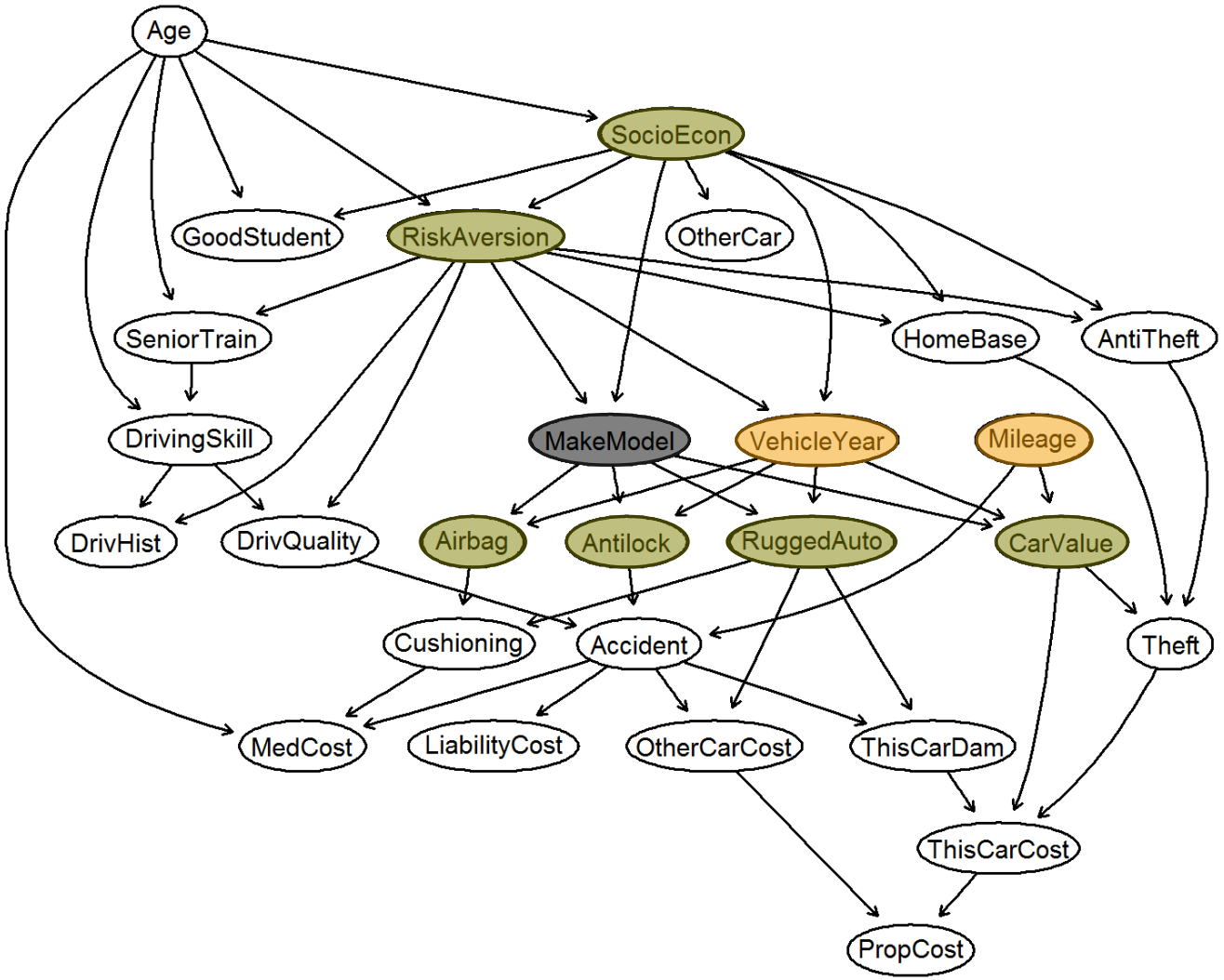
What is the Markov Blanket of `MakeModel` with latent variables?

- Nodes in boxes not measured
- **Subtleties**: paths are to be calculated on the marginal of the Bayesian Network (called a Maximal Ancestral Graph)
- Cushioning becomes a Child of *T* in the marginal network
  - Needs theory of marginal of Bayesian Networks: Maximal Ancestral Graphs
- **Markov Blanket**
  - All nodes connected to *T* with a collider path



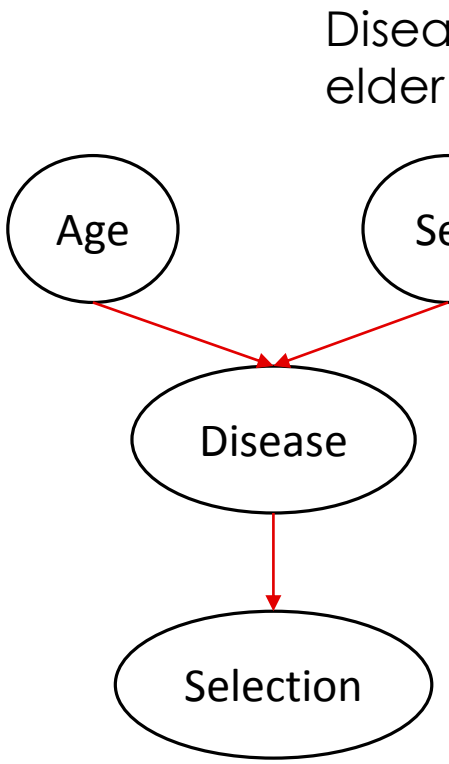
Feature Selection Intrinsically Related to Causality!

- Causalities determine the solution to the **feature selection** problem
- Explains theoretically **why** Feature Selection is used for Knowledge Discovery
- Feature selection becomes a **causal discovery problem**



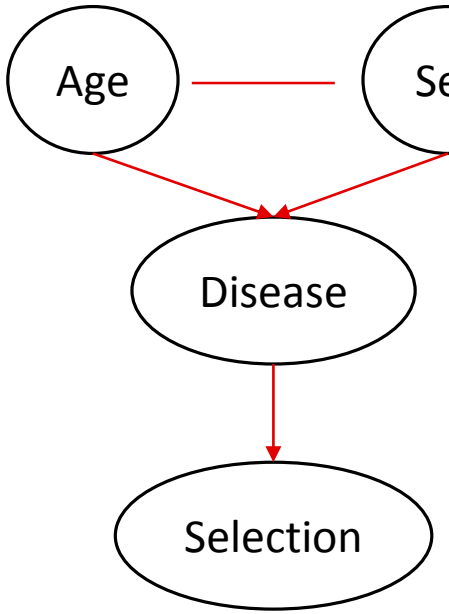
Selection Bias

- Extensions to the theory required under selection bias
- **Markov Blanket** needs to consider selection bias
- Case-Control studies have **selection bias by design**
- Selection bias modeled with additional node
- Selected **distribution** is P(data | Selection = 1)



Selection Bias

- Extensions to the theory required under selection bias
- **Markov Blanket** needs to consider selection bias
- Case-Control studies have **selection bias by design**
- Selection bias modeled with additional node
- Selected **distribution** is P(data | Selection = 1)



Spurious association (not in the general population) due to selection process

Selection = 1 for all samples
P(S=1 | Disease = True) = high
P(S=1 | Disease = False) = low

Causal Models in Biology

- **Pros**: identify the connections between feature selection and causality
- **Pros**: inspire us to design feature selection algorithms
- **Cons**: several other subtle assumptions to discover causality (see Geris, L. and Gomez-Cabrero, D. (2016). *Uncertainty in Biology*. Springer International Publishing, Chapter 3 for more)
- **Cons**: Bayesian Networks do not consider feedback cycles, selection bias, latent variables.
- **Pros**: However, major recent advances in causality remove assumptions
- More Material:
  - Tsamardinos KDD talk [http://videolectures.net/kdd2017_tsamardinos_feature_selection/]
  - MXM **R** Package with numerous algorithms for all types of data

Stability and Replaceability

Replaceable Features and Knowledge Discovery

- Suppose genes {X, Y, Z} are a Markov Blanket of *T*
- Suppose genes {A, B, C, D} are also a Markov Blanket of *T*
  - both **minimal** and **optimally** predictive
  - {X, Y, Z} **replaceable** by {A, B, C, D}
- It is **misleading** to report to biologists "all you need to predict *T* is X, Y, Z, forget the rest
- **Report all Markov Blankets**
- Need Feature Selection algorithms that identify all solutions

Stability of Selection

- Even measuring stability is tricky and hard [Nogueira, L., Sechidis, K. and Brown, G.(2018) *On the Stability of Feature Selection Algorithms*, JMLR 18(174):1−54, 2018.]
- Replaceable features cause instability of selection, even asymptotically
- Example: *X* and *Z* exact copies of each other and belong to a Markov Blanket of *T*
  - For algorithms that guarantee a Markov Blanket (minimality of selection): During Cross-Validation either *X* or *Z* will be selected, but not both
  - **Point**: **Can't throw away features that are selected "unstably"**
  - Other algorithms: *X* and *Z* both selected, but importance weight split between the two
    - Lasso coefficients, Random Forest importance score
  - **Point**: **Can't throw away features with low importance**

More on Replaceable Features

- Replaceable features do not necessarily strongly correlate
  - Some algorithms try to cluster together strongly correlated features

    [Grace, T.H., Tsamardinos, I., Raghu, V., Kaminski, N. and Benos V.P. (2015)T-RECS: Stable Selection of Dynamically Formed Groups of Features with Application to Prediction of Clinical Outcomes, Pac Symp Biocomput. 2015;20:431-442]

    [Klasen, J., Barbez, E., Meier, L., Meinshausen, N., Bühlmann, P., Koornneef, M., Busch, W. and Schneeberger, K. (2016). A multi-marker association method for genome-wide association studies without the need for population structure correction. Nature Communications, 7, p.13299.]
- $T = C$
- $X = C + D_1 + \varepsilon_1$
- $Z = C + D_2 + \varepsilon_2$
- Noise terms normally distributed and independent
- X and Z replaceable for T provided that D_i and ε_i have the same variance
- X and Z share a common predictive component for T and a distinct component.
- Correlation between X and Z can ranges within (0, 1].

Addressing Replaceability

- Use algorithms that return all solutions
  - **TIE*** [Statnikonv, A. and Lytkin I. N. (2014). *Algorithms for Discovery of Multiple Markov Boundaries*, JMLR. 2013 Feb; 14: 499–566.]
  - **Lasso for multiple solutions** [Pantazis, Y., Lagani, V., Charonyktakis, P., Tsamardinos, I. (2018) *Multiple Equivalent Solutions for the Lasso*. arXiv: 1710.04995]
  - **Statistically Equivalent Signatures** [Lagani, V., Athineou, G., Farcomeni, A., Tsagris, M. and Tsamardinos, I. (2017) *Feature Selection with the R Package MXM: Discovering Statistically Equivalent Feature Subsets*. Journal of statistical software]
  - **Our new upcoming algorithm, stay tuned**
- Importance (added value) of a feature
  - Assessed in a Markov Blanket (minimality imposed)
  - Build a model with and without the variable (individual contribution in the context of all other selected variables)

Applying Feature Selection

My advice to Practitioners (I)

- Good choice to try for most cases
  - Lasso [Tibshirani, *Journal of the Royal Statistical Society. Series B Vol. 58, No. 1* 1996], typically returns more features, but better performance. Linear.
- Large sample, very few features
  - Use exhaustive search
- Large sample, few features
  - Use exact methods [Bertsimas et al., *Best Subset Selection via a Modern Optimization Lens*, Annals of Statistics, 44, 2016]
- Large sample relative to total number of features:
  - Use Backward Search

My advice to Practitioners (II)

- Large sample relative to the expected size of Markov Blanket
  - Use Forward-Backward with Early Dropping (**FBED**[1]) and 2 runs (Borboudakis & Tsamardinos, 2017, https://arxiv.org/abs/1705.10770)
  - **Generalized versions of Orthogonal Matching Pursuit** [Tsagris, M. (2018) Guide on performing selection with R package, https://cran.r-project.org/web/packages/MXM/vignettes/FS_guide.pdf]
- Small sample, can only condition with enough statistical power on *k* features
  - Use **MMPC**, **SES** [Tsamardinos et al., : ACM SIGKDD, 2003]
- Huge sample, huge dimensionality: our latest algorithm for **Big Data Feature Selection** [Tsamardinos et al. *Massively-Parallel Feature Selection for Big Data*, https://arxiv.org/abs/1708.07178]
- **After feature selection**: Give problem to **power classifiers** (SVMs, Random Forests, Gradient Boosting Trees, GPs)

Reminder

- Feature Selection is part of the pipeline
  - Needs to be **CVed** and **tuned**
- Which features to return: the ones selected by the optimal configuration on all data
- Try numerous algorithms

The MXM R Package

- Efficient implementations of (some) tests and feature selection algorithms
- **Algorithms**: Backward Search, Forward-Backward, FBED, MMPC, MMMB, SES (for multiple solutions)
- Conditional Independence Tests available

Target	Predicting features	Test
Continuous	Continuous	Pearson (robust) Correlation or Spearman
Continuous	Categorical/continuous	Linear (robust) regression or quantile (median) regression
Categorical	Categorical	G^2 test of independence
Proportions (between 0 and 1)	Categorical/continuous	Beta regression or linear (robust) regression or quantile (median) regression
Counts	Categorical/continuous	Poisson or Negative binomial regression
Zero inflated counts	Categorical/continuous	Zero inflated Poisson regression
Survival	Categorical/continuous	Cox, Weibull or exponential regression
Binary	Categorical/continuous	Logistic regression
Nominal	Categorical/continuous	Multinomial regression
Ordinal	Categorical/continuous	Ordinal regression
Clustered continuous, binary or counts	Continuous	Mixed models
Case-control	Categorical/continuous	Conditional logistic regression

https://CRAN.R-project.org/package=MXM

Summary

- Feature selection is a major primary task
- Features are partitioned to indispensable, replaceable, redundant, and irrelevant
- A Markov Blanket is a minimal-size, optimally predictive set; the solution to the feature selection problem
- Typically, there exist (or are statistically equivalent) multiple Markov Blankets!
- Causal modeling connects feature selection and causality
- Don't just throw away features with low importance weight
- Stability should consider the presence of multiple solutions
- Practical advise was provided

Hyper-parameter search strategies

Problem definition

- Identifying the hyper-parameters configuration $\theta^* \in \Theta$ that provides the best performance on $\mathbf{D} = \{\langle \mathbf{x}_i, y_i \rangle\}$

- Main issues:
  - The number of possible configurations $|\Theta|$ is high or infinite
  - Not all hyper-parameter configurations are admissible or meaningful (conditional hyper-parameters)
  - Evaluating a single configuration could be time consuming

Number of hyper-parameters

- Multiple learners, each with their own set of hyper-parameters

- Example I: Weka Software
  - 27 learners (up to 10 hyper-params each)
  - 10 meta-methods
  - 2 ensemble method
  - At least **786** hyper-params in total

- Example II: scikit-learn
  - 15 learners (59 hyper-params in total)
  - 13 feature pre-processor
  - 4 data pre-processor
  - At least **110** hyper-params in total

Conditional hyper-parameters

- Some hyper-params are meaningful only conditionally to the activations of other hyper-parameters
- **Example**: Support Vector Machines (SVMs)
  - Unconditional hyper-parameter: cost factor C
  - Unconditional hyper-parameter: kernel function (e.g., RBF, polynomial)
    - Conditional hyper-param: tuning factor γ (only for RBF kernel)
    - Conditional hyper-param: degree d(only for polynomial kernel)

Evaluating single configurations

- A performance estimation protocol is required, e.g.:
  - hold-out
  - cross validation

- Evaluating a single configuration can take from < 1 sec to hours, days or more (depending by the problem)

How to identify the optimal hyper-parameter configuration θ^*

- Exhaustively evaluating all configurations is not feasible

- A strategy for efficiently search in the space of possible configurations Θ is required

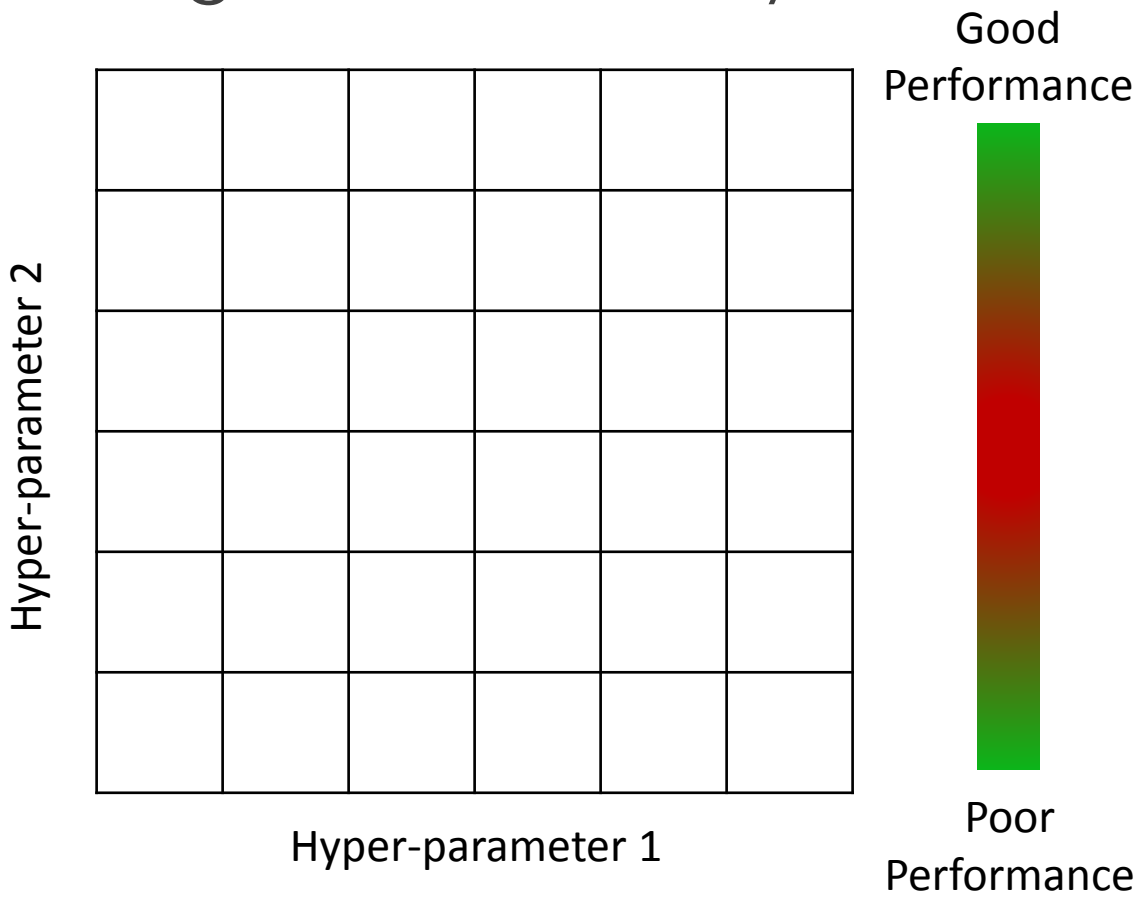
Commonly used hyper-parameter search strategies

- **Grid search**: static
- **Random search**: dynamic, but naïve
- **Optimization methods**:
  - Bayesian optimization, dynamic

Grid search

- Evaluating a fixed number of configurations, usually regularly distributed across Θ

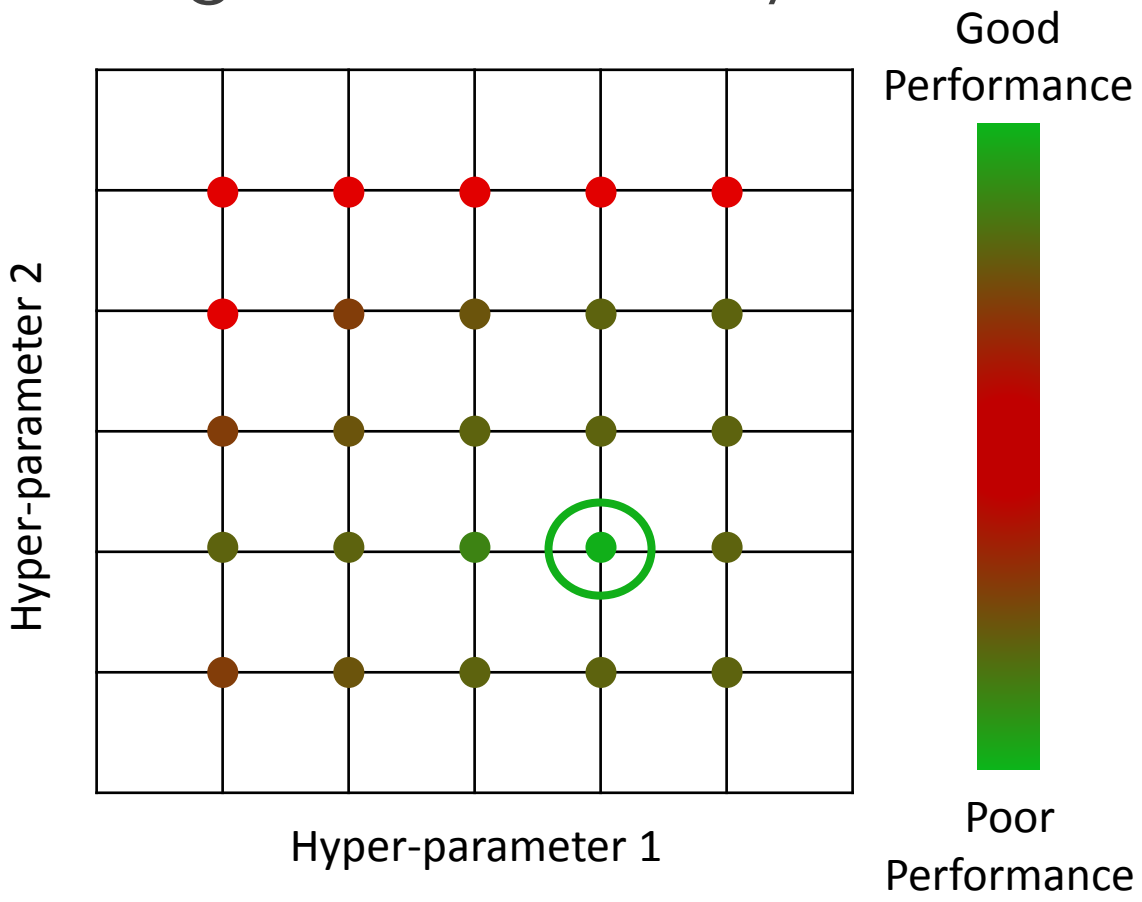
- Simple example:
  - 2 real-value hyper-parameters
  - 5 values to investigate per each hyper-parameter
  - 25 configurations in total



Grid search

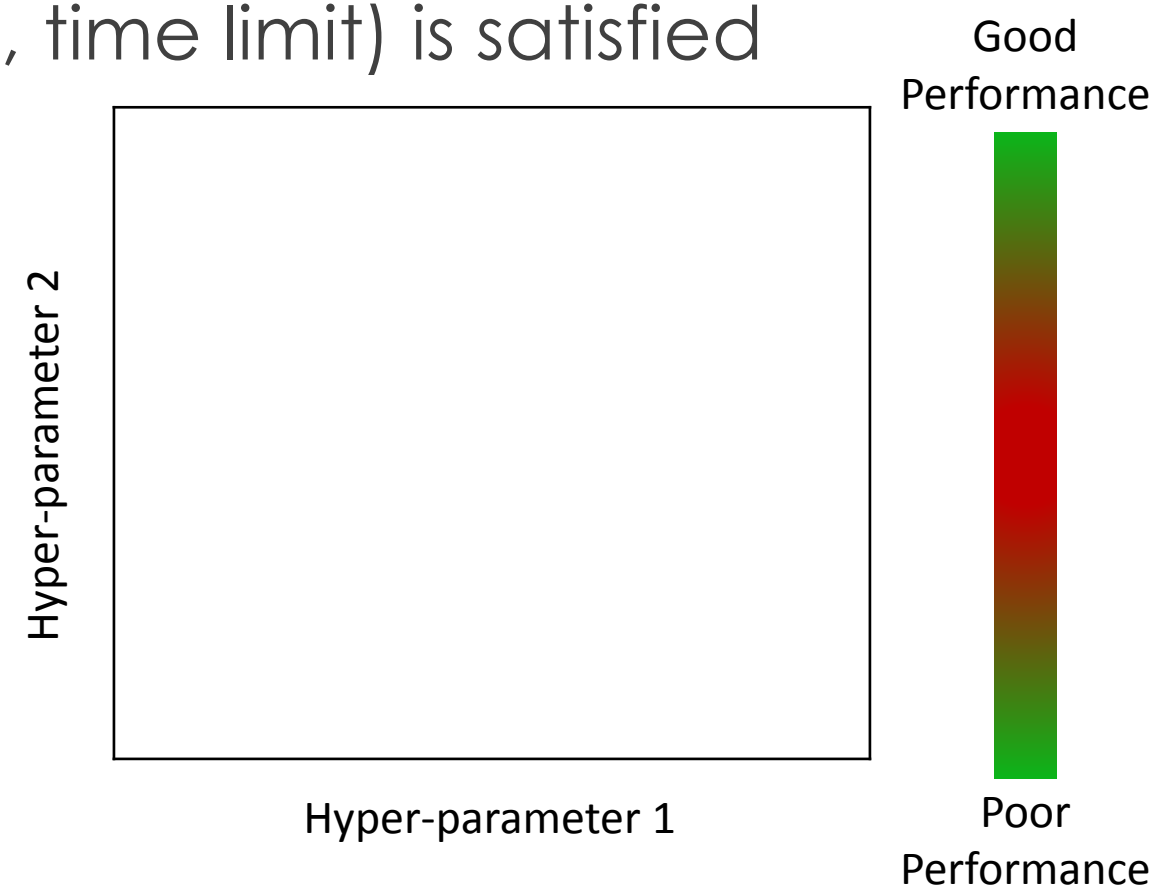
- Evaluating a fixed number of configurations, usually regularly distributed across Θ

- Simple example:
  - 2 real-value hyper-parameters
  - 5 values to investigate per each hyper-parameter
  - 25 configurations in total



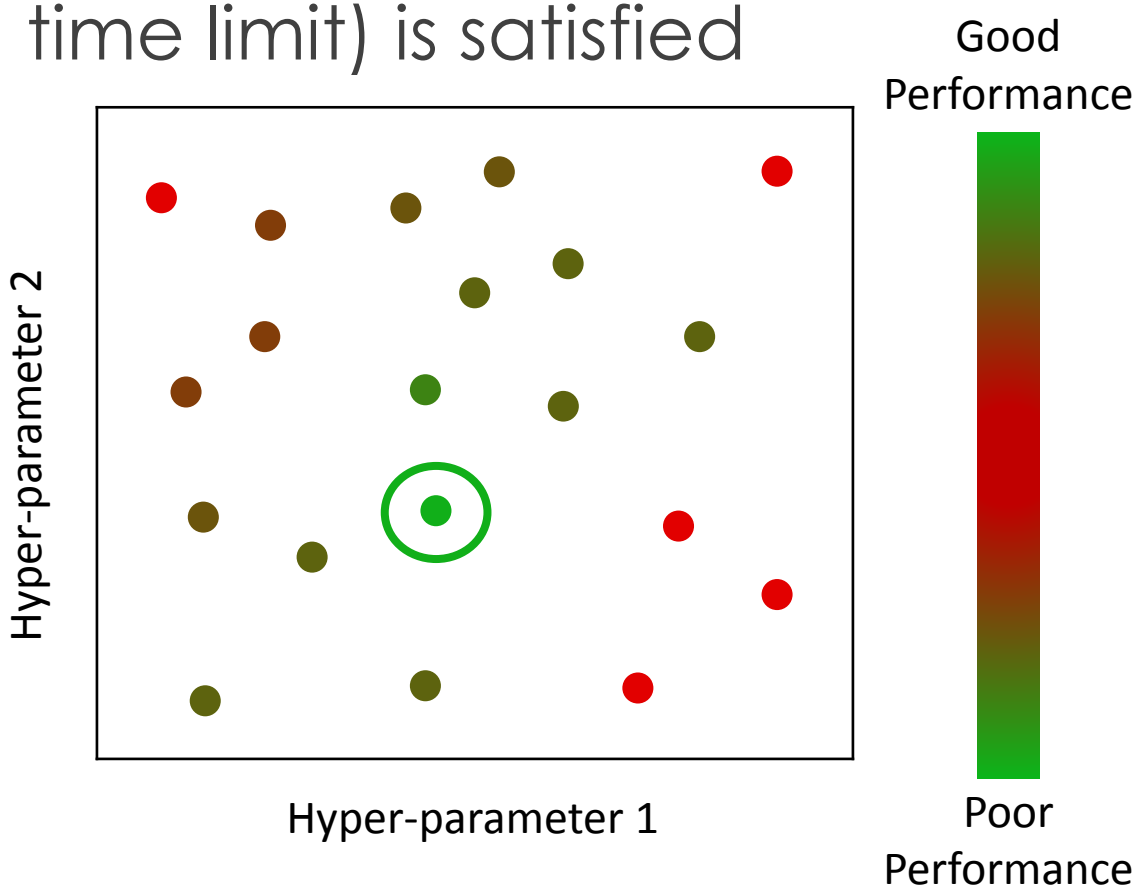
Random search

- Picking configurations at random across Θ, until some criterion (e.g., time limit) is satisfied



Random search

- Picking configurations at random across Θ, until some criterion (e.g., time limit) is satisfied



Optimization methods

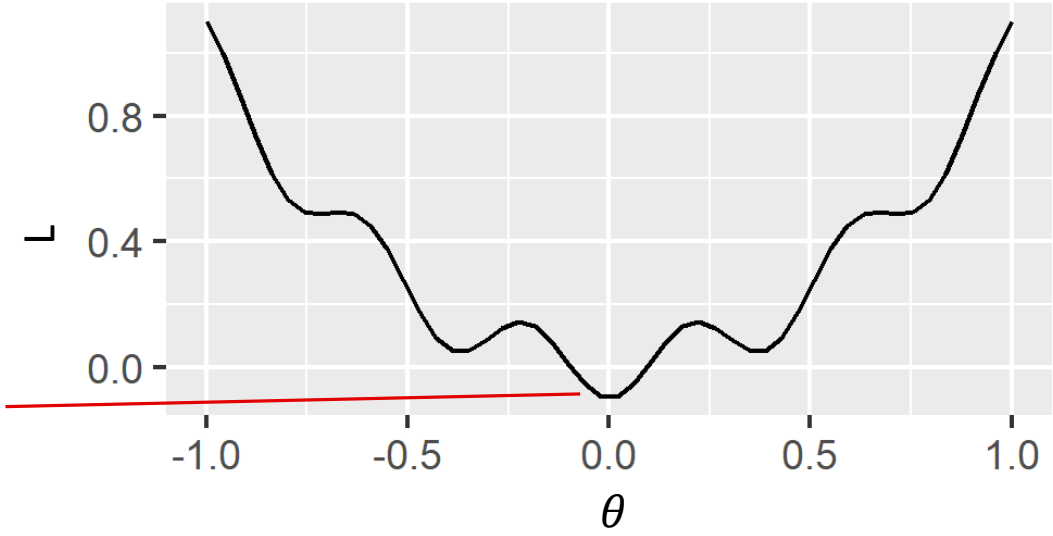
- A whole branch of mathematics / engineering focus on identifying optimal solution(s) θ^* out of a candidate set Θ.
- Optimization methods applied on the problem of hyper-parameter settings include
  - Genetic algorithm [Olson et at., GECCO '16, 2016]
  - Particle swarm optimization [Ye, PLoS ONE 12(12), 2017]
  - Bayesian (global) optimization [Hutter et al., Learning and Intelligent Optimization, pp. 507–523, 2011]
- Major difference from standard optimization: the objective function value has **uncertainty**!

Bayesian optimization

- Bayesian optimization (BO) methods have proven to be particularly effective for hyper-parameter search
- BO algorithm general schema:
  1. Select a configuration θ_i to evaluate
  2. Compute the performance value p_i corresponding to θ_i
  3. Use $\{\langle\theta_1, p_1\rangle, \ \ldots, \langle\theta_i, p_i\rangle\}$ to estimate the function $\Phi: \Theta \rightarrow R$ linking configurations to performance estimates
  4. If some criterion (e.g. time limit) is satisfied, return the best configuration θ^*; otherwise go back to 1.

BO intuitive example

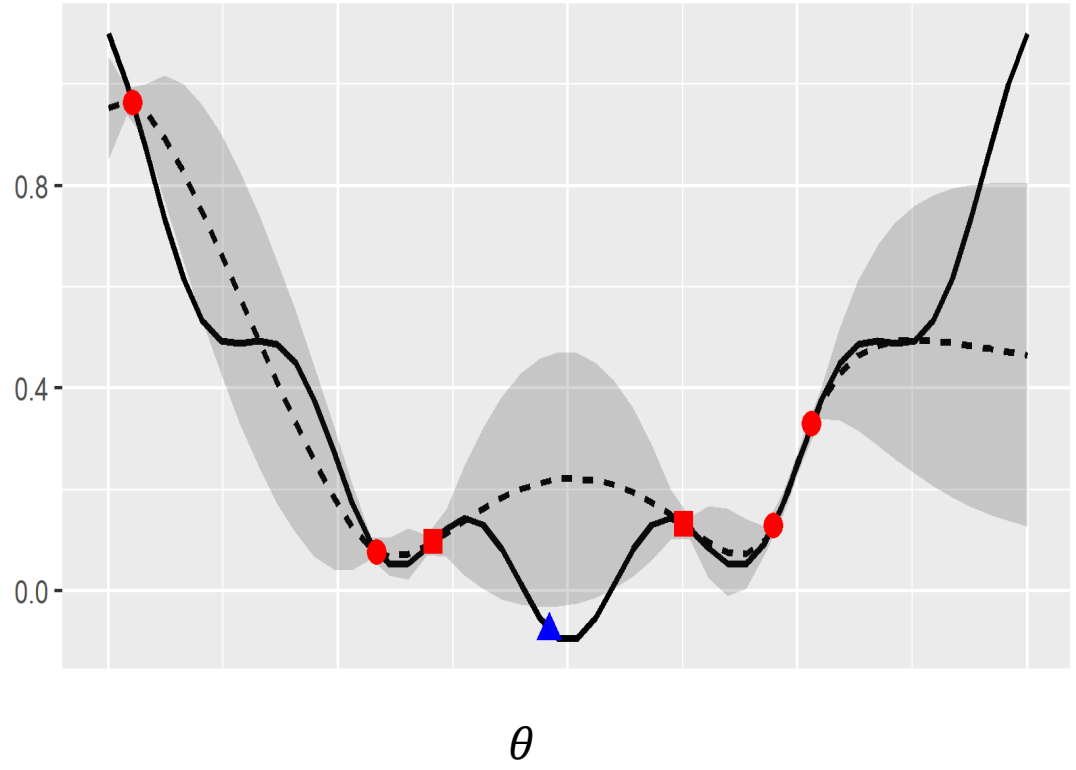
- A single hyper-parameter θ
- The performance is computed in terms of loss L
- The function $\Phi: x \rightarrow y$ is unknown and must be estimated



Optimal, minimal loss for $\theta = 0$

BO operation at iteration i = 7

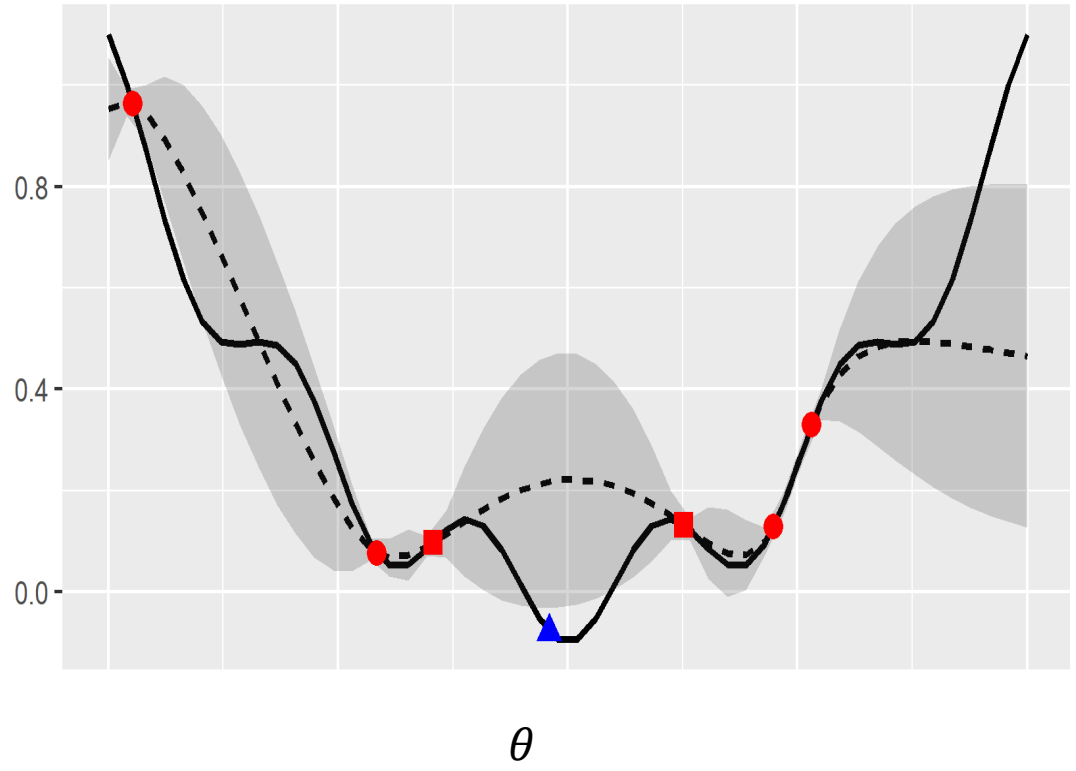
- At iteration i = 7, there are already 6 θ values where L has been evaluated (red points)
- The 6 red points $\{\langle\theta_1, L_1\rangle, \ldots, \langle\theta_6, L_6\rangle\}$ allow to approximate the function Φ (solid line) with $\hat{\Phi}$ (dotted line)
- The grey area indicate the uncertainty around the estimated $\hat{\Phi}$



BO operation at iteration i = 7

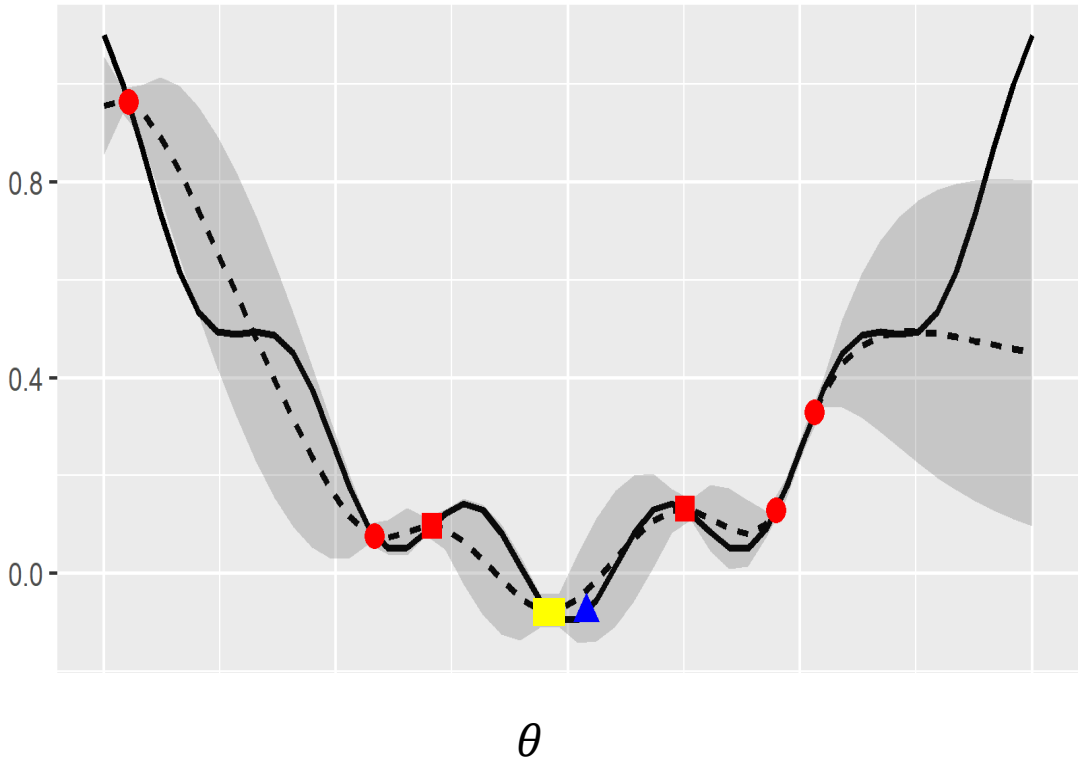
- The BO algorithm next suggests to evaluate L for the θ value marked in blue

- The blue point is identified taking into account:
  - The expected reduction in loss (the larger the better, i.e., exploitation)
  - The uncertainty of $\hat{\Phi}$ (the larger the better, i.e., exploration)



BO operation at iteration i = 8

- L was evaluated at the point proposed at iteration 7 (now in yellow)
- $\hat{\Phi}$ was re-estimated, with considerably less uncertainty
- A new point (blue) is again suggested for the next iteration



Take Home Messages

- **Hyper-parameter search** is an **important step** in machine learning
  - Average performance improvement of 45% [Thornton et al., Auto-WEKA, 2013]

- While optimizing hyper-parameters can be a daunting tasks, **efficient** and **effective** solution for automating this process are under continuous development.

References

- Pearl, *on logic and probability*, Comput. Intel.1988
- Welch, *The Welch-James Approximation to the Distribution of the Residual Sum of Squares in a Weighted Linear Regression*, Biometrika, 69(2),1982
- Olson R.S., Urbanowicz R.J., Andrews P.C., Lavender N.A., Kidd L.C., Moore J.H. (2016) Automating Biomedical Data Science Through Tree-Based Pipeline Optimization. In: Squillero G., Burelli P. (eds) Applications of Evolutionary Computation. EvoApplications 2016. Lecture Notes in Computer Science, vol 9597. Springer, Cham
- Ye F (2017) Particle swarm optimization-based automatic parameter selection for deep neural networks and its applications in large-scale and high-dimensional data. PLoS ONE 12(12): e0188746.
- Hutter, Frank, Holger H. Hoos, and Kevin Leyton-Brown. "Sequential model-based optimization for general algorithm configuration." *International Conference on Learning and Intelligent Optimization*. Springer, Berlin, Heidelberg, 2011.
- Thornton, Chris, et al. "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms." *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013.
- Kohavi & John, "Wrappers for feature subset selection"Artificial Intelligence, 97, 1-2,1997.
- Geris, L. and Gomez-Cabrero, D. (2016). *Uncertainty in Biology*. Springer International Publishing, Chapter 3

References

- Peña J, Nilsson R, Björkegren J, Tegnér J. Towards scalable and data efficient learning of markov boundaries. International Journal of Approximate Reasoning. 2007;45(2):211–232
- TIE* : Statnikonv, A. and Lytkin I. N. (2014). *Algorithms for Discovery of Multiple Markov Boundaries*, JMLR. 2013 Feb; 14: 499–566
- Statistically Equivalent Signatures: Lagani, V., Athineou, G., Farcomeni, A., Tsagris, M. and Tsamardinos, I. (2017) *Feature Selection with the R Package MXM: Discovering Statistically Equivalent Feature Subsets*. Journal of statistical software
- Tsamardinos KDD talk [http://videolectures.net/kdd2017_tsamardinos_feature_selection/]
- MXM R Package (https://cran.r-project.org/web/packages/MXM/index.html)
- Nogueira, L., Sechidis, K. and Brown, G.(2018) *On the Stability of Feature Selection Algorithms*, JMLR 18(174):1−54, 2018
- Grace, T.H., Tsamardinos, I., Raghu, V., Kaminski, N. and Benos V.P. (2015)T-RECS: Stable Selection of Dynamically Formed Groups of Features with Application to Prediction of Clinical Outcomes, Pac Symp Biocomput. 2015;20:431-442
- Klasen, J., Barbez, E., Meier, L., Meinshausen, N., Bühlmann, P., Koornneef, M., Busch, W. and Schneeberger, K. (2016). A multi-marker association method for genome-wide association studies without the need for population structure correction. Nature Communications, 7, p.13299

References

- Lasso for multiple solutions: Pantazis, Y., Lagani, V., Charonyktakis, P., Tsamardinos, I. (2018) *Multiple Equivalent Solutions for the Lasso*. arXiv: 1710.04995
- Tibshirani, *Journal of the Royal Statistical Society. Series B Vol. 58, No. 1* 1996
- Bertsimas et al., *Best Subset Selection via a Modern Optimization Lens*, Annals of Statistics, 44, 2016
- Borboudakis & Tsamardinos, 2017, Forward-Backward Selection with Early Dropping: https://arxiv.org/abs/1705.10770
- Tsagris, M. (2018) Guide on performing selection with R package, https://cran.r-project.org/web/packages/MXM/vignettes/FS_guide.pdf
- Tsamardinos et al., Time and sample efficient discovery of Markov blankets and direct causal relations ACM SIGKDD, 2003
- Tsamardinos et al. Massively-Parallel Feature Selection for Big Data, https://arxiv.org/abs/1708.07178

End of Part III