Bayesian Regression with Factorization Machines for Risk Management and Robust Decision Making

Pablo Angulo, Víctor Gallego, David Gómez-Ullate, Pablo Suárez, Carlos García-Gutiérrez

Valencia, EURO, July 10th, 2018

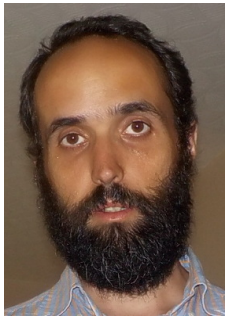
The team

Pure math You consider the topic interesting.

Applied math You consider the topic might be useful to others.

Industrial math Someone has asked you to work on that topic.

- We are an **industrial mathematics** research group.
- We come from different backgrounds.
- We also do applied math, but not together.
- We are part of a larger group mostly based at ICMAT.





POLITÉCNICA

ICMAT Instituto de Ciencias Matemáticas

ICMAT Instituto de Ciencias Matemáticas

POLITÉCNICA

The job

annalect

OmnicomMediaGroup

OMD phd

- Annalect asesorates their customers on **how to spend the marketing budget** (among other things).
- In order to trade *bulk agreements*, decisions must often be made quite in advance (tipically **one year ahead**).
- They want a **data-driven decision process** that is **replicable** for different customers and scenarios, with different regressors and different levels of detail (among other things).
- The final product must be flexible, robust and interactive.

Strategic budget allocation

Tipically once every year, a firm puts together all of its data, and decides where to spend their *marketing budget* for the next period **in order to maximize conversions (sales)**.
A common approach in the industry:

1. Build an **econometric model** that *fits the data well*:

$$Conversions = C^*(a, b) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

where a are the **uncontrollable** parameters, b are the **investment levels** and σ^2 is the **unexplained variance**.

2. Find the investment levels that **maximize** conversions:

$$b^* = \arg\max_b C^*(a, b)$$

3. Make "small" adjustments later (inertia).

Strategic budget allocation

Tipically once every year, a firm puts together all of its data, and decides where to spend their *marketing budget* for the next period **in order to maximize conversions (sales)**.
A common approach in the industry:

1. Build an **econometric model** that *fits the data well*:

$$Conversions = C^*(a, b) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

   where a are the **uncontrollable** parameters, b are the **investment levels** and σ^2 is the **unexplained variance**.

2. Find the investment levels that **maximize** conversions:

$$b^* = \arg\max_b C^*(a, b)$$

3. Make "small" adjustments later, if necessary.

Strategic budget allocation

Tipically once every year, a firm puts together all of its data, and decides where to spend their *marketing budget* for the next period **in order to maximize conversions (sales)**.
A common approach in the industry:

1. Build an **econometric model** that *fits the data well*:

$$Conversions = C^*(a, b) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

   where a are the **uncontrollable** parameters, b are the **investment levels** and σ^2 is the **unexplained variance**.

2. Find the investment levels that **maximize** conversions:

$$b^* = \arg\max_b C^*(a, b)$$

3. Make "small" adjustments later, if necessary.

Strategic budget allocation

Tipically once every year, a firm puts together all of its data, and decides where to spend their *marketing budget* for the next period **in order to maximize conversions (sales)**.
A common approach in the industry:

1. Build an **econometric model** that *fits the data well*:

$$Conversions = C^*(a, b) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

   where a are the **uncontrollable** parameters, b are the **investment levels** and σ^2 is the **unexplained variance**.

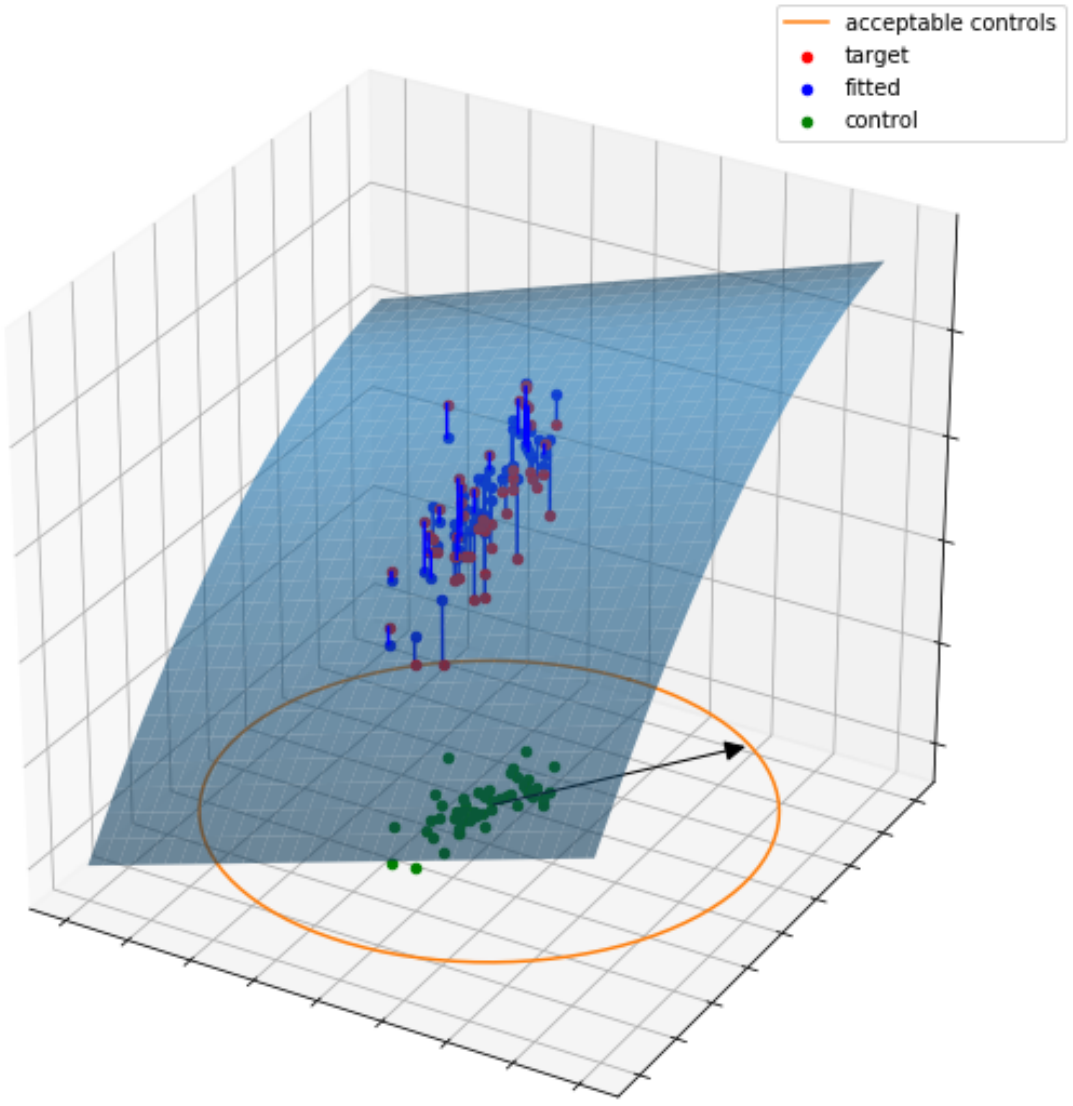
2. Find the investment levels that **maximize** conversions:

$$b^* = \arg\max_b C^*(a, b)$$

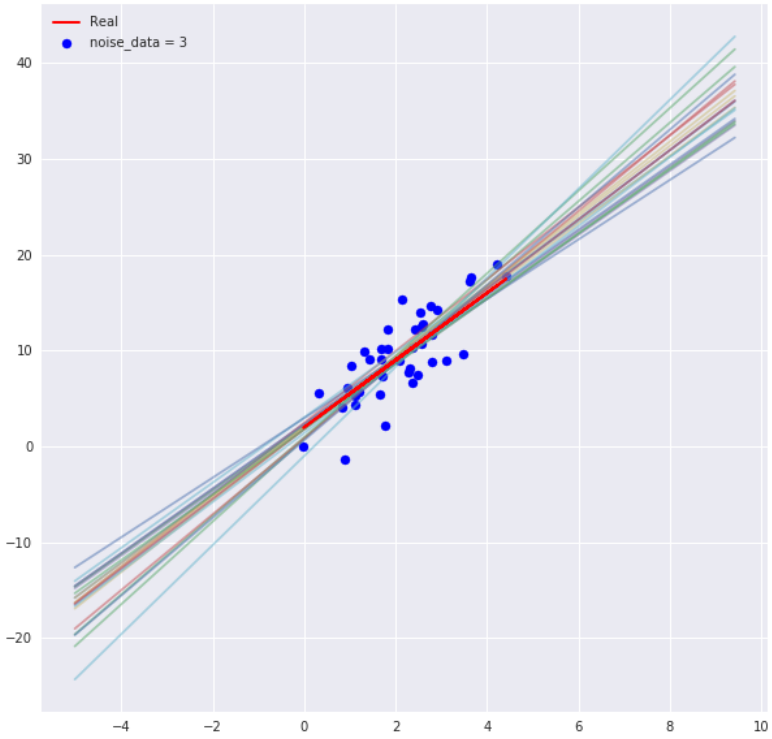
3. Make “small” adjustments later, if necessary.

Fit a model then follow the gradient

But this strategy fails: *follow the gradient and you'll always end up far away from the data, where your model can not be trusted...*



The solution: Bayesian regression



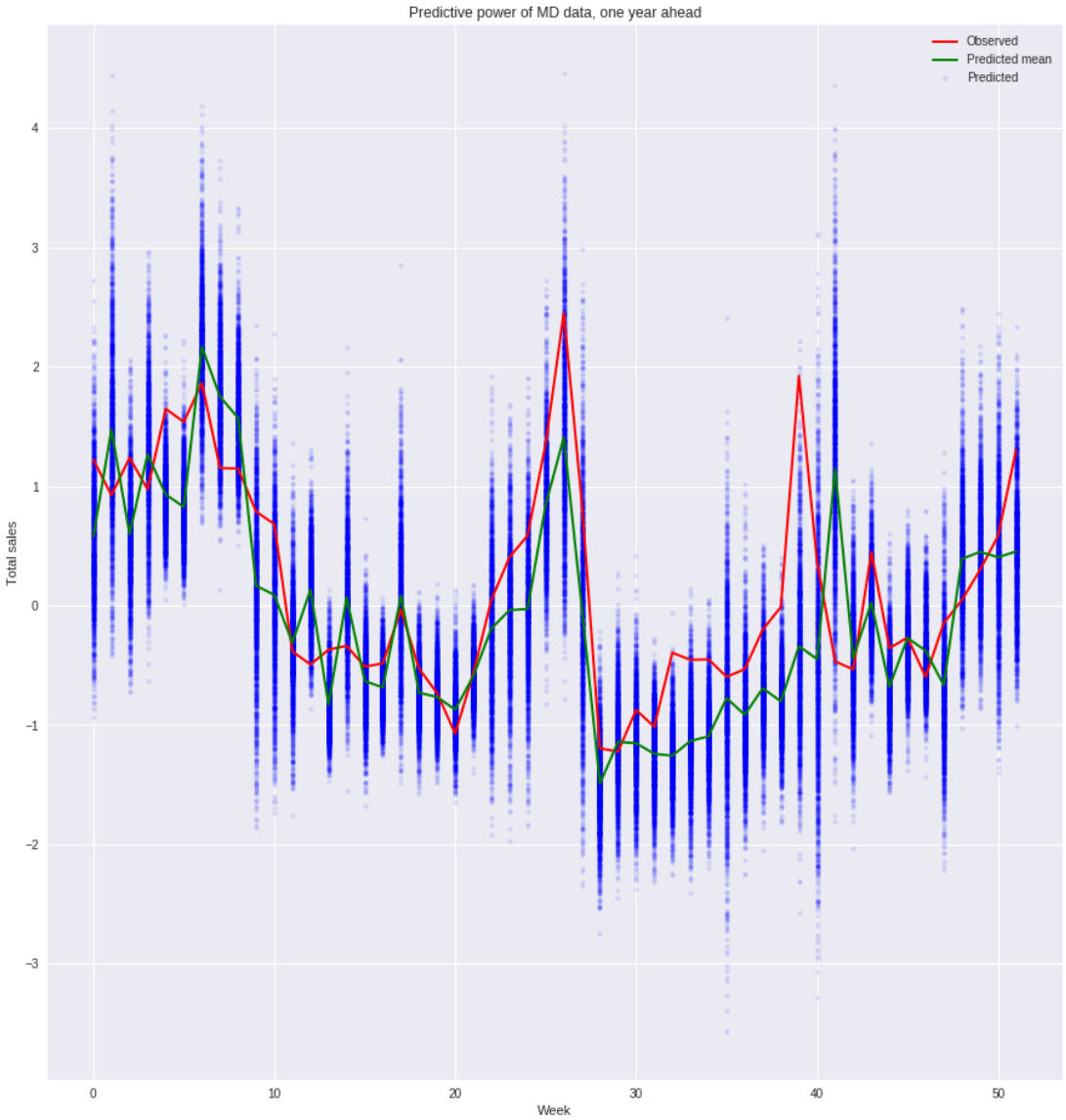
$$\textit{Conversions} = C_\theta(a,b) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

There is uncertainty both in ϵ and in the parameters θ.

- If the data is better, we have less uncertainty about θ.
- If the data is better, we can do with a smaller σ (we *explain more*).
- The uncertainty in ϵ is the same for all (a, b), but $C_\theta(a, b)$ *has smaller uncertainty near the data points*.

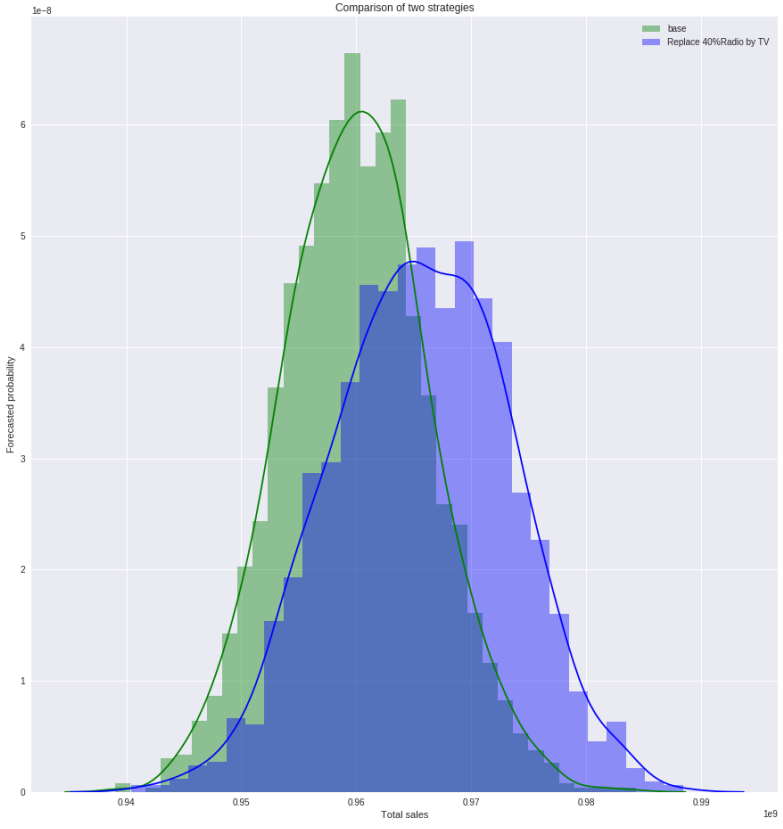
The goal (I): One year-ahead predictions

We sample an ensemble of models and use them to compute predictions of the sales in each week in the coming year.



The goal (II): Compare two strategies

For two different strategies we have different forecasts.



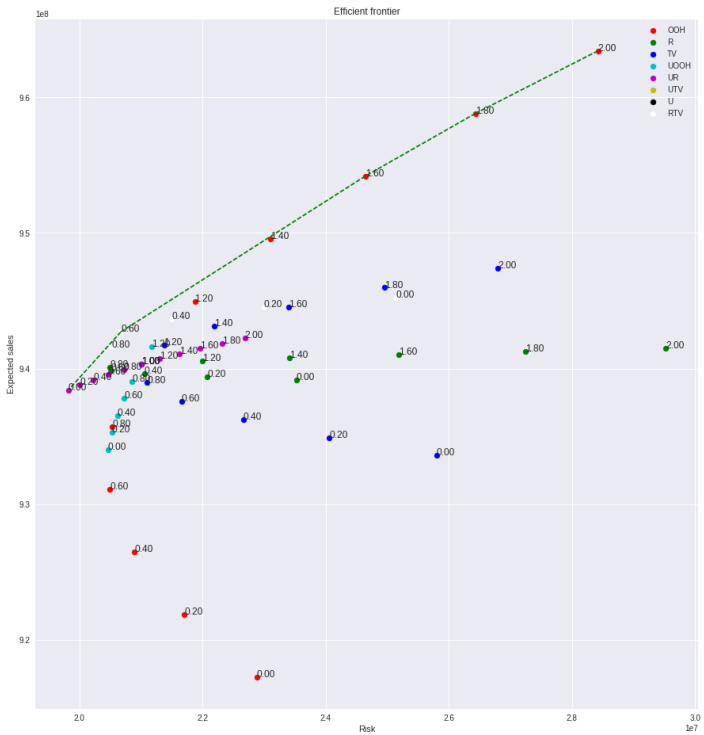
The goal (III): Compare many strategies

We can choose among many competing strategies in a simple way if we only consider expected return and risk.

We plot them in a **risk return spectrum**, similar to the classical ones in portfolio optimization:

The goal (IV): Find out the best strategies

A **risk return spectrum**, with the Pareto front highlighted, simplifies the decision process.
Each point in the Pareto front corresponds to a *single Pareto optimal* media plan (under suitable conditions).
There are many techniques for aproximating the set of Pareto optimal estrategies...



- If the model is linear, it can be computed exactly and quickly.
- Otherwise, it can be found as a series of single-objective optimization problems.
- There are other techniques based on swarms, GAs, etc, but we didn’t use them.

Expectations are computed against the probability of each model.

The final goal: How bad is your data?

It is common in the media industry to work with datasets that have some, or all, of the following problems:

- Data is **aggregated** over weeks, even months.
- Data is also **aggregated spatially** (national sales, national temperature average ...).
- **Important data is missing**. (promotions...)
- Some variables are only activated for a few weeks (but they are important: e.g., soccer world championship)

We can sometimes improve the dataset, but...

The final goal

Ultimately, the procedure must help decide if data-driven decision making is possible at all!

The final goal: How bad is your data?

It is common in the media industry to work with datasets that have some, or all, of the following problems:

- Data is **aggregated** over weeks, even months.
- Data is also **aggregated spatially** (national sales, national temperature average ...).
- **Important data is missing**. (promotions...)
- Some variables are only activated for a few weeks (but they are important: e.g., soccer world championship)

We can sometimes improve the dataset, but...

The final goal

Ultimately, the procedure must help decide if data-driven decision making is possible at all!

How?

- Clean the data
- Gather more data
- Choose the model
- Choose the explanatory variables
- Choose the priors
- Choose the computational tool and software library
- Check performance on the test set
- Choose your objectives (risk metrics)
- Use the posterior to build visualizations
- Show those around, discuss
- Iterate

Data

Our first dataset (a franchise of restaurants) is typical in the industry:

- Data was **aggregated** over weeks and restaurants.
- **Important data is missing**: e.g. is this advertisement a promotion?
- Some variables are only activated for a few weeks (but they are important: e.g., soccer world championship)
- Missing or wrong values, some important data is poorly estimated: **average** temperature for the whole country....

We could improve this a little:

- **EDA** (we use **pandas**).
- **Download data** from AEMET or other official sources.
- **Ask** for **more detailed data** (per day?, per restaurant?)

Data

Our first dataset (a franchise of restaurants) is typical in the industry:

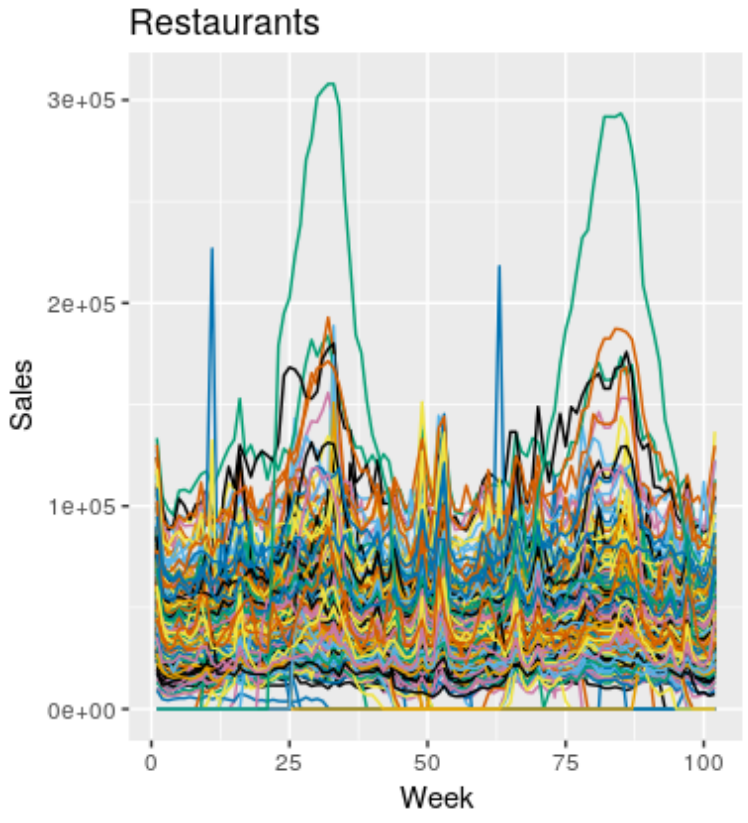
- Data was **aggregated** over weeks and restaurants.
- **Important data is missing**: e.g. is this advertisement a promotion?
- Some variables are only activated for a few weeks (but they are important: e.g., soccer world championship)
- Missing or wrong values, some important data is poorly estimated: **average** temperature for the whole country....

We could improve this a little:

- **EDA** (we use **pandas**).
- **Download data** from AEMET or other official sources.
- **Ask** for **more detailed data** (per day?, per restaurant?)

A second dataset

Eventually asking worked, and they gave us more data: **sales per week and per restaurant**.



So we move from $\sim$ 240 to $\sim$ 100000 data points.

Not small data any more.

To select variables, or not to select variables

We got a huge list of regressors. Similar GOF metrics can be obtained with less variables. But it's worse than that:

- Suppose **two variables are highly correlated**. **One is a control variable** (e.g. investment in a certain channel), **the other is not** (e.g. temperature).
- Naturally, we get *similar metrics if we use one or the other*.
- If we *drop the control variable*, we declare it has *no effect*.
- If we *drop the other variable*, we *overestimate the effect* of the control variable.

Bayesian Regression solves this problem without special effort:

If two variables are positively correlated, their weights in linear bayesian regression are negatively correlated.
Interpretation: *the model can explain some part of the variance with either variable*. **The data doesn't explain if the effect is due to one or the other.**

This can be shown using the conjugate prior...

To select variables, or not to select variables

We got a huge list of regressors. Similar GOF metrics can be obtained with less variables. But it's worse than that:

- Suppose **two variables are highly correlated**. **One is a control variable** (e.g. investment in a certain channel), **the other is not** (e.g. temperature).
- Naturally, we get *similar metrics if we use one or the other*.
- If we *drop the control variable*, we declare it has *no effect*.
- If we *drop the other variable*, we *overestimate the effect* of the control variable.

Bayesian Regression solves this problem without special effort:

If two variables are positively correlated, their weights in linear bayesian regression are negatively correlated.
Interpretation: *the model can explain some part of the variance with either variable*. **The data doesn't explain if the effect is due to one or the other.**

This can be shown using the conjugate prior...

To select variables, or not to select variables

We got a huge list of regressors. Similar GOF metrics can be obtained with less variables. But it's worse than that:

- Suppose **two variables are highly correlated**. **One is a control variable** (e.g. investment in a certain channel), **the other is not** (e.g. temperature).
- Naturally, we get *similar metrics if we use one or the other*.
- If we *drop the control variable*, we declare it has *no effect*.
- If we *drop the other variable*, we *overestimate the effect* of the control variable.

Bayesian Regression solves this problem without special effort:

If two variables are positively correlated, their weights in linear bayesian regression are negatively correlated.
Interpretation: *the model can explain some part of the variance with either variable*. **The data doesn't explain if the effect is due to one or the other.**

This can be shown using the conjugate prior...

To select variables, or not to select variables

We got a huge list of regressors. Similar GOF metrics can be obtained with less variables. But it's worse than that:

- Suppose **two variables are highly correlated**. **One is a control variable** (e.g. investment in a certain channel), **the other is not** (e.g. temperature).
- Naturally, we get *similar metrics if we use one or the other*.
- If we *drop the control variable*, we declare it has *no effect*.
- If we *drop the other variable*, we *overestimate the effect* of the control variable.

Bayesian Regression solves this problem without special effort:

If two variables are positively correlated, their weights in linear bayesian regression are negatively correlated.
Interpretation: *the model can explain some part of the variance with either variable*. **The data doesn't explain if the effect is due to one or the other.**

This can be shown using the conjugate prior...

To select variables, or not to select variables

We got a huge list of regressors. Similar GOF metrics can be obtained with less variables. But it's worse than that:

- Suppose **two variables are highly correlated**. **One is a control variable** (e.g. investment in a certain channel), **the other is not** (e.g. temperature).
- Naturally, we get *similar metrics if we use one or the other*.
- If we *drop the control variable*, we declare it has *no effect*.
- If we *drop the other variable*, we *overestimate the effect* of the control variable.

Bayesian Regression solves this problem without special effort:

If two variables are positively correlated, their weights in linear bayesian regression are negatively correlated.
Interpretation: *the model can explain some part of the variance with either variable*. **The data doesn't explain if the effect is due to one or the other.**

This can be shown using the conjugate prior...

To select variables, or not to select variables

We got a huge list of regressors. Similar GOF metrics can be obtained with less variables. But it's worse than that:

- Suppose **two variables are highly correlated**. **One is a control variable** (e.g. investment in a certain channel), **the other is not** (e.g. temperature).
- Naturally, we get *similar metrics if we use one or the other*.
- If we *drop the control variable*, we declare it has *no effect*.
- If we *drop the other variable*, we *overestimate the effect* of the control variable.

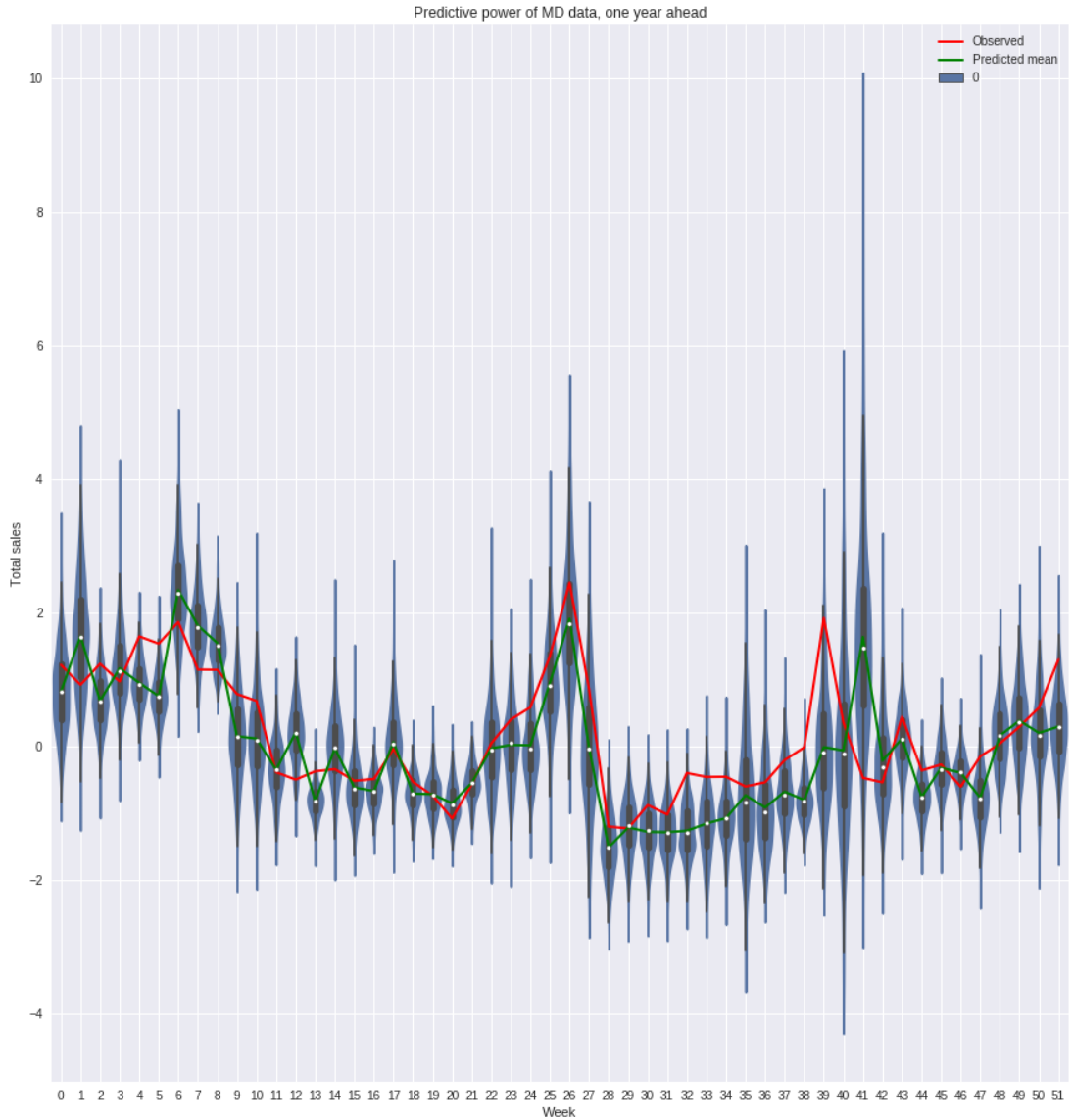
Bayesian Regression solves this problem without special effort:

If two variables are positively correlated, their weights in linear bayesian regression are negatively correlated.
Interpretation: *the model can explain some part of the variance with either variable*. **The data doesn't explain if the effect is due to one or the other.**

This can be shown using the conjugate prior...

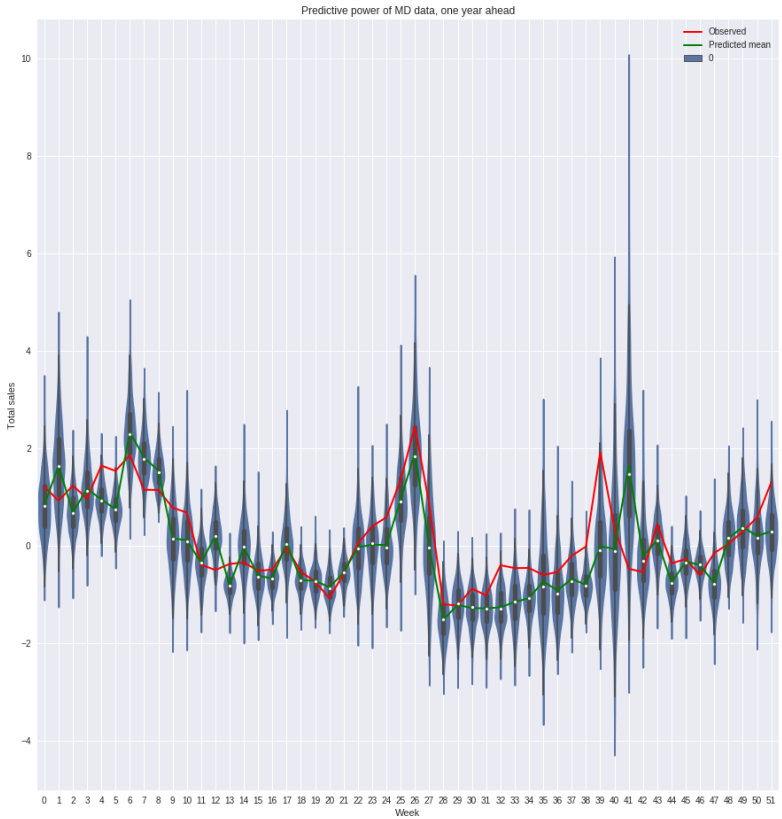
Priors

Did I mention *some variables are only activated for a few weeks*?
Look at week 41 for this prediction:



Priors

At week 41 for this prediction, a variable that *had never been activated in the training set* became active, so the **priors** for the coefficients of that variable **are added to the posterior unmodified** (assume the model is linear).



If some variables are important, but are only active during a few weeks, or not at all during the training set, the prior goes unmodified onto the posterior, so a non-informative prior does not work.

Some degree of **expert ellicitation** is necessary:

- possible impact of an explanatory var.
- positive correlation for the weights for similar variables.

The model

- For linear and nonlinear dense models, **conjugate priors** exists and are very useful:
  - Interpretation of the priors
  - Speed up the computations
  - Understand some details on a higher level (like correlated explanatory variables)
- For aggregated datasets, linear models are just fine, possibly with a few higher order terms (cherry picking).
- For "matrix" data (e.g., sales per restaurant and week), we used **Factorization Machines**. There is actually a **conjugate prior** for this model too, but it needs extra work...
  See also *"Bayesian Dynamic Tensor Regression" (M. Billio, R. Casarin, M. Iacopini)*

The model

- For linear and nonlinear dense models, **conjugate priors** exists and are very useful:
  - Interpretation of the priors
  - Speed up the computations
  - Understand some details on a higher level (like correlated explanatory variables)
- For aggregated datasets, linear models are just fine, possibly with a few higher order terms (cherry picking).
- For "matrix" data (e.g., sales per restaurant and week), we used **Factorization Machines**. There is actually a **conjugate prior** for this model too, but it needs extra work...
  See also *"Bayesian Dynamic Tensor Regression" (M. Billio, R. Casarin, M. Iacopini)*

The model

- For linear and nonlinear dense models, **conjugate priors** exists and are very useful:
  - Interpretation of the priors
  - Speed up the computations
  - Understand some details on a higher level (like correlated explanatory variables)
- For aggregated datasets, linear models are just fine, possibly with a few higher order terms (cherry picking).
- For "matrix" data (e.g., sales per restaurant and week), we used **Factorization Machines**. There is actually a **conjugate prior** for this model too, but it needs extra work...
  See also *"Bayesian Dynamic Tensor Regression" (M. Billio, R. Casarin, M. Iacopini)*

The model

- For linear and nonlinear dense models, **conjugate priors** exists and are very useful:
  - Interpretation of the priors
  - Speed up the computations
  - Understand some details on a higher level (like correlated explanatory variables)
- For aggregated datasets, linear models are just fine, possibly with a few higher order terms (cherry picking).
- For "matrix" data (e.g., sales per restaurant and week), we used **Factorization Machines**. There is actually a **conjugate prior** for this model too, but it needs extra work...
  See also *"Bayesian Dynamic Tensor Regression" (M. Billio, R. Casarin, M. Iacopini)*

The model

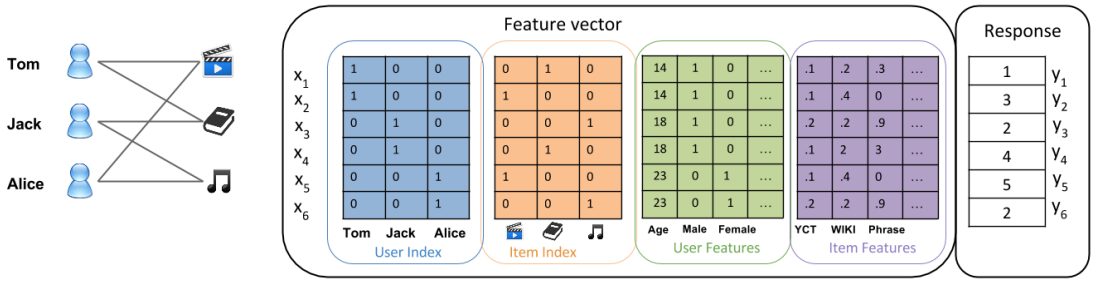
- For linear and nonlinear dense models, **conjugate priors** exists and are very useful:
  - Interpretation of the priors
  - Speed up the computations
  - Understand some details on a higher level (like correlated explanatory variables)
- For aggregated datasets, linear models are just fine, possibly with a few higher order terms (cherry picking).
- For "matrix" data (e.g., sales per restaurant and week), we used **Factorization Machines**. There is actually a **conjugate prior** for this model too, but it needs extra work...
  See also *"Bayesian Dynamic Tensor Regression" (M. Billio, R. Casarin, M. Iacopini)*

The model

- For linear and nonlinear dense models, **conjugate priors** exists and are very useful:
  - Interpretation of the priors
  - Speed up the computations
  - Understand some details on a higher level (like correlated explanatory variables)
- For aggregated datasets, linear models are just fine, possibly with a few higher order terms (cherry picking).
- For “matrix” data (e.g., sales per restaurant and week), we used **Factorization Machines**. There is actually a **conjugate prior** for this model too, but it needs extra work...
  See also *“Bayesian Dynamic Tensor Regression” (M. Billio, R. Casarin, M. Iacopini)*

Factorization Machines

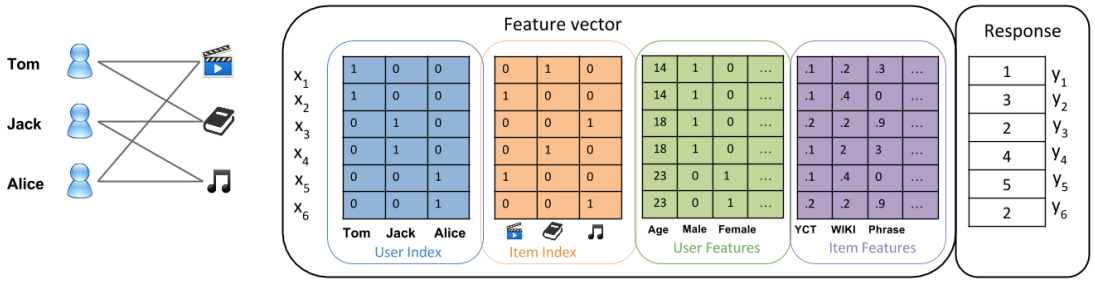
A Factorization Machine is a state-of-the-art machine learning technique:



- A nice trick makes it potentially **very efficient**.
- First used for **recommendation systems**.
- Designed to work with many predictor categorical variables, which are codified as 0/1 variables, most of which are zero (one-hot encoding).
- Works with continuous variables too.

Factorization Machines

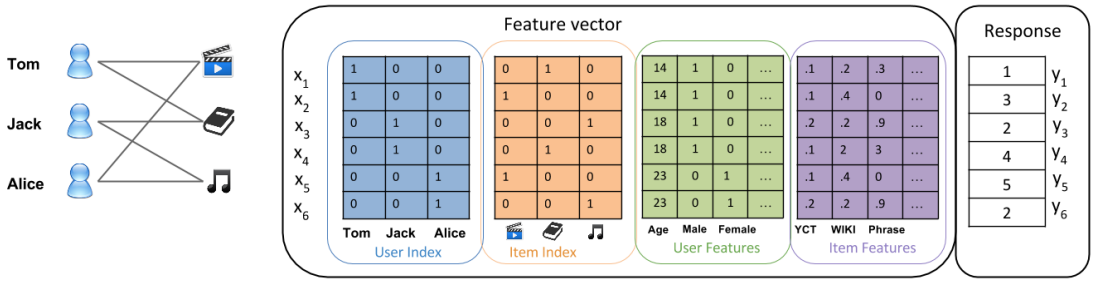
A Factorization Machine is a state-of-the-art machine learning technique:



- A nice trick makes it potentially **very efficient**.
- First used for **recommendation systems**.
- Designed to work with many predictor categorical variables, which are codified as 0/1 variables, most of which are zero (one-hot encoding).
- Works with continuous variables too.

Factorization Machines

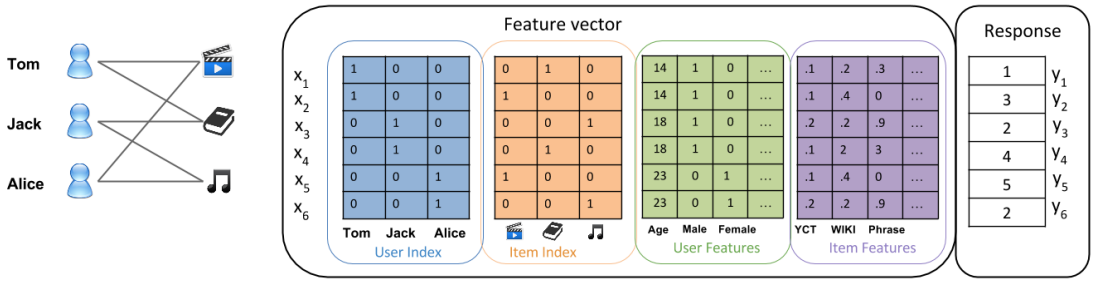
A Factorization Machine is a state-of-the-art machine learning technique:



- A nice trick makes it potentially **very efficient**.
- First used for **recommendation systems**.
- Designed to work with many predictor categorical variables, which are codified as 0/1 variables, most of which are zero (one-hot encoding).
- Works with continuous variables too.

Factorization Machines

A Factorization Machine is a state-of-the-art machine learning technique:

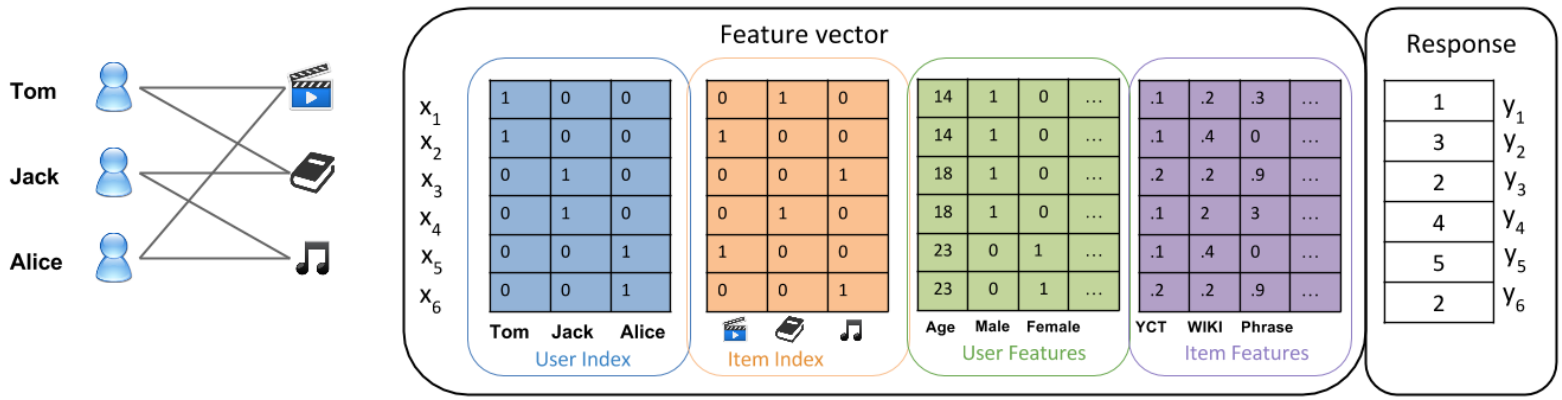


- A nice trick makes it potentially **very efficient**.
- First used for **recommendation systems**.
- Designed to work with many predictor categorical variables, which are codified as 0/1 variables, most of which are zero (one-hot encoding).
- Works with continuous variables too.

Factorization Machines

A Factorization Machine is a **quadratic model** (can be of higher order, but this is not common), but the quadratic part has **low rank**:

$$\hat{y}^{FM}(x) := w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j>i} \langle v_i, v_j \rangle x_i x_j$$



A bounded rank makes it **cheaper** to compute, and forces it to **generalize**, instead of overfit, in abundance of predictor variables.

Factorization Machines

If we use a *dense* quadratic matrix, the coefficient for each interaction restaurant $\times$ regressor will only be trained when that regressor is active for that restaurant's data.

If, for instance, *a new restaurant opens in winter*, we don't have a clue how it will perform in summer.

But its performance in winter may be enough to clasiffy it as a *"summer restaurant"*.

Factorization Machines

If we use a *dense* quadratic matrix, the coefficient for each interaction restaurant $\times$ regressor will only be trained when that regressor is active for that restaurant's data.

If, for instance, *a new restaurant opens in winter*, we don't have a clue how it will perform in summer.

But its performance in winter may be enough to clasiffy it as a *"summer restaurant"*.

Computational approach

- Split the data set into a train and a test set.
- For linear and nonlinear dense models, **conjugate priors** exists and are useful.
- For Factorization Machines, we used **Markov Chain Monte Carlo (MCMC)**: a *random walk* is *guided* by the posterior probability distribution so that the time spent in each region of the parameter space is proportional to its probability.
  We obtain an ensemble of models $\{C_{\theta_i}\}_{i=1}^{K}$ that is *an approximate sample from the posterior probability distribution*.
- We have also used **Stochastic Variational Inference**: the posterior is **approximated by a member of a parametric family** with finitely many parameters.

Computational approach

- Split the data set into a train and a test set.
- For linear and nonlinear dense models, **conjugate priors** exists and are useful.
- For Factorization Machines, we used **Markov Chain Monte Carlo (MCMC)**: a *random walk* is *guided* by the posterior probability distribution so that the time spent in each region of the parameter space is proportional to its probability.
  We obtain an ensemble of models $\{C_{\theta_i}\}_{i=1}^{K}$ that is *an approximate sample from the posterior probability distribution*.
- We have also used **Stochastic Variational Inference**: the posterior is **approximated by a member of a parametric family** with finitely many parameters.

Computational approach

- Split the data set into a train and a test set.
- For linear and nonlinear dense models, **conjugate priors** exists and are useful.
- For Factorization Machines, we used **Markov Chain Monte Carlo (MCMC)**: a *random walk* is *guided* by the posterior probability distribution so that the time spent in each region of the parameter space is proportional to its probability.
  We obtain an ensemble of models $\{C_{\theta_i}\}_{i=1}^{K}$ that is *an approximate sample from the posterior probability distribution*.
- We have also used **Stochastic Variational Inference**: the posterior is **approximated by a member of a parametric family** with finitely many parameters.

Computational approach

- Split the data set into a train and a test set.
- For linear and nonlinear dense models, **conjugate priors** exists and are useful.
- For Factorization Machines, we used **Markov Chain Monte Carlo (MCMC)**: a *random walk* is *guided* by the posterior probability distribution so that the time spent in each region of the parameter space is proportional to its probability.
  We obtain an ensemble of models $\{C_{\theta_i}\}_{i=1}^{K}$ that is *an approximate sample from the posterior probability distribution*.
- We have also used **Stochastic Variational Inference**: the posterior is **approximated by a member of a parametric family** with finitely many parameters.

Software libraries

Some software for bayesian regression and/or factorization machines.

fastfm Oficially, it does "*Bayesian Factorization Machines*", and it does MCMC walks, but is not really for bayesian regression.

libfm The reference implementation, it also does "*Bayesian Factorization Machines*". Allows for block decompositions, not for bayesian regression.

polylearn Does higher order interactions, easy to use, not bayesian.

vowpal wabbit Very fast, multicore, no python interface, not bayesian.

fmpytorch FM on pytorch is a good idea, but it didn't work for us.

pytorch + pyro Very *promising*, does **Stochastic Variational Inference**, but *still beta*.
pyro also finds the optimal σ^2 while training (max likelihood).

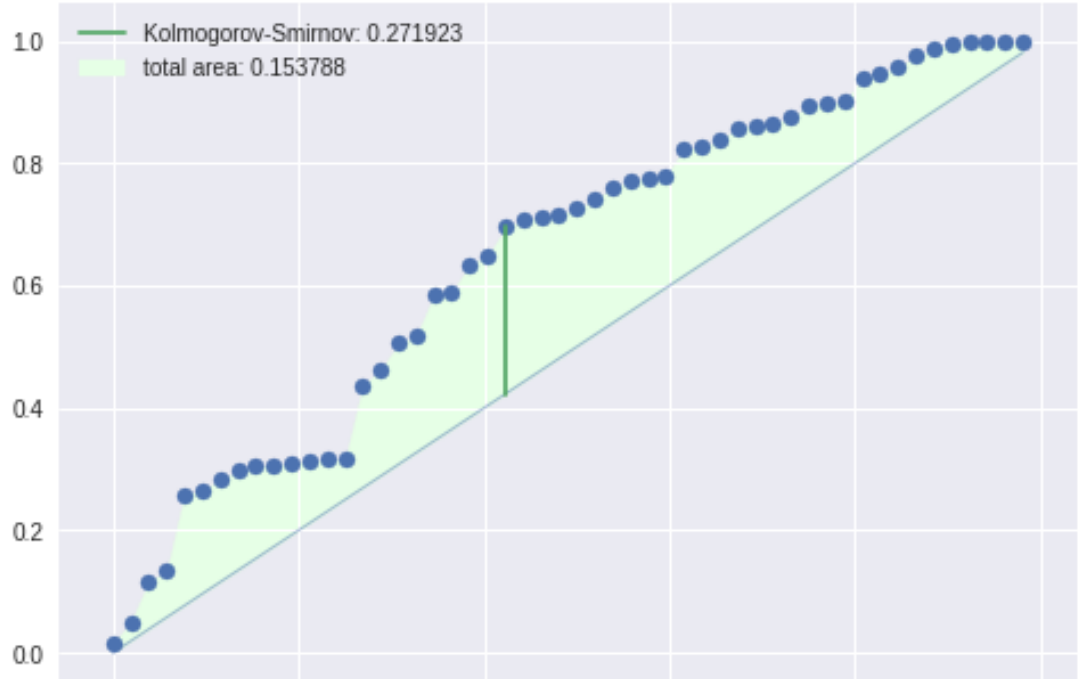
Using the test set for validation

Using the **test set**, we can *validate our predictions*:

- Compare the mean of the prediction with real sales in each week of the test period.
- But we also need to estimate risk!

Problem: each observed sales value must be compared to a prediction from a different distribution.

Solution: the quantiles of the real sales within the MCMC sample should come from a uniform distribution: we can test this with a P-P plot.



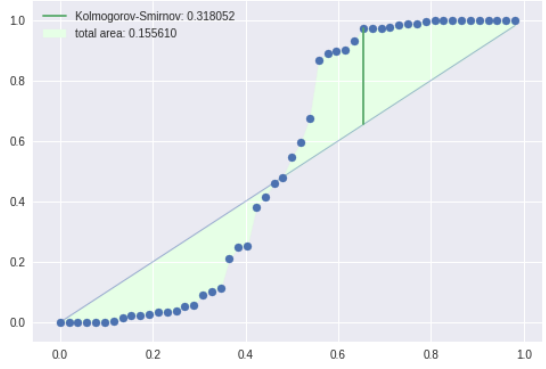
Unexplained variance

The P-P plot can detect a prediction that:

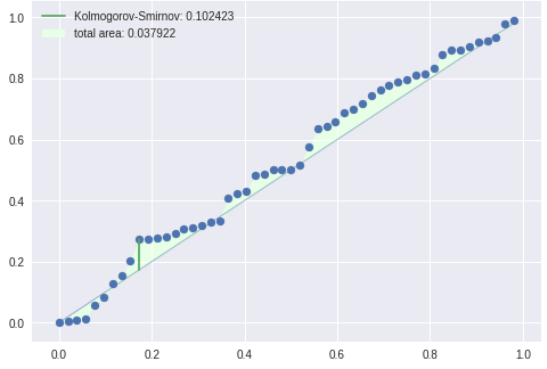
underestimates risk The observed data will often be **above**, or **below**, the predictions of all the ensemble.

overestimates risk The observed data will be in the **mid quantiles** too often.

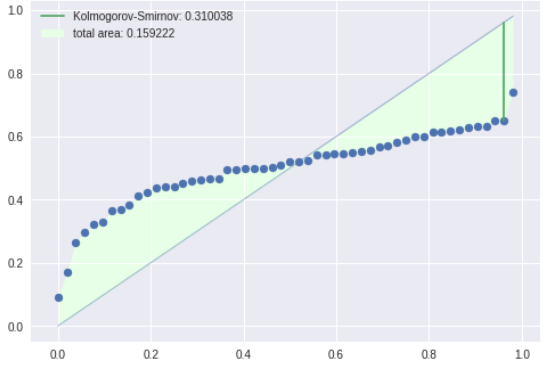
If uncertainty is underestimated, we can improve our predictions adding an independent noise:



A P-P plot that underestimates risk



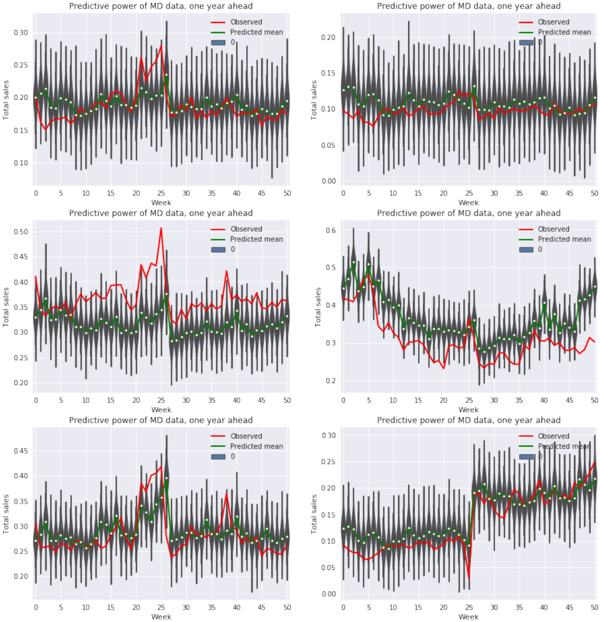
A P-P plot with optimal noise



A P-P plot with too much noise

In other words, add the noise level that *minimizes the* ***Kolmogorov-Smirnov*** *statistic*.

Predictions for each restaurant



Restaurants-at-risk

We can now enlarge our decision space.

The restaurant chain is a franchise

A plan with good total expected sales and good total variance could be discarded if it upsets too many restaurant owners.

Restaurant's Risk. $RR_i = P(Sales_i < Q_i(0.05; base); new)$

- Q_i is the quantile function for restaurant i under the base investment plan.
- $P(E; new)$ is the probability of E under the new investment plan.

Restaurants at Risk : Expected number of restaurants whose sales figure with the new plan will be below the 5% quantile of the base plan.
It is also $\sum RR_i$.

The number of "*Restaurants-At-Risk*" is the expected number of restaurant owners that will complain about the new strategy.

Restaurants-at-risk

We can now enlarge our decision space.

The restaurant chain is a franchise

A plan with good total expected sales and good total variance could be discarded if it upsets too many restaurant owners.

Restaurant's Risk. $RR_i = P(Sales_i < Q_i(0.05; base); new)$

- Q_i is the quantile function for restaurant i under the base investment plan.
- $P(E; new)$ is the probability of E under the new investment plan.

Restaurants at Risk : Expected number of restaurants whose sales figure with the new plan will be below the 5% quantile of the base plan.
It is also $\sum RR_i$.

The number of "*Restaurants-At-Risk*" is the expected number of restaurant owners that will complain about the new strategy.

Restaurants-at-risk

We can now enlarge our decision space.

The restaurant chain is a franchise

A plan with good total expected sales and good total variance could be discarded if it upsets too many restaurant owners.

Restaurant's Risk. $RR_i = P(Sales_i < Q_i(0.05; base); new)$

- Q_i is the quantile function for restaurant i under the base investment plan.
- $P(E; new)$ is the probability of E under the new investment plan.

Restaurants at Risk : Expected number of restaurants whose sales figure with the new plan will be below the 5% quantile of the base plan.
It is also $\sum RR_i$.

The number of "*Restaurants-At-Risk*" is the expected number of restaurant owners that will complain about the new strategy.

Conclusions

- FMs are non-linear, but #parameters is linear in #features.
- FMs can mix continuous and categorical variables.
- Some questions can be answered with confidence, others can't.
- The model has built-in risk estimation, but we can use the test set to validate.
- We can decide if a particular data set supports good decisions, or not.
- The model is general and can be adapted to very different scenarios.

Annalect is already applying Linear Bayesian Regression, for aggregated datasets.

Conclusions

- FMs are non-linear, but #parameters is linear in #features.
- FMs can mix continuous and categorical variables.
- Some questions can be answered with confidence, others can't.
- The model has built-in risk estimation, but we can use the test set to validate.
- We can decide if a particular data set supports good decisions, or not.
- The model is general and can be adapted to very different scenarios.

Annalect is already applying Linear Bayesian Regression, for aggregated datasets.

Conclusions

- FMs are non-linear, but #parameters is linear in #features.
- FMs can mix continuous and categorical variables.
- Some questions can be answered with confidence, others can't.
- The model has built-in risk estimation, but we can use the test set to validate.
- We can decide if a particular data set supports good decisions, or not.
- The model is general and can be adapted to very different scenarios.

Annalect is already applying Linear Bayesian Regression, for aggregated datasets.

Conclusions

- FMs are non-linear, but #parameters is linear in #features.
- FMs can mix continuous and categorical variables.
- Some questions can be answered with confidence, others can't.
- The model has built-in risk estimation, but we can use the test set to validate.
- We can decide if a particular data set supports good decisions, or not.
- The model is general and can be adapted to very different scenarios.

Annalect is already applying Linear Bayesian Regression, for aggregated datasets.

Conclusions

- FMs are non-linear, but #parameters is linear in #features.
- FMs can mix continuous and categorical variables.
- Some questions can be answered with confidence, others can't.
- The model has built-in risk estimation, but we can use the test set to validate.
- We can decide if a particular data set supports good decisions, or not.
- The model is general and can be adapted to very different scenarios.

Annalect is already applying Linear Bayesian Regression, for aggregated datasets.

Conclusions

- FMs are non-linear, but #parameters is linear in #features.
- FMs can mix continuous and categorical variables.
- Some questions can be answered with confidence, others can’t.
- The model has built-in risk estimation, but we can use the test set to validate.
- We can decide if a particular data set supports good decisions, or not.
- The model is general and can be adapted to very different scenarios.

Annalect is already applying Linear Bayesian Regression, for aggregated datasets.

Conclusions

- FMs are non-linear, but #parameters is linear in #features.
- FMs can mix continuous and categorical variables.
- Some questions can be answered with confidence, others can’t.
- The model has built-in risk estimation, but we can use the test set to validate.
- We can decide if a particular data set supports good decisions, or not.
- The model is general and can be adapted to very different scenarios.

Annalect is already applying Linear Bayesian Regression, for aggregated datasets.

Conclusions

- FMs are non-linear, but #parameters is linear in #features.
- FMs can mix continuous and categorical variables.
- Some questions can be answered with confidence, others can’t.
- The model has built-in risk estimation, but we can use the test set to validate.
- We can decide if a particular data set supports good decisions, or not.
- The model is general and can be adapted to very different scenarios.

Annalect is already applying Linear Bayesian Regression, for aggregated datasets.

FAQs

Questions?

Quote from “Robust Portfolio Optimization and Management”

FAQ #1: Is this special to the media industry?

There are **fundamental differences** between the media industry and the financial industry, but this problem is common to both.

> Although advanced optimization software is widely available, many asset managers have problems applying optimization methodology or avoid it altogether. One reason is that in practical applications **portfolio optimization is very sensitive to the inputs** (e.g., expected returns of assets and their covariances), and **“optimal” portfolios frequently have extreme or non-intuitive weights** for some assets.
>
> Generally, the practitioner’s solution to this problem has been to *add constraints to the original optimization problem in order to limit nonintuitive results*. However, as a result, the constraints—instead of forecasts—often determine the portfolio, **making the risk-return optimization process pointless**.
>
> Fabozzi et al,**Robust Portfolio Optimization and Management**.

... but this is even more general.

Why bayesian?

FAQ #2: Why does it have to be bayesian?

- We need a probability distribution for each possible strategy, based on everything we have: data and modelling knowledge.
- Most of the predictions are for situations that will never be observed: subjective probability.
- Thus we are subject to Bayes theorem, so **even if we choose bootstrap**, we could look at the prior and critize it.
- Why not use prior experience?

Why bayesian?

FAQ #2: Why does it have to be bayesian?

- We need a probability distribution for each possible strategy, based on everything we have: data and modelling knowledge.
- Most of the predictions are for situations that will never be observed: subjective probability.
- Thus we are subject to Bayes theorem, so **even if we choose bootstrap**, we could look at the prior and critize it.
- Why not use prior experience?

Why bayesian?

FAQ #2: Why does it have to be bayesian?

- We need a probability distribution for each possible strategy, based on everything we have: data and modelling knowledge.
- Most of the predictions are for situations that will never be observed: subjective probability.
- Thus we are subject to Bayes theorem, so **even if we choose bootstrap**, we could look at the prior and critize it.
- Why not use prior experience?

Why bayesian?

FAQ #2: Why does it have to be bayesian?

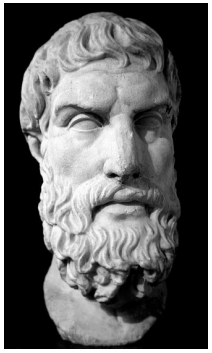
- We need a probability distribution for each possible strategy, based on everything we have: data and modelling knowledge.
- Most of the predictions are for situations that will never be observed: subjective probability.
- Thus we are subject to Bayes theorem, so **even if we choose bootstrap**, we could look at the prior and critize it.
- Why not use prior experience?

Occam - Epicurus - Bayes

FAQ #3: How do you explain Bayesian Statistics to a Company?

William of Ockham

Of all the possible explanations of a phenomenon, choose the simplest.



Epicurus

Of all the possible explanations of a phenomenon, keep them all.

Bayes & Laplace

Of all the possible explanations of a phenomenon, assign a probability to each of them.

An ensemble of experts

Another way to think of Bayesian Regression: we do not ask our questions to a single model, but to an ensemble.

Interpretation of Bayesian Regression

If all the models give **similar answers**, we are **confident in our predictions**.
If the models give wildly **different answers**, the **data does not support any conclusion**.
It may happen that *some questions get precise answers*, while others do not.