University of Novi Sad
Faculty of Sciences
Department of mathematics and informatics

Milica Brkić

Crop yield prediction by data fusion using matrix factorization

Master thesis

Supervisor:

dr. Sanja Brdar

2018, Novi Sad

Abstract

The aim of this work is the crop, in particular maize and soybean, yield prediction. Accurate prediction is important in order to be able to choose the best hybrid for given location. We can look at that task as recommendation system, where we want to "recommend" the best hybrid for field. The definition of the best here means hybrid that will give the highest yield in given circumstances. Algorithm that is used to help us do that is known as Data fusion by matrix factorization algorithm (DFMF algorithm). This thesis provides the description of the DFMF algorithm and its application on data that comes from Syngenta Crop Challenge. Complete preprocessing, processing and visualisation of the data is performed in Python.

Contents

LIST OF FIGURES

LIST OF TABLES

1 Introduction

Predicting crop yield is a very hard task. There are so many factors that influence final outcome: soil properties (content of organic matter, soil pH value, percentage of clay, silt, sand in soil, cation exchange capacity, etc), weather conditions (precipitation, temperature, solar radiation, etc.), amount and the type of fertilizers and pesticides used, genetic characteristics of planted hybrid, planting date, and many others. Complex interaction of all these factors determines the final crop yield.

As world's population is growing so does the demand for food is increasing. To tackle challenges of increased demand for food, seed industries and breeders are seeking the way to develop and improve seed varieties. Yield is one of the best indicator for making the decision which seed varieties would be suitable for the given location, so we need to be able to predict yield.

Idea for choosing this topic came from Syngenta Crop Challenge (`https://www.ideaconnection.com/syngenta-crop-challenge/`). Syngenta (`https://www.syngenta.com/`) is one of companies who is trying hard to bring new technologies in agriculture. Every year from 2016 it opens challenges in order to invite people all around the world to help them in creating smarter production. The goal is to apply various machine learning and mathematical models on historical data in order to help industry breed better seeds. Discovering patterns may help scientist to evaluate seeds more accurately and choose the best candidates for further improvements, and can also help farmers to optimize the production just by smart seed selection.

The experimental part of this thesis includes the analyzes performed on corn data sets that originate from Syngenta Crop Challenge 2018 and on soybean data sets that came from Syngenta Crop Challenge 2017.

Today data is generated more easily than ever before, thanks to devices for high-throughput screening, sensors, cameras on drones, satellites. It offers a plenty of information only if we know how to extract it from raw data. Difficulties that come on way are how to jointly observe all heterogeneous input spaces so that they could benefit from each other. Overcoming this problem is vital in order to improve prediction accuracy. When we have multiple relations, which are represented as multiple matrices, we want to exploit information from one relation when predicting another.

In the thesis the main tool used for analyzing is *Data fusion by matrix fac-*

torization algorithm (DFMF) which is a penalized matrix tri-factorization model that collectively tri-factorizes many data matrices such that each data matrix is decomposed into a product of tree latent matrices. The algorithm was introduced in 2015 in the doctoral dissertation of Marinka Žitnik. The algorithm is flexible, requires minimal input data transformations and it can handle both multiple relations and multiple object type data.

The problem considered in the thesis is maize and soybean yield prediction. There is a limited number of locations on which breeders can plant hybrids and that causes uncertainty when trying to choose the best hybrids for the growers to plant. The task is to accurately predict the performance of each individual hybrid so that maize and soybean breeders could make better decisions on which hybrids to move forward and provide to growers, that would finally increase productivity to meet the world's growing demands. What would be a yield of particular hybrid on particular field? Answer on that question is expected from DFMF algorithm.

2 Basic concepts

2.1 Matrix factorization

Data is the most often represented in the form of matrix. Very powerful mathematical tool for analyzing data that can be expressed as a matrix is matrix factorization. It is used for many problems that arise in data science. For instance, building recommendation systems, dimensionality reduction, clustering, image compression, discovering underlying structures...

Let our data be organized in a form of matrix $X \in \mathbb{R}^{n \times m}$. *Matrix factorization* or matrix decomposition is a factorization of a matrix into a product of matrices. Let us consider matrix two-factorization of X, which is a way of approximating X by a product of two matrices UV^T, where $U \in \mathbb{R}^{n \times k}$ and $V^T \in \mathbb{R}^{k \times m}$. If we denote the rows of X by X_i we know that rows of X can be represented as linear combinations of the rows of V^T, $U_i V^T$. We think of the rows of V^T as latent factors and the entries of U as coefficients of the linear combinations. Another way of seeing this is as like rows of X are approximated by a k-dimensional linear subspace which is spanned by the rows of V^T. On the other hand, each column of X is a linear combination of the columns of U. U and V are called *latent matrices* or *latent factor* matrices.

Matrix X can be *exactly factored* as $\hat{X} = UV^T$ if its rank is at most k and if we do not impose additional constraints. When approximating a matrix X by a matrix $\hat{X}$ we need to have some measure that tells us how good our approximation is and then our task is to make that approximation error as smaller as possible. Depending of the application different measures are used.

Frobenius norm is most widely used measure of discrepancy, both in unconstrained and in constrained factorization, between the original matrix X and it's approximation $\hat{X}$. Frobenius norm of the difference between X and $\hat{X}$ is defined in the following way:

$$||X - \hat{X}||^2_{Fro} = \sum_i \sum_j (X_{ij} - \hat{X}_{ij})^2 \tag{1}$$

It is well known that k-rank matrix $\hat{X}$ which is given by the k leading components of the singular value decomposition gives the best possible approximation in term of Frobenius norm.

In constraint factorization we put additional constraints on the factor matrices. On that way we are removing the degrees of freedom on the factorization

UV^T of a reconstructed $\hat{X}$. We do that in order to ease the interpretation of the factor matrices or in order to reduce the complexity of the model.

Definition 2.1. A matrix factorization algorithm can be defined in the following way:

1. Data weights $W \in \mathbb{R}_+^{m \times n}$ (optional),
2. Prediction link $f : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$,
3. Hard constraints on factors $(U, V) \in \mathcal{C}$,
4. Weight loss between X and $\hat{X} = f(UV^T)$, $\mathcal{D}(X||\hat{X}, W) \geq 0$
5. Regularization penalty, $\mathcal{R}(U, V) > 0$

For the model $X \sim f(UV^T)$ the optimization problem is:

$$\underset{U,V \in \mathcal{C}}{argmin}[\mathcal{D}(X||f(UV^T), W) + \mathcal{R}(U, V)]. \tag{2}$$

where f can be nonlinear.

Example of weighted loss function could be the following:

$$\mathcal{D}_W(X, \hat{X}) = ||W \odot (X - UV^T)||_{Fro}^2$$

where $\odot$ denotes the element-wise product of matrices.

2.2 Matrix completion

A concept which is very similar to matrix factorization is *matrix completion*. The aim of matrix completion is to recover unknown entries from the known ones.

Matrix completion often seeks to find the lowest rank matrix that agrees with the partially observed matrix. Assumptions that are made is that the

sampling of the observed matrix are uniformly at random and on the number of sampled entries. Also there must be known at least one observed entry per row and column.

The most famous example of matrix completion is Netflix problem or the movie-rating problem. Predicting movie rating accurately was so important that Netflix offered million dollars price for the first algorithm that would be better than its own recommendation system by 10%. Given the matrix, entry (i, j) represents the rating of movie j by user i if user i has watched (and rated) movie j and is otherwise missing. The task is to predict missing values in order to make good recommendation in what movie should particular user watch next. The assumption here is that rating matrix is expected to be low-rank since user's preferences can be described with few factors, such as movie genre and actors playing in the movie.
Criterion for evaluating the results was Root Mean Square Error (RMSE):

$$\text{RMSE } = \frac{1}{|R|}\sqrt{\sum_{(i,j)\in R}(\hat{r}_{ij} - r_{ij})^2},$$

where $\hat{r}_{ij}$ is predicted rating and r_{ij} is true rating of user i on film j.
Task it to make good recommendation system. One way of solving this problem is via optimization. As RMSE is the measure that tells us how good our recommendation system is, the main idea is to minimize RMSE in order to get the best recommendation. We want to make good recommendations on movies that people haven't yet seen. Given user by movie matrix R the goal is to represent R as a product of two matrices Q and P, $R \approx QP^T$, where $R \in \mathbb{R}^{n\times m}$, $Q \in \mathbb{R}^{n\times k}$ and $P^T \in \mathbb{R}^{k\times m}$ ($k << n, m$). We can think of every row of Q as k dimensional representation of a given user and every column of P as k dimensional representation of a given movie. The missing rating of user i for movie j is estimated by the dot product of i-th row of Q and j-th column of P^T. This method discovers latent factors (or latent dimensions) in which the user can be mapped according to matrix Q and movies can be mapped accourding to matrix P.
The optimization problem that needs to be solved to get good recommendation is the following:

$$\min_{P,Q} \sum_{(i,j)\in R}(r_{ij} - q_i p_j^T)^2$$

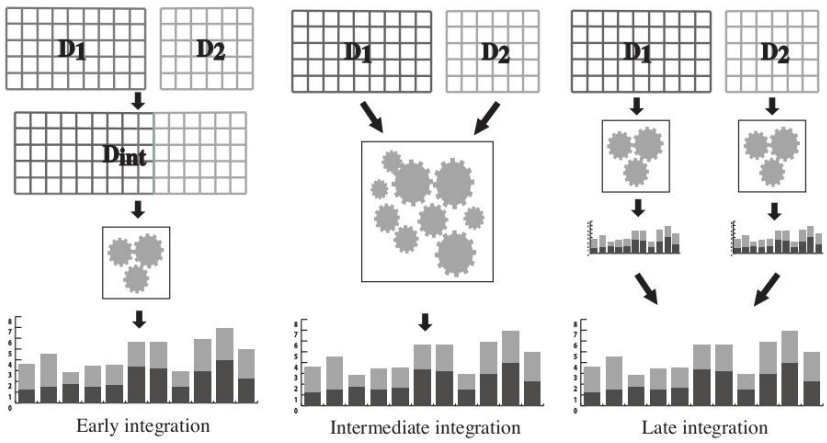
The sum goes over the known entries of matrix R, so there is no need for filling the missing values. Optimization problem can be solved using gradient method.

2.3 Data Fusion

Data fusion is a process in which we want to use multiple data sources and combine them in order to produce information that is more consistent and accurate than that provided by any individual data source. Depending on the processing stage at which fusion takes place, data fusion approaches can be categorized into three categories (Figure 1):

- early (or full) integration,
- late (or decision) integration,
- intermediate (or partial) integration.

In **early integration** all data sets are concatenated into single, feature-based table before applying algorithm. Here transformation of data sets is required which in some cases may cause information loss.
In **late integration** each data set is treated separately. Each data source gives rise to a separate algorithm. By combining the results from all models we get prediction. When treating all data sets separately we are not able to find mutual interactions and that often leads to lower performance of the model.
In **intermediate integration** there is a single model that takes data from different sources by keeping the initial structure. In such a setting there is no need for transformation of input data or only minimal transformation is needed.



Source: M. Žitnik, Learning by fusing heterogenous data, 2015; page: 36

Figure 1: Types of data fusion.

2.3 Data Fusion

Nowadays there are many well-established feature-based machine learning and data mining algorithms for early and late integration, but on the other hand there are only few inference algorithms for partial integration. In the thesis it will be represented a method for intermediate data fusion based on constrained matrix tri-factorization, called Data fusion by matrix factorization (DFMF) algorithm.

DFMF algorithm was introduced for the first time in doctoral thesis of Marinka Žitnik in 2015. She tested the algorithm on few data sets where she showed how powerful algorithm is. One of the tasks were to predict gene function. The results were compared to state-of-the-art multiple kernel learning algorithm and achieved higher accuracy than can be obtain from any single data source considered alone.

Method that was proposed by Wang ([2]) is conceptually similar to method that was proposed by Žitnik. He considerates aslo both inter-type and intra-type relations but requires relations to be symmetric and all relations must be present.

3 Fusing heterogeneous data

3.1 Data fusion by collective matrix factorization

Data fusion by matrix factorization algorithm (DFMF) considers r object types $\varepsilon_1, ..., \varepsilon_r$ and a collection of data sources, each relating a pair of object types $(\varepsilon_i, \varepsilon_j)$. Matrix $R_{ij} \in \mathbb{R}^{n_i \times n_j}$ relates object types ε_i and ε_j, where there are n_i objects of type ε_i and n_j objects of type ε_j. In the movie example object types are movie, user, genre, actor and relations are user's ratings of movie, movies genres, actor's roles in movies (Figure 2). Matrices R_{ij} and R_{ji} don't need to be symmetric. Matrix $Q_i \in \mathbb{R}^{n_i \times n_i}$ is constraint matrix it represents a relation between objects of the same type ε_i. In the movie example there is no such intra-type information as constraints.

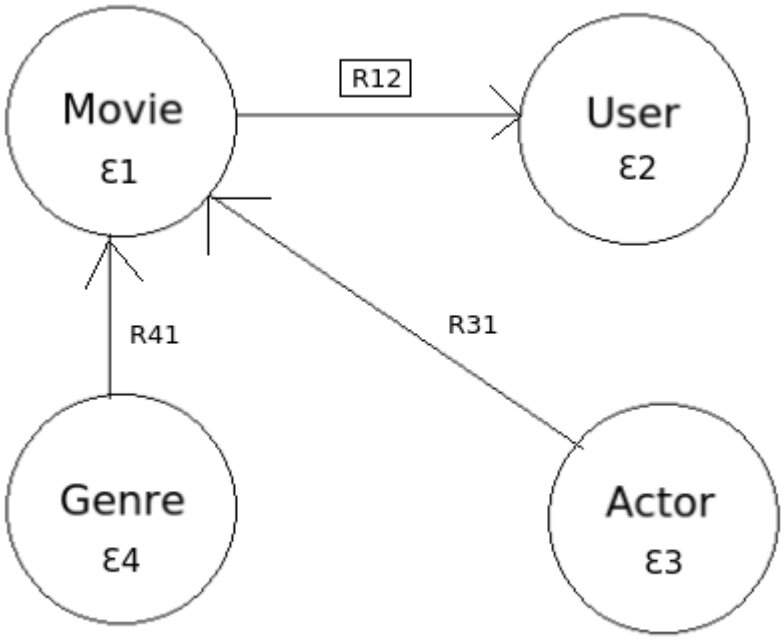
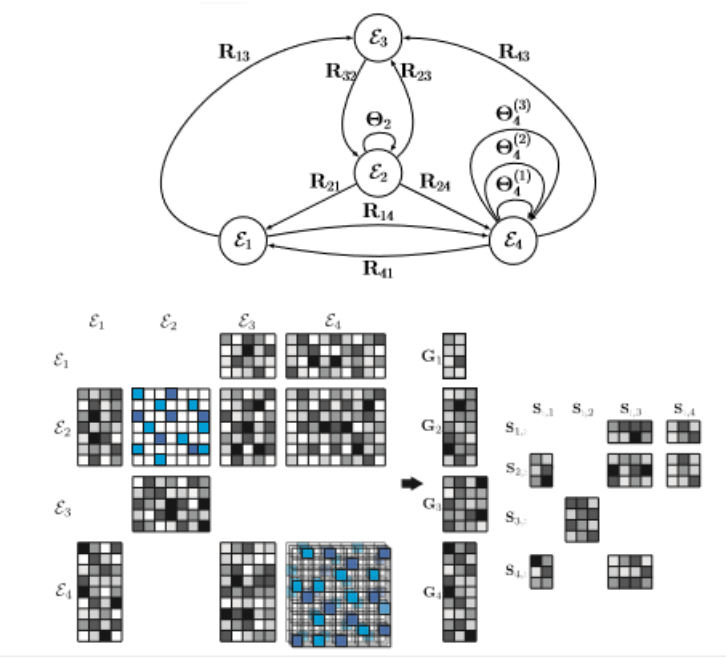


Figure 2: Nodes represent object types and edges correspond to relation and constraint matrices.

For better understanding imagine that we have the scenario from Figure 3. There it is represented with both the graph of relations between object types and the block-based matrix structure. We can see that some relations are completely missing, for instance R_{34}. Constraints are denoted with loops

and in example from Figure 3 are provided for object types ε_2 (one constraint matrix) and ε_4 (three constraint matrices).



Source: M.Žitnik, Learning by fusing heterogeneous data, 2015, page: 135

Figure 3: Graphical representation of fusion configuration with object types $\varepsilon_1, \varepsilon_2, \varepsilon_3$ and ε_4.

Data fusion algorithm represented here works even in the case when not all relations between all pairs of object types are present and that is often happening because in real-world scenarios it is hard to have access to all of relations. The only premise is that underlying graph of relations between object types is connected.

All available relational matrices R_{ij} are simultaneously factorized such that R_{ij} is represented as product of three matrices $G_i \in \mathbb{R}^{n_i \times k_i}$, $G_j^T \in \mathbb{R}^{k_j \times n_j}$ and $S_{ij} \in \mathbb{R}^{k_i \times k_j}$, $R_{ij} \approx G_i S_{ij} G_j^T$, approximation of the matrix R_{ij} is regularized though constraint matrices $Q_i \in \mathbb{R}^{n_i \times n_i}$ and $Q_j \in \mathbb{R}^{n_j \times n_j}$. Entry $R_{ij}(p, q)$ is approximated by an inner product of the p-th row of matrix G_i and linear combination of the columns of matrix S_{ij}, weighted by the q-th column of G_j. The matrix S_{ij} has relatively few vectors in comparison to R_{ij} ($k_i << n_i, k_j << n_j$) and it is used to represent many data vectors. Good approximation lies behind the premise that there exist some correspondence among difference input spaces, that the latent structure is present in the

original data.

Data fusion by matrix factorization algorithm has the following characteristics:

- DFMF can model the multiple relations and multiple object types,
- not all relations between object types have to be present, some can be completely missing,
- for every object type we can have multiple constraint matrices,
- there are no assumptions about structural properties of relations.

The goal is to improve predictive accuracy in inferring relations between two target object types, ε_i and ε_j, in movie example the goal is to infer relation between movie and user and all other object types and relations are there to help us in that task. The relation between two object types ε_i and ε_j is represented by matrix R_{ij} and assumption is that it is $[0, 1]$-matrix. Entries in the target matrix indicate how strong is relation, where 0 denotes no relation and 1 denotes the strongest relation. Entries that we want to predict are reconstructed through matrix factorization.

In the following few subsections you can find factorization model and objective function that are used and also see how updating rules for optimization are derivated.

3.2 Factorization

Input to data fusion by matrix factorization algorithm is a block matrix R, where the matrix in i-th row and j-th column represents the relationship between object types ε_i and ε_j:

$$R = \begin{bmatrix} 0 & R_{12} & \dots & R_{1r} \\ R_{21} & 0 & \dots & R_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ R_{r1} & R_{r2} & \dots & 0 \end{bmatrix} \tag{3}$$

As it mentioned earlier not all relations are required to be present and $R_{ij} \neq R_{ji}$.

Also constrains that relate objects of the same type can be considerate or

in other words intra-type information as constraints. Each object type may have several constraints. Assume that there are t_i data sources for object type ε_i, those data sources are represented by a set of constraint matrices $Q_i^{(t)}$ where $t \in \{1, 2, ..., t_i\}$. Constraints are collectively encoded in a set of constraint block diagonal matrices $Q^{(t)}$ for $t \in \{1, 2, ..., \underset{i}{max} t_i\}$:

$$Q^{(t)} = \begin{bmatrix} Q_1^{(t)} & 0 & \dots & 0 \\ 0 & Q_2^{(t)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q_r^{(t)} \end{bmatrix}$$

The i-th block on the main diagonal of $Q^{(t)}$ is zero if $t > t_i$. There are two type of constraints:

- *must-link constraints* - can be seen as rewards that reduce the optimization function,
- *cannot-link constraints* - impose penalties on the current approximation.

Must-link constraints relate to objects that are similar and entries in constraint matrices are negative for that objects. Entries in constraint matrices are positive for dissimilar objects. These constraints will adapt the objective function to include penalties for violating constraints.

The block matrix R_{ij} is decomposed into matrix factors G and S:

$$G = \begin{bmatrix} G_1^{n_1 \times k_1} & 0 & \dots & 0 \\ 0 & G_2^{n_2 \times k_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & G_r^{n_r \times k_r} \end{bmatrix}, \tag{4}$$

$$S = \begin{bmatrix} 0 & S_{12}^{k_1 \times k_2} & \dots & S_{1r}^{k_1 \times k_r} \\ S_{21}^{k_2 \times k_1} & 0 & \dots & S_{2r}^{k_2 \times k_r} \\ \vdots & \vdots & \ddots & \vdots \\ S_{r1}^{k_r \times k_1} & S_{r2}^{k_r \times k_2} & \dots & 0 \end{bmatrix}.$$

Matrix S has the same block structure as matrix R, if a relation is missing in R it is also missing in its corresponding matrix factor S. And in general it is asymmetric ($S_{ij} \neq S_{ji}$).

To every object type ϵ_i it is assigned a factorization rank k_i. The latent relation between object types ϵ_i and ϵ_j is represented in factor S_{ij} and factor G_i is specific to object type ϵ_i and we use it in reconstructing every relation where this object types is present. In that way, every relation R_{ij} obtains its own factorization $G_i S_{ij} G_j^T$ where factor $G_i(G_j)$ is shared across all relations which involve object types $\epsilon_i(\epsilon_j)$. So our reconstructed block matrix GSG^T can be represented in the following way:

$$\begin{bmatrix} 0 & G_1 S_{12} G_2^T & \dots & G_1 S_{1r} G_r^T \\ G_2 S_{21} G_1^T & 0 & \dots & G_2 S_{2r} G_r^T \\ \vdots & \vdots & \ddots & \vdots \\ G_r S_{r1} G_1^T & G_r S_{r2} G_2^T & \dots & 0 \end{bmatrix}.$$

3.3 Objective function

The goal is to make as good approximation as possible of the input data by adherenting to must-link and cannot-link constraints. The objective function is defined on the following way:

$$\begin{aligned} \text{minimize } J(G,S) = & \sum_{R_{ij} \in \mathcal{R}} ||R_{ij} - G_i S_{ij} G_j^T||_F^2 \\ & + \sum_{t=1}^{maxt_i} tr(G^T Q^{(t)} G) \\ \text{subject to } G \geq 0. & \end{aligned} \tag{5}$$

where $||\cdot||_F$ is Frobenius norm and $tr(\cdot)$ is trace of matrix (sum of main diagonal entries). Sum goes over all present relations, the objective function is constructed such that it allows some relations to be completely missing. On that way we are not forced to replace the missing relations with zero matrices or some other. This enables better optimization, reduction of the value of objective function. In the objective function are also incorporated intra-type information.

The optimization problem defined in equation (5) is solved using Data fusion by matrix factorization (DFMF) algorithm shown in next subsection. The algorithm first initializes matrix G and then iteratively updates matrix factors by alternating between fixing G and updating S and then fixing S and updating G. This is done until convergence or predefined time limit is reached. Updating G and S using the rules given in the algorithm converge to a local minimum of the given problem in equation (5).

The objective function $J(G,S)$ in equation (5) can be expanded as:

$$\begin{aligned} J(G,S) = & \sum_{R_{ij}\in\mathcal{R}} tr(R_{ij}^T R_{ij} - 2G_j^T R_{ij}^T G_i S_{ij} \\ & + G_i^T G_i S_{ij} G_j^T G_j S_{ij}^T) \\ & + \sum_{t=1}^{\underset{i}{max}\, t_i} \sum_{i=1}^{r} tr(G_i^T Q_i^{(t)} G_i). \end{aligned} \tag{6}$$

Multiplicative update rules for regularized decomposition of relation matrices are derived by fixing one matrix and considering the roots of the partial derivative with respect to other matrix factor of Lagrangian function.

The method of Lagrange multipliers is used to find the solution for optimization problems constrained to one or more equalities and (or) inequalities. The problem can be formulated as:

$$\begin{aligned} & x^* = \underset{x}{argmin} f(x) \\ & \text{subject to } h_i = 0, \forall i = 1,2,...,m \\ & \text{subject to } g_i \leq 0, \forall i = 1,2,...,n \end{aligned} \tag{7}$$

The Lagrangian function is than the following:

$$L(x,\lambda,\mu) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{i=1}^{n} \mu_i g_i(x) \tag{8}$$

where λ_i and μ_i are Lagrangian multipliers.
If x^* is a local optimum and functions f, h_i and g_i in (7) are continuously differentiable then there are Lagrange multiplier vectors λ and μ such that the optimization problem satisfies KKT conditions:

1.*Stationarity*

$$\bigtriangledown_x f(x^*) + \sum_{i=1}^{m} \lambda_i \bigtriangledown_x h_i(x^*) + \sum_{i=1}^{n} \mu_i \bigtriangledown_x g_i(x^*) = 0$$

2.*Primal feasibility*

$$\begin{aligned} h_i(x^*) &= 0, \text{ for } \forall i = 1,2,...,m \\ g_i(x^*) &\leq 0, \text{ for } \forall i = 1,2,...,n \end{aligned}$$

3.*Dual feasibility*

$$\mu_i \geq 0, \text{ for } \forall i = 1,2,...,m$$

4. *Complementary slackness*

$$\mu_i g_i(x^*) = 0, \text{ for } \forall i = 1, 2, ..., m$$

Correctness of the algorithm is guaranteed by the following theorem.

Theorem 3.1. *If the update rule of G and S from DFMF algorithm converge, then the final solution satisfies the KKT (Karuch-Kuhn-Tucker) optimality conditions.*

Proof. Following the theory the Lagrangian multipliers $\mu_1, \mu_2, ..., \mu_r$ are introduced and the Lagrangian function is constructed:

$$L = J(G, S) - \sum_{i=1}^{r} tr(\mu_i 1_{n_i \times k_i} G_i^T). \tag{9}$$

Then for i, j such that $R_{ij} \in \mathcal{R}$:

$$\frac{\partial L}{\partial S_{ij}} = -2G_i^T R_{ij} G_j + 2G_i^T G_i S_{ij} S_j^T G_j \tag{10}$$

and for $i = 1, ..., r$

$$\begin{aligned} \frac{\partial L}{\partial G_i} = & \sum_{j:R_{ij} \in \mathcal{R}} (-2R_{ij} G_j S_{ij}^T + 2G_i S_{ij} G_j^T G_j S_{ij}^T) \\ & + \sum_{j:R_{ji} \in \mathcal{R}} (-2R_{ji}^T G_j S_{ji} + 2G_i S_{ji}^T G_j^T G_j S_{ji}) \\ & + \sum_{t=1}^{\substack{max_i \\ i}} 2Q_i^{(t)} G_i - \mu_i 1_{n_i \times k_i} \end{aligned} \tag{11}$$

Fixing $G_1, G_2, ..., G_r$ and letting $\frac{\partial L}{\partial S_{ij}} = 0$ for all $i, j = 1, 2, ..., r$, we get the following:

$$S = (G^T G)^{-1} G^T R G (G^T G)^{-1} \tag{12}$$

Fixing S and let $\frac{\partial L}{\partial G_i} = 0$ for $i = 1, 2, ..., r$. Next we get an expression for the KKT multiplier μ_i from equations (10) and (11). The KKT complementary slackness condition for the nonnegativity of G_i is:

$$\begin{aligned} 0 = & \mu_i 1_{n_i \times k_i} \circ G_i = \\ = & \Bigg[\sum_{j:R_{ij} \in \mathcal{R}} (-2R_{ij} G_j S_{ij}^T + 2G_i S_{ij} G_j^T G_j S_{ij}^T) \\ & + \sum_{j:R_{ji} \in \mathcal{R}} (-2R_{ji}^T G_j S_{ji} + 2G_i S_{ji}^T G_j^T G_j S_{ji}) \\ & + \sum_{t=1}^{\substack{max_i \\ i}} 2Q_i^{(t)} G_i \Bigg] \circ G_i \end{aligned} \tag{13}$$

Notion $\circ$ is Hadamard product. The Hadamard product is binary operation that takes two matrices of the same dimensions, and produces another matrix where each element (i,j) is the product of elements (i,j) of the original two matrices. Let with Γ_i denote $\Gamma_i = \mu_i \circ G_i$. Equation (13) is a fixed point equation that the solution must satisty at convergence. Therefore, let:

$$\begin{aligned} Q_i^{(t)} &= [Q_i^{(t)}]^+ - [Q_i^{(t)}]^- \\ R_{ij}G_jS_{ij}^T &= (R_{ij}G_jS_{ij}^T)^+ - (R_{ij}G_jS_{ij}^T)^- \\ S_{ij}G_j^TG_jS_{ij}^T &= (S_{ij}G_j^TG_jS_{ij}^T)^+ - (S_{ij}G_j^TG_jS_{ij}^T)^- \\ R_{ji}^TG_jS_{ji} &= (R_{ji}^TG_jS_{ji})^+ - (R_{ji}^TG_jS_{ji})^- \\ S_{ji}^TG_j^TG_jS_{ji} &= (S_{ji}^TG_j^TG_jS_{ji})^+ - (S_{ji}^TG_j^TG_jS_{ji})^- \end{aligned}$$

all matrices on the right-hand side are nonnegative. By initial guess of G_i and the successive updates of G_i using equations (14)-(16) in DFMF algorithm we converge to a local minimum of the objective function given in equaition (4). Using such a rule, at convergence, G_i satisfies $\Gamma_i \circ G_i = 0$. As G_i is nonnegative we have that $\Gamma_i = 0$ □

Theorem 3.2. *(Convergence of Data fusion by matrix factorization algorithm): The objective function $J(G,S)$ given by equation (5) is nonincreasing under the updating rules for matrix factors G and S in Algorithm given in the next subsection.*

The proof can be found in [1].

Function from equation (5) is noncovex and thus it has multiple local minima. The global minimum of multi-relational system remains unreachable. However DFMF algorithm converges to a local minimum of equation (5).

Our goal is to infer relation betweeen two target objects ϵ_i and ϵ_j. Therefore the stopping criterion only includes the target matrix R_{ij}. The convergence criteria is the following:

$$||R_{ij} - G_iS_{ij}G_j^T||^2 < \epsilon,$$

where ϵ is user defined parameter. In experiment presented in this thesis it is set to 10^{-5}. To reduce time needed for computation, the convergence criteria is considerate in every fifth iteration.

3.4 DFMF Algorithm

Input:

- a set $\mathcal{R}$ of relation matrices R_{ij},
- constraint matrices $Q^{(t)}$ for $t \in \{1, 2, ..., max_i t_i\}$,
- factorization ranks $k_1, k_2, ..., k_r$.

Output:

- matrix factors S and G

I Initialize G_i for $i = 1, 2, ..., r$.

II Repeat until convergence or a time limit is reached:

1. Construct R and G using definitions in equation (3) and (4) .

2. Update S using:

$$S \leftarrow (G^T G)^{(-1)} G^T R G (G^T G)^{(-1)}$$

3. Set $G_i^{(e)} \leftarrow 0$ for $i = 1, 2, ..., r$..

4. Set $G_j^{(d)} \leftarrow 0$ for $i = 1, 2, ..., r$..

5. For $R_{ij} \in \mathcal{R}$:

$$\begin{aligned} G_i^{(e)} &+= (R_{ij} G_j S_{ij}^T)^+ + G_i (S_{ij} G_j^T G_j S_{ij}^T)^- \\ G_i^{(d)} &+= (R_{ij} G_j S_{ij}^T)^- + G_i (S_{ij} G_j^T G_j S_{ij}^T)^+ \\ G_j^{(e)} &+= (R_{ij}^T G_i S_{ij})^+ + G_j (S_{ij}^T G_i^T G_i S_{ij})^- \\ G_j^{(d)} &+= (R_{ij}^T G_i S_{ij})^- + G_j (S_{ij}^T G_i^T G_i S_{ij})^- \end{aligned} \tag{14}$$

6. For $t = 1, 2, ..., \underset{i}{max} t_i$:

$$\begin{aligned} G_i^{(e)} &+= [Q_i^{(t)}]^- G_i \text{ for } i = 1, 2, ..., r \\ G_i^{(d)} &+= [Q_i^{(t)}]^+ G_i \text{ for } i = 1, 2, ..., r \end{aligned} \tag{15}$$

7. Construct G as:

$$G \leftarrow G \circ Diag(\sqrt{\frac{G_1^{(e)}}{G_1^{(d)}}}, \sqrt{\frac{G_2^{(e)}}{G_2^{(d)}}}, ..., \sqrt{\frac{G_r^{(e)}}{G_r^{(d)}}}), \tag{16}$$

where $\circ$ is Hadamard product. The $\sqrt{\cdot}$ and $\div$ are entry-wise operations. $X^+(p,q)$ is defined as $X(p,q)$ if $X(p,q) \geq 0$ else it is 0 and $X^-(p,q)$ is $-X^-(p,q)$ if $X(p,q) \leq 0$ else it is set to 0. Therefore, both X^+ and X^- are nonnegative matrices.

Factorization model represented here is quite sensitive to the initialization of matrix G. The proper initialization is thus very important and reduces the number of iterations needed to obtain matrix factors of equal quality. G is initialized by separately initializing every matrix G_i. Matrix S doesn't have to be initialized as it is computed from G. Some authors used random initialization, but random Acol initialization was shown to be better option. In random Acol columns of G_i are computed as an element-wise average of a random subset of columns in R_{ij}.

Input to Data fusion by matrix factorization algorithm are facotorization ranks $k_1, k_2, ..., k_r$. The algorithm is sensitive to ranks so it is imporatant to choose the ones that will minimize the error. In our corn and soybean example we tried different ranks and picked the onces that gave us the best result in terms of RMSE error and coefficient of determination.

In the next two sections it is represented experimental part of the thesis. You can see the application of DFMF algorithm on maize and soybean yield prediction.

4 Corn yield prediction

4.1 Data description

The data in this thesis comes from Syngenta Crop Challenge. The aim of Syngenta Crop Challenge is to solve problems in agriculture throughout data analytics. The 2018 challenge was the third one and it focused on developing a quantitative framework for predicting maize hybrid performance in new, untested locations. On the challenge hybrid performance is measured as the yield difference over a competitive benchmark, known as the check yield, but in this thesis hybrid performance is measured as yield.

The data contains three separate datasets: the performance dataset, the environment dataset and the genetic dataset (Table 1). The performance dataset consists of hybrid name, year, location information (latitude, longitude and location ID), yield, check yield, yield difference (difference between yield and check yield) and maturity group which is value assigned to hybrids based on how many days that particular hybrid takes to become mature. Performance data of various hybrids is provided from 2008 to 2016. The environment dataset contains the recorded weather and soil conditions for selected growing region. Soil and weather attribute names were masked and their value was scaled which made job even harder. In the environment dataset there are 8 soil attributes (s1-s8) and 72 weather attributes, 6 characteristics for every month. Weather data is provided from 2001 to 2016 and the soil attributes doesn't change much over the years so they are the same for that period. The genetic dataset provides genetic information for 2.267 experimental hybrids. It contains information for 19.465 unique genetic markers.

Table 1: List of features

Performance Dataset	Hybrid name Year Location (Latitude, Longitude, ID) Yield Check Yield Yield difference Maturity group
Enviromental Dataset	Location (Latitude, Longitude, ID) s1-s8 (soil attributes) Year wij (characteristic i in month j)
Genetic Dataset	Hybrid name Genetic markers

4.2 Preprocessing stage

In the preprocessing phase there were detected multiple entries with the same latitude and longitude value but with a different ID, as each latitude and longitude combination is represented by a unique location ID, duplicates were removed and on the end there were 2.238 locations (fields). Most of the farms are located in the American Midwest. Figure 1 shows where those farms are.

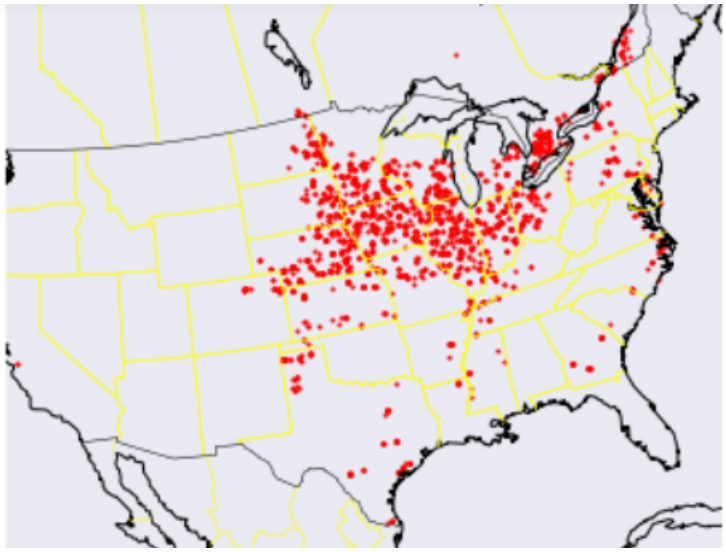


Figure 4: Red dots are the region mostly in the United States where farms are located.

4.2 Preprocessing stage

By analyzing correlation between yield and individual features we get the most straightforward approach about the problem complexity. Correlation measures the strength of association between two variables and the direction of the relationship. In terms of the strength of relationship, the value of the correlation coefficient varies between -1 and $+1$. A value of ± 1 indicates a perfect degree of association between the two variables. As the correlation coefficient value goes towards 0, the relationship between the two variables will be weaker. The direction of the relationship is indicated by the sign of the coefficient; a $+$ sign indicates a positive relationship and a $-$ sign indicates a negative relationship. The Figure 5 shows Pearson's correlation between attributes.
Pearson correlation coefficient is calculated on the following way:

$$\rho(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y},$$

where $cov(\cdot,\cdot)$ is the covariance, $\sigma_X(\sigma_Y)$ is the standard deviation of $X(Y)$, $\mu_X(\mu_Y)$ is mean of $X(Y)$ and E is expectation.

Pearson's correlation is the most often used correlation coefficient, but there is aslo Spearman's correlation.
Spearman's correlation is calculated:

$$r_s(X,Y) = 1 - \frac{6\sum_{i=1}^{N} d_i}{N(N^2-1)},$$

where N denotes the total number of samples and d_i denotes the difference between ranks of predicted and real values at position i.

4.2 Preprocessing stage

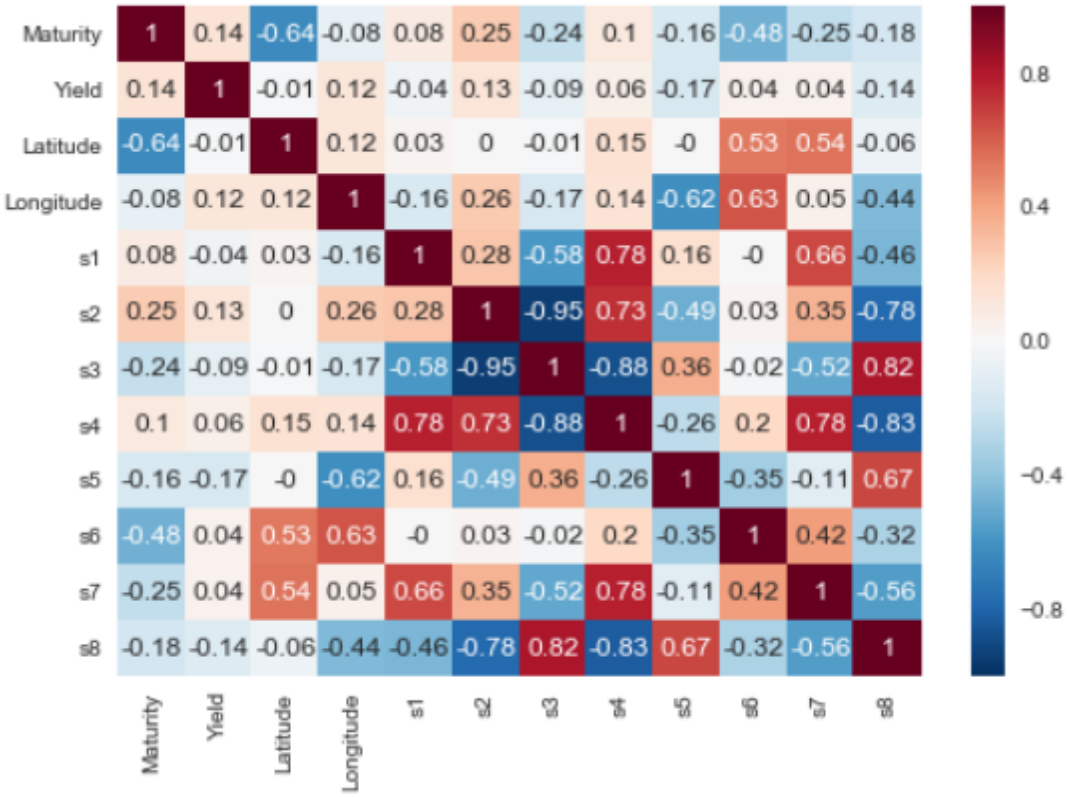


Figure 5: Pearson's correlation between attributes.

There were no significant neither positive nor negative correlation between yield and soil features and between yield and latitude and longitude. Also by analyzing Pearson's correlation between yield and weather attributes there were no any strong relationship. Also Spearman correlation didn't give us any significant information. It is their complex interaction that should be uncovered in order to get good yield prediction.

One can see from the Figure 6 that distribution is far from uniform regarding amount of data that is presented per year, some years are richer with the data.

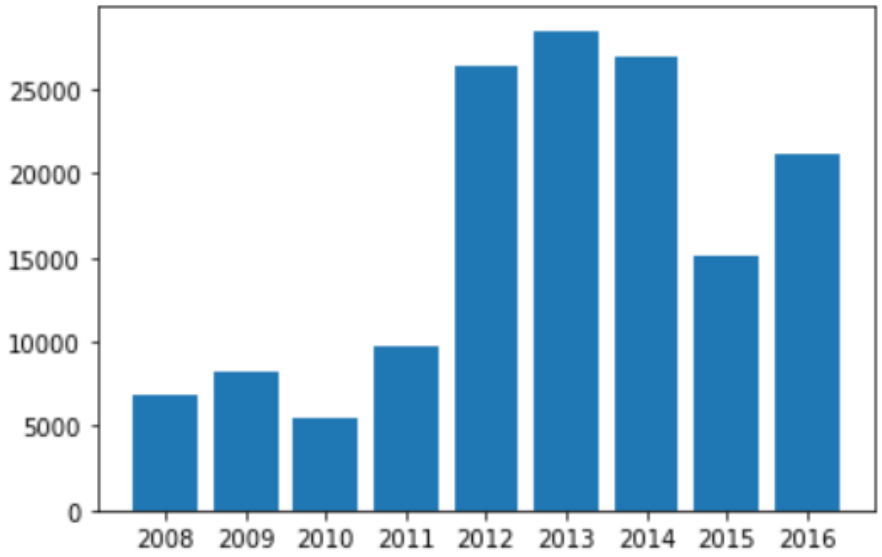


Figure 6: Number of crops per year.

On the other hand it doesn't mean that percentage of unknown values is smaller for those years that have more data as not all years have all hybrids on all locations present, so Figure 7 gives us better picture. Figure 7 shows percentage of unknown values per year.

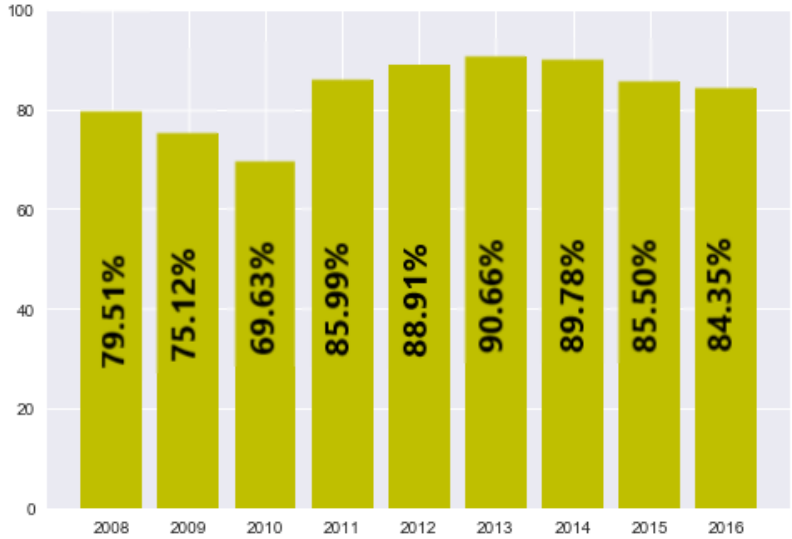


Figure 7: Percentage of unknown values per year

4.2 Preprocessing stage

In 2008 we have 163 hybrids planted on 137 locations and on 2014 we have 486 hybrids planted on 358 fields. Number of planted corn varieties (red) and number of fields (blue) occurring per year is shown in Figure 8.

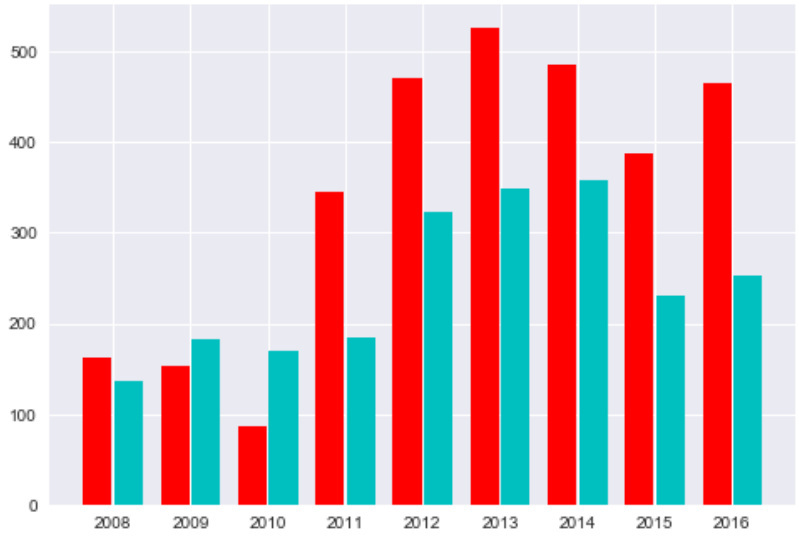


Figure 8: Number of maize varieties (red) and number on fields (blue) per year.

In Figure 9 and 10 it is shown the performance (yield) of each hybrid planted in year 2012 and in 2010. The minimum yield in 2012 was 20.09 bushels per acre and the maximum yield was 238.67 bushels per acre. In 2010 minimum was 20.93 and maximum 228.36 bushels per acre. Such drastic difference in yield is present for all other years also. Various reasons may be responsable for that. Amount of rain and temperature in growing season, quantity and type of fertilizers and persticide used, soil pH value, water holding capacity, electric conductivity, organic carbon content in soil, hybrid planted and many other. This great deviation even within one year stresses the difficulty in predicting performance.

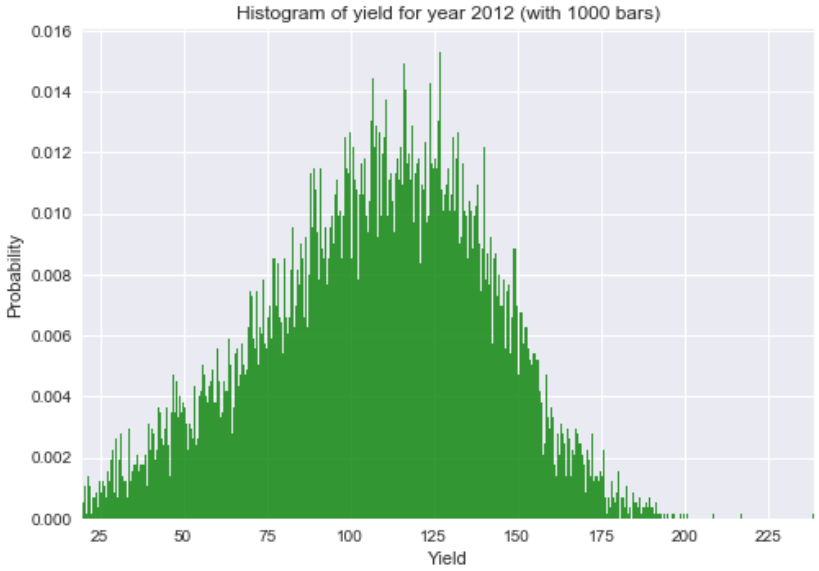


Figure 9: Yield in year 2012.

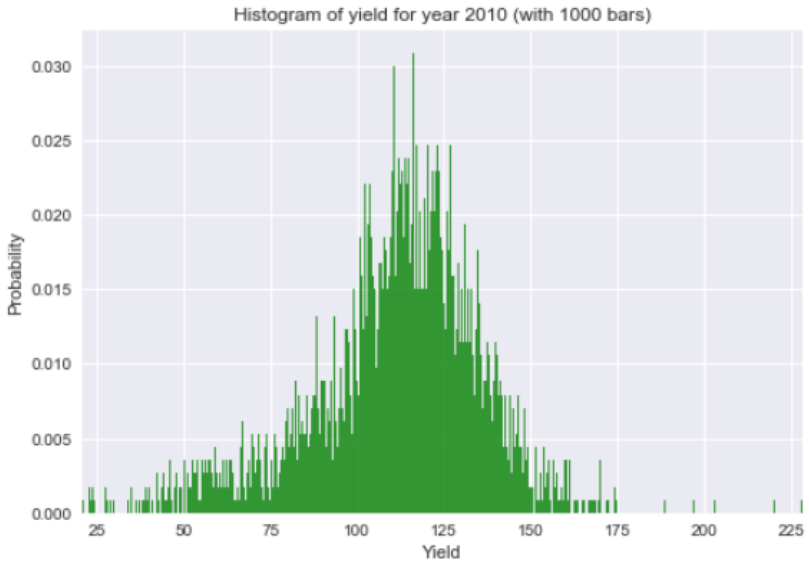


Figure 10: Yield in year 2010.

4.3 Results

In data set there were nine years present, from 2008 to 2016. Every year needs to be analized separately and we decided to choose two for analizing

in this thesis: 2010 and 2012. The reason for choosing these two years was because of their diversity where we wanted to check performance of DFMF algorithm in different settings. In 2010 there were $69,62\%$ of unknown values and 87 hybrids planted on 170 locations and in 2012 there were $88,90\%$ of missing values and 471 hybrids planted on 322 fields. Mean yield in 2010 was 111.35 and in 2012 was 107.58 bushels per acre.
On Figure 11 it visualized how many times each of the hybrids appeared in year 2010 (left) and 2012 (right), one can see that some of the hybrids are way more planted than the others.

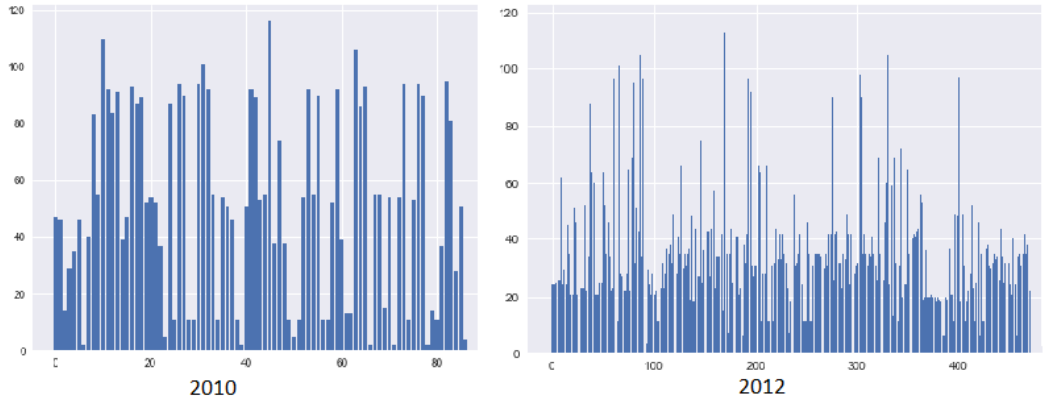


Figure 11: Distribution of hybrids.

On Figure 12 it visualized how many times each of the fields appeared in year 2010 and 2012.

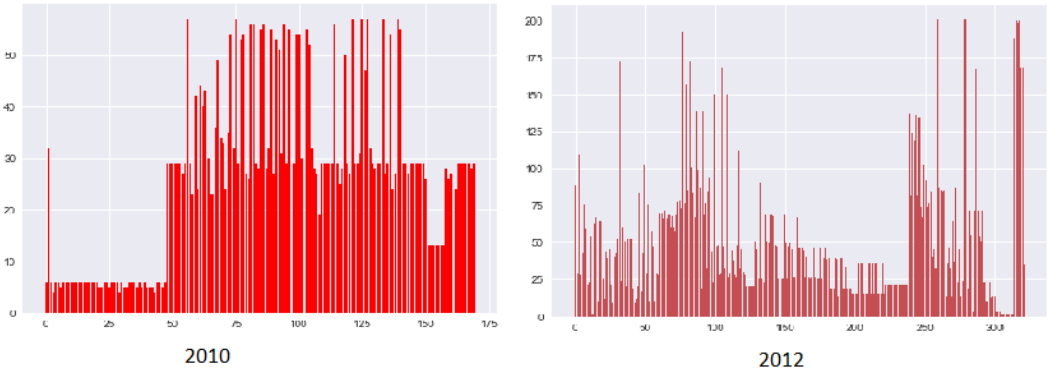


Figure 12: Distribution of fields.

DFMF will be applied on given data. For measuring the results of the model it will be used root mean square error (RMSE) and R squared (R^2).

RMSE is a frequently used measure of the differences between values predicted by a model and orginal values.
Root mean square error:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}$$

where y_i is orginal value and $\hat{y}_i$ is value predicted by model and N is the number of samples in test set.

R squared (coefficient of determination) is the proportion of the variance in the dependent variable that is predictable from the independent variables. If a data set has N values $y_1, y_2, ..., y_N$ and values predicted by model are denoted with $\hat{y}_1, \hat{y}_2, ..., \hat{y}_N$ respectively, then residuals are $e_i = y_i - \hat{y}_i$.
The coefficient of determination is defined:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

where

$$SS_{res} = \sum_i (y_i - \hat{y}_i)^2 = \sum_i e_i^2$$

is the residual sum of squares,

$$SS_{tot} = \sum_i (y_i - \bar{y}_i)^2$$

is the total sum of squares and

$$\bar{y}_i = \frac{1}{N}\sum_{i=1}^{N} y_i$$

is the mean of observed data. Values of coefficient of determination can be negative and the best possible value is 1.

On the Figure 13 are shown object types and their relations. Four object types are present: Hybrid (ϵ_1), Field (ϵ_2), Soil (ϵ_3) and Weather (ϵ_4). Genetic data set could not be used as it contains missing values and DFMF algorithm is not supporting relations with missing entries on matrices other than target matrix. Task is to complete relation R_{12}, between Hybrid (object type ϵ_1) and Field (object type ϵ_2).

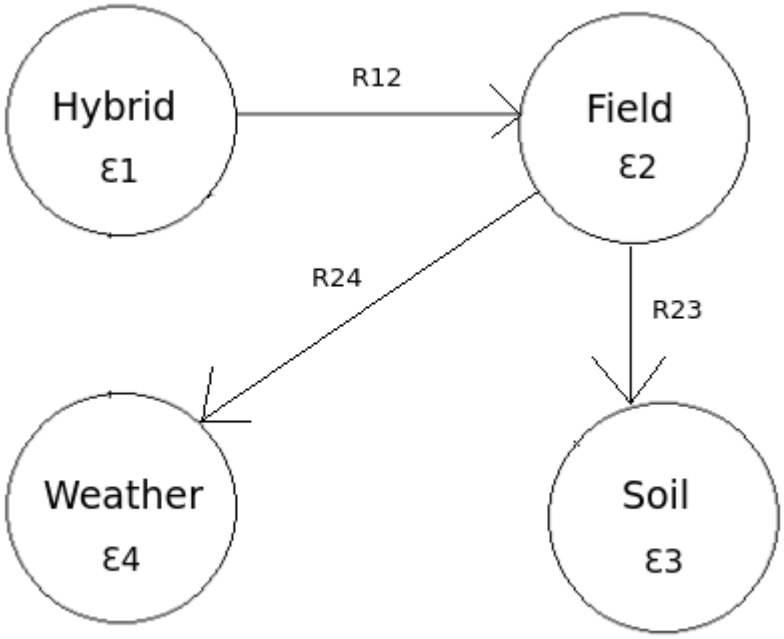


Figure 13: The fusion configuartion.

R_{12} is the target matrix, the one which we need to complete using the known entries from R_{12} relation and all other present relations. Rows of R_{12} are hybrids planted in the particular year and columns are locations present in that particular year that we analyze. Rows of R_{23} and R_{24} are same like columns of R_{12}, columns of R_{23} are the soil features and columns of R_{24} are weather attributes present in data set.

Whole data set R_{12} is divided into two subsets, training and test set. Training set contains around 90% of known values and the rest is take to be for testing the model.

Firstly only two object types will be observed, Hybrid and Field. We will try to reconstuct missing values in the matrix just by values that are known. Table 2 shows results that we got from DFMF algorithm for year 2010 and 2012.

Table 2: Results per year with objects Hybrid and Field

Year	RMSE	R^2
2010	19.78	0.4523
2012	21.28	0.5275

When we include Soil object type, which contains information about soil (8 attributes) and latitude and longitude values and Weather object type which contain weather information present in the data set we get results that are shown in Table 3.

Table 3: Results per year with objects Hybrid, Field, Soil and Weather

Year	RMSE	R^2
2010	15.24	0.6131
2012	19.97	0.5907

We can see that after adding information which are meaningful in predicting yield our results improved. RMSE got smaller and R squared higher as we expected to happen.

Ranks that we used for year 2010: Hybrid: 10, Field: 17, Soil: 6, Weather: 6 and for 2012: Hybrid: 47, Field: 32, Soil: 6, Weather: 5.

5 Soybean yield prediction

5.1 Soybean data description

Data about soybean came in form of one matrix, but we split it in three data sets (matrices) such that we could create objects same like in corn example (Figure 13). In Table 4 you can see list of features and their explanations.

Table 4: List of features

Performance Dataset	Year	Year in which hybrid was planted
	Location ID	Every field has unique ID
	Hybrid	
	Yield	Amount of grain per unit of land
Soil Dataset	Location ID	
	Soil class	Soil class category
	CEC	Cation exchange capacity
	Organic matter	Percentage of soil made up of organic matter
	Clay	Percentage of clay in soil
	Silt	Percentage of silt in soil
	Sand	Percentage of sand in soil
	Area	Propability of growing soybeans in the subregion
	PI	Soil productivity index
	Ph	log of H+ concentration in soil
Weather Dataset	Year	
	Location(ID, Latitude, Longitude)	
	Temperature	Sum of daily temperatures in C° in growing season
	Precipitation	Sum of daily precipitations in mm in growing season
	Solar radiation	Sum of daily solar radiation in W/m^2 in growing season

In this data set there were 174 soybean varieties and their performance is measured between 2009 and 2015 at 205 fields.
Figure 14 shows number of crops and Figure 15 shows percentage of unknown values per year.

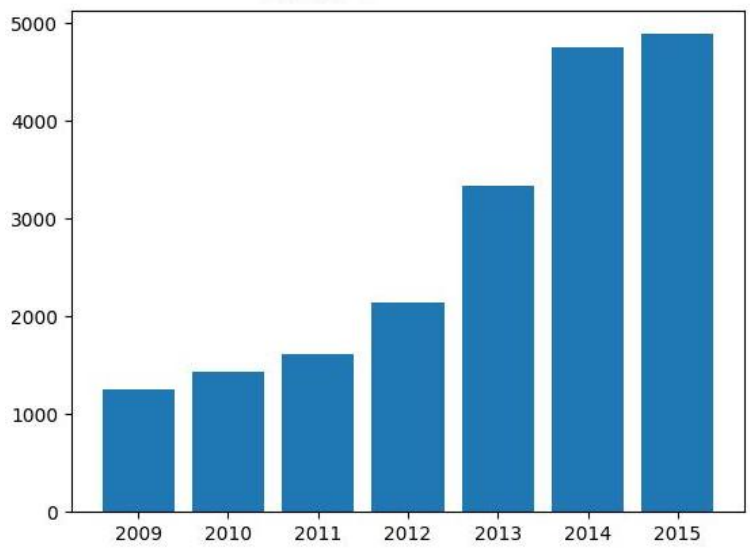


Figure 14: Number of crops per year.

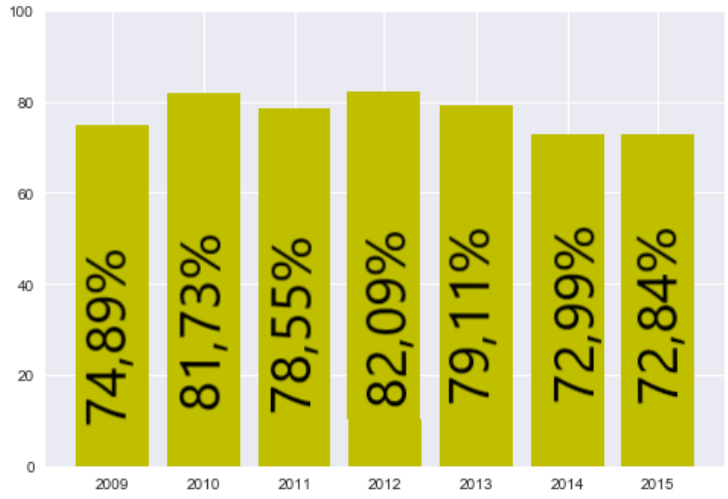


Figure 15: Percentage of unknown values per year.

On Figure 16 green dots represent farm locations.

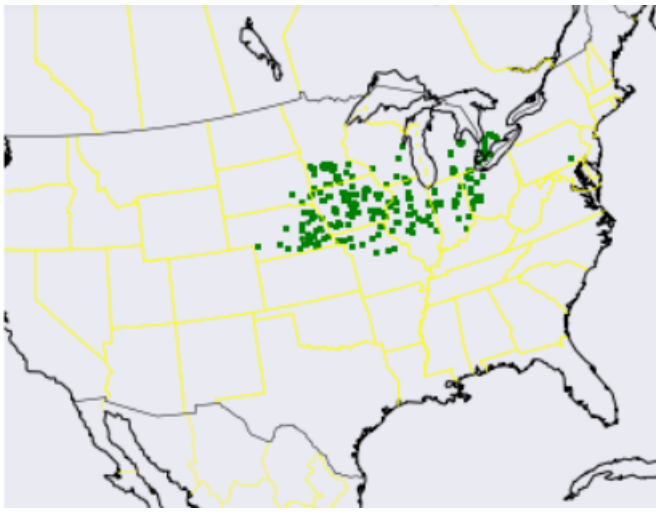


Figure 16: Green dots represent where farms are located.

5.2 Results

The analysis is conducted on year 2014 and 2015. Here we have chosen these two years because of their similarity where we wanted to see how DFMF algorithm is functioning in similar settings. In 2014 there were 123 hybrids planted on 143 locations and in 2015 there were 114 hybrids planted on 148 farms. Soybean yield is not so drastically changing within one year as we saw happening at corn. Minimum yield was 21.89 and maximum was 110.47 in 2015. Figure 17 is representing histogram of yield for 2014 and Figure 18 for year 2015.

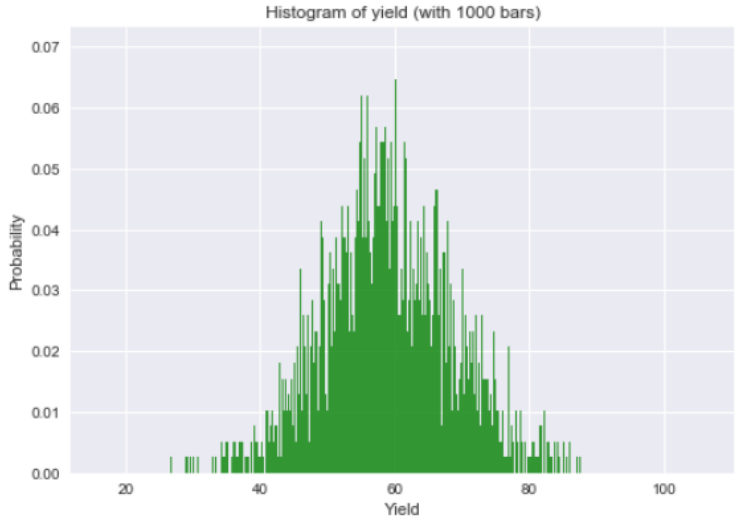


Figure 17: Soybean yield for 2014.

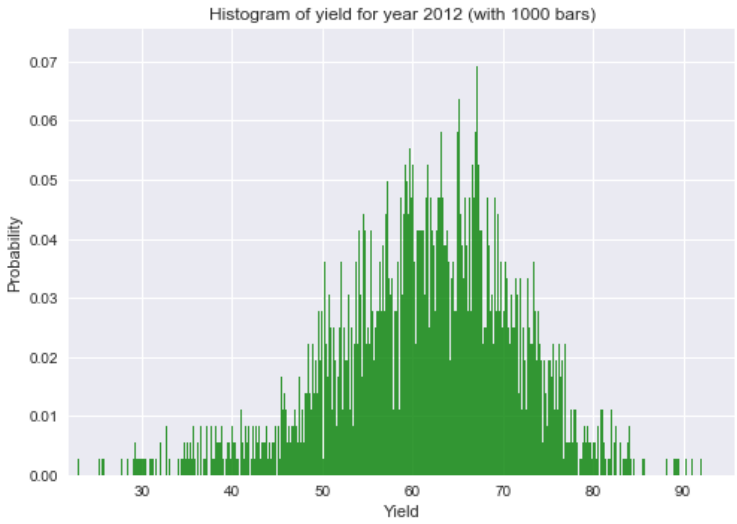


Figure 18: Soybean yield for 2015.

Again same like in corn example we have that distribution of hybrids and fields is not close to uniform. Figure 19 shows the distribution for year 2014 and Figure 20 shows the distribution for year 2015.

5.2 Results

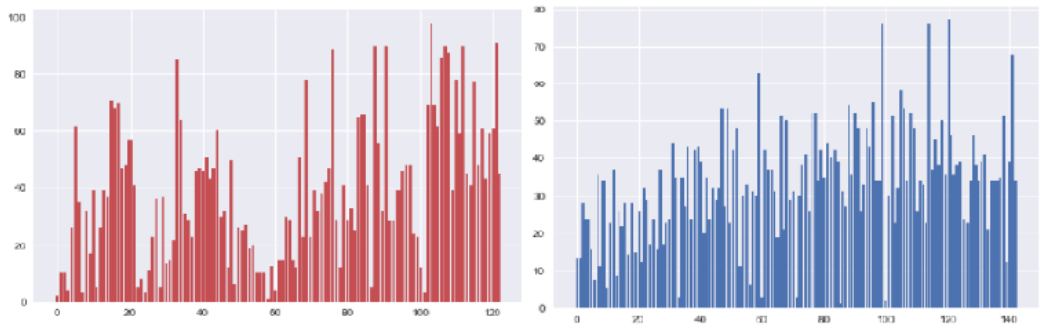


Figure 19: Distribution of hybrids (left) and fields (right) for year 2014.

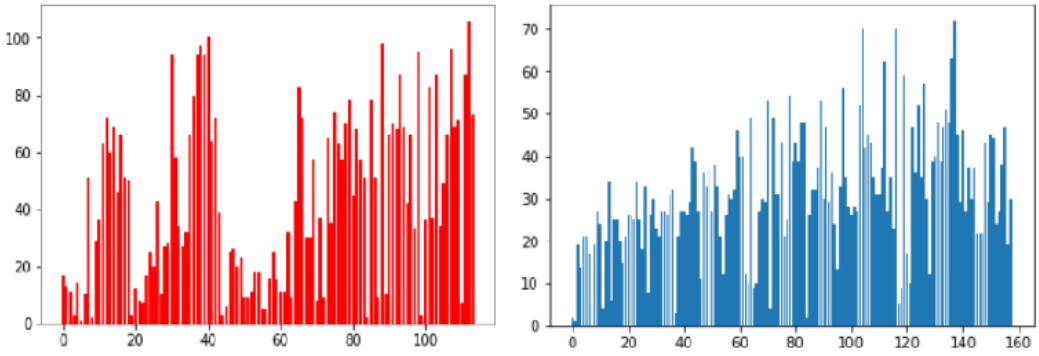


Figure 20: Distribution of hybrids (left) and fields (right) for year 2015.

By analyzing Pearson's correlation only on data from year 2015 on Figure 21 we can see that there is no significant correlation between yield and other soil and weather attributes. Figure 22 shows us Spearman's correlation where we also cannot find any strong relation between yield and other features. For year 2014 also we don't get any strong correlation regarding yield in comparison to other features.

5.2 Results

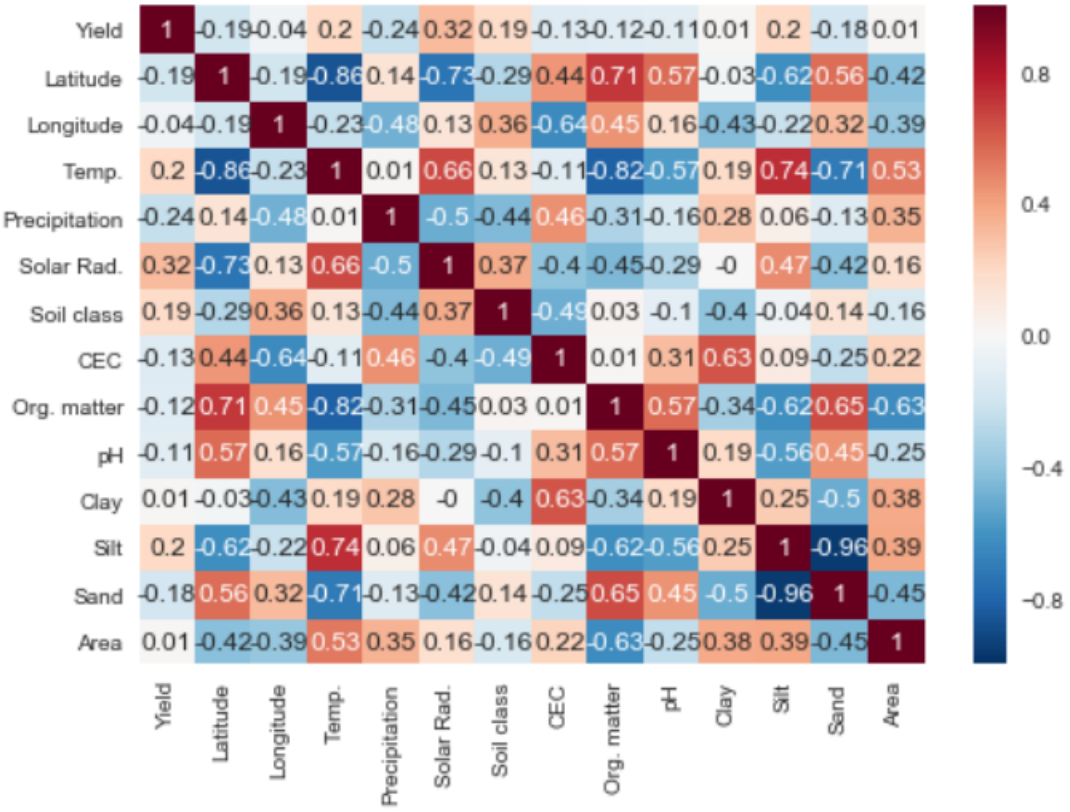


Figure 21: Pearson's correlation between attributes.

5.2 Results

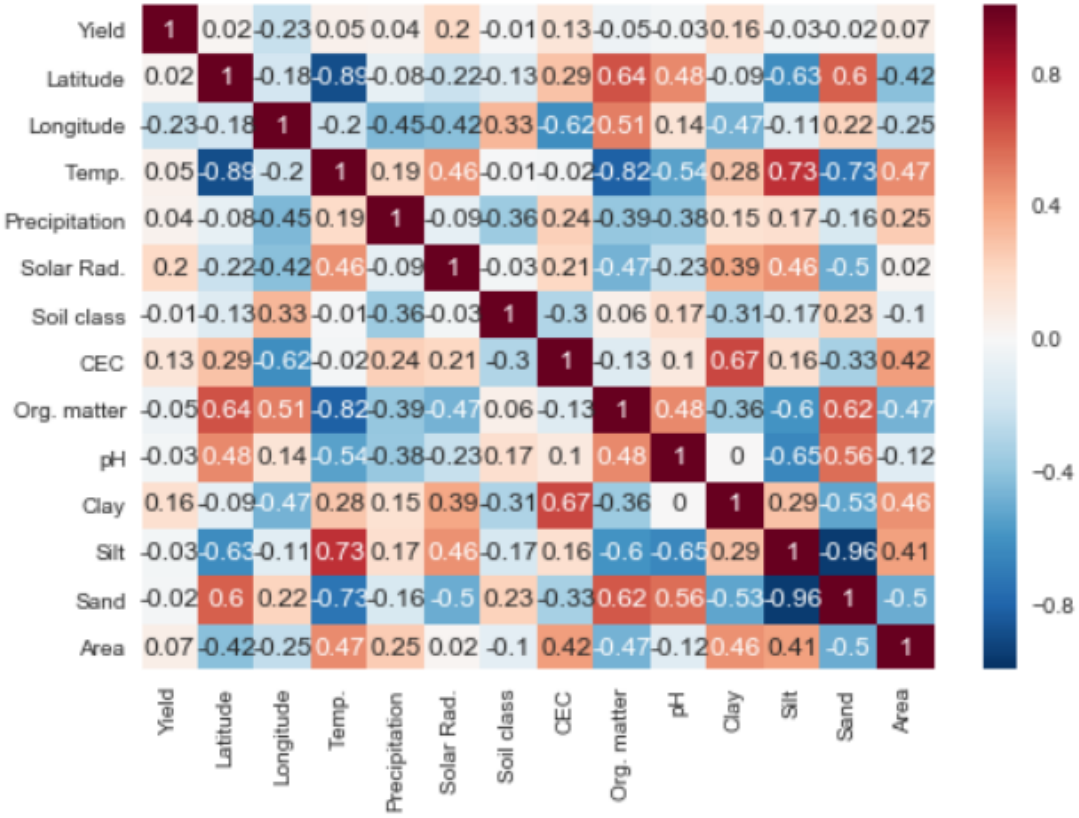


Figure 22: Spearman's correlation between attributes.

Performance of Data fusion by matrix factorization (DFMF) algorithm is measured by using root mean squared error (RMSE) and coefficient of determination (R squared) as in corn example. Table 5 shows the results that we got by using DFMF algorithm just on two objects: Hybrid and Field. Table 6 shows the results when we incorporated weather and soil information to help us predict yield.

We have matrix R_{12} (which connects objects ϵ_1 and ϵ_2) whose rows are hybrids and columns are fields present in particular year, matrix R_{23} (which connects objects ϵ_2 and ϵ_3) whose rows are fields and coulmns are soil attibutes and matrix R_{24} (connects objects ϵ_2 and ϵ_4) whose rows are fileds and columns are weather attributes.

Rank values which gave us the best result both for year 2014 and 2015 were: 20 for Hybrid, 18 for Field, 6 for Soil and 5 for Weather.

Table 5: Results for 2014 and 2015 with objects Hybrid and Field

Year	RMSE	R^2
2014	5.15	0.7178
2015	5.51	0.7428

Table 6: Results for 2014 and 2015 with objects Hybrid, Field, Weather and Soil

Year	RMSE	R^2
2014	4.89	0.7450
2015	4.63	0.7886

Figure 20 shows the histogram of real soybean yield values for year 2015 and Figure 21 shows the histogram of predicted soybean yield values for year 2015.

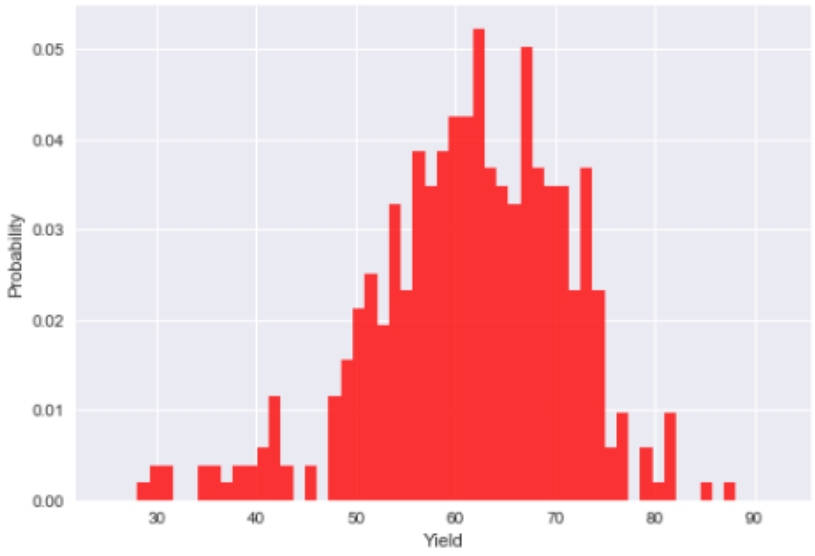


Figure 23: Histogram of real soybean yield for year 2015.

5.2 Results

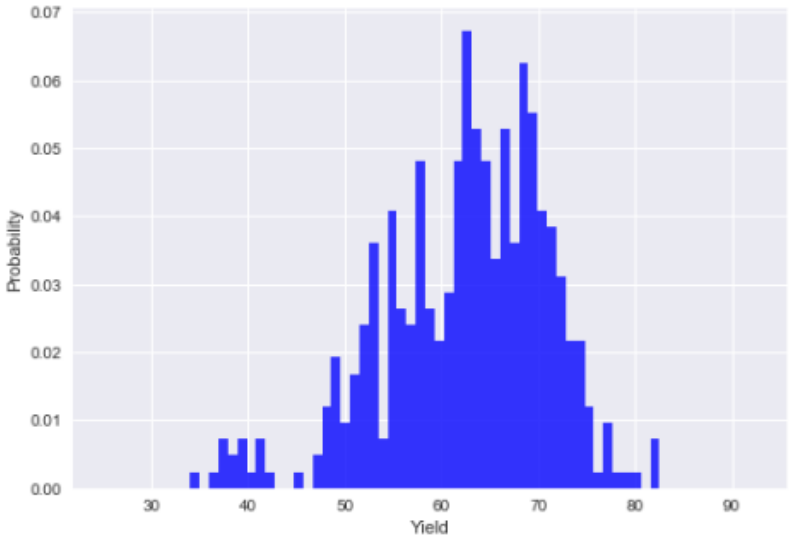


Figure 24: Histogram of predicted soybean yield for year 2015.

Mean value is 61.47 and standard deviation is 9.69 for real soybean data from test set for year 2015 and for predicted values mean is 62.51 and standard deviation is 8.48.

Conclusion

Today various machine learning algorithms are used to predict crop yield for the next season. Often problem is lack of historical data to feed the model. Because of the limited number of locations on which breeders can plant hybrids, as well as experimental and time costs, it is impossible to have information on the performance of all hybrids on all fields. In this thesis we tried to enrich historical data sets about performance of different corn and soybean hybrids.

Data fusion by matrix factorization was extensively used in the field of medicine and bioinformatics and proved to be very successful. Problems coming from agriculture can also be tackled with such approach. That was the reason to test the algorithm in maize and soybean yield prediction.

The main ambition underlying experimental part was to evaluate performance in predicting yields of varieties on the locations where breeding company did not run tests. Successful accomplishment of that task would enable us to enrich data set and to develop better yield predictions models for the next season. For the purpose of evaluation we mask part of the available data to compare it with values predicted by the algorithm.

The algorithm accomplish good results on soybean and moderate results for maize yield prediction. The reason for this can be in the sparseness of data sets. Better results were obtained for soybean yield prediction where percentage of missing values was smaller. In maize data there were a lot of missing values and furthermore we had a high number of different hybrids which rarely appeared and locations where only small number of hybrids are planted. From the previous two histograms we can see that algorithm has difficulties in predicting very low and high yields. The reason for this extreme values can be extensive use of fertilizers or some weather disasters and as that information is not present in our data set, it is not surprising that used algorithm missed to predict them.

Fusing more data sources (yields, soil, weather) through joint factorization increased accuracy of algorithm in predicting both, soybean and maize yields. Appropriate usage of genetic data that is available for maize could further improve the results. More detailed information about weather in soybean data would result in more accurate prediction. Also it would be very useful to have data about irrigation, fertilizers and pesticides used, as that information is important in yield prediction.

Obtained results are promising and a new challenges are identified. Our future work will include extensive experiments with factorization parameters (ranks and initializations), adding more data from external sources regarding weather and vegetation indexes, comparison of obtained results with some other machine learning models. Final goal is to use obtained results from DFMF algorithm and try to estimate crop yield for next season.

References

[1] M. Žitnik: Learning by fusing heterogeneous data, 2015.

[2] F. Wang, T. Li, C. Zhang: Semi-supervised clustering via matrix factorization, 2008.

[3] M. Žitnik, B. Župan: Data fusion by matrix factorization, 2013.

[4] A. Singh, G. Gordon: Relational Learning via Collective Matrix Factorization, 2008.

[5] D. Greene, P. Cunningham: A Matrix Factorization Approach for Integrating Multiple Data Views, 2009.

[6] J. Leskovec, A. Rajaraman, J. D. Ullman: Mining of massive datasets, 2014.

[7] L. Trefethen, D. Bau: Numerical linear algebra, 1997.

[8] M. J. Zaki, W. Meira: Data Mining and Analysis: Fundamental Concepts and Algorithms, 2014.

[9] W. McKinney: Python for Data Analysis, 2011.

[10] O. Marko, S. Brdar, M. Panić, P. Lugonja, V. Crnojević: Soybean varieties portfolio optimisation based on yield prediction, 2016.

[11] O. Marko, S. Brdar, M. Panić, I. Šašić, D. Despotović, M. Knežević, V. Crnojević: Portfolio optimisation for seed selection in diverse weather scenarios, 2017.

[12] M. Žitnik: Pristop matricne faktorizacije za gradnjo napovednih modelov iz heterogenih podatkovnih virov, 2012.

[13] Lee, Daniel D. and Seung, H. Sebastian: Algorithms for non-negative matrix factorization, 2000.

[14] A. P. Singh, G. J. Gordon: A Unified View of Matrix Factorization Models, 2008.

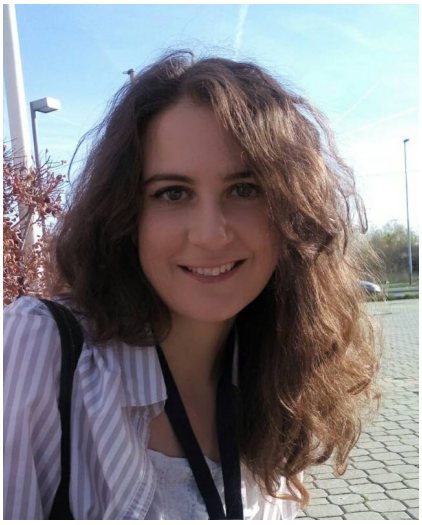
[15] J. Nocedal, S. J. Wright: Numerical optimization, 2006.

[16] S. Boyd, L. Vandenberghe: Convex optimization, 2009.

[17] M. Žitnik, B. Župan: Matrix factorization-based data fusion for drug-induced liver injurz prediction, 2014.

[18] Y.X. Wang and Y.J. Zhang: Nonnegative matrix factorization: A comprehensive review, 2013.

Biography



Milica Brkić was born on the 21th of March 1993 in Novi Sad. She finished elementary school "Nikola Tesla" and the Gymnasium "Svetozar Marković" in Novi Sad. She received her BSc degree in Applied Mathematics in 2016 from the Faculty of Sciences, University of Novi Sad, Serbia and she continued her Master studies in the field of Data Science at the same faculty. Since March 2018, Milica is working as Junior Researcher at the BioSense Institute.

Novi Sad, August 2018 Milica Brkić

UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET
KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj:
RBR
Identifikacioni broj:
IBR
Tip dokumentacije: monografska dokumentacija
BF
Tip zapisa: tekstualni štampani materijal
TZ
Vrsta rada: Master rad
VR
Autor: Milica Brkić
AU
Mentor: dr Sanja Brdar
MN
Naslov rada: Predikcija prinosa useva pomoću fuzije podataka koristeći faktorizaciju matrice
NR
Jezik publikacije: engleski
JP
Jezik izvoda:s/e
JI
Zemlja publikovanja: Republika Srbija
ZP
Uže geografsko područje: Vojvodina
UGP
Godina: 2018.
GO
Izdavač: autorski reprint
IZ Mesto i adresa: Novi Sad, Trg Dositeja Obradovića 4
MA
Fizički opis rada: text
FO
Naučna oblast: matematika
NO
Naučna disciplina: primenjena matematika
ND
Ključne reči: fuzija podataka, faktorizacija matrice

PO
UDK
Čuva se: u biblioteci Departmana za matematiku i informatiku, Prirodno-matematičkog fakulteta, u Novom Sadu
ČU
Važna napomena:
VN
Izvod: Sa rastom svetske populacije raste i potreba za hranom. Mnoge industrije koje proizvode semena traže način da razviju i unaprede sorte semena. Prinos je najbolji indikator koji će nam reći koja sorta semena je pogodna za koju lokaciju, iz tog razloga veoma je važno da izvršimo predikciju prinosa. Danas se podaci mogu lakše generisati nego ikada ranije. Ukoliko imamo podatke o sistemu koji posmatramo iz različitih perspektiva problem na koji nailazimo je kako da posmatramo udruženo sve te ulazne podatke na način da mogu da imaju koristi jedan od drugog. Kada imamo mnogo relacija, koje su predstavlje pomoću matrice, cilj nam je da iskoristimo informacije od jedne relacije kako bismo predvideli drugu. Istraživanja koja su sproveda na različitim podacima su pokazala prednost korišćenja DFMF algoritma u odnosu na standardne algoritme mašinskog učenja za ranu i kasnu integraciju i algoritme parcijalne integracije kao što je Multiple kernel learning algoritam. U ovom radu opisan je algoritam fuzije podataka pomoću faktorizacije matrice (Data fusion by matrix factorization algorithm ili skraćeno DFMF algoritam). DFMF algoritam koristi penalizovanu kolektivnu faktorizaciju matrice, gde je svaka matrica predstavljena kao proizvod tri matrice. U eksperimentalnom delu DFMF algoritam je korišćen za predikciju prinosa useva. Podaci korišćeni u tezi potiču sa Sindžentinog takmičenja.
IZ
Datum prihvatanja teme od strane NN veća: 03.09.2018.
DP
Datum odbrane:
DO
Članovi komisije:
KO
Predsednik: dr Ivica Bošnjak, vanredni profesor
Član: dr Vladimir Crnojević, redovni profesor
Član: dr Sanja Brdar, naučni saradnik

UNIVERSITY OF NOVI SAD
FACULTY OF SCIENCES
KEY WORD DOCUMENTATION

Accession number:
ANO
Identification number:
INO
Document type: monograph type
DT
Type of record: printed text
TR
Contents code: Master thesis
CC
Author: Milica Brkić
AU
Mentor: Sanja Brdar, PhD
MN
Title: Crop yield prediction by data fusion using matrix factorization
XI
Language of text: English
LT
Language of abstract: s/e
LA
Country of publicatin: Republic of Serbia
CP
Locality of publication: Vojvodina
LP
Publication year: 2018.
PY
Publisher: author's reprint
PU
Publ. place: Novi Sad, Trg Dositeja Obradovića 4
PP
Physical description: text
PD
Scientific field: mathematics
SF
Scientific discipline: applied mathematics

SD
Key words: data fusion, matrix factorization
UC
Holding data: Department of Mathematics and Informatics' Library,
Faculty of Sciences,
Novi Sad
HD
Note:
N
Abstract: As the global population is growing so does foods demand is increasing. Many seed industries are seeking the way to develop and improve seed varieties. Yield is one of the best indicator for making the decision which seed varieties would be suitable for the given location, but depends on complex interplay between weather, soil, genetics and other parameters. To be able to predict yield we need advanced algorithms. Today data is generated more easily than ever before. If we have data about system that is observed from various perspectives, the challange is how to jointly observe all those input spaces in such a way that they can benefit from each other. That is vital for improving accuracy. When we have multiple relations, which are represented as multuple matrices, we want to exploit information from one relation when predicting another.In this thesis Data Fusion by Matrix Factorization (DFMF) algorithm is used in order to achieve that task. DFMF algorithm uses a penalized matrix tri-factorization model that collectively tri-factorizes many data matrices such that each data matrix is decomposed into a product of tree latent matrices. Studies that were conducted on various datasets have shown that DFMF algorithm has high predictive power compared to standard machine learning and data mining algorithms for early and late integration and algorithms for intermediate integration like multiple kernel learning. In the experimental part DFMF algorithm is used in order to predict crop yield. Data from the thesis comes from the third Syngenta Crop Challenge.
AB
Accepted by the Scientific Board on: 03.09.2018.
ASB
Defended:
DE
Thesis defend board:
DB
President: Ivica Bošnjak, PhD
Member: Vladimir Crnojević, PhD
Member: Sanja Brdar, PhD