From lasso to collaborative filtering and recommender systems

Pierre-Alexandre Mattei

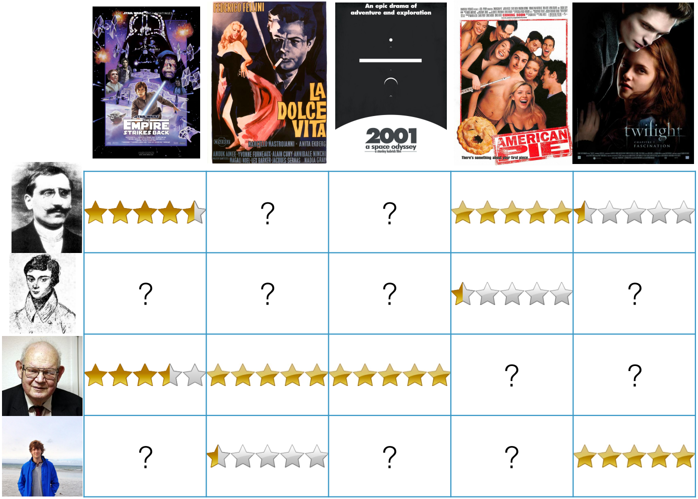
http://pamattei.github.io – @pamattei
pierre-alexandre.mattei@inria.fr

MSc Data Science

Collaborative filtering: basics

Collaborative filtering: basics

Collaborative filtering



Collaborative filtering

Problem: We partly observe a user-product matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$. Specifically, we only know the value of x_{ij} (e.g. the score of the movie) for a set of indices Ω.
We assume that there exist an (unknown) matrix $\mathbf{Z} \in \mathbb{R}^{m \times n}$, which is the "true user-product matrix", that we want to recover given our observations. We therefore have

$$\forall (i, j) \in \Omega,\ x_{ij} = z_{ij},$$

or more realistically

$$\forall (i, j) \in \Omega,\ x_{ij} \approx z_{ij}.$$

Exercise: How could we model $x_{ij} \approx z_{ij}$?

Collaborative filtering

Problem: We partly observe a user-product matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$. Specifically, we only know the value of x_{ij} (e.g. the score of the movie) for a set of indices Ω.
We assume that there exist an (unknown) matrix $\mathbf{Z} \in \mathbb{R}^{m \times n}$, which is the "true user-product matrix", that we want to recover given our observations. We therefore have

$$\forall (i,j) \in \Omega,\ x_{ij} = z_{ij},$$

or more realistically

$$\forall (i,j) \in \Omega,\ x_{ij} \approx z_{ij}.$$

Exercise: How could we model $x_{ij} \approx z_{ij}$? For example, we could say that $(x_{ij} - z_{ij})^2$ is minimised. A probabilistic approach would be

$$\forall (i,j) \in \Omega,\ x_{ij} = z_{ij} + \varepsilon_{ij},\ \text{with } \varepsilon_{ij} \sim \mathcal{N}(0, \sigma).$$

Collaborative filtering

Problem: We partly observe a user-product matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$. Specifically, we only know the value of x_{ij} (e.g. the score of the movie) for a set of indices Ω.
We assume that there exist an (unknown) matrix $\mathbf{Z} \in \mathbb{R}^{m \times n}$, which is the "true user-product matrix", that we want to recover given our observations.

This is clearly an ill-posed problem!(for now)

To be able to solve it, we need to assume that $\mathbf{Z}$ has some sort of simple structure. We need a notion of "simple structure" for matrices. Any idea?

Collaborative filtering

To be able to solve it, we need to assume that $\mathbf{Z}$ has some sort of simple structure. We need a notion of "simple structure" for matrices.

A first simple notion of simplicity is a constant matrix: all entries are identical. In a collaborative filtering context, it means all users have the exact same taste, and all products are equivalent... This is clearly overly simple!

Collaborative filtering

To be able to solve it, we need to assume that $\mathbf{Z}$ has some sort of simple structure. We need a notion of "simple structure" for matrices.

A first simple notion of simplicity is a constant matrix: all entries are identical. In a collaborative filtering context, it means all users have the exact same taste, and all products are equivalent... This is clearly overly simple!

A still very simple, but more resonable option, would be to assume that all columns are the same, or that all rows are the same.
We can be a bit more general and maybe assume that all columns (or rows) are proportional.

Collaborative filtering

To be able to solve it, we need to assume that $\mathbf{Z}$ has some sort of simple structure. We need a notion of "simple structure" for matrices.

A first simple notion of simplicity is a constant matrix: all entries are identical. In a collaborative filtering context, it means all users have the exact same taste, and all products are equivalent... This is clearly overly simple!

A still very simple, but more resonable option, would be to assume that all columns are the same, or that all rows are the same.
We can be a bit more general and maybe assume that all columns (or rows) are proportional.

The notion of rank of a matrix generalises these simple heuristics!

What is the rank of a matrix?

The rank is a measure of complexity for matrices. A matrix $\mathbf{Z} \in \mathbb{R}^{n \times n}$ has rank $r \in \{1, ..., n\}$ if there exists two matrices $\mathbf{U} \in \mathbb{R}^{n \times r}$ and $\mathbf{V} \in \mathbb{R}^{n \times r}$ such that

$$\mathbf{Z} = \mathbf{U}\mathbf{V}^T.$$

Intuitively, it means that we can describe the matrix with only $2nr$ coefficients, rather than n^2. So when r is small there is a huge gain...

An equivalent definition is that the rank of $\mathbf{Z}$ is the number of nonzero eigenvalues of $\mathbf{Z}^T\mathbf{Z}$.

What is the rank of a matrix?

- The only matrix with rank 0 is the null matrix.

What is the rank of a matrix?

- The only matrix with rank 0 is the null matrix.
- $\mathbf{Z}$ and $\mathbf{Z}^T$ have same rank.

What is the rank of a matrix?

- The only matrix with rank 0 is the null matrix.
- $\mathbf{Z}$ and $\mathbf{Z}^T$ have same rank.
- A matrix $\mathbf{Z}$ of rank 1 can be written $\mathbf{Z} = \mathbf{u}\mathbf{v}^T$. This gives

$$\mathbf{Z} = \mathbf{u}(v_1, ..., v_n) = (v_1\mathbf{u}, ..., v_n\mathbf{u}),$$

which means that the columns of $\mathbf{Z}$ are proportional.

What is the rank of a matrix?

- The only matrix with rank 0 is the null matrix.
- $\mathbf{Z}$ and $\mathbf{Z}^T$ have same rank.
- A matrix $\mathbf{Z}$ of rank 1 can be written $\mathbf{Z} = \mathbf{u}\mathbf{v}^T$. This gives

$$\mathbf{Z} = \mathbf{u}(v_1, ..., v_n) = (v_1\mathbf{u}, ..., v_n\mathbf{u}),$$

which means that the columns of $\mathbf{Z}$ are proportional. Similarly, we can write

$$\mathbf{Z} = (u_1, ..., u_n)^T\mathbf{v} = (u_1\mathbf{v}, ..., u_n\mathbf{v})^T,$$

which means that the rows of $\mathbf{Z}$ are proportional.

Collaborative filtering with low-rank matrices

A way to have a well posed-problem: add the assumption that the "true matrix" should have low-rank k! This makes sense because it means that there are just a few factors of variation. In particular, we recover the naive solutions of the previous slides.

$$\hat{\boldsymbol{Z}} \in \operatorname{argmin}_{\boldsymbol{Z} \in \mathbb{R}^{m \times n}} \sum_{(i,j) \in \Omega} (z_{ij} - x_{ij})^2 + \lambda \operatorname{rank}(\boldsymbol{Z}).$$

Collaborative filtering with low-rank matrices

$$\hat{\boldsymbol{Z}} \in \mathrm{argmin}_{\boldsymbol{Z} \in \mathbb{R}^{m \times n}} \sum_{(i,j) \in \Omega} (z_{ij} - x_{ij})^2 + \lambda \mathrm{rank}(\boldsymbol{Z}),$$

is unfortunately a super hard combinatorial problem to solve, because the rank is the number of non-zero singular values of Z.
Exercise:Can we replace this number of nonzeros by something smooth and convex?

Collaborative filtering with low-rank matrices

$$\hat{\boldsymbol{Z}} \in \operatorname{argmin}_{\boldsymbol{Z} \in \mathbb{R}^{m \times n}} \sum_{(i,j) \in \Omega} (z_{ij} - x_{ij})^2 + \lambda \operatorname{rank}(\boldsymbol{Z}),$$

is unfortunately a super hard combinatorial problem to solve, because the rank is the number of non-zero singular values of Z.
Exercise:Can we replace this number of nonzeros by something smooth and convex?

Yes! The ℓ_1 norm of the singular values, it's called the nuclear norm:

$$||Z||_* = \sum_{i=1}^{\min(n,m)} \lambda(Z)_i.$$

Like for the lasso, we replaced a number of nonzero elements by a ℓ_1 norm.

Collaborative filtering with the nucelar norm

The new problem is then:

$$\hat{\boldsymbol{Z}} \in \text{argmin}_{\boldsymbol{Z} \in \mathbb{R}^{m \times n}} \sum_{(i,j) \in \Omega} (z_{ij} - x_{ij})^2 + \lambda ||Z||_*.$$

It's convex and very fast algorithms exist!

We will use one called softimpute (Hastie, Mazumder, Lee, & Zadehin, JMLR 2015) in today's lab.