Variable Screening for High-dimensional Discriminant Analysis

With Food Authenticity Applications

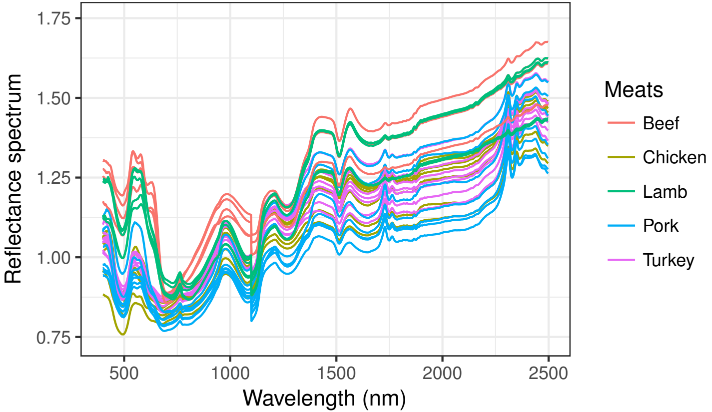
Pierre-Alexandre Mattei

IT University of Copenhagen
Conférence Chimiométrie XIX, Conservatoire National des Arts et Métiers

Joint work with **Charles Bouveyron** (Université Côte d'Azur & INRIA),
Michael Fop & **Thomas Brendan Murphy** (University College Dublin)

IT UNIVERSITY OF COPENHAGEN

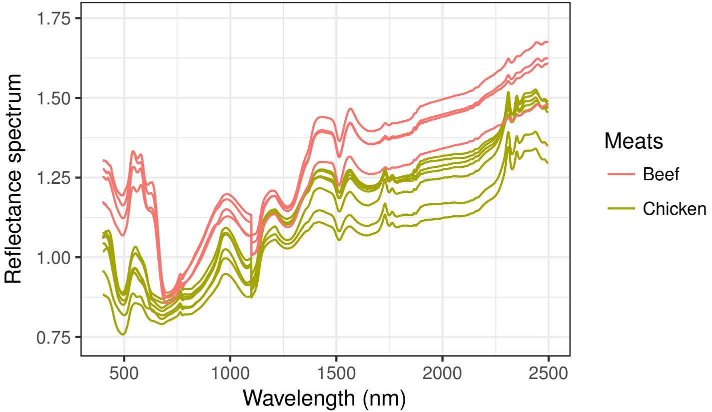
Recognising the right meat is a high-dimensional classification problem



Data from McElhinney, Downey &Fearn (JNIRS'99)

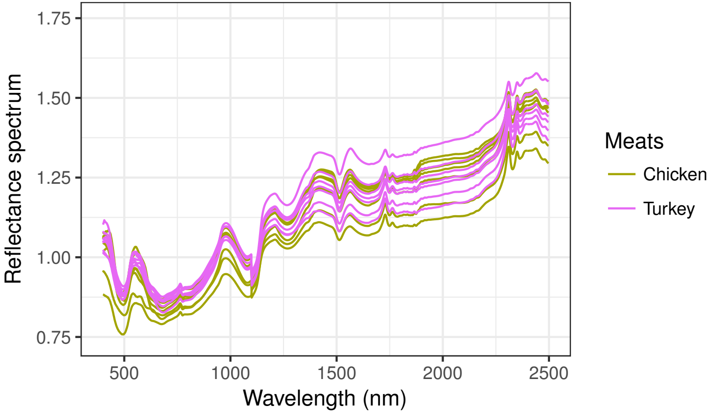
5 classes, 231 observations, 1050 wavelengths...

Recognising the right meat is a high-dimensional classification problem



Although $p = 1050$ is much larger than $n = 231$ (and the **curse of dimensionality** is to be reckoned with), **the classes look pretty well separated...**

Recognising the right meat is a high-dimensional classification problem



...or do they?

Some subproblems are much harder to discriminate than others.

General data pattern we'll consider today

- **high-dimensional data**
- **more than two classes**
- **some subproblems are harder than others**
- **some variables are more useful than others**

Those are some common features of spectral food authenticity data.

On these data sets, Gaussian discriminant with variable selection works *really* well

A greedy algorithm introduced simultaneously by Murphy, Dean & Raftery (AOAS'10) and by Maugis, Celeux & Martin-Magniette (JMVA'11).

Method	**Out-of-sample misclassification rate**
Gaussian discriminant analysis with variable selection and updating	**6.1% (3.5)**
Transductive SVMs	42.6% (5.7)
Random Forests	20.1% (3.8)
AdaBoost	20.3% (4.8)
Bayesian Multinomial Regression	34.2% (5.8)
Heavy preprocessing	5.6%-13.9%

Extremely good empirical results **BUT unfit for problems with p larger than a few dozens!**

Is is possible to scale up the technique to enable it to treat thousands of variables easily ?

Variable screening is a way to dramatically scale up expensive algorithms

Introduced by Fan & Lv (JRSSB'08), the idea is to

- compute **cheap marginal scores** for all variables,
- use these scores to **rank the variables,**
- **keep only the top-K variables** and feed them to the expensive algorithm.

These scores are typically **marginal correlations** between the individual variables $\mathbf{x}_j$ and a response $\mathbf{y}$.

Their low computational price comes from the fact that they **ignore completely the correlations between the variables** $\mathbf{x}_1, \ldots, \mathbf{x}_p$.

Is marginal screening fit for multiclass classification?

If there is an easier classification subproblem (like red vs. white meat), **any marginal ranking is going to give the highest scores to the variables suitable for this easier problem.**

Consequently, we would like to rely on a **more refined scheme than a single marginal ranking.**

Our solution: compute several rankings.

Computing one ranking for each partition of the classes

Let $\mathcal{C}$ the set of all C possibles classes. A **partition** of $\mathcal{C}$ is a set of nonempty subsets of $\mathcal{C}$ such that every element of $\mathcal{C}$ is exactly in one of these subsets.

Examples: {white meats, red meats }; {poultry, {beef, pork, lamb} }
{beef, {chicken, pork, lamb, turkey} };
{ {chicken}, {turkey}, {beef, pork, lamb} }

There are B_c possible partitions, B_c is the c-th **Bell number.** The first Bell numbers are

$$B_0 = B_1 = 1,\ B_2 = 2,\ B_3 = 5,\ B_4 = 15,\ B_5 = 52,\ B_6 = 203...$$

Computing one ranking for each partition of the classes using Bayes Factors

Given a nontrivial partition $\rho = \{\rho_1, ..., \rho_K\}$ of cardinal $K \in \{2, ..., C\}$ and a variable $j \in \{1, ..., p\}$, we wish to **measure the usefulness of variable j to discriminate the classes induced by ρ**. We will use **Bayes factors between two competing models** to this end.

We will compare the models:

- **model $\mathcal{M}_\rho^j$: j is discriminative**
- **model $\mathcal{M}_0^j$: j is not discriminative.**

Computing one ranking for each partition of the classes using Bayes Factors: defining the models

Model $\mathcal{M}_\rho^j$: given some parameters $\boldsymbol{\tau} \in \Delta^C$, $\mu_1, ..., \mu_K \in \mathbb{R}$ and $\sigma_1, ..., \sigma_K \in \mathbb{R}^+$, we define

$$\mathcal{M}_\rho^j : \begin{cases} z \sim \text{Cat}(\tau) \\ x_j | \{z \in \rho_k\} \sim \mathcal{N}(\mu_k, \sigma_k). \end{cases} \tag{1}$$

Model $\mathcal{M}_0^j$: for $\boldsymbol{\tau} \in \Delta^C$, $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$, we define

$$\mathcal{M}_0^j : \begin{cases} z \sim \text{Cat}(\tau) \\ x_j \sim \mathcal{N}(\mu, \sigma). \end{cases} \tag{2}$$

To obtain Bayesian models, we use **normal-inverse-gamma priors.** Hyperparameters are chosen following the **unit information paradigm** of Kass & Wasserman (JASA'95).

Computing one ranking for each partition of the classes using Bayes Factors: defining the score

Our score for variable j and partition ρ will be

$$\log \mathsf{BF}(\mathcal{M}_\rho^j, \mathcal{M}_0^j) = \log p(\mathbf{x}_j, \mathbf{z} | \mathcal{M}_\rho^j) - \log p(\mathbf{x}_j, \mathbf{z} | \mathcal{M}_0^j),$$

which is very cheap to compute, and is exactly the **Bayesian evidence in favor of $\mathcal{M}_\rho^j$** (Kass & Raftery, JASA'95).

We have defined scores for all partitions and variables.

From several rankings to a single subset of variables

Partitions	{white meats, red meats }	{poultry, rest }	...
Top variables	122	546	...
	245	239	...
	189	108	...
	112	808	...
	...	...	

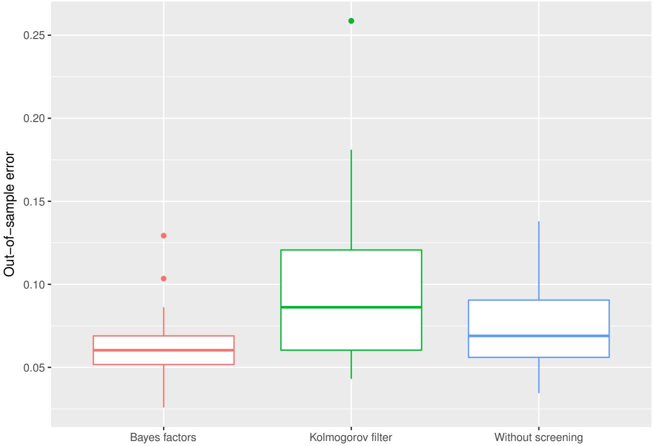
We seek a **single subset of variables that would take into account all these rankings.**

From several rankings to a single subset of variables

Partitions	{white meats, red meats }	{poultry, rest }	. . .
Top variables	122	546	. . .
	245	239	. . .
	189	108	. . .
	112	808	. . .
	. . .	. . .	

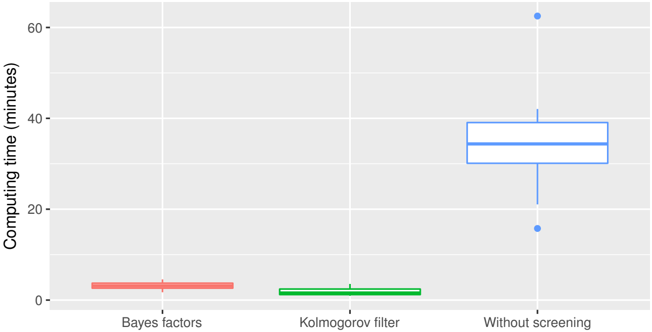
We keep the **top-k variables for each partition,** until we end up with the desired amount of variables.

BF screening vs. no screening vs. Kolmogorov filter (ranking-based, Mai & Zou, AOS'15)



The Kolmogorov filter makes no mistakes for *white vs. red meat*, but a lot of mistakes for harder partitions.

BF screening vs. no screening vs. Kolmogorov filter (ranking-based, Mai & Zou, AOS'15)



Both screening methods are **much faster!**