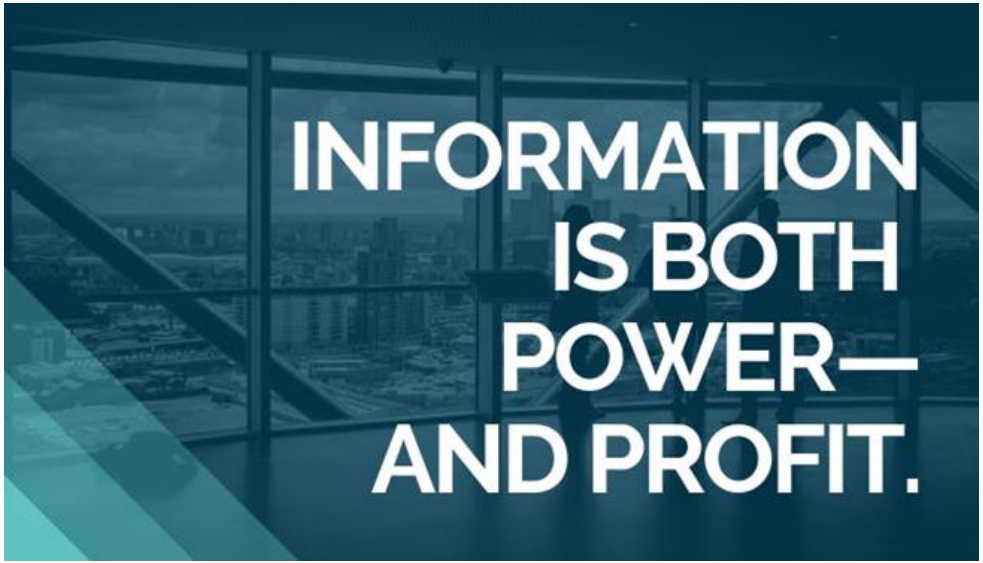
Learning from data:

Technical Nuances of Machine Learning in Plant Breeding

Alencar Xavier, 02/04/2021
Research Scientist at Corteva Biostatistics
Adjunct professor at Purdue University

Adequate use of



Outline

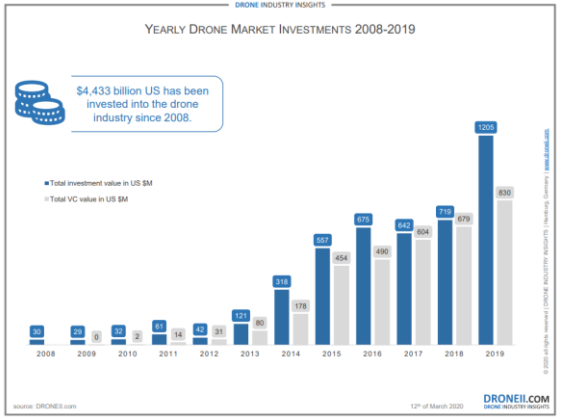
Three faces on machine learning

1. Good
2. Bad
3. Ugly

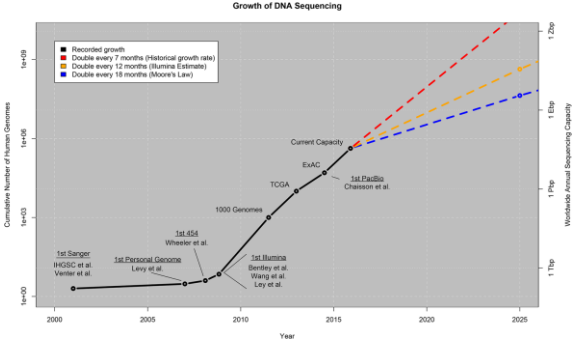
1. **Good**
   - Intro and motivation
   - Filters in PB
2. **Bad**
   - Metrics of success
   - Adequate validation
3. **Ugly**
   - Optimization
   - From RR to NN

Part 1 – Good learners

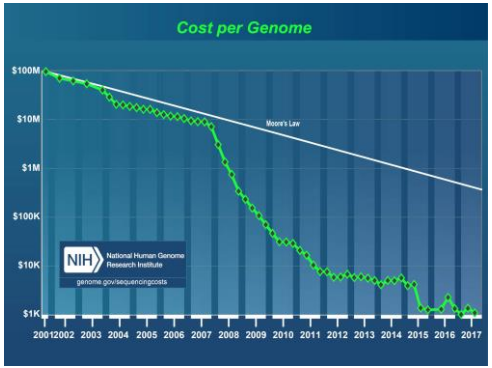
More Pheno



More Geno



The Cost of Sequencing a Human Genome. NIH. https://www.genome.gov/27565109/the-cost-of-sequencing-a-human-genome/



Stephens, Z. D.et al. (2015). Big data: astronomical or genomical? *PLoS biology*, *13*(7), e1002195.

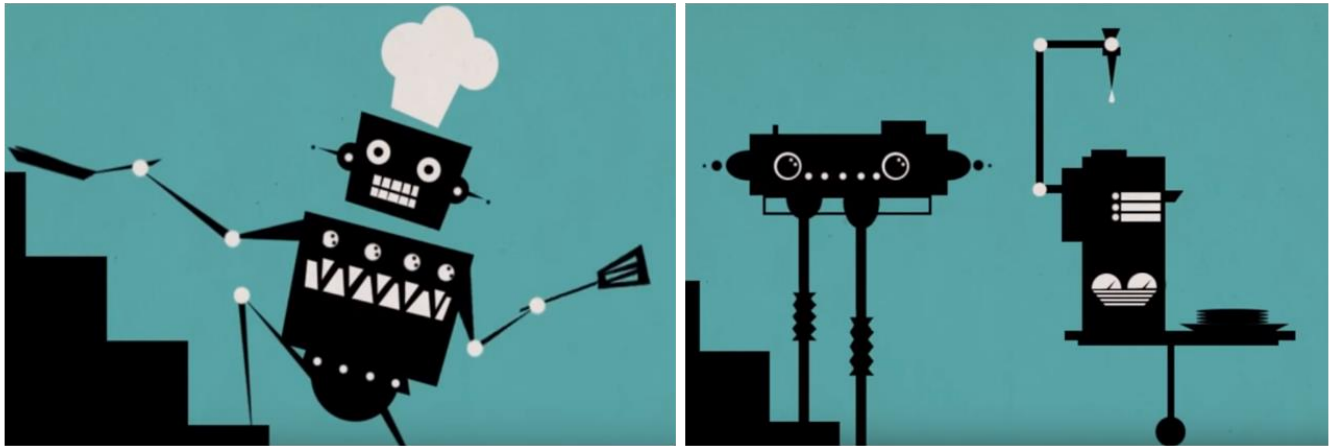
More Env

- **UC Merced - GridMET**
- **NWS - NOAA**
- **NASA - GISS**
- **Harmonized SoilDB**
- **USDA - SSURGO**

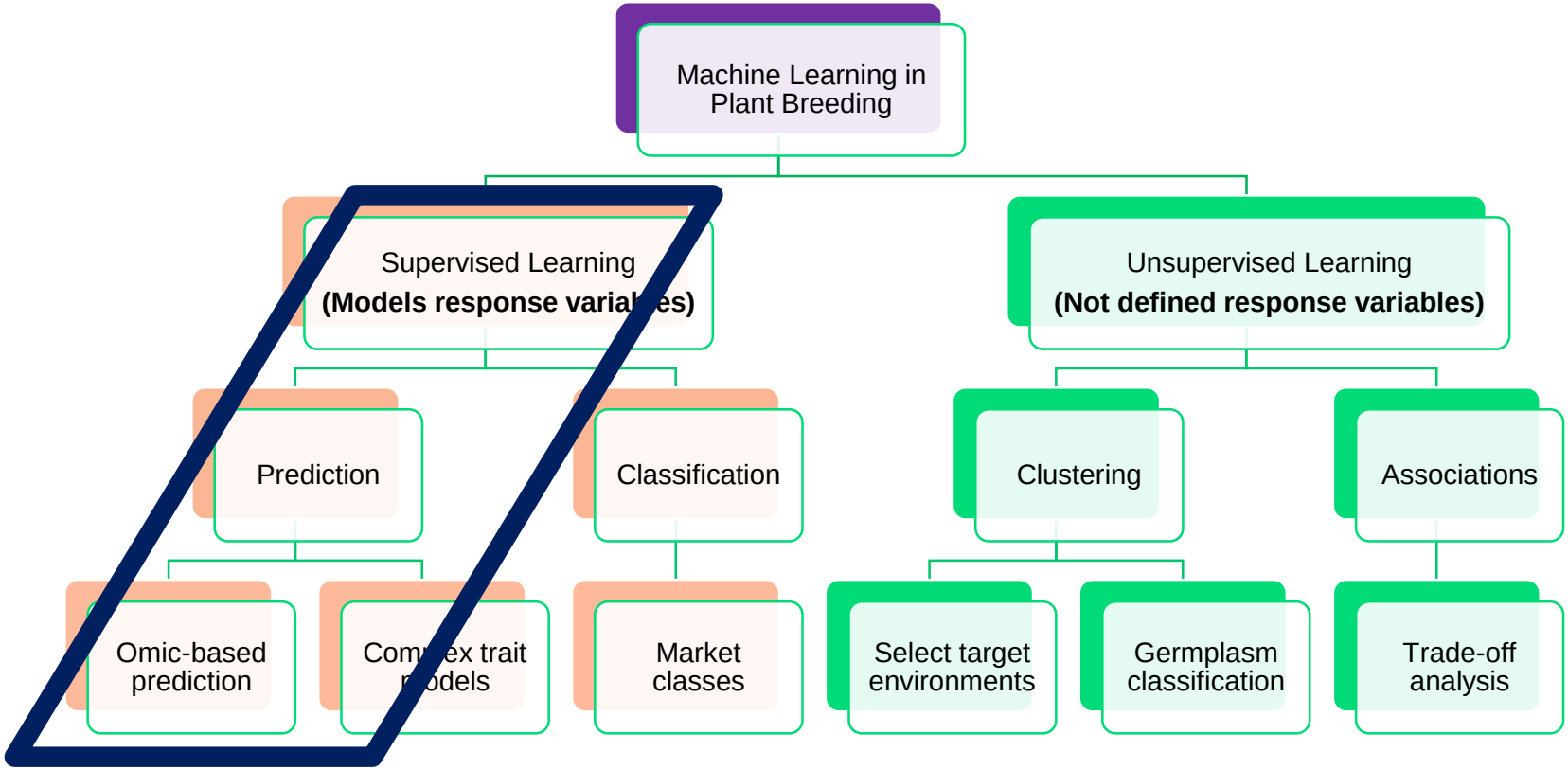
More Computing

Why is machine learning good for?

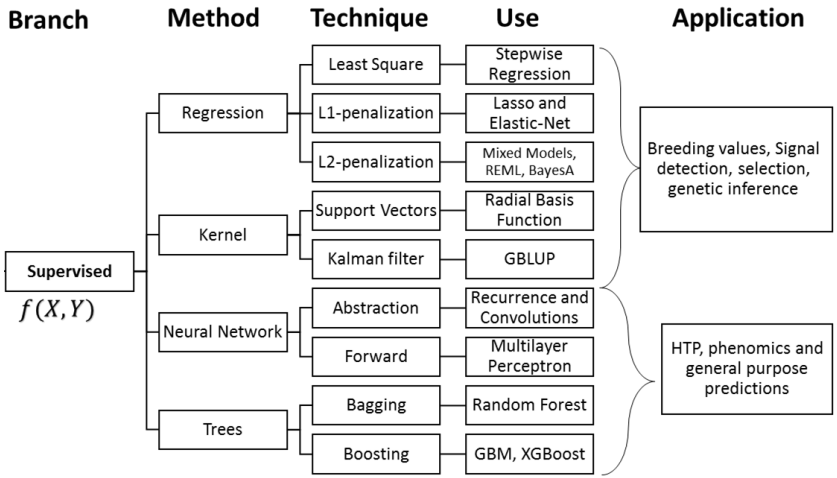
Good for solving **single** **well-defined** **problem**

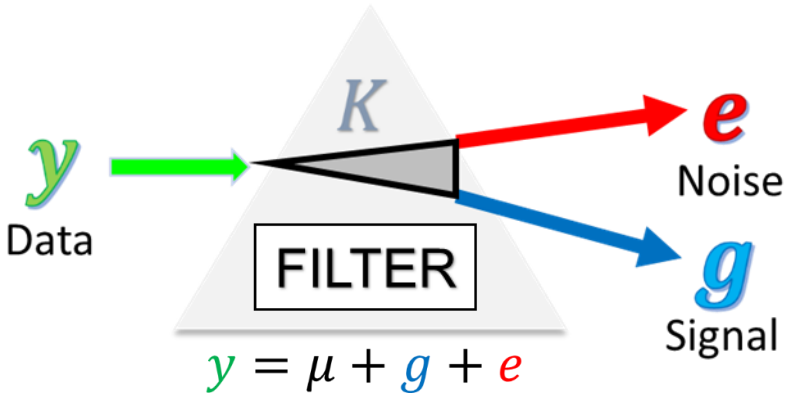


Source: https://www.youtube.com/watch?v=MPR3o6Hnf2g

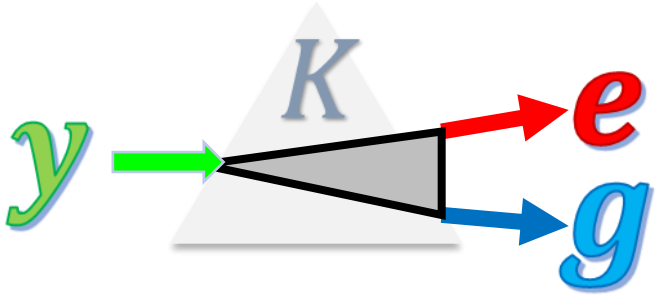

Machine Learning in Plant Breeding
Supervised Learning
(Models response variables)
Unsupervised Learning
(Not defined response variables)
Prediction
Classification
Clustering
Associations
Omic-based prediction
Complex trait models
Market classes
Select target environments
Germplasm classification
Trade-off analysis

Key of supervised learning: FILTERING

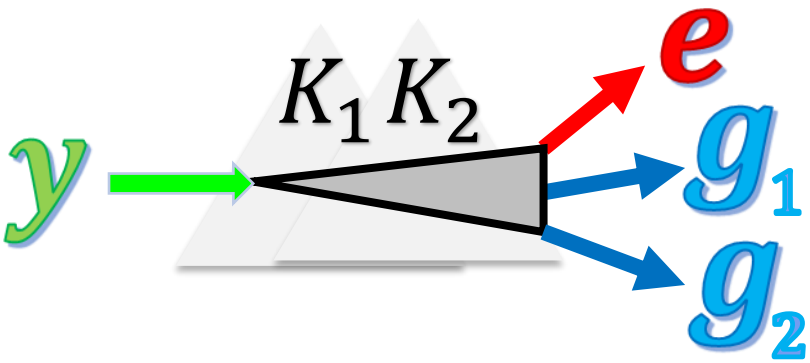




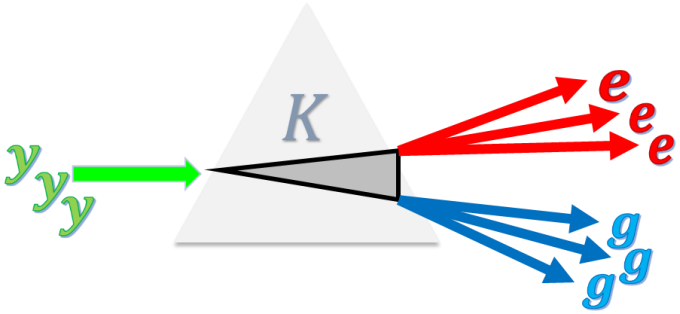
Generalizations of simple filters



1) Multiple filters

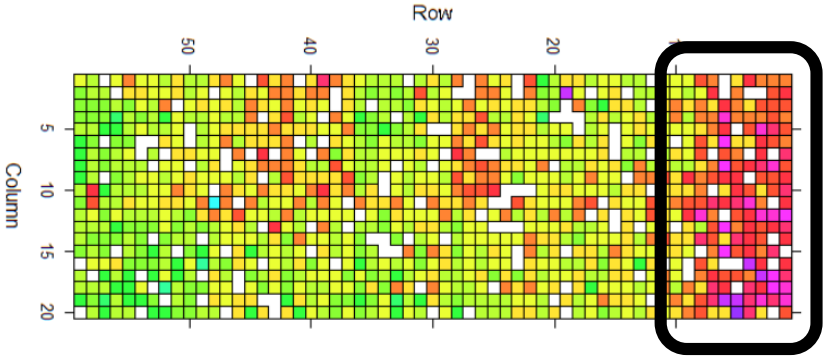


2) Multi-response filters

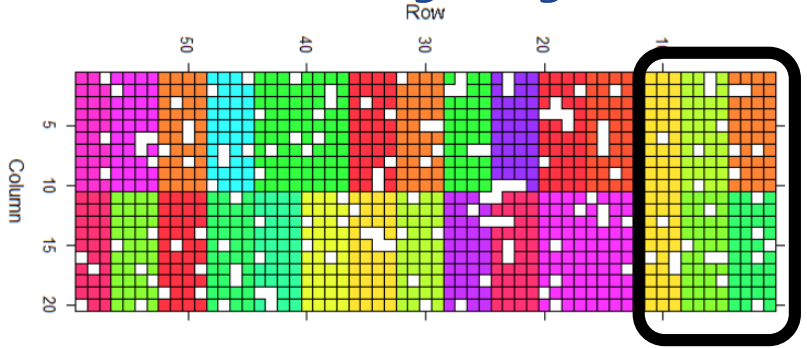


1) When bother with multiple filters?

Field variation



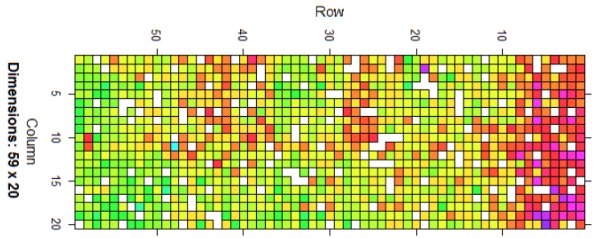
Family layout



Some families were placed on unfavorable side of the field…

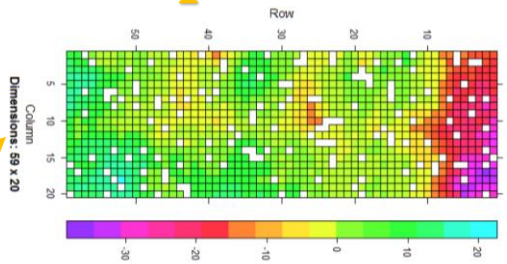
SoyNAM field,
Indiana 2014

Pheno

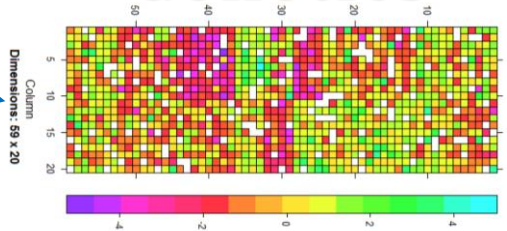


G, R

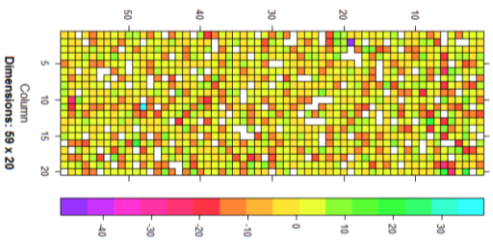
Spatial



Genetics

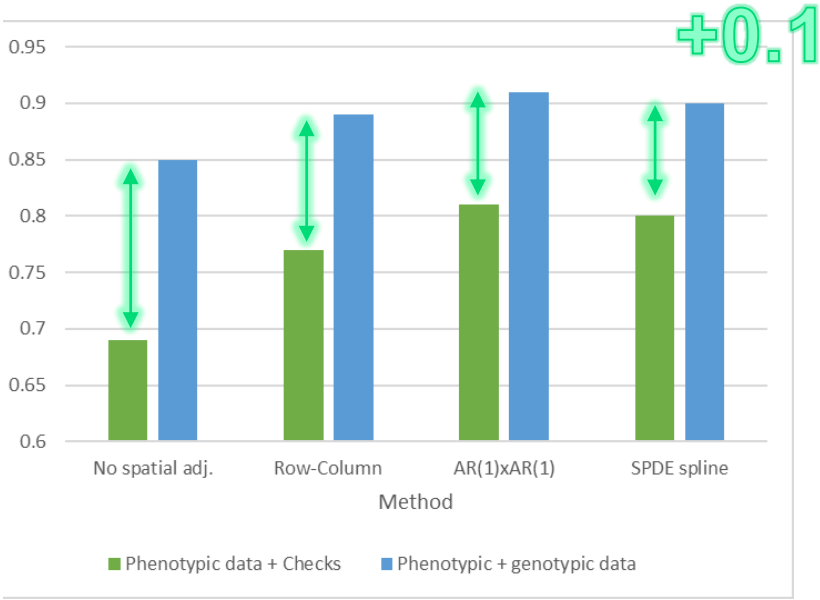


Residuals

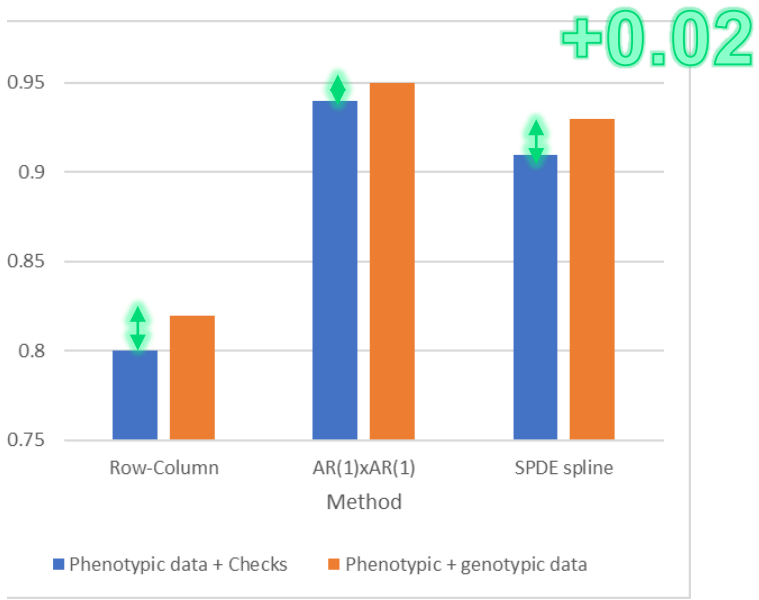


Multiple filters benefit separability of signals

Improvement in genetic accuracy



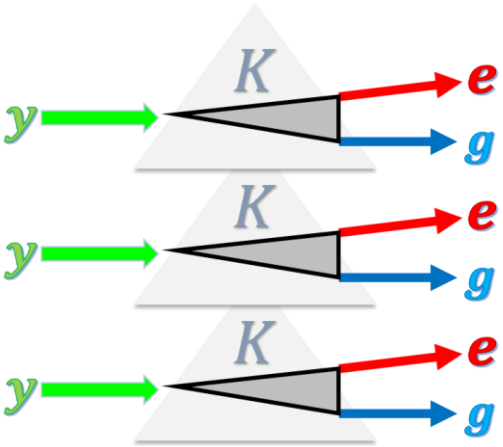
Improvement in spatial accuracy



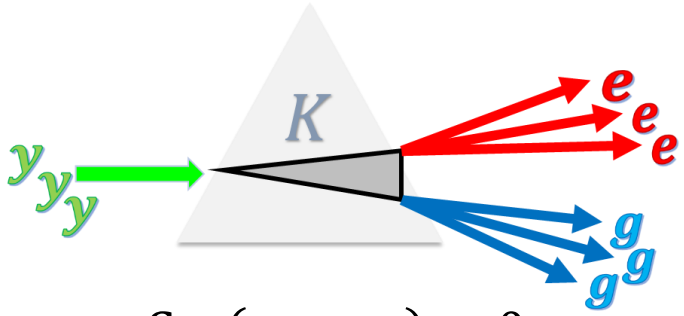
Results derived from simulation of field with1500 plots, 1275 non-replicated entries + checks, from single trials

2) When bother with multi-response filters?

Filter one input at a time
(parallelizable)



Filter passing multiple input at once



$$\mathrm{Cor}(g_A, g_B,) \neq 0$$

New information

Genetic correlation table

	g_1	g_2	g_3
g_1	1	ρ_{12}	ρ_{13}
g_2	ρ_{21}	1	ρ_{23}
g_3	ρ_{31}	ρ_{32}	1

- **Bivariate model**

Objective gain of information!

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \qquad y \sim N(0, G \otimes \Sigma_a + I \otimes \Sigma_e)$$

- Model equation

$$\begin{bmatrix} Z_1'\Sigma_e^{11}Z_1 + G^{-1}\Sigma_a^{11} & Z_1'\Sigma_e^{12}Z_2 + G^{-1}\Sigma_a^{12} \\ Z_2'\Sigma_e^{12}Z_1 + G^{-1}\Sigma_a^{12} & Z_2'\Sigma_e^{22}Z_2 + G^{-1}\Sigma_a^{22} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} Z_1'(\Sigma_e^{11}y_1 + \Sigma_e^{12}y_2) \\ Z_2'(\Sigma_e^{22}y_2 + \Sigma_e^{12}y_1) \end{bmatrix}$$

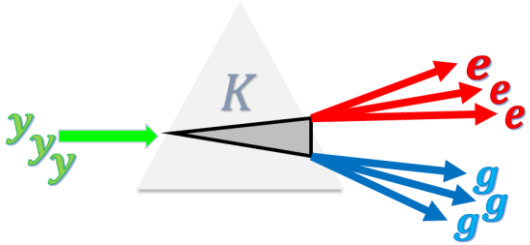
- **Univariate vs bivariate solution for a given predictor**

$$g_1 = (Z_1'\Sigma_e^{11}Z_1 + G^{-1}\Sigma_a^{11})^{-1}(Z_1'\Sigma_e^{11}y_1)$$

EXTRA INFORMATION

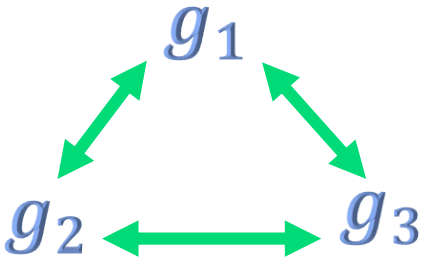
$$g_1|g_2 = (Z_1'\Sigma_e^{11}Z_1 + G^{-1}\Sigma_a^{11})^{-1}(Z_1'(\Sigma_e^{11}y_1 + \Sigma_e^{12}y_2) - (Z_1'\Sigma_e^{12}Z_2 + G^{-1}\Sigma_a^{12})g_2)$$

Sparse & Directed Filtering



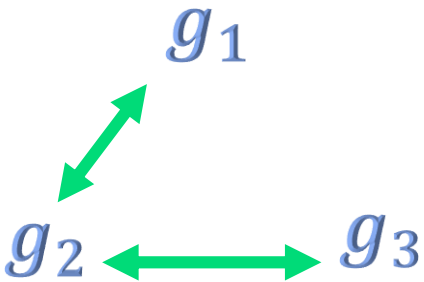
Genetic correlation table

	g_1	g_2	g_3
g_1	1	ρ_{12}	ρ_{13}
g_2	ρ_{21}	1	ρ_{23}
g_3	ρ_{31}	ρ_{32}	1



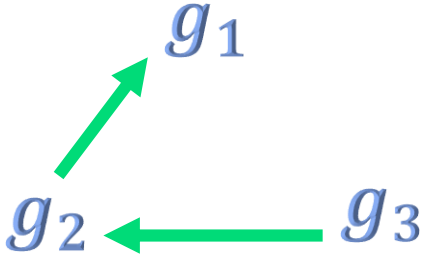
Sparse

	g_1	g_2	g_3
g_1	1	ρ_{12}	–
g_2	ρ_{21}	1	ρ_{23}
g_3	–	ρ_{32}	1



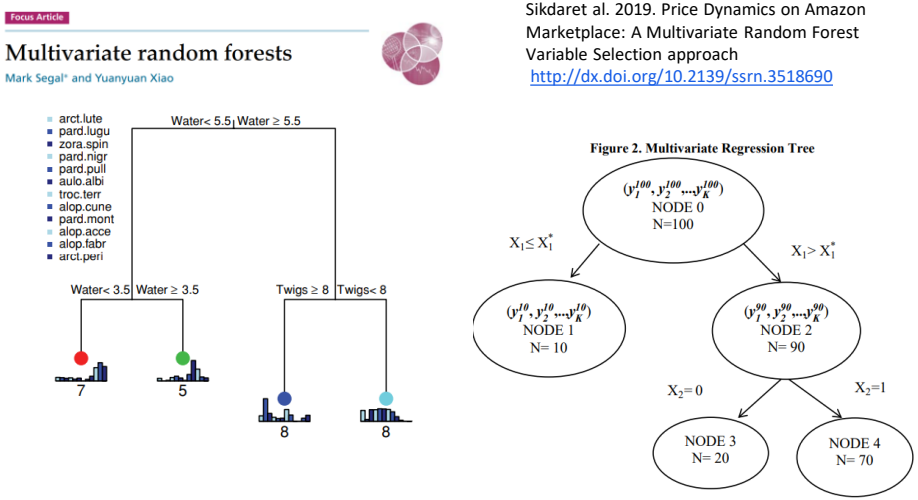
Directed

	g_1	g_2	g_3
g_1	1	–	–
g_2	ρ_{21}	1	–
g_3	–	ρ_{32}	1



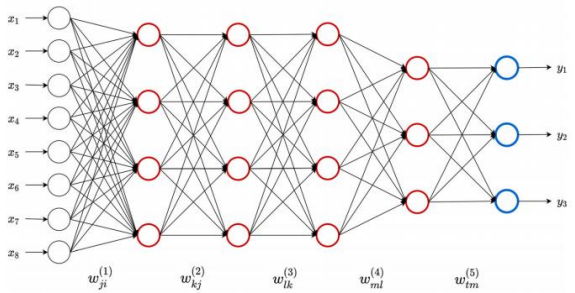
Modern multi-response machinery

MVRF feature splits occur
for all responses together

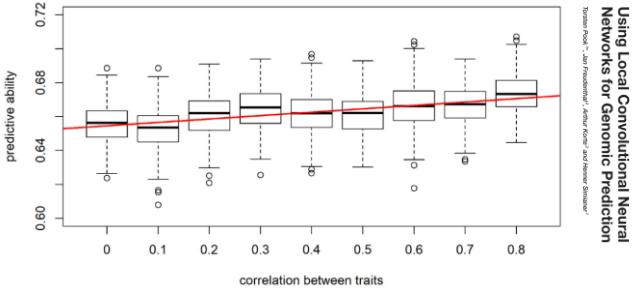


Sikdaret al. 2019. Price Dynamics on Amazon Marketplace: A Multivariate Random Forest Variable Selection approach
http://dx.doi.org/10.2139/ssrn.3518690

Multi-response comes natural for DNNs



Montesinos-López et al. BMC Genomics (2021) 22:19
https://doi.org/10.1186/s12864-020-07319-x



Using Local Convolutional Neural Networks for Genomic Prediction

Modern multi-response machinery

MegaLMM: Mega-scale linear mixed models for genomic predictions with thousands of traits

Daniel E Runcie[1], Jiayi Qu[1], Hao Cheng[1] and Lorin Crawford[2]

MegaLMM (SEM-like approach)

$$Y = F\Lambda + XB + U + E$$

$$F = X_F B_F + U_F + E_F$$

$$Y \sim N(0, \Psi_G + \Psi_R + \Lambda'\Lambda)$$

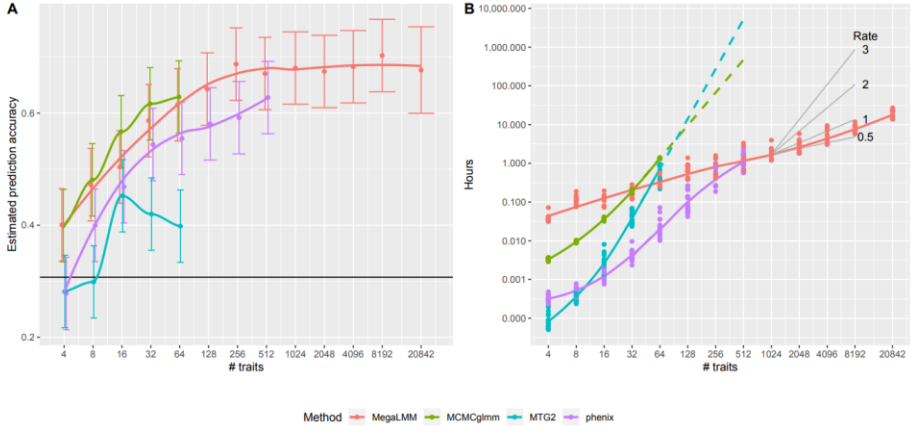
$$U \sim N(0, K, G)$$

$$G = \Psi_G + \Lambda'\Psi_{FG}\Lambda$$

$$E \sim N(0, I, R)$$

$$R = \Psi_R + \Lambda'\Psi_{FE}\Lambda$$

*Solved using Gibbs sampling



"Classical" mixed model framework has been evolving in this area

Fast Computation of Polygenic Effects via Multivariate Ridge Regression

A Preprint

David Habier
Department of Biostatistics
Corteva Agriscience
Johnston, IA 50131
david.habier@corteva.com

Alencar Xavier
Department of Biostatistics
Corteva Agriscience
Johnston, IA 50131
alencar.xavier@corteva.com

Tilde-Hat and Gauss-Seidel (THAGS)

Statistical model

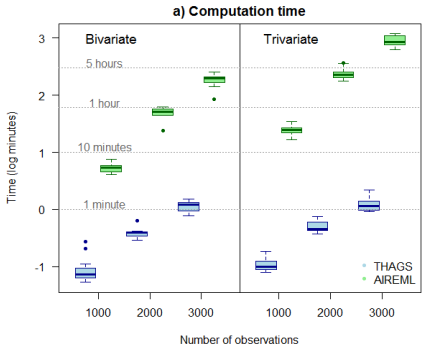
- $y_k = \mu_k + X_k \beta_k + e_k$, for trait k
- $\beta \sim N(0, I \otimes \Sigma_\beta)$
- $e \sim N(0, I \otimes \Sigma_e)$, $\text{cov}(e_i, e_j) = 0$

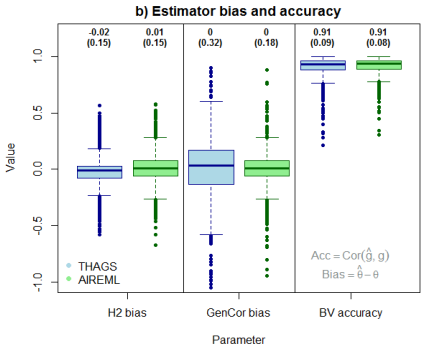
Coefficient updates: Multivariate Gauss-Seidel

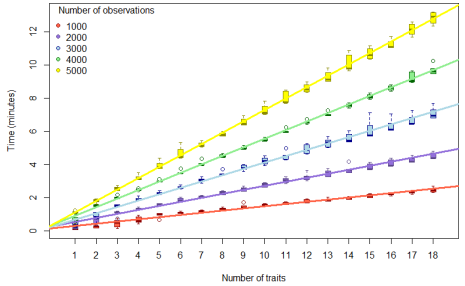
- $\beta_j^{t+1} = \left(X_j'\Sigma_e^{-1}X_j + \Sigma_\beta^{-1}\right)^{-1}\Sigma_e^{-1}X_j'(X_j\beta_j^t + e^t)$
- $e^{t+1} = e^t - X_j'(\beta_j^{t+1} - \beta_j^t)$

Covariance updates: Tilde-Hat method

- $\Sigma_{\beta(A,B)} = \frac{(y_A - \tilde{\mu}_A)'(X_A\tilde{\beta}_B) + (y_B - \tilde{\mu}_B)'(X_B\tilde{\beta}_A)}{n_A\sum_{j=1}^{t}\sigma^2_{x(A)_j} + n_B\sum_{j=1}^{t}\sigma^2_{x(B)_j}}$
- $\Sigma_{e(i,i)} = \frac{(y_i - \tilde{\mu}_i)'\tilde{e}_i}{n_i - 1}$







1. **Good**
   - Intro and motivation
   - Filters in PB
2. **Bad**
   - Metrics of success
   - Adequate validation
3. **Ugly**
   - Optimization
   - From RR to NN

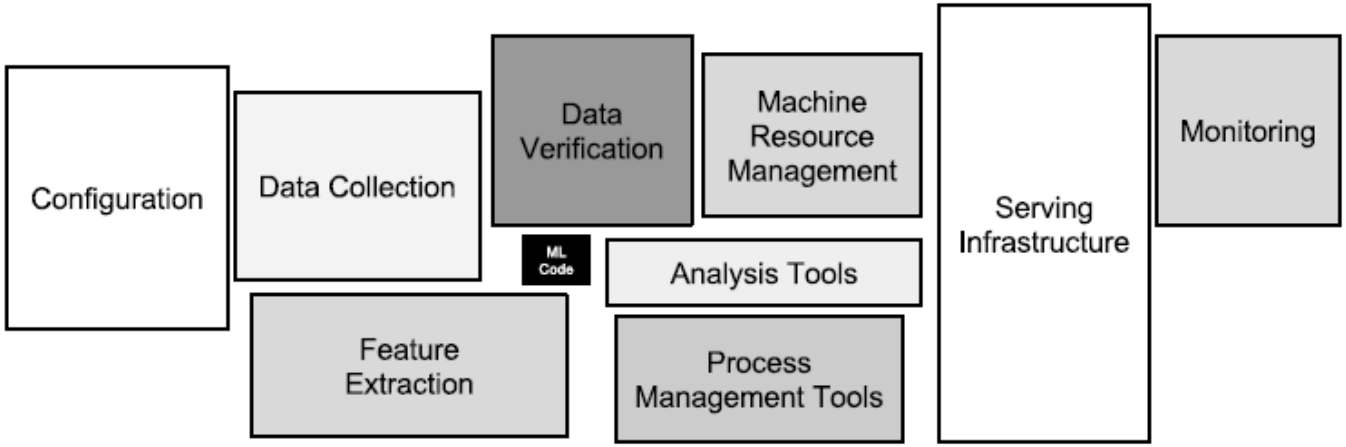
(avoiding)

Part 2 – Bad learning

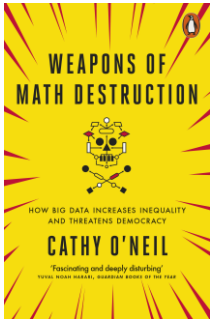
There is usually more to ML than a proof of concept with cross-validations

Hidden Technical Debt in Machine Learning Systems

doi/10.5555/2969442.2969519



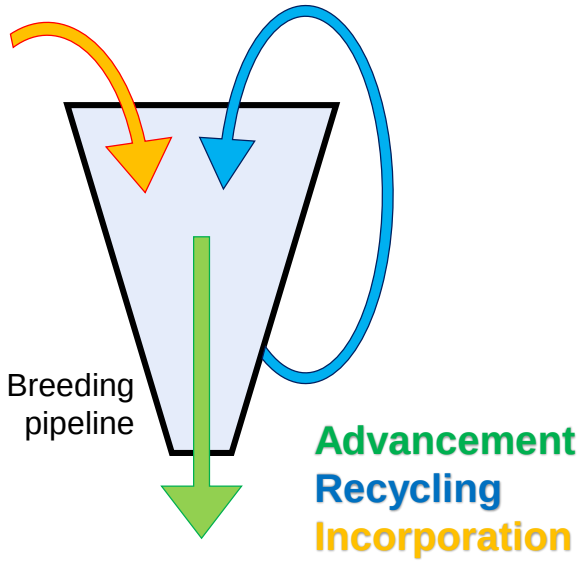
- How easily can an entirely new algorithmic approach be tested at full scale?
- What is the transitive closure of all data dependencies?
- How precisely can the impact of a new change to the system be measured?
- Does improving one model or signal degrade others?
- How quickly can new members of the team be brought up to speed?



Chasing the right signal

- Breeding value (**GEBV**)
  - *Pattern*: ADDITIVE GENETICS
  - *Method*: GBLUP, BayesABC, LASSO
  - *Suits*: **RECYCLING**, **ADVANCEMENT**

- Genomic value (**EGV**)
  - *Pattern*: ANY GENETICS
  - *Method*: RKHS, DNN, Random Forest
  - *Suits*: **ADVANCEMENT**



Defining the problem: Metrics for success

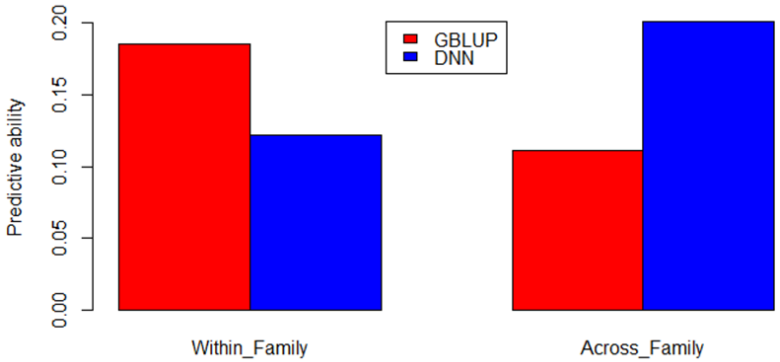
1. Scientist (**why**): to define the problem mathematically (easy to get it wrong)

2. Metric (**what**): Correlations, MSPE, Jaccard index, Accuracy, Success (1|0)

3. Testing (**how**): Simulation or cross-validation (CV) on real data? WF vs AF?

How to design an adequate cross-validation???

Two metrics, two different answers



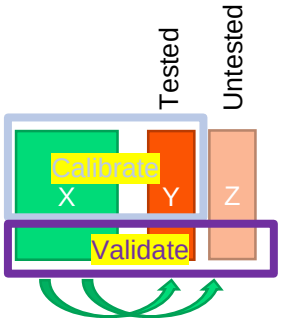
Grain yield on the SoyNAM population: Populations 1:8 predicting populations 9:15

Testing machines for different scenarios of genomic prediction

	Genotype	Environment	Prediction Difficulty
CV00	New	New	*****
CV0	Observed	New	***
CV1	New	Observed	***
CV2	Observed	Observed	*

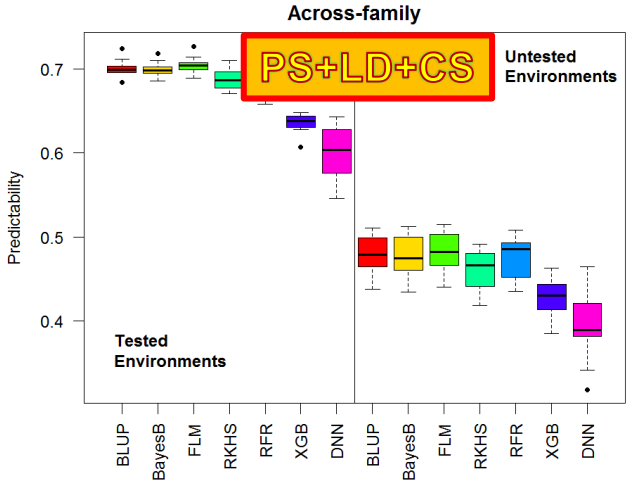
Adapted from Crossa et al. (2017) doi.org/10.1016/j.tplants.2017.08.011

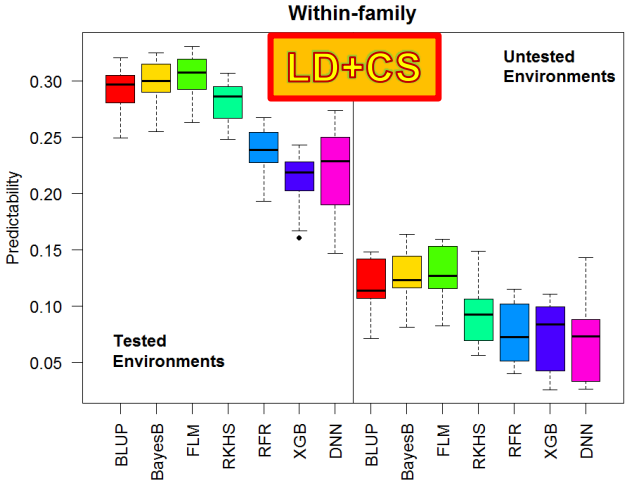
CV scheme

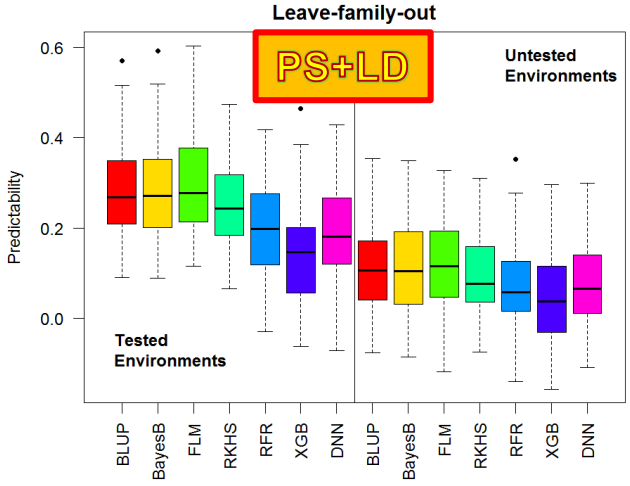


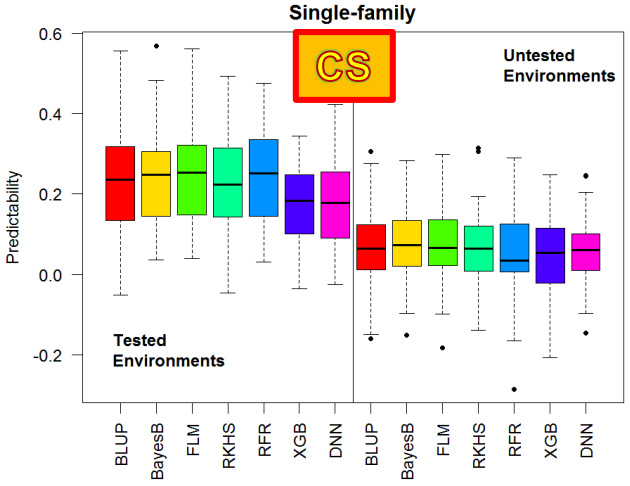
Type of information captured by SNP

- Population structure (PS)
- Linkage disequilibrium (LD)
- Cosegregation / Haplotype (CS)









SoyNAM data
ES: 2012 (7 loc)
PS: 2013 (4 loc)
#Fam = 40
Genos = 5600
SNPs = 4300
Obs: 3k-5k obs/loc

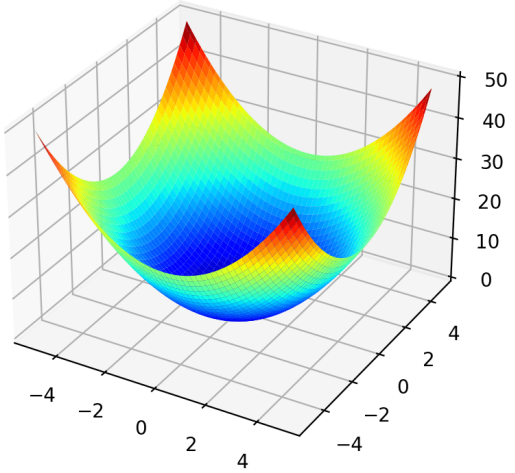
CORTEVA™ agriscience
Alencar.Xavier@Corteva.com
Corteva Biostatistics, Methods group

1. **Good**
   - Intro and motivation
   - Filters in PB
2. **Bad**
   - Metrics of success
   - Adequate validation
3. **Ugly**
   - Optimization
   - From RR to NN

Part 3 – Ugly learning

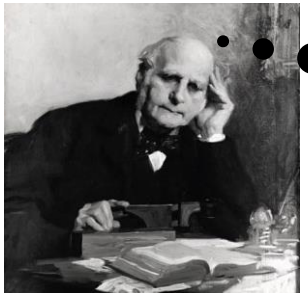
Divergency in philosophy

- In quantitative genetics:
  - Parameters: Variances + Regression coefficients
  - Function: Likelihood (complex and convex)
  - Tuning: Generally not needed
  - **Method**: First order (EM, MCMC), second order (AI, MIVQUE)
- In machine learning:
  - Parameters: Regression coefficients
  - Function: MSE, L2 (simple)
  - Tuning: Cross validations, need secondary objective function
  - **Method:** First order: coordinate & gradient descent



Solving: $y = Xb + e$

$Finding \rightarrow argmin(e'e + \lambda b'b)$



I've created a monster!!

- Coordinate descent

  (Use diagonals of LHS)

$$b_j^{t+1} = \frac{x_j'(y - X_{-j}b_{-j})}{x_j'x_j + \lambda}$$

Used for WGR (RR, BayesA)

glmnet, BGLR, bWGR, GS3

- Gradient descent

  (Does not build LHS)

$$b^{t+1} = b^t - \frac{2r}{n}[X'(y - Xb^t) + \lambda b^t]$$

Used for Deep Neural Nets

TF/Keras, PyTorch, MXNet, h2o

- Second order

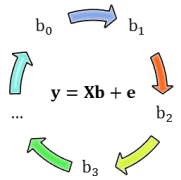
  (Builds entire LHS)

$$b = (X'X + \lambda)^{-1}(X'y)$$

Used for everything else

ASREML, lme4, SAS, BLUPF90

Coordinate descent of a RRBLUP



$$f(b_{RR}) \rightarrow \text{argmin}\left(\sum_{i=1}^{n} e^2 + \lambda \sum_{j=1}^{p} b^2\right)$$

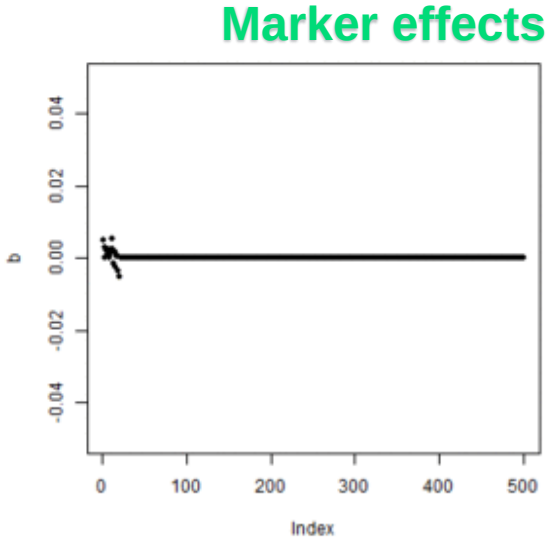
$$SS = e'e + \lambda(b'b)$$
$$SS = (y - xb)'(y - xb) + \lambda(b'b)$$
$$SS = y'y + (xb)'(xb) - 2(y'xb) + \lambda(b'b)$$
$$SS = y'y + b'(x'x)b - 2(y'xb) + \lambda(b'b)$$

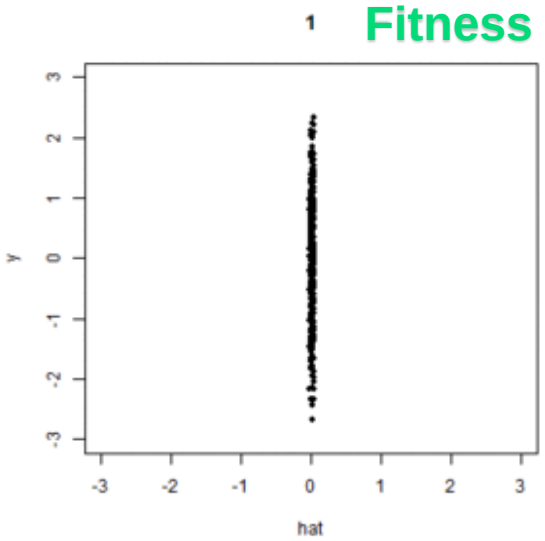
$$\frac{\partial SS}{\partial b} = \cancel{y'y} + 2(x'x)b^{\cancel{2}} - 2(y'x)\cancel{b} + 2\lambda b^{\cancel{2}}$$

$$0 = 2b(x'x) - 2(y'x) + 2b\lambda$$

$$-2b\,(x'x + \lambda) = -2\,(y'x)$$

$$b = \frac{-2(y'x)}{-2(x'x + \lambda)} = \frac{y'x}{x'x + \lambda}$$





CORTEVA™ agriscience
Alencar.Xavier@Corteva.com
Corteva Biostatistics, Methods group

Gradient descent of RRBLUP

- Model (Ridge)

$$y = X\beta + e$$

- Gradient descent

$$\beta^{t+1} = \beta^{t} - \alpha\nabla$$

- Where

$$\alpha = \text{Learning rate}$$

$$f = (y - Xb)'(y - Xb) + \lambda b'b$$

$$\nabla = f' = \frac{\partial f}{\partial b} = -2X'(y - Xb) + 2\lambda b$$

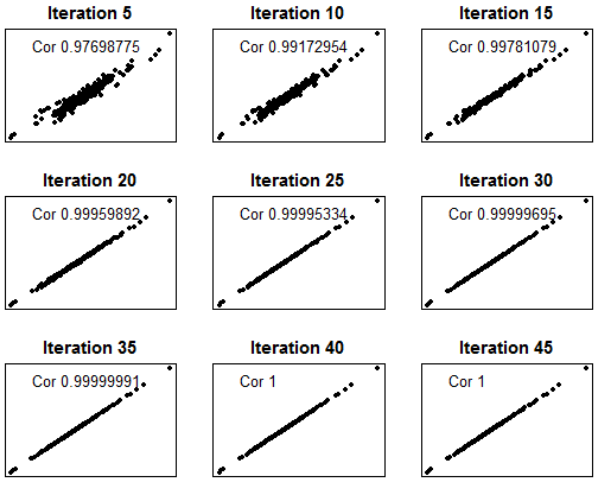
- Thus

$$\nabla = n^{-1}(-2X'e + 2\lambda\beta^{t}) = -2n^{-1}(X'e - \lambda\beta^{t})$$

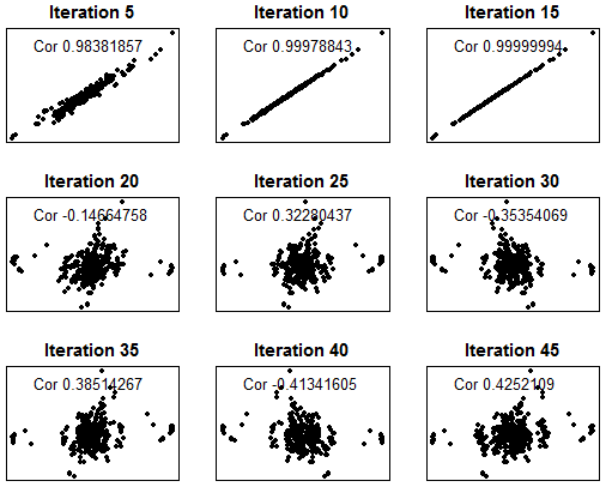
$$\boldsymbol{\beta^{t+1} = \beta^{t} + \frac{2\alpha}{n}(X'e - \lambda\beta^{t})}$$

Scatter plot between true solution and solution at iteration x

Small learning rate



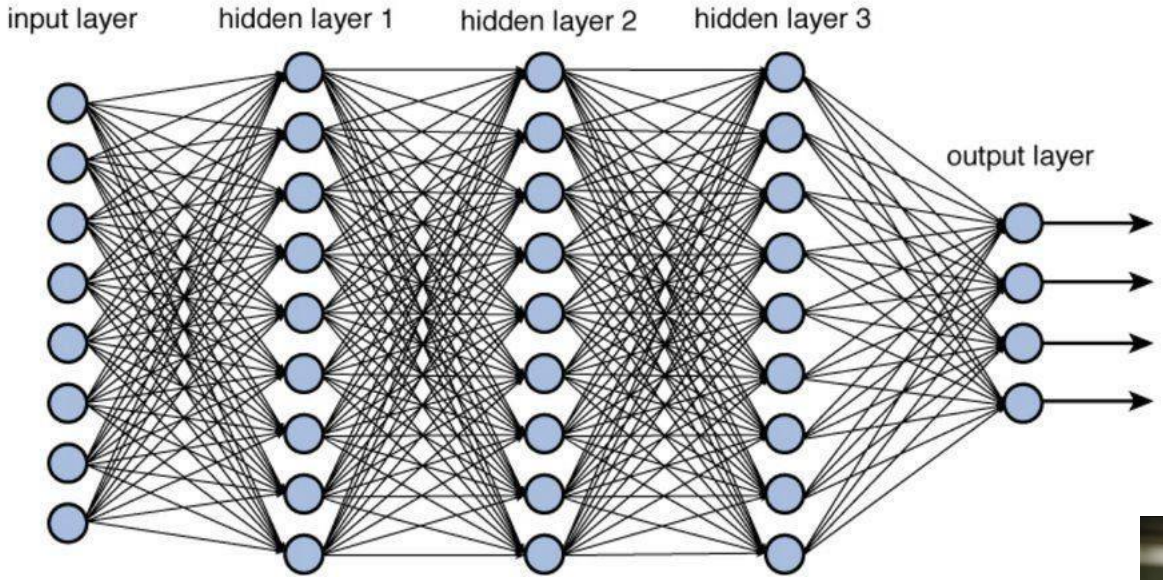
Large learning rate

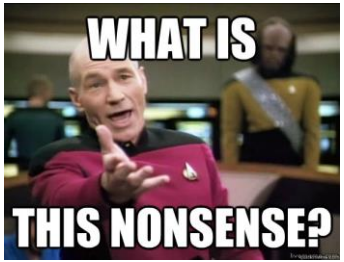


CORTEVA™ agriscience

Alencar.Xavier@Corteva.com
Corteva Biostatistics, Methods group

Deep Neural Network





Alencar.Xavier@Corteva.com
Corteva Biostatistics, Methods group

From Ridge Regression to Deep Neural Net

Models illustrated without intercept

Linear model

$$y = Xb + e$$

PLS/PCR model

$$y = (XB_1)b_2 + e$$

NN model

$$y = \alpha(XB_1)b_2 + e$$

Deep NN model

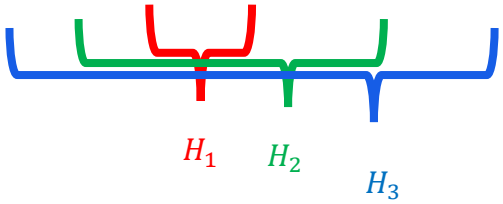
$$y = \alpha(\alpha(XB_1)B_2)b_3 + e$$

α = activation function

Alencar.Xavier@Corteva.com
Corteva Biostatistics, Methods group

The ~~Scary~~ Deep Neural Network

$$Y = \alpha(\alpha(XB_1)B_2)B_3 + E$$



- **Fit layers**
  - $H_1 = \alpha(XB_1)$
  - $H_2 = \alpha(H_1B_2)$
  - $H_3 = H_2B_3$
- **Compute gradients** (aka. get residuals)
  - $E_3 = Y - H3$
  - $E_2 = \alpha(E_3B_3')$
  - $E_1 = \alpha(E_2B_2')$
- **Update coefficient**
  - $B_1 = B_1 - \gamma\left(\frac{2r_1}{n}[\,X'E_1 - \lambda B_1]\right)$
  - $B_2 = B_2 - \gamma\left(\frac{2r_2}{n}[\,H_1'E_2 - \lambda B_2]\right)$
  - $B_3 = B_3 - \gamma\left(\frac{2r_3}{n}[\,H_2'E_3 - \lambda B_3]\right)$

Top-to-bottom code ~ 30 lines

```
# DNN Function
dnn = function(y, X, nit=1000, batch=250,RELU=FALSE, Leak=0.1,
               dropout=0, Lambda=0.1,LrnRate = 1,Nodes1=4, Nodes2=4){
  # Normalization
  if(is.null(ncol(y))) y = matrix(y)
  muY = colMeans(y,na.rm=T);  sdY = apply(y,2,sd,na.rm=T);  y = apply(y,2,scale)
  ActFun = tanh;  if(RELU) ActFun = function(x){x[x<0]=x[x<0]*Leak;return(x)}
  DropOut = function(x,prc=dropout){x[sample(length(x),length(x)*prc)]=0;return(x)}
  # Dimensions & Learning settings
  n = nrow(X);  p = ncol(X);  k = ncol(y)
  n1 = Nodes1; n2 = Nodes2;  lmb = Lambda; rate = LrnRate/c(p,n1,n2)
  b1 = matrix(rnorm(n1*p,0,1/p),p,n1)
  b2 = matrix(rnorm(n1*n2,0,1/n1),n1,n2)
  b3 = matrix(rnorm(n2,0,1/n2),n2,k)
  CNV1 = CNV2 = c()
  for(i in 1:nit){ # Iterations (backprop)
    if((i+1)%%100==1) cat('iteration',i,'\n') # Sample batch
    w = sample(n,batch,replace=T);  y0 = y[w,]; X0 = X[w,]
    # Fit hidden layers (1) and Gradients (2)
    H1 = ActFun(X0%*%b1); H2 = ActFun(H1%*%b2); H3 = H2%*%b3
    e3 = y0-H3; if(anyNA(e3)) e3[is.na(e3)]=0
    e2 = ActFun(e3 %*% t(b3));  e1 = ActFun(e2 %*% t(b2))
    # Update coefficients
    b1 = b1 + DropOut(t(X0)%*%(e1)-lmb*b1)*(2/n)*rate[1]
    b2 = b2 + DropOut(t(H1)%*%(e2)-lmb*b2)*(2/n)*rate[2]
    b3 = b3 + DropOut(t(H2)%*%(e3))*(2/n)*rate[3]
    CNV1 = c(CNV1,mean(apply(e3,2,var,na.rm=T)))
    CNV2 = c(CNV2,mean(diag(cor(H3,y0,use='p'))))
  }
  out = list(af=ActFun,b1=b1,b2=b2,b3=b3,conv_MSE=CNV1,conv_GOF=CNV2,mu=muY,sd=sdY)
  class(out) = 'smalldnn'; return(out)}

predict.smalldnn = function(object,newdata){
  x = object; h = x$af(x$af(newdata%*%x$b1)%*%x$b2)%*%x$b3
  for(i in 1:ncol(h)) h[,i] = x$mu[i] + x$sd[i]*h[,i]
  return(h)}
```


Alencar.Xavier@Corteva.com
Corteva Biostatistics, Methods group

Thank you for your attention!

Remarks:

1) ML can be a powerful tool for plant breeding
2) Cross-validation must be carefully design to understand the machinery
3) The scary black boxes are usually simple methods

Questions??

Alencar Xavier

Alencar.Xavier@Corteva.com

Some free lit to look up!