Bayesian construction of perceptrons to predict phenotypes from 584K SNP data.

Luc Janss, Bert Kappen
Radboud University Nijmegen Medical Centre
Donders Institute for Neuroscience

Introduction

- Genetic prediction still difficult
  - But also done very simple, testing single predictors....
- Here: Bayesian multivariate model building
  - Based on “Variable Selection” - George & McCullogh 1993
  - Constructs complete “perceptrons”: the simultaneous bit-pattern for selected predictors
- Example: human data 632 phenotypes + orginally 710K SNPs (p/n >1000)
  - > Using 3 chromosomes, 104K SNPs
  - > Cross validation using 80:20 split

Difficult prediction of genetics: common practice now

- GWAS studies: univariate SNPs associations
  - Based on Least Squares (un-shrunken estimates)
  - Requires p values 10^{-5} – 10^{-7}
  - Significant ones are highly overestimated
  - But OK for testing association
- Prediction: use univariate tested SNPs from GWAS
  - Large false negative rate (simultaneous testing)
  - Can't trust effects estimates (use “positive allele index”)
  - Models so put to together don't predict well

Failure to predict human height: "the case of the missing heritability"

The case of the missing heritability

When scientists opened up the human genome, they expected to find the genetic components of common traits and diseases. But they were nowhere to be seen. **Brendan Maher** shines a light on six places where the missing loot could be stashed away.

Nature, 6 Nov. 2008

- Height is 80-90% heritable, so with DNA markers one should be able to predict a person's height well
- Large scale studies used 500K SNP arrays, and came up with 40 markers (individually tested in LS regression models and requiring $P<10^{-7}$)
- Yet, these markers only explained 5% of variation ....

Better ideas for prediction?

- GWAS may be OK for testing association but looks too simplistic for model building for prediction
  - Common dogma in statistics is that prediction needs shrunken estimates (e.g. Bayesian or mixed models)
  - Predictors are often colinear requiring multivariate model building
- With large p huge space to find multivariate models!
  - > Machine learning techniques or Bayesian models using Markov chain Monte Carlo

Bayesian multivariate model building

- Observation vector $\mathbf{y}$ (size n)
- Sets of possible predictors (covariates) $\mathbf{x}_j$ (j=1, .. p)
- A particular model can be presented using the perceptron (bit pattern, size p) that indicates which $\mathbf{x}_j$ are selected:

  M_i = (0 0 0 1 0 1 1 0 0 0 1 0 1 1 1 0 1 0 0 1)

  $\mathcal{M}$ is set of all possible bit patterns (2^p)

- How to search through $\mathcal{M}$?
- How to determine "best" multivariate model?

Bayesian variable selection

- Bayesian procedures would typically consider to add/remove one predictor, or make one "point mutation" in the perceptron:

$M_i = (\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ )$

$M_{i*} = (\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ )$

- This is a switch between models of different dimension → would require Reversible Jump
  - Difficult to tune, need to generate "good" proposals

Gibbs sampling alternative

(George & McCullogh, 1993)

- Does not switch in/out predictors but switches them between mild/heavy shrinkage using a mixture model
- Considers a model including *all* predictors:

$$\mathbf{y} = \mathbf{1}\mu + b_1\mathbf{x}_1 + b_2\mathbf{x}_2 + \ldots b_p\mathbf{x}_p + \mathbf{e}$$

$$b_j \sim \begin{cases} \pi_0 \mathrm{N}(0,\tau_0) & \text{If "unselected" (bit=0)} \\ \pi_1 \mathrm{N}(0,\tau_1) & \text{If "selected" (bit=1)} \end{cases}$$

Hierarchical model on b_j's

- τ_0 is set very small applying very heavy shrinkage (effectively removing) for "unselected" predictors
- π_0 is set big (>0.99) to "remove" most predictors

Gibbs sampling

- The G&McC alternative can be implemented using Gibbs sampling to "point-mutate" the bits.
- Posterior density is:

$$\underbrace{(\sigma_e^2)^{-n/2}\exp(-(y-1\mu-\sum b_j x_j)^T(...)/2\sigma_e^2)}_{\text{Likelihood part}}\underbrace{\prod(\tau_{\gamma_j})^{(-1/2)}\exp(-b_j^2/2\tau_{\gamma_j})}_{\text{Shrinkage of } b_j \text{ according to bit indicator } \gamma_j.}\prod f(\gamma_j)$$

- Leads to a Bernouilli distribution to resample the bit indicator γ_j.
- Switching the bits using this sampling step works well (the bit is switched without changing the b_j effect or model fit)

Model settings

- Set $p\tau_0$ to be <1% of var(**y**)
  - "Switched off" predictors fit about 1% of noise, here needs $\tau_0 = 2\text{x}10^{-7}\text{var}(\mathbf{y})$
- From idea of true # predictors (e.g. 50-200 SNPs)
  - Set τ_1, or estimate with a weak prior (done here), prior is here $3\text{x}10^{-2}\text{var}(\mathbf{y})$ with 3DF
  - Set π_1 :here 0.04% (4/10K, 40/100K)
- Conditional probs for γ_j involve $\sqrt{\tau_1/\tau_0}\,\pi_0/\pi_1$

Construction of prediction models

- Based on marginal posterior probability for inclusion of a predictor in the perceptrons
  - Simple, maybe too simple with colinear predictors?
- Based on one or more (or most likely) perceptron generated in the MCMC
  - Looks ideal, but difficult in practice: many small variations between generated perceptrons, identical ones are hardly ever produced (with large p).
  - Use MCMC cycles producing low residual variance
- Using all generated perceptrons (basically weighted according to posterior probability)

Re-fit of selected markers in training

- For use of single perceptron or markers selected according to marginal probabilities, a non-mixture model was re-trained:

  $\mathbf{y} = \mathbf{1}\mu + b_1\mathbf{x}_1 + b_2\mathbf{x}_2 + ... b_p\mathbf{x}_p + \mathbf{e}$

  $b_j \sim \mathrm{N}(0, \sigma_b^2), \sigma_b^2 \sim U(0, \infty)$

  with selected predictors (p<50)

- Works also for co-linear predictors

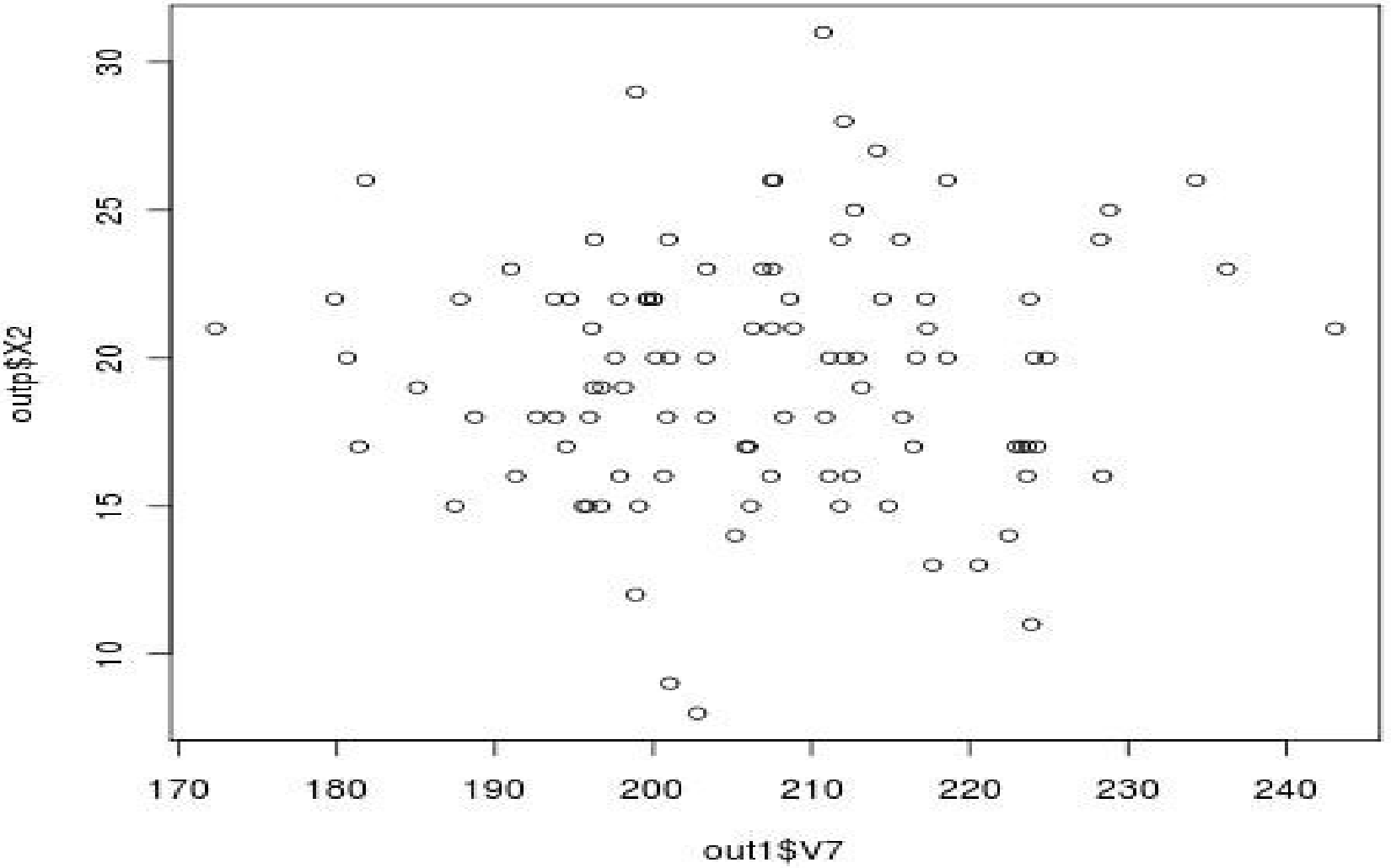
Example data set

- Human data with continuous psychological traits on 632 children (attention, concentration scores)
  - Medium-good heritabiliy (genetics contribute 30-50%)
- 704K SNPs, using 3 chromosomes only which are known to have important regions → 104K SNPs
- Data was split once 80:20 (524:108) for training and testing in cross validation
- A factor Language (5 classes) was fitted in training data and estimates also added to prediction

Prediction using perceptrons

- Averaged over perceptrons
  - Prediction correlation 0.1852
  - Variance explained in training data 11.4%
  - Using 8-31 SNPs per perceptron out of 1064
- “Best” perceptron explained in training data 26.6% with 21 SNPs
  - Refit in non-mixture model gave 15.1% explained in training
  - Prediction correlation 0.1826

Explained variance and number of used markers from MCMC



Prediction based on sets of predictors formed using marginal probabilities

Top #	PostProb cut off	Prediction corr
5	0.50	0.215
6	0.20	0.191
9	0.15	0.156
12	0.10	0.096
18	0.07	0.051
21	0.06	0.028

- GWAS (Least Squares) had the same #1 and #3, but selected #2 in a somewhat different position, and missed #4, #5 (and further ...).

Future work

- Aim to identify “most probable” multivariate model
  - MCMC generates nearly all unique patterns
  - Clustering procedures under development
- Testing “significance” in some way, e.g. checking bits switched on in 3 regions:

>= 1	0	0	21.7%
1	0	0	17.5%
>= 1	>= 1	0	16.3%
0	0	0	13.2%
>= 1	0	>= 1	12.8%
>= 1	>= 1	>= 1	10.7%

Conclusions

- Need to separate testing association and prediction
- “Marginal” selection of predictors was good too, but:
  - After re-estimation in Bayesian shrinkage model
  - And only for small top-list (not for larger top lists)
- Selection of multivariate model not optimal yet, current choice fits more variance in training but not in testing
- Use of Least Squares remains unsatisfactory for prediction
- Variable selection still worked with p/n around 200, no large signs of overfit.