Mildly penalized maximum likelihood estimation of genetic covariances matrices without tuning

Karin Meyer

Animal Genetics and Breeding Unit
University of New England, Armidale

21st Conference of the Association for Animal Breeding and Genetics, Lorne, September 28-30, 2015

kmeyer@une.edu.au

Objectives

What?
• Obtain ‘better’ estimates of genetic parameters

How?
• Reduce sampling variances & ‘loss’
  ~⇒ penalty on REML likelihood

Why?
• Estimates are closer to true values
  ~⇒ more genetic progress

Penalized REML

- Shrink canonical eigenvalues $\lambda_i$ towards mean $\bar{\lambda}$
- Assume: $\lambda_i \sim \text{Beta}(\alpha, \beta)$
  $$\mathcal{P} \propto \sum_i (\alpha - 1) \log(\lambda_i) + (\beta - 1) \log(1 - \lambda_i)$$
  $$\alpha = 1 + \bar{\lambda}(N_E - 2) \quad \beta = 1 + (1 - \bar{\lambda})(N_E - 2)$$
- Pen. likelihood: $\log \mathcal{L}_P = \log \mathcal{L} - \mathcal{P}$
- Use $N_E = \alpha + \beta$ to regulate strength of penalty
  ~⇒ ‘effective sample size’

Criterion

- **Loss** measures matrix divergence
  ~⇒ Estimates $\hat{\Sigma}$ vs. population values $\Sigma$
  $$L(\Sigma, \hat{\Sigma}) = \text{tr}(\Sigma^{-1}\hat{\Sigma}) - \log|\Sigma^{-1}\hat{\Sigma}| - q$$

Simulation

- $q = 9$ traits, 72 sets of pop. values, 500 replicates
- Paternal half-sib design with 10 progeny per sire
  - Small: 100 sires, Large: 1000 sires

Results

- Substantial reductions in loss feasible
  - Mean PRIAL $\approx 50\%$ for small sample
- Spread increases with $N_E$
  - Low PRIAL if pop. values $\neq$ assumed distribution
  - Negative PRIAL flags too stringent penalization
- Estimation of $N_E$ from data not very successful
  - Laborious & often too low
- Default value $N_E \approx 4$ suited to wide range of pop. values & sample sizes
  - Worthwhile reductions in loss for many cases
  - No detrimental effects

Conclusions

- Mild penalty on REML estimates of covariance matrices recommended
  - can yield substantial & large proportions of possible reductions in loss
  - can identify adequate defaults for tuning-free application
- No increase in computational complexity ⇒ suitable for routine use

Implemented in WOMBAT