Cryptography & Statistics: a short introduction

Louis J. M. Aslett (louis.aslett@durham.ac.uk)

Department of Mathematical Sciences
Durham University

Short Course, Part II

54th Gregynog Statistical Conference
March 2018

Outline

1. Existing Statistical Methodology
   - Survey of literature using cryptographic methods with (mostly) standard statistics methodology.
2. Machine Learning
   - Homomorphic encryption.
   - Novel variant on random forests (and naïve Bayes).
3. Arbitrary Bayesian Models
   - Homomorphic secret sharing.
   - Theoretically arbitrary low-dimensional model fitting on securely pooled data. Model security also guaranteed.

Motivation

Security in statistics and machine learning applications is a growing concern:

- computing in a ‘hostile’ environment (e.g. cloud computing);
- donation of sensitive/personal data (e.g. medical/genetic studies);
- complex models on constrained devices (e.g. smart watches)
- running confidential algorithms on confidential data (e.g. engineering reliability)

Existing Statistical Methodology

Existing Statistical Methodology

A lightning tour of some of the encrypted statistics literature.

We will see the majority of work so far is existing algorithms simply refactored to run homomorphically.

Existing Statistical Methodology

A lightning tour of some of the encrypted statistics literature.

We will see the majority of work so far is existing algorithms simply refactored to run homomorphically.

Call to arms! Statisticians can develop novel approaches to approximate otherwise currently intractable statistical techniques.

Graepel et al. (2012) Best name! 'ML Confidential'

Assume know which observations are same class, so separated operations.

Linear means classifier, binary $b \in \{-1, +1\}$. Let $s_y = \sum_{i \in I_y} x_i$

$$\left.\begin{array}{ll} \mu_y = n_y^{-1} s_y & w = \mu_{+1} - \mu_{-1} \\ c = (\mu_{+1} - \mu_{-1})^T (\mu_{+1} - \mu_{-1})/2 & \end{array}\right\} \text{predict:} \quad \text{sign}(w^T x^\star - c)$$

Graepel et al. (2012) Best name! 'ML Confidential'

Assume know which observations are same class, so separated operations.

Linear means classifier, binary $b \in \{-1, +1\}$. Let $s_y = \sum_{i \in I_y} x_i$

$$\left.\begin{array}{ll} \mu_y = n_y^{-1} s_y & w = \mu_{+1} - \mu_{-1} \\ c = (\mu_{+1} - \mu_{-1})^T (\mu_{+1} - \mu_{-1})/2 & \end{array}\right\} \text{predict:} \quad \text{sign}(w^T x^\star - c)$$

Transform:

$$\begin{aligned} \tilde{w} &= n_{-1} s_{+1} - n_{+1} s_{-1} \\ &= n_{+1} n_{-1} w \\ \tilde{c} &= (m_{-1} s_{+1} - m_{+1} s_{-1})^T (m_{-1} s_{+1} + m_{+1} s_{-1}) \\ &= 2 n_{+1} n_{-1} c \end{aligned}$$

$$\text{predict:} \quad \text{sign}(2 n_{+1} n_{-1} \tilde{w}^T x^\star - \tilde{c})$$

Graepel et al. (2012) Best name! 'ML Confidential'

Assume know which observations are same class, so separated operations.

Linear means classifier, binary $b \in \{-1, +1\}$. Let $s_y = \sum_{i \in I_y} x_i$

$$\left.\begin{array}{ll} \mu_y = n_y^{-1} s_y & w = \mu_{+1} - \mu_{-1} \\ c = (\mu_{+1} - \mu_{-1})^T (\mu_{+1} - \mu_{-1})/2 & \end{array}\right\} \text{predict:} \quad \text{sign}(w^T x^\star - c)$$

Transform:

$$\begin{aligned} \tilde{w} &= n_{-1} s_{+1} - n_{+1} s_{-1} \\ &= n_{+1} n_{-1} w \\ \tilde{c} &= (m_{-1} s_{+1} - m_{+1} s_{-1})^T (m_{-1} s_{+1} + m_{+1} s_{-1}) \\ &= 2 n_{+1} n_{-1} c \end{aligned}$$

$$\text{predict:} \quad \text{sign}(2 n_{+1} n_{-1} \tilde{w}^T x^\star - \tilde{c})$$

Similar approach for Fisher's Linear Discriminant Classifier

Wu & Haven (2012) : Linear Regression (low-d)

Observe that mean vector and covariance matrix of design matrix can be computed with low multiplicative depth.

$$\Sigma = \frac{1}{n^2}\left(nX^TX - (n\mu)(n\mu)^T\right)$$

However, store numerator and denominator of fraction as separate ciphertexts.

They propose Cramer's rule for inverse:

$$(X^TX)^{-1} = \frac{1}{\det(X^TX)}\mathrm{Adj}(X^TX)$$

Problem is explosion in multiplicative depth. Consider max 5 dimensions.

Implement Chinese Remainder Theorem for SIMD evaluation of matrix product.

Esperança, Aslett & Holmes (2017)

We propose accelerated gradient descent methods. For linear regression, this can be written:

$$\beta^{[k]} = \sum_{n=1}^{k} (-1)^{n+1} \binom{k}{k-n} \delta^n (X^T X)^{n-1} X^T y$$

Prove this is oscillatory $\implies$ van Wijngaarden transformation can be applied.

Competitive with Nesterov acceleration and, more importantly, for k steps we prove

$$\begin{aligned} \text{max depth std gradient descent} &= 2k \\ \text{max depth van Wijngaarden} &= 2k+1 \\ \text{max depth Nesterov} &= 3k \end{aligned}$$

For low fixed multiplicative depth, vWT often *outperforms* Nesterov (factor of 2 in error norm for very low depth).

Esperança, Aslett & Holmes (2017) (contd.)

$\mathring{m}(x) = \sum_{i=0}^{2^{d-1}-1} a_i x^i \in R_t$

$n = \max\{i : a_i > 0\}$

Lemma (FV parameter requirements for GD)

If data is represented in binary decomposed polynomial form, then after running the ELS–GD *algorithm the degree and coefficient value of the encrypted regression coefficients is bound by:*

$$\deg(\tilde{\beta}^{[k]}) \leq \max\{4n + \deg(\tilde{\beta}^{[k-1]}), (4k-1)n\}$$

where $\deg(\beta^{[1]}) \leq 3n$ *and* $n \equiv (\phi+1)\log_2(10)$*;*

$$\text{and } ||\tilde{\beta}^{[k]}||_\infty \leq (4n + (n+1)^2)NP||\tilde{\beta}^{[k-1]}||_\infty + (4k-3)n(n+1)N$$

where $||\tilde{\beta}^{[1]}||_\infty \leq n(n+1)N$

Esperança, Aslett & Holmes (2017) (contd.)

Tested up to 25 dimensions.

Real examples:

	Mood Stability	Prostate Cancer
N	28	97
P	2	8
$\|\|\hat{\beta}\|\|_\infty$	0.04	0.26
Memory	15 MB	3.5 GB
Time	12 secs	30 mins

Gascón et al. (2017)

Multiparty computing version of linear regression problem: different variables held by different parties (ie vertially partitioned design matrix). Uses new fixed-point precision conjugate gradient descent. algorithm.

Achieves substantial performance improvement in computing inner product compared to standard MPC approach using OT.

$N > 1,000,000$, $P = 100$ fitted in under 1 hour.

Work has led to software suite for MPC algorithms.

Lauter et al. (2014)

Pearson goodness-of-fit, linkage disequilibrium, EM-algorithm for haplotyping and Cochran-Armitage testing for genomic data. Main contribution is data representation allowing algorithms to be recast.

Genotype encoding: (AA, Aa, aa) usually represented $(0, 1, 2)$. Instead, each locus for each person encoded:

$$\begin{aligned}
AA \text{ (value 0)}: &\quad x_0 \leftarrow \mathsf{Enc}(k_p, 1), \quad x_1 \leftarrow \mathsf{Enc}(k_p, 0), \quad x_2 \leftarrow \mathsf{Enc}(k_p, 0) \\
Aa \text{ (value 1)}: &\quad x_0 \leftarrow \mathsf{Enc}(k_p, 0), \quad x_1 \leftarrow \mathsf{Enc}(k_p, 1), \quad x_2 \leftarrow \mathsf{Enc}(k_p, 0) \\
aa \text{ (value 2)}: &\quad x_0 \leftarrow \mathsf{Enc}(k_p, 0), \quad x_1 \leftarrow \mathsf{Enc}(k_p, 0), \quad x_2 \leftarrow \mathsf{Enc}(k_p, 1) \\
\text{missing}: &\quad x_0 \leftarrow \mathsf{Enc}(k_p, 0), \quad x_1 \leftarrow \mathsf{Enc}(k_p, 0), \quad x_2 \leftarrow \mathsf{Enc}(k_p, 0)
\end{aligned}$$

Phenotype encoding:

$$\begin{aligned}
\text{unaffected (value 0)}: &\quad z_0 \leftarrow \mathsf{Enc}(k_p, 1), \quad z_1 \leftarrow \mathsf{Enc}(k_p, 0) \\
\text{affected (value 1)}: &\quad z_0 \leftarrow \mathsf{Enc}(k_p, 0), \quad z_1 \leftarrow \mathsf{Enc}(k_p, 1) \\
\text{missing}: &\quad z_0 \leftarrow \mathsf{Enc}(k_p, 0), \quad z_1 \leftarrow \mathsf{Enc}(k_p, 0)
\end{aligned}$$

Bost et al. (2015)

Focus on encrypted prediction only (assume model is already trained).

- Hyperplane decision
- Naïve Bayes
- Decision Trees

Main contribution is a communication intensive method to compute

$$\arg\max_i \{x_1, \ldots, x_k\}$$

using only Paillier (1999), $M = \mathbb{Z}_N, \mathcal{F}_M = \{+\}, N \approx 2^{1024}$, and Goldwasser & Micali (1982), $M = \mathbb{F}_2, \mathcal{F}_M = \{+\}$.

Involves an information theoretically secure method to switch between schemes.

Machine Learning

Machine Learning Encrypted?

Lots of constraints! Are traditional machine learning techniques out of reach to run on encrypted data? We've looked at a semi-parametric naïve Bayes and a variant of random forests.

Machine Learning Encrypted?

Lots of constraints! Are traditional machine learning techniques out of reach to run on encrypted data? We've looked at a semi-parametric naïve Bayes and a variant of random forests.

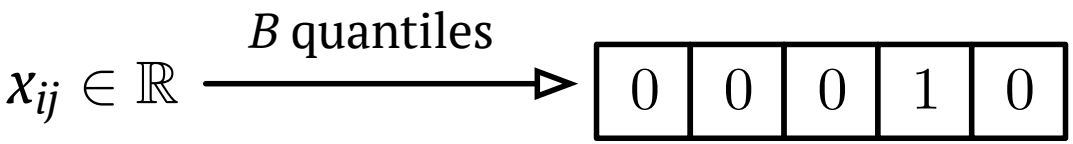
So, want to build a random forest on encrypted data … but, recall from part I:

- No comparisons possible to evaluate splits
- No max possible to find highest class vote
- No division possible to do average votes
- …

Thus random forests (and other methods) need to be tailored for encrypted computation. This is where statistics and machine learning community can get involved!

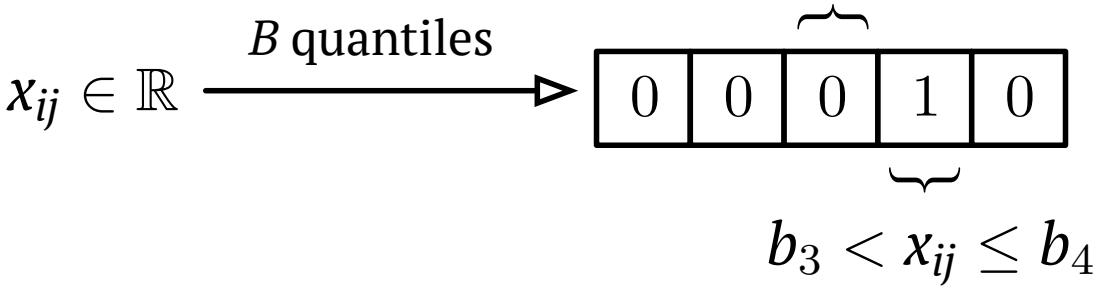
Completely Random Forests (CRFs) — Data encoding

1



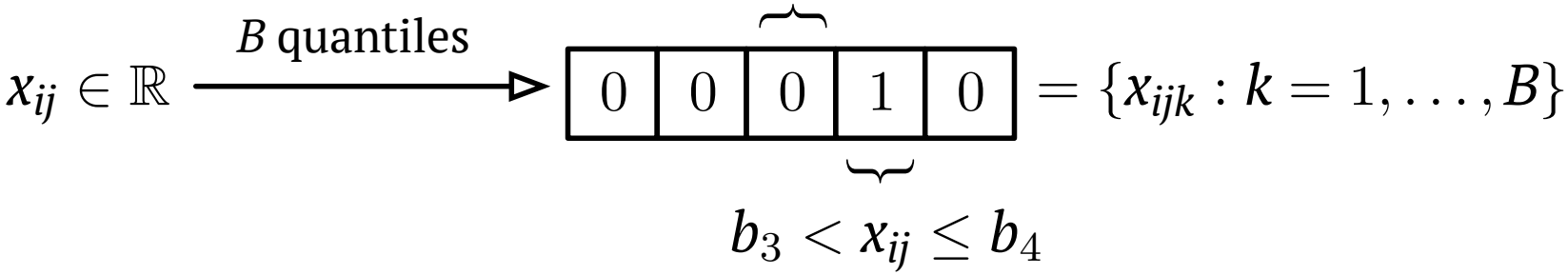
Completely Random Forests (CRFs) — Data encoding

1. $b_0 := -\infty$ $\qquad$ $b_B := \infty$



Completely Random Forests (CRFs) — Data encoding

1. $b_0 := -\infty$ $\qquad$ $b_B := \infty$



Completely Random Forests (CRFs) — Data encoding

1. $$b_0 := -\infty \qquad\qquad b_B := \infty$$

$$b_2 < x_{ij} \leq b_3$$

$$x_{ij} \in \mathbb{R} \xrightarrow{B \text{ quantiles}} \underbrace{\boxed{0}\boxed{0}\overbrace{\boxed{0}}^{}\underbrace{\boxed{1}}_{}\boxed{0}}_{} = \{x_{ijk} : k = 1, \ldots, B\}$$

$$b_3 < x_{ij} \leq b_4$$

2. Then,

$$\mathbb{I}(x_{ij} \leq b_l) = \sum_{k=1}^{l} x_{ijk} \quad \text{and} \quad \mathbb{I}(x_{ij} > b_l) = \sum_{k=l+1}^{B} x_{ijk}$$

Completely Random Forests (CRFs) — Data encoding

1. $$b_0 := -\infty \qquad\qquad b_B := \infty$$

$$b_2 < x_{ij} \leq b_3$$

$$x_{ij} \in \mathbb{R} \xrightarrow{B \text{ quantiles}} \boxed{0}\boxed{0}\boxed{0}\boxed{1}\boxed{0} = \{x_{ijk} : k = 1, \ldots, B\}$$

$$b_3 < x_{ij} \leq b_4$$

2. Then,

$$\mathbb{I}(x_{ij} \leq b_l) = \sum_{k=1}^{l} x_{ijk} \quad \text{and} \quad \mathbb{I}(x_{ij} > b_l) = \sum_{k=l+1}^{B} x_{ijk}$$

3. Similarly encode response category c, $y_i \to y_{ic} \in \{0, 1\}$.

Completely Random Forests (CRFs) — Data encoding

1. $$b_0 := -\infty \qquad\qquad b_B := \infty$$

$$b_2 < x_{ij} \leq b_3$$

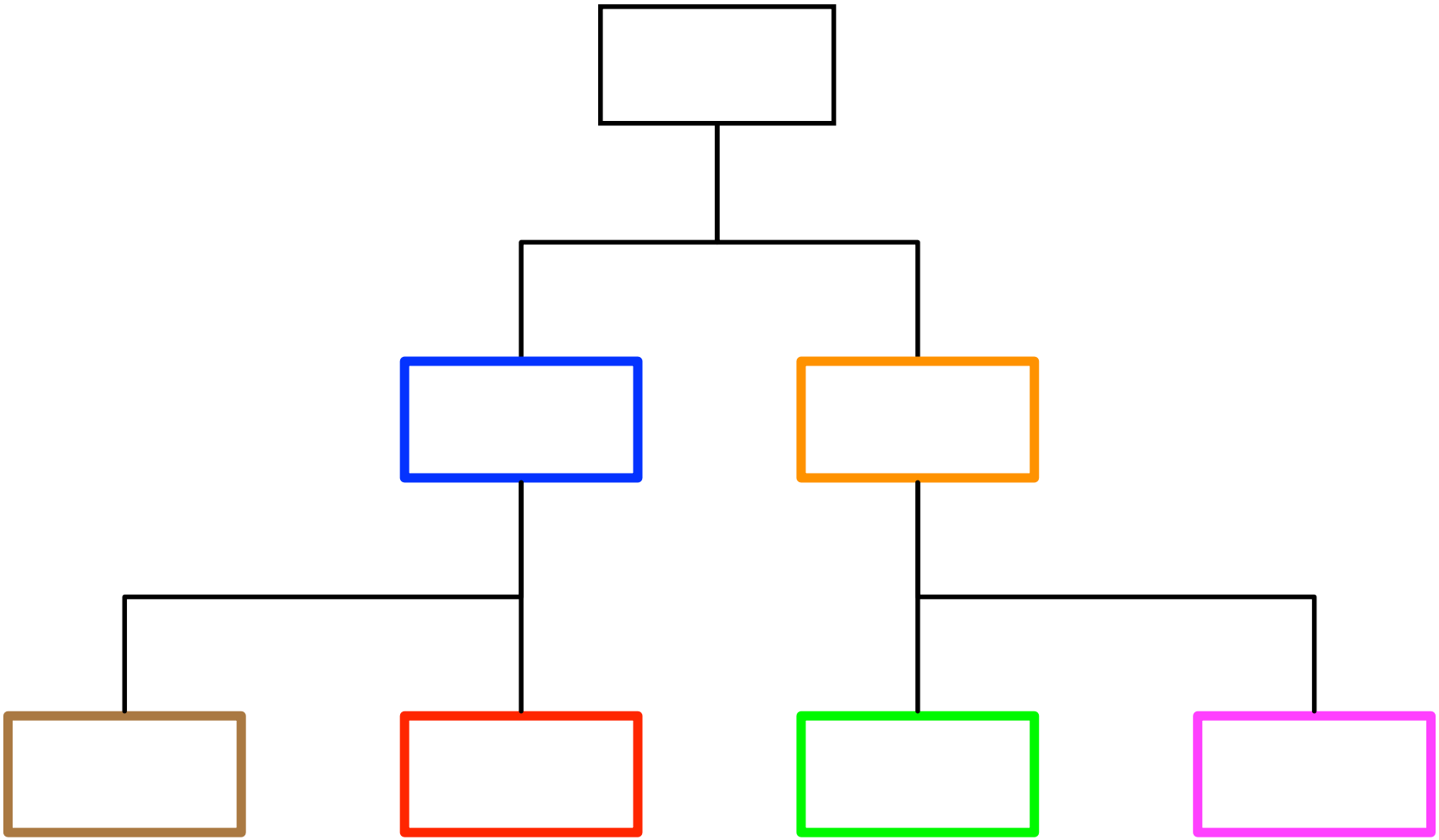
$$x_{ij} \in \mathbb{R} \xrightarrow{B \text{ quantiles}} \underbrace{\boxed{0}\boxed{0}\boxed{0}\boxed{1}\boxed{0}}_{b_3 < x_{ij} \leq b_4} = \{x_{ijk} : k = 1, \ldots, B\}$$

2. Then,

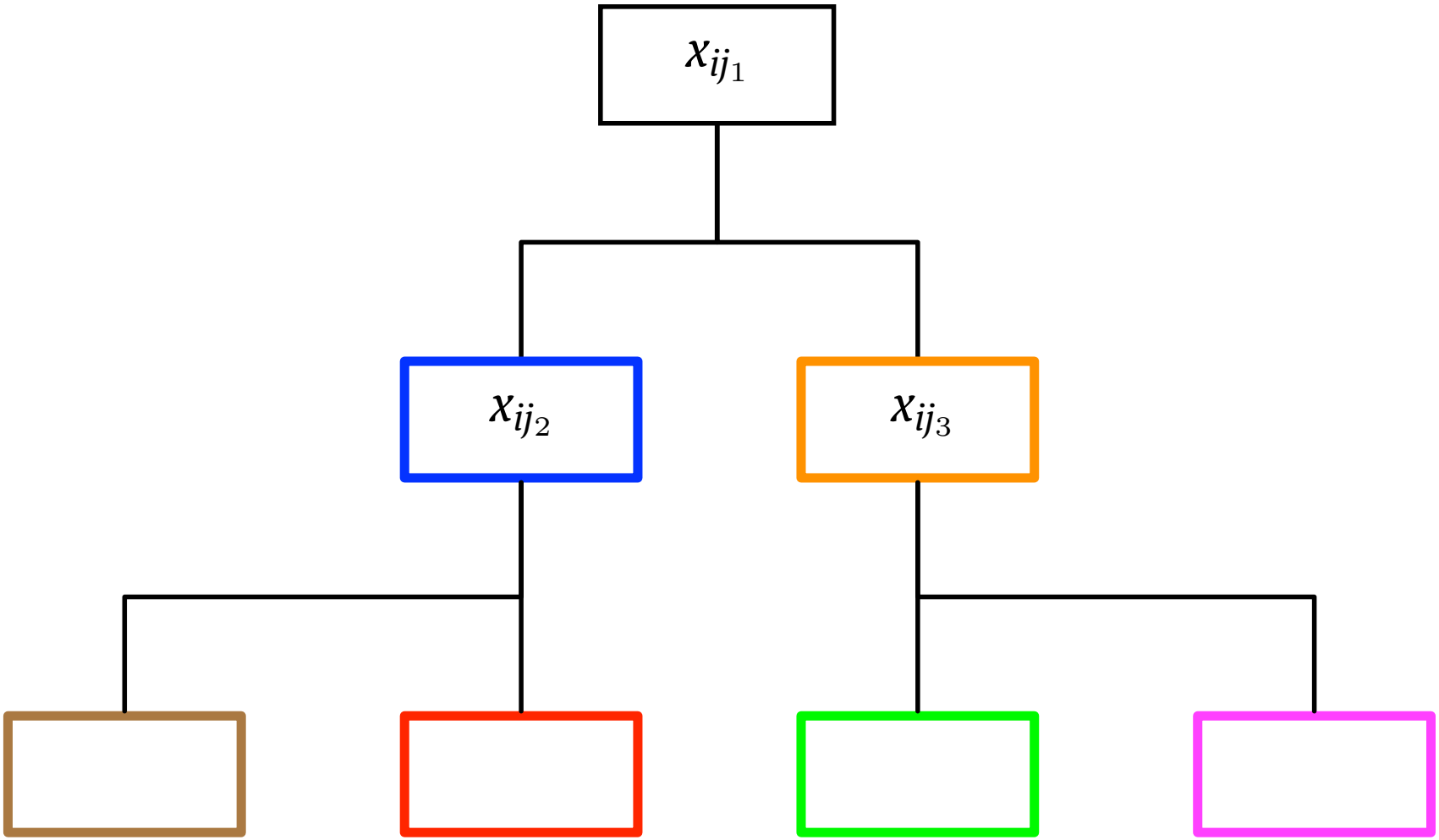
$$\mathbb{I}(x_{ij} \leq b_l) = \sum_{k=1}^{l} x_{ijk} \quad \text{and} \quad \mathbb{I}(x_{ij} > b_l) = \sum_{k=l+1}^{B} x_{ijk}$$

3. Similarly encode response category c, $y_i \to y_{ic} \in \{0, 1\}$.
4. Build a decision tree selecting variable j and split point b_l *completely* at random to a fixed depth.

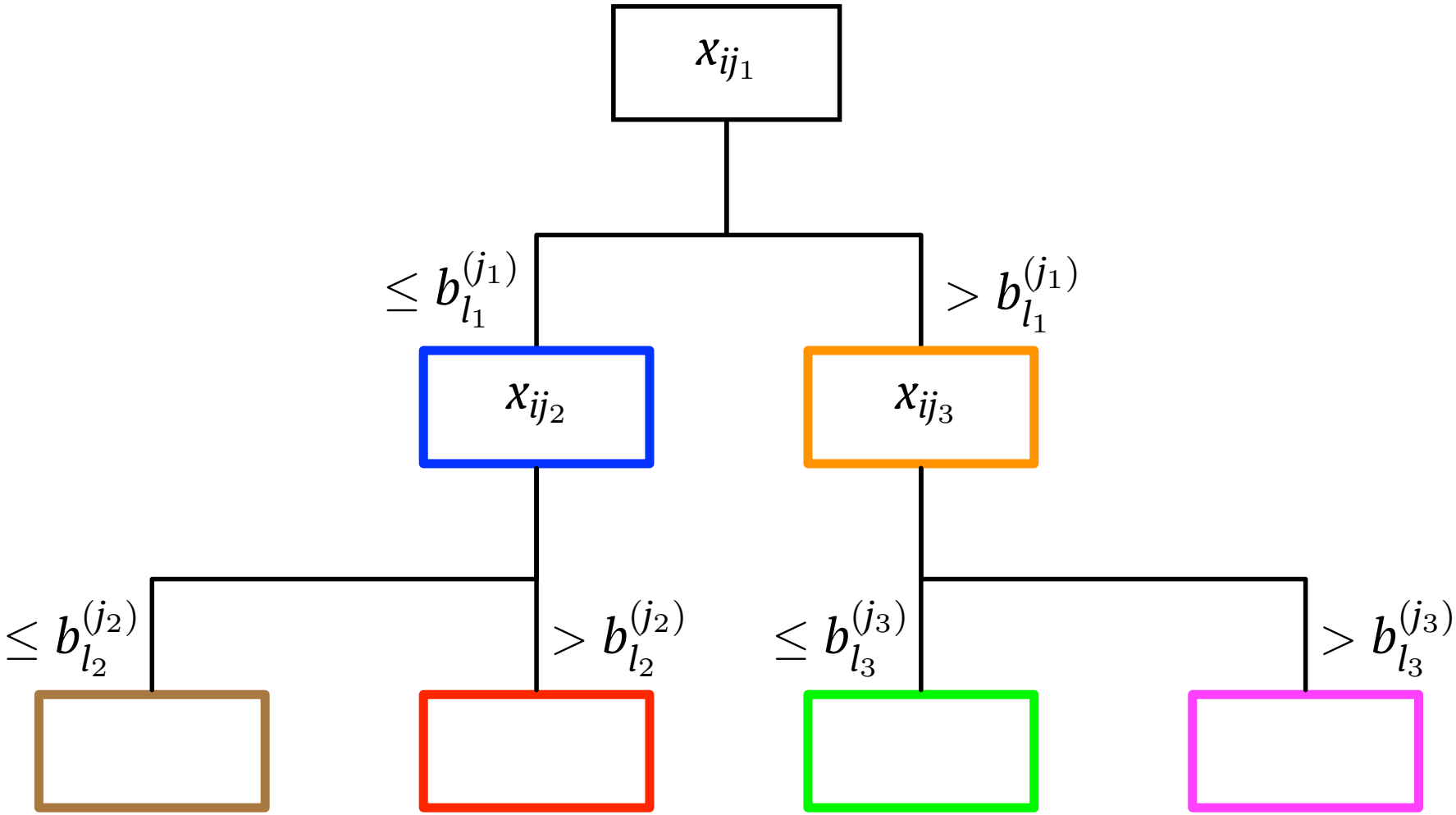
CRFs — Tree ‘fitting’, I



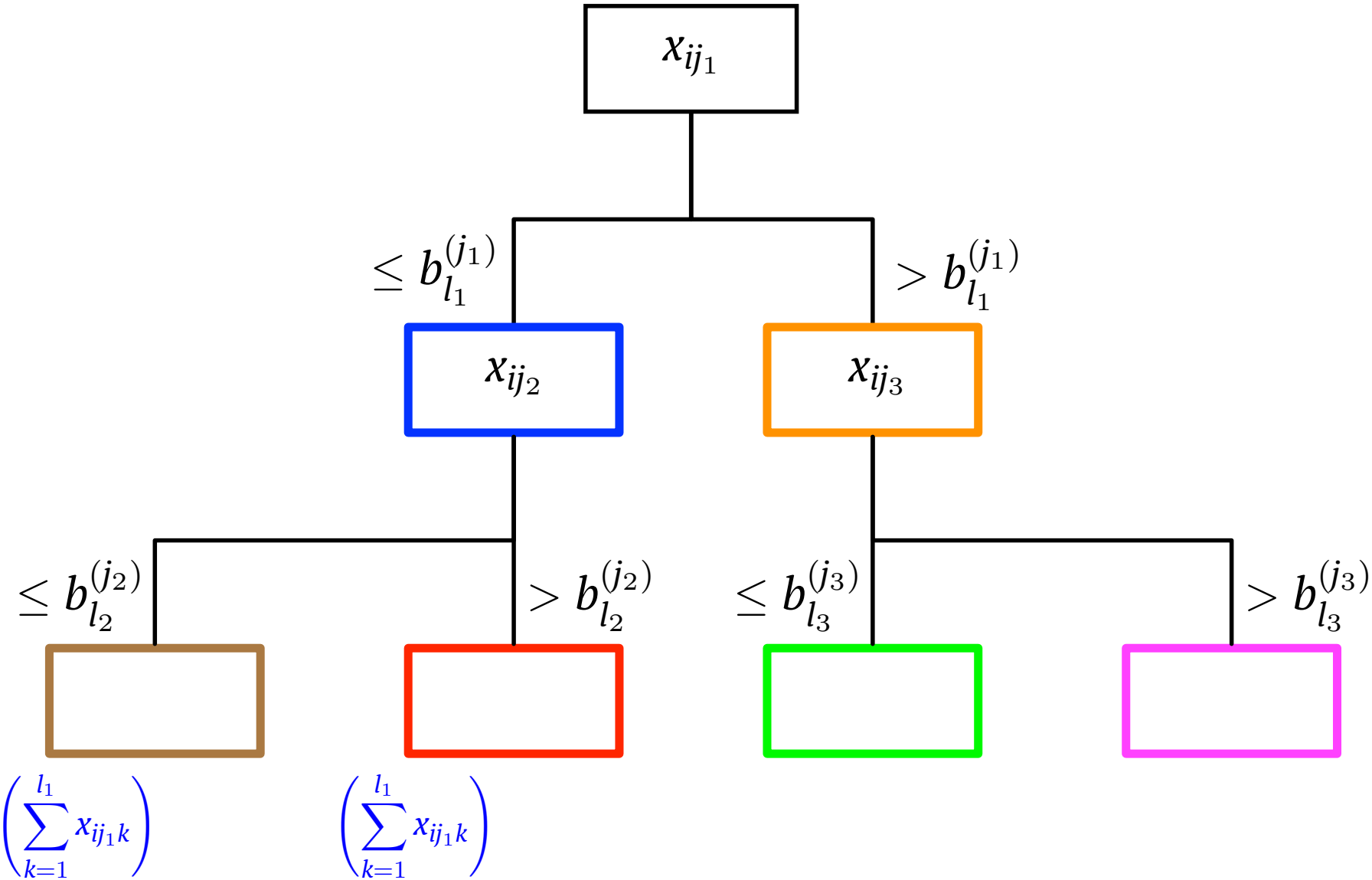
CRFs — Tree 'fitting', I



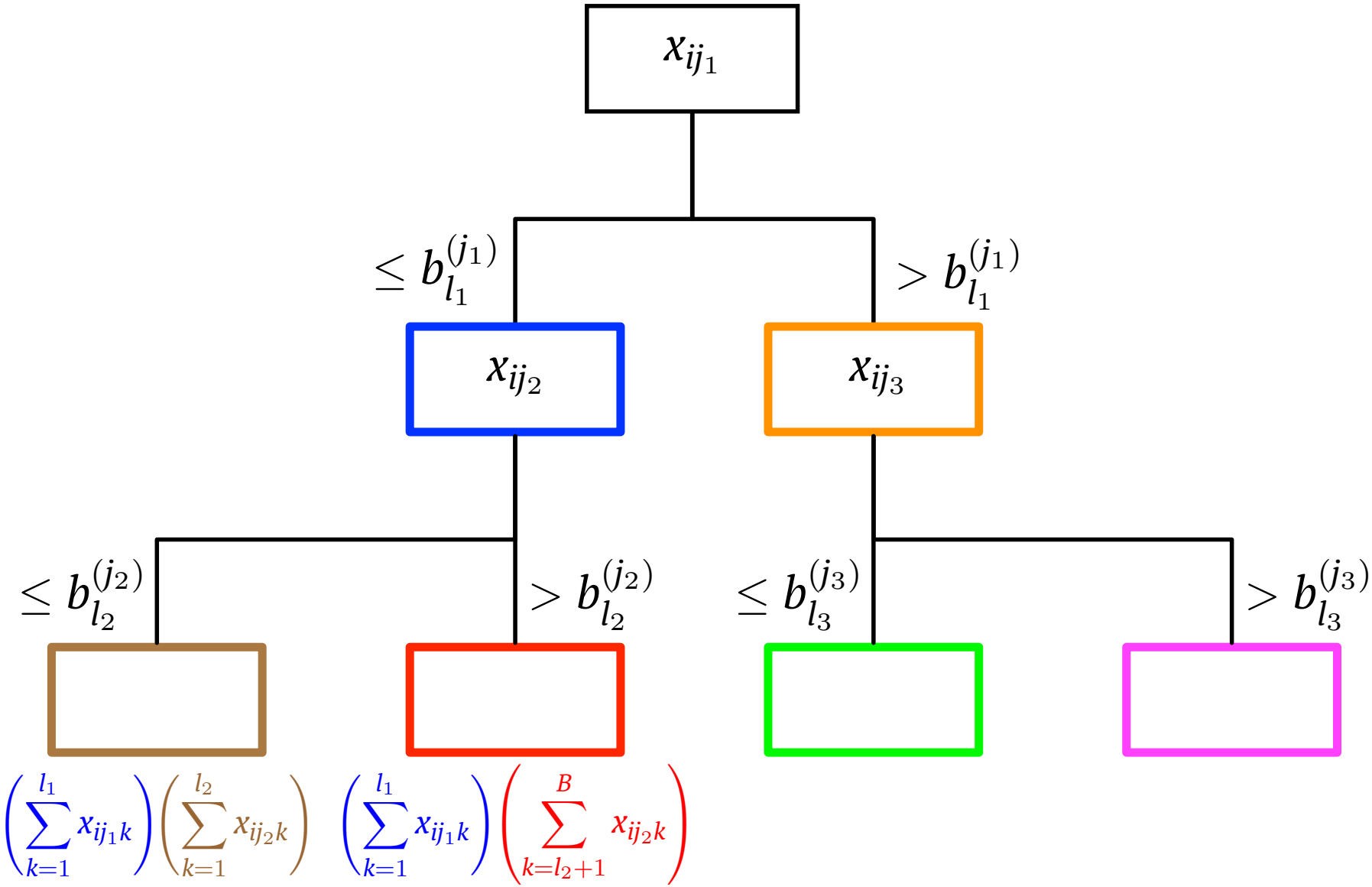
CRFs — Tree 'fitting', I



CRFs — Tree 'fitting', I

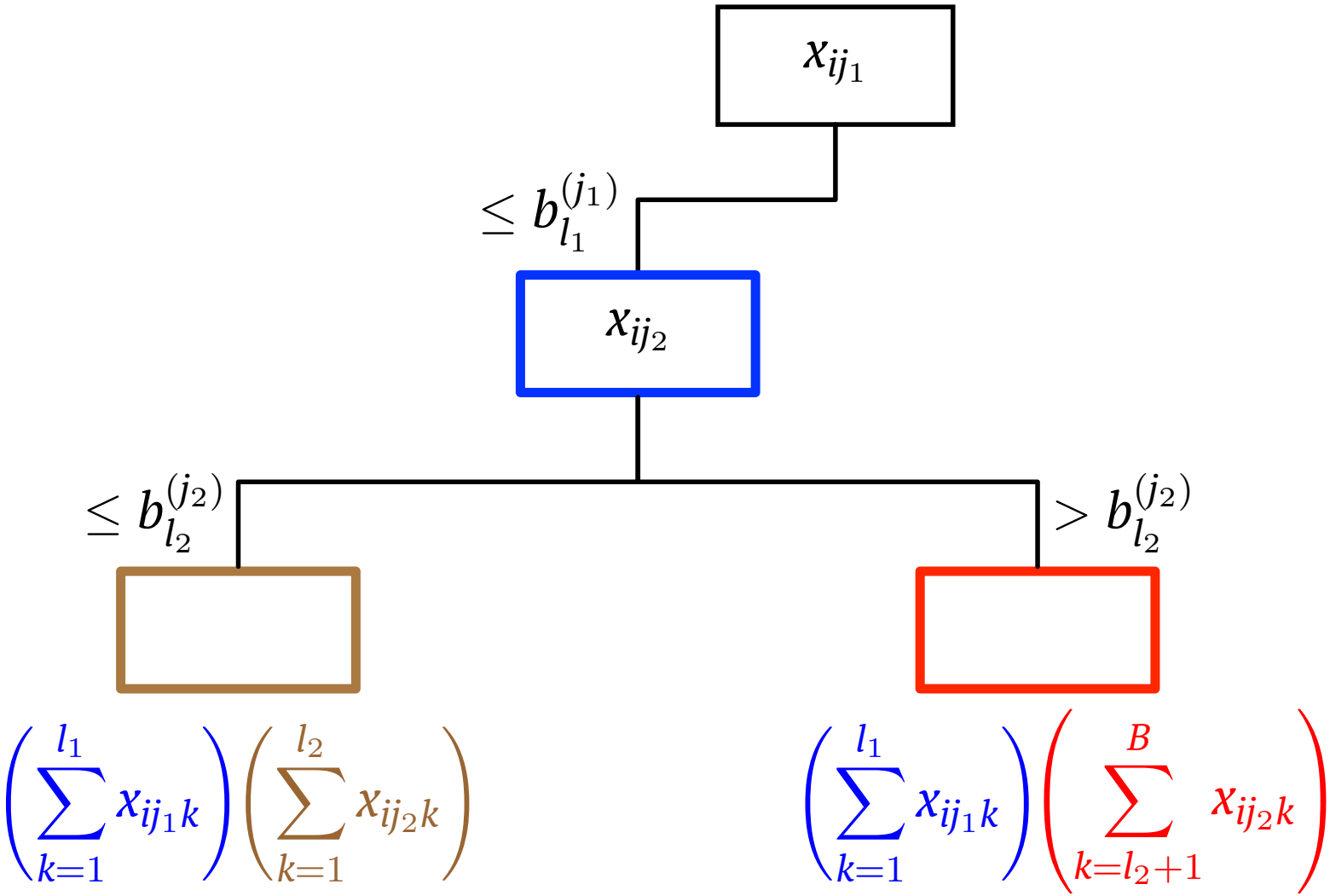


CRFs — Tree 'fitting', I

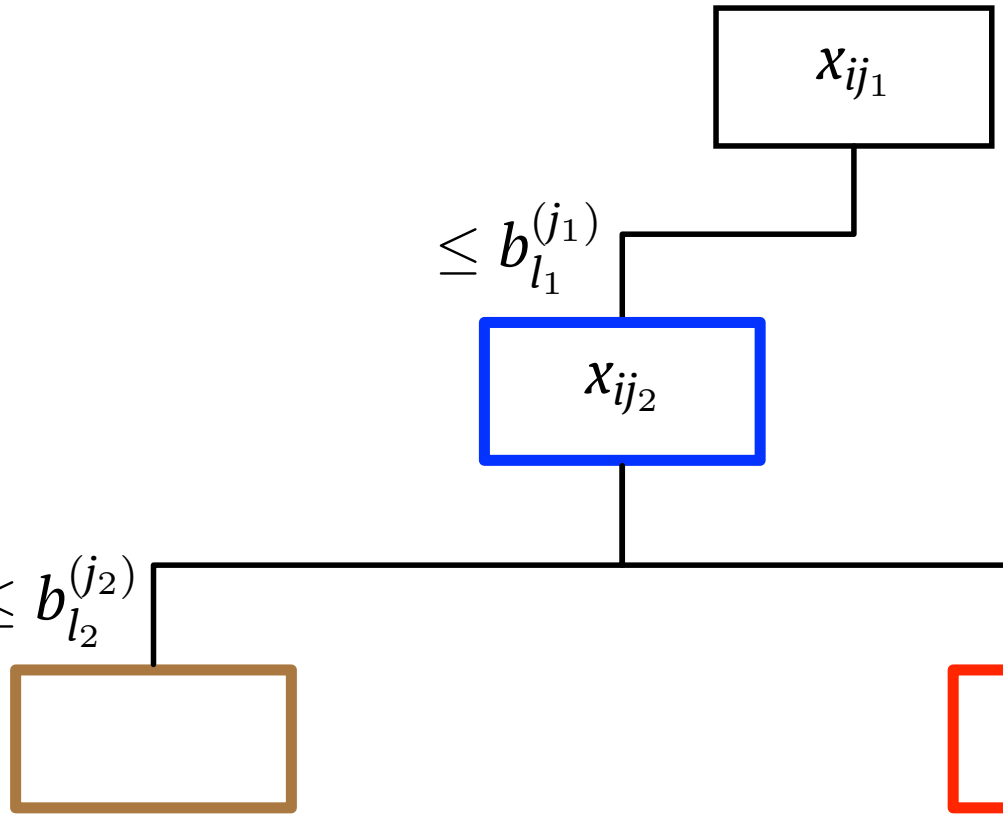


Exactly one terminal leaf indicator evaluates to 1, encrypted.

CRFs — Tree 'fitting', II



CRFs — Tree 'fitting', II



$$\nu_{ic} = y_{ic}\left(\sum_{k=1}^{l_1} x_{ij_1k}\right)\left(\sum_{k=1}^{l_2} x_{ij_2k}\right) \qquad \nu_{ic} = y_{ic}\left(\sum_{k=1}^{l_1} x_{ij_1k}\right)\left(\sum_{k=l_2+1}^{B} x_{ij_2k}\right)$$

NB Must evaluate *all* branches and categories as blindfold.

CRFs — Prediction

Prediction involves:

- evaluating a new observation through all branches;
- taking product with corresponding vote totals for each class;
- summing across trees and across leaves to get total votes for each class.

CRFs — Prediction

Prediction involves:

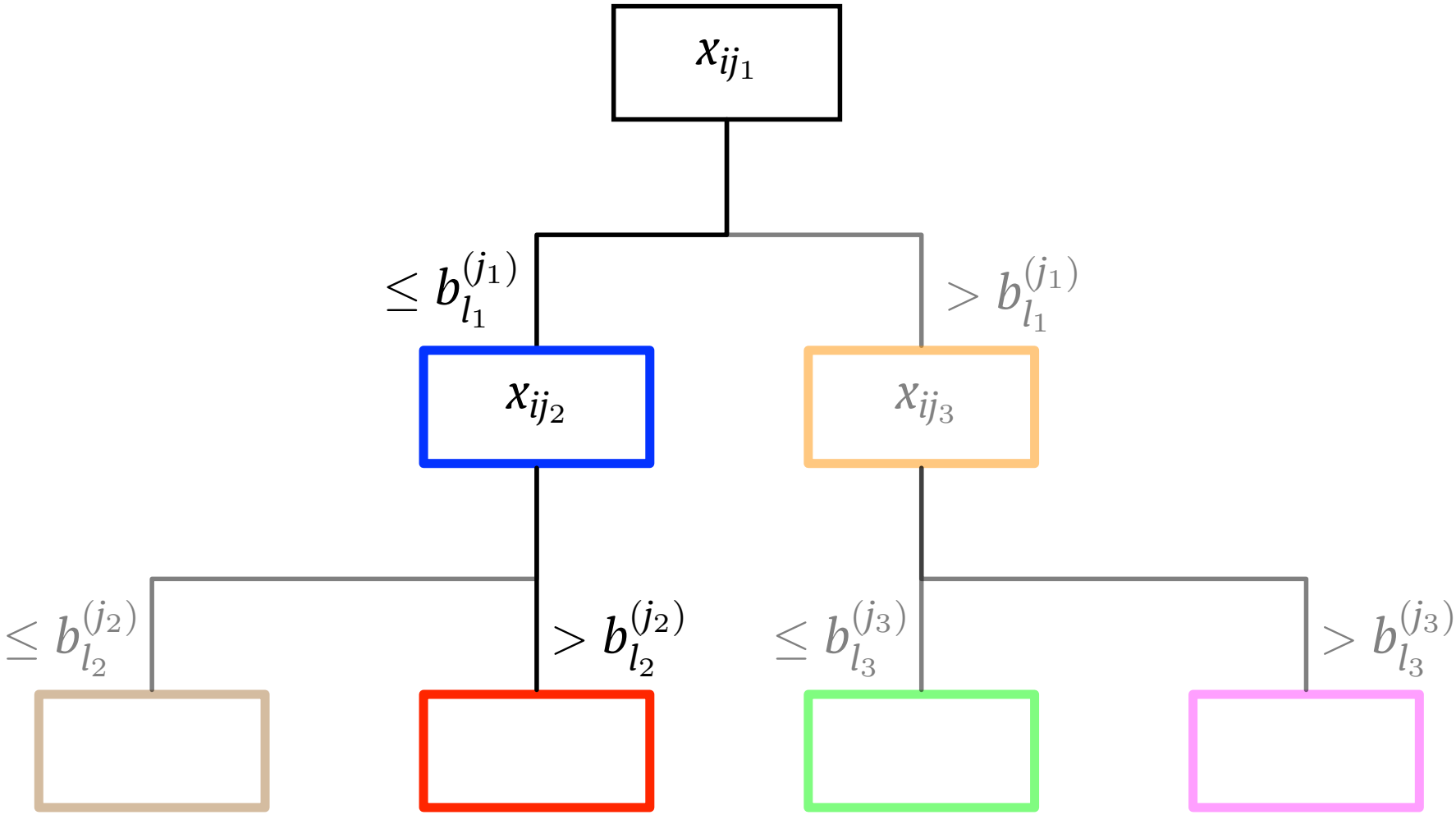
- evaluating a new observation through all branches;
- taking product with corresponding vote totals for each class;
- summing across trees and across leaves to get total votes for each class.

Random Forests usually use:

1. single vote per tree (requires comparison to find max)
2. relative class frequencies (requires division and $[0, 1]$ value)

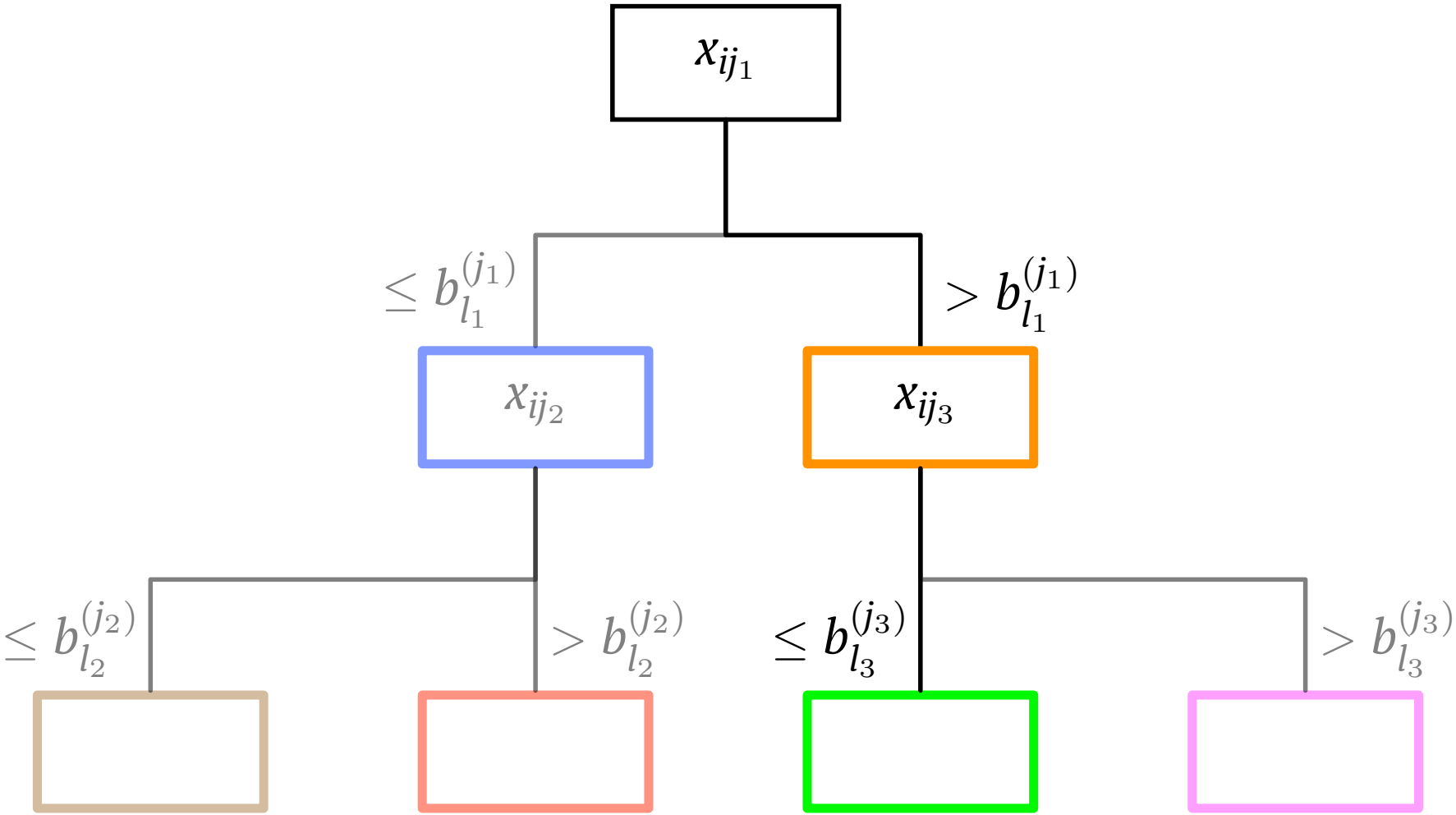
But here trees contribute raw 'vote' totals to the prediction: confused leaves with many votes can overwhealm certain ones with few.

CRFs — Raw votes problem



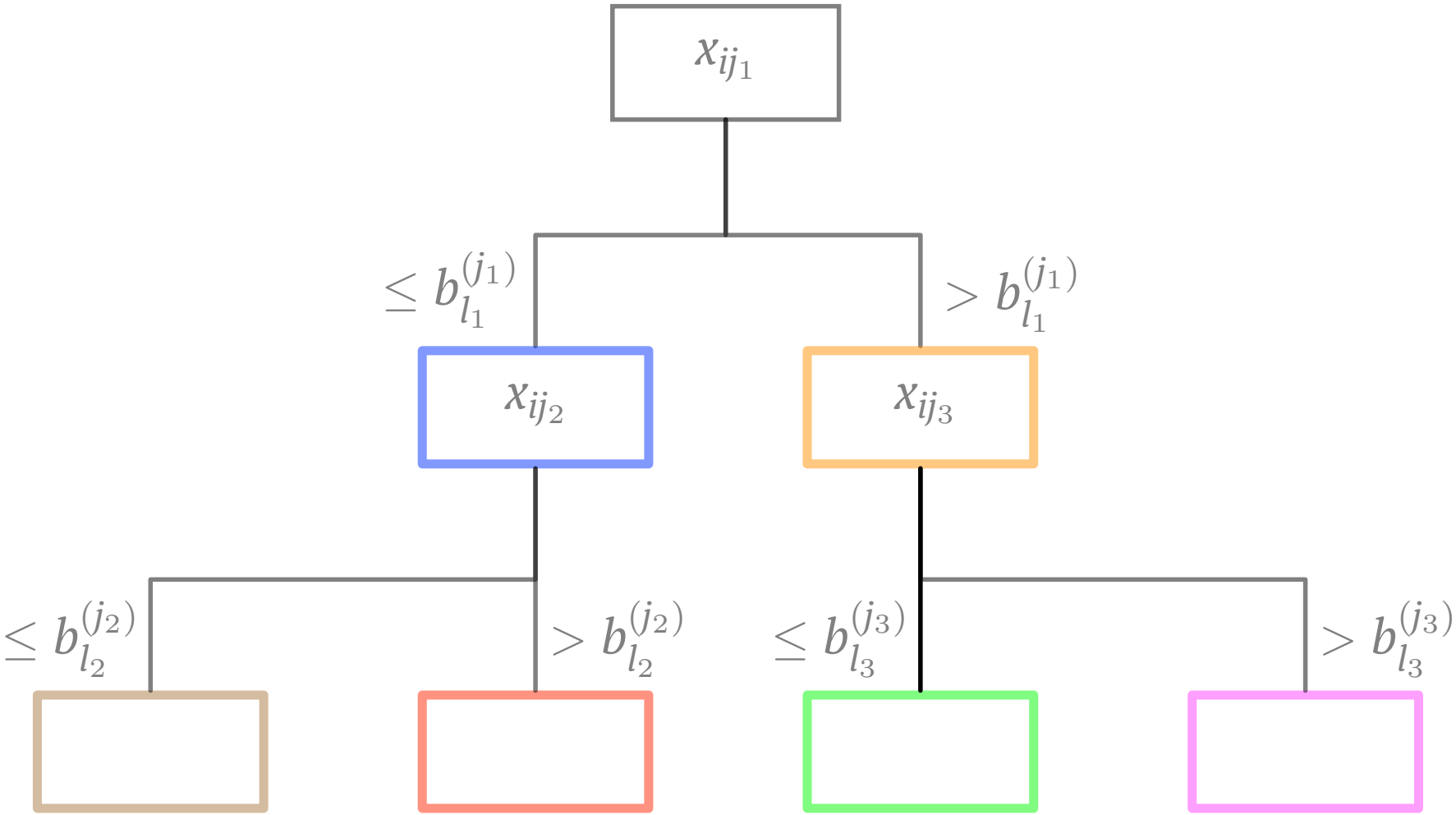
class	A	B	C
votes, $\sum_i \nu_{ic}$	78	7	15

CRFs — Raw votes problem



class	A	B	C
votes, $\sum_i \nu_{ic}$	1007	1080	1015

CRFs — Raw votes problem



$\Rightarrow$ overall votes, $\sum_i \nu_{ic}$

class	A	B	C
	1085	1087	1030

Relative class frequencies

Let ν_c be the number of votes for class c in a leaf. The relative class frequency contribution should be:

$$\frac{\nu_c}{\sum_c \nu_c}$$

But, this belongs to $[0, 1]$ which we can't represent and involves division.

Relative class frequencies

Let ν_c be the number of votes for class c in a leaf. The relative class frequency contribution should be:

$$\frac{\nu_c}{\sum_c \nu_c}$$

But, this belongs to $[0, 1]$ which we can't represent and involves division.Target equivalently:

$$\nu_c \left\lfloor \frac{N}{\sum_c \nu_c} \right\rfloor$$

where N is the number of training observations.

- By construction $\sum_c \nu_c \leq N$, so $0 \leq \frac{\sum_c \nu_c}{N} \leq 1$
- Recall, $X \sim \text{Geometric}(p) \implies \mathbb{E}[X] = p^{-1}$

Stochastic fraction estimate (I)

Thus, an unbiased approximation to fraction is draw from Geometric distribution with probability $\frac{\sum_c \nu_c}{N}$.

Not really helping ... any better than division?!

Stochastic fraction estimate (I)

Thus, an unbiased approximation to fraction is draw from Geometric distribution with probability $\frac{\sum_c \nu_c}{N}$.

Not really helping ... any better than division?!

Crucial observation: $\nu_c := \sum_{i=1}^{N} \nu_{ic}$ where $\nu_{ic} \in \{0, 1\} \; \forall \, i, c$.

(recall ν_{ic} is 1 if training obs. i was of class c and fell in this leaf of the decision tree ... leaf indices supressed)

Stochastic fraction estimate (I)

Thus, an unbiased approximation to fraction is draw from Geometric distribution with probability $\frac{\sum_c \nu_c}{N}$.

Not really helping ... any better than division?!

Crucial observation: $\nu_c := \sum_{i=1}^{N} \nu_{ic}$ where $\nu_{ic} \in \{0, 1\} \; \forall \, i, c$.

(recall ν_{ic} is 1 if training obs. i was of class c and fell in this leaf of the decision tree ... leaf indices supressed)

$\implies$ blind sampling with replacement from $\{\sum_c \nu_{ic} : i = 1, \ldots, N\}$ will produce an encrypted 1 with probability exactly $\frac{\sum_c \nu_c}{N}$.

$\implies$ can blind sample the latent bernoulli process underlying a Geometric$\left(p = \frac{\sum_c \nu_c}{N}\right)$ random variable.

Stochastic fraction estimate (II)

New problem! count number of leading zeros in an encrypted Bernoulli process.

Stochastic fraction estimate (II)

New problem! count number of leading zeros in an encrypted Bernoulli process.

Inspiration from CPU hardware algorithm for renormalising the mantissa of an IEEE floating point number.

Let $\xi_1, \ldots, \xi_M$ be a resampled vector ($\xi_i = \sum_c \eta_{cj}$, some j) and assume M is a power of 2.

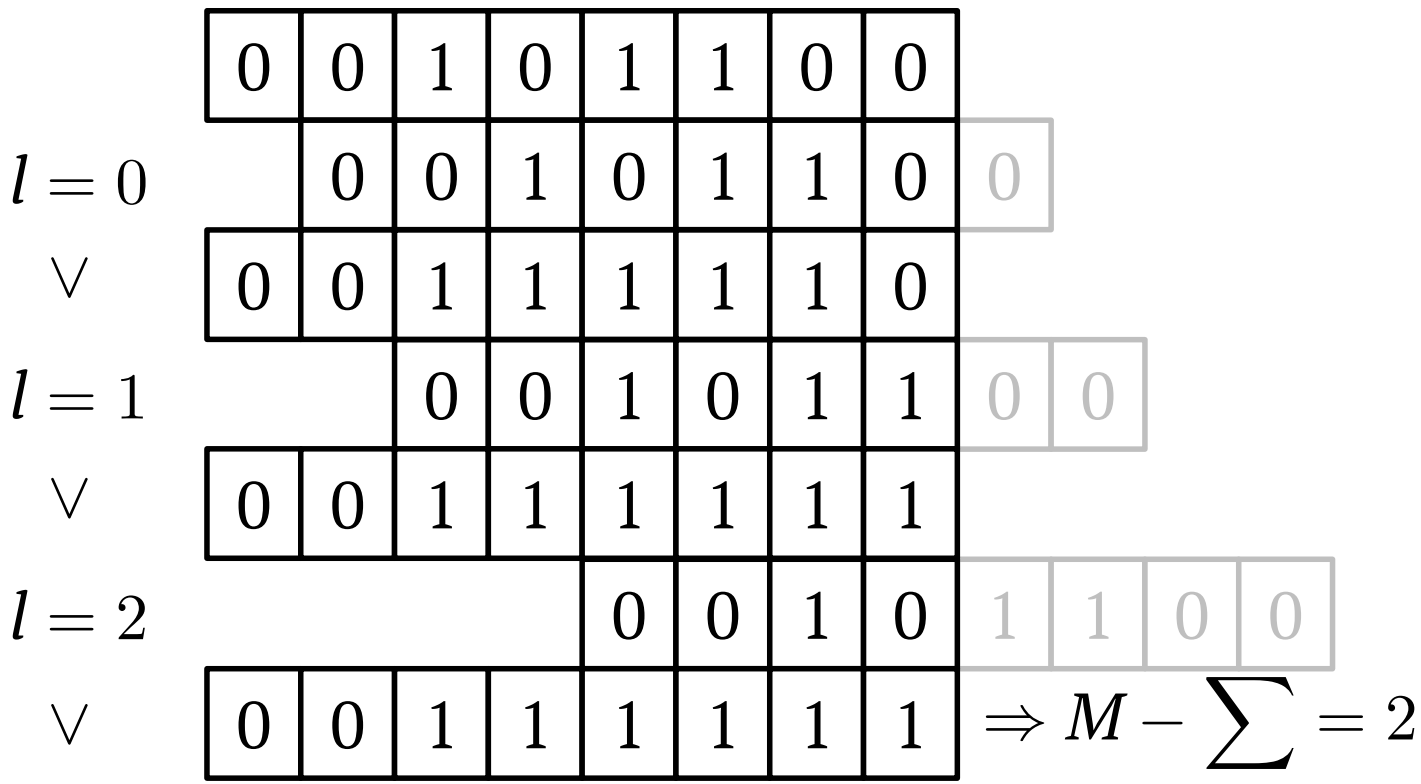
1. For $l \in \{0, \ldots, \log_2(M) - 1\}$:
   - Set $\xi_i = \xi_i \vee \xi_{i-2^l} = \xi_i + \xi_{i-2^l} - \xi_i \xi_{i-2^l} \quad \forall\, 2^l + 1 \leq i \leq M$
2. The number of leading zeros is $M - \sum_{i=1}^{M} \xi_i$

Corresponds to increasing power of 2 bit-shifts OR'd with itself, all computable encrypted.

$$\implies \left\lfloor \frac{N}{\sum_c \nu_c} \right\rfloor \approx M - \sum_{i=1}^{M} \xi_i + 1$$

Stochastic fraction estimate (III)

CPU hardware algorithm for mantissa normalisation



Stochastic fraction estimate (IV)

Bias

Clearly, since blindfolded can't sample *until* a 1 observed, so choose a fixed M and accept small bias.

Stochastic fraction estimate (IV)

~~Bias~~ Shrinkage

Clearly, since blindfolded can't sample *until* a 1 observed, so choose a fixed M and ~~accept~~ celebrate small ~~bias~~ shrinkage.

Stochastic fraction estimate (IV)

~~Bias~~ Shrinkage

Clearly, since blindfolded can't sample *until* a 1 observed, so choose a fixed M and ~~accept~~ celebrate small ~~bias~~ shrinkage.

The shrinkage is mild unless there are fewer than $\frac{N}{M}$ observations in the leaf, in which case the shrinkage is more extreme: this is desirable because it shrinks the influence of underpopulated leaves.

e.g. $N = 1000, M = 32 \implies$ heavy shrinkage for leaves with < 31 observations.

Stochastic fraction estimate (IV)

~~Bias~~ Shrinkage

Clearly, since blindfolded can't sample *until* a 1 observed, so choose a fixed M and ~~accept~~ celebrate small ~~bias~~ shrinkage.

The shrinkage is mild unless there are fewer than $\frac{N}{M}$ observations in the leaf, in which case the shrinkage is more extreme: this is desirable because it shrinks the influence of underpopulated leaves.

e.g. $N = 1000, M = 32 \implies$ heavy shrinkage for leaves with < 31 observations.

Computational consideration

Multiplicative depth of this algorithm is M, which must be factored into tree building.

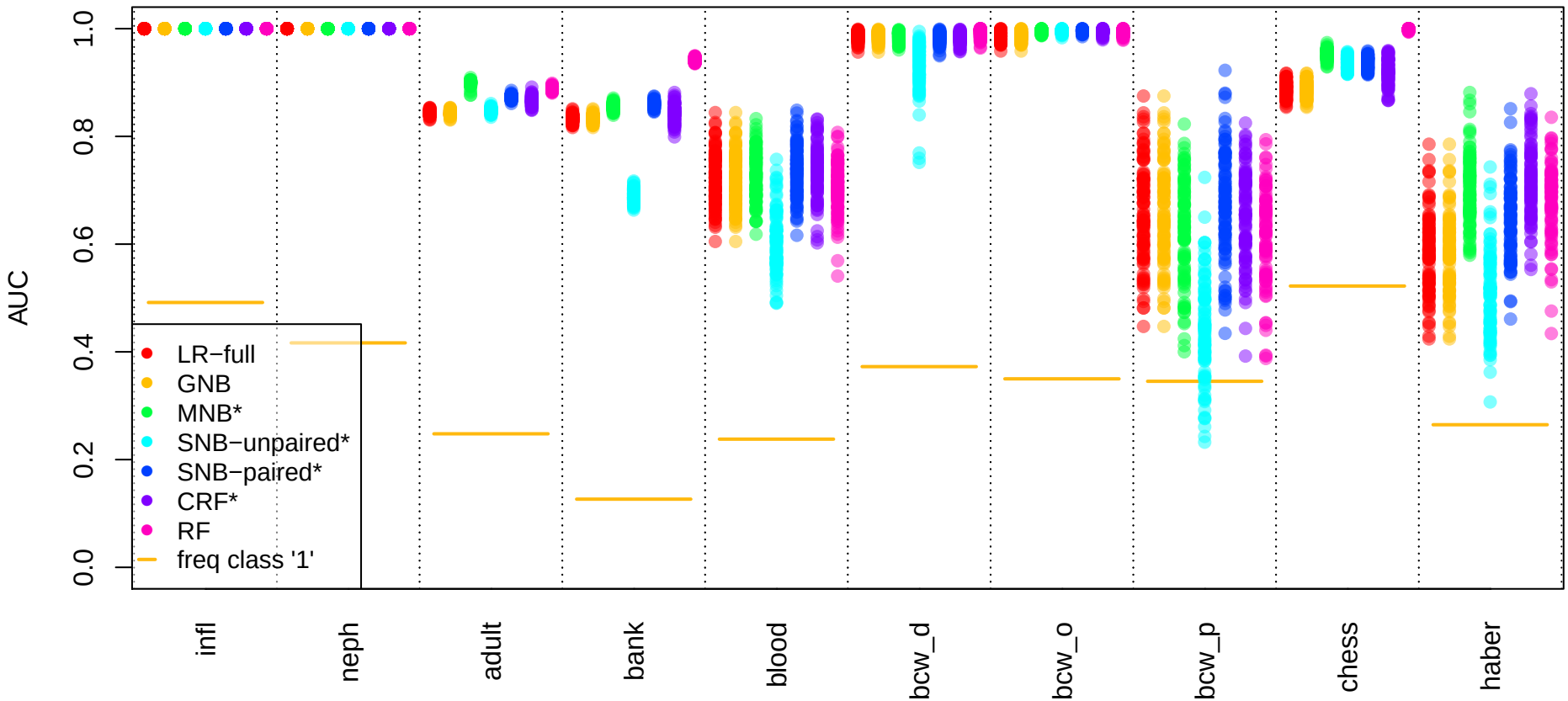
Theoretical homomorphic scheme requirements

To build a forest of trees with L levels, the homomorphic encryption scheme must support:

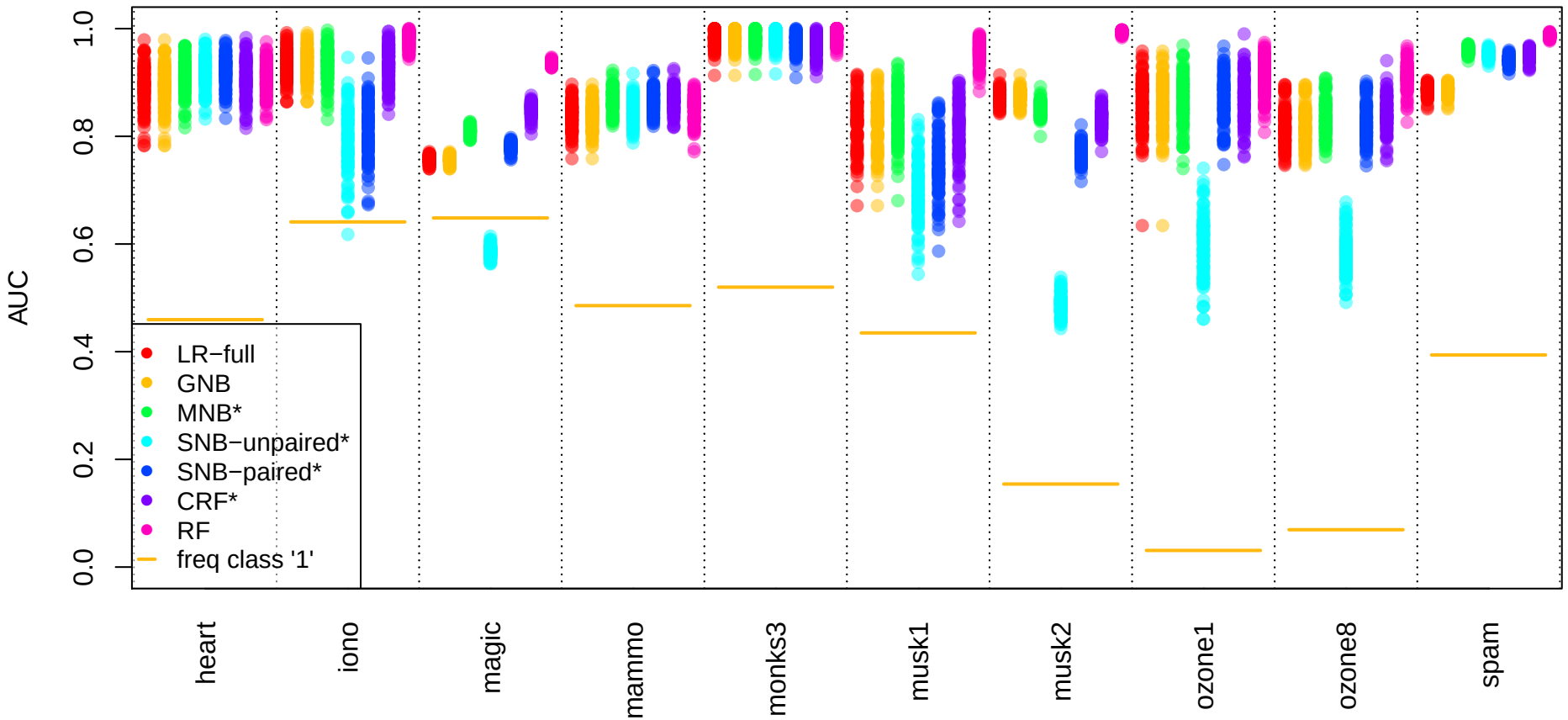
- depth L multiplications for tree building
- depth $L + M$ for stochastic fraction adjustment
- depth $2L + M$ for building, adjustment and prediction.

Furthermore, for the current generation of Ring Learning With Errors encryption schemes where the message space is a polynomial ring, it must support coefficients up to $T \max\{\sum_i y_{ic} : c = 1, \ldots, |\mathcal{C}|\}$.

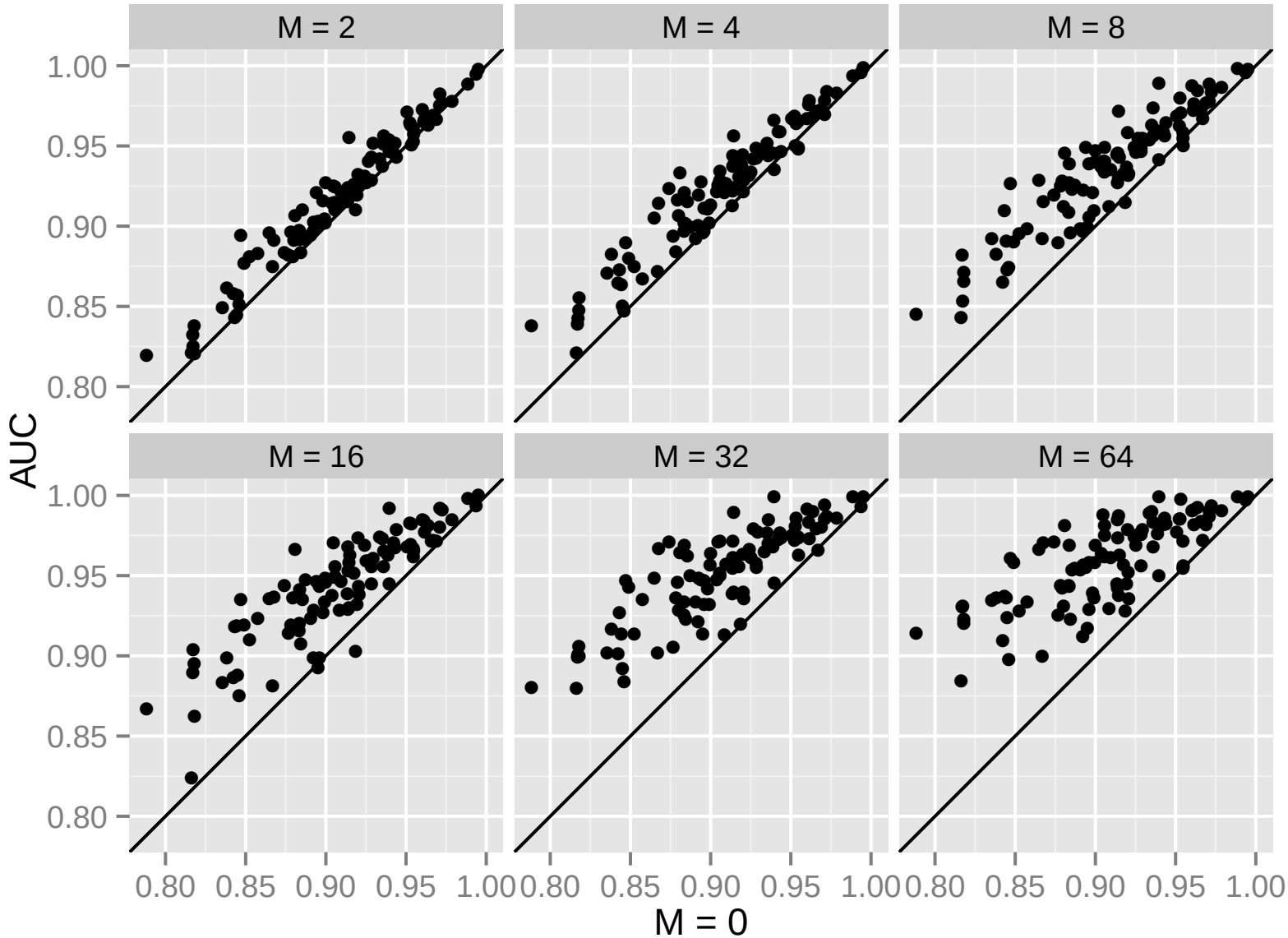
Results (I)


AUC
0.0
0.2
0.4
0.6
0.8
1.0
LR−full
GNB
MNB*
SNB−unpaired*
SNB−paired*
CRF*
RF
freq class '1'
infl
neph
adult
bank
blood
bcw_d
bcw_o
bcw_p
chess
haber

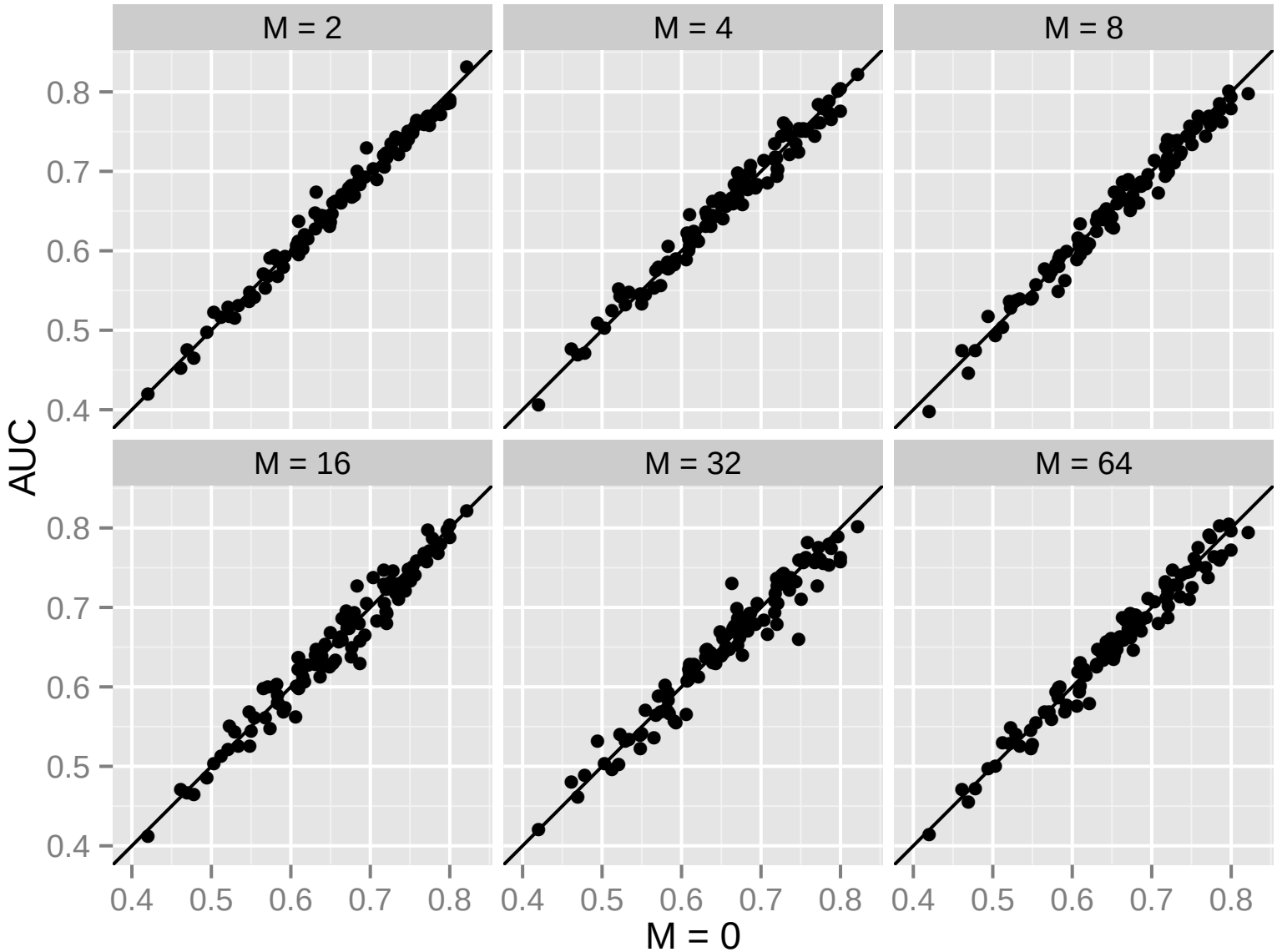
Results (II)


AUC
0.0
0.2
0.4
0.6
0.8
1.0
LR−full
GNB
MNB*
SNB−unpaired*
SNB−paired*
CRF*
RF
freq class '1'
heart
iono
magic
mammo
monks3
musk1
musk2
ozone1
ozone8
spam

Stochastic fraction effect (best)



Stochastic fraction effect (worst)



Computational considerations

Note that CRFs are parallelisable right down to the individual observation, which helps with ameliorating the cost of encrypted computation.

Computational considerations

Note that CRFs are parallelisable right down to the individual observation, which helps with ameliorating the cost of encrypted computation.

Wisconsin data ($N = 547$)

- Launched
  2×18 servers $\times 32$ cores $= 1,152$ CPU core cluster on Amazon EC2
  $\Rightarrow$ 576 Dublin & 576 São Paulo
- Fit 50 trees in Dublin, 50 in São Paulo
  - unique `set.seed()` for each region
- Data split into 17 shards of 32 obs + 1 shard 3 obs $\Rightarrow$ 1 datum per core!
- Reduction sum of votes in each region and combine regions $\Rightarrow$ 100 tree forest

Computational considerations

Note that CRFs are parallelisable right down to the individual observation, which helps with ameliorating the cost of encrypted computation.

Wisconsin data ($N = 547$)

- Launched
  2×18 servers $\times 32$ cores $= 1,152$ CPU core cluster on Amazon EC2
  $\Rightarrow$ 576 Dublin & 576 São Paulo
- Fit 50 trees in Dublin, 50 in São Paulo
  - unique `set.seed()` for each region
- Data split into 17 shards of 32 obs + 1 shard 3 obs $\Rightarrow$ 1 datum per core!
- Reduction sum of votes in each region and combine regions $\Rightarrow$ 100 tree forest

1h 36m

US$ 23.86

Arbitrary Bayesian Models

Perspectives on “privacy”

- Differential privacy
  - on outcomes of ‘statistical queries’
  - guarantees of privacy for individual observations

Perspectives on “privacy”

- Differential privacy
  - on outcomes of ‘statistical queries’
  - guarantees of privacy for individual observations
- Data privacy
  - at rest
  - during fitting
  - data pooling

Perspectives on “privacy”

- Differential privacy
  - on outcomes of ‘statistical queries’
  - guarantees of privacy for individual observations
- Data privacy
  - at rest
  - during fitting
  - data pooling
- Model privacy
  - prior distributions
  - model formulation

The perspective for today ...

- **Eve** has a private model, including prior information which may itself be private.
- **Cain** and **Abel** have private data which is relevant to the fitting of Eve's model.

Can Eve fit a model, pooling data from Cain and Abel without observing their raw data and without revealing her model and prior information? Abel also doesn't trust Cain ...

$\pi(\cdot \mid \psi)$
$\pi(\psi)$

$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=1}^{n_1}$

$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=n_1+1}^{N}$

Back to the problem ...

$\pi(\cdot \mid \psi)$
$\pi(\psi)$

$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=1}^{n_1}$

$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=n_1+1}^{N}$

Back to the problem ...

$$\pi(\cdot \mid \psi)$$
$$\pi(\psi)$$

$$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=1}^{n_1}$$

$$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=n_1+1}^{N}$$

$$\mathbf{x}_i^\star = \mathrm{Enc}(k_p, \mathbf{x}_i)$$

Back to the problem ...

$$\pi(\cdot \mid \psi)$$
$$\pi(\psi)$$

$$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=1}^{n_1}$$

$$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=n_1+1}^{N}$$

$$\pi(\psi \mid X) \propto$$

$$\text{Dec}\left[k_s, \prod_{i=1}^{N} \pi(\mathbf{x}_i^\star | \text{Enc}(k_p, \psi)) \times \right.$$

$$\left. \text{Enc}(k_p, \pi(\psi)) \right]$$

$$\mathbf{x}_i^\star = \text{Enc}(k_p, \mathbf{x}_i)$$

Back to the problem ...

$$\pi(\cdot \mid \psi)$$
$$\pi(\psi)$$

$$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=1}^{n_1}$$

$$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=n_1+1}^{N}$$

$$\pi(\psi \mid X) \propto$$

$$\mathrm{Dec}\left[k_s, \prod_{i=1}^{N} \pi(\mathbf{x}_i^\star \mid \mathrm{Enc}(k_p, \psi)) \times \right.$$

$$\left. \mathrm{Enc}(k_p, \pi(\psi))\right]$$

$$\mathbf{x}_i^\star = \mathrm{Enc}(k_p, \mathbf{x}_i)$$

✗ Likelihood restricted to low degree polynomials
✗ Can only handle very small N due to multiplicative depth
✗ MAP/posterior? How? MCMC?

✗ Who holds secret key?

Eve, Cain & Abel

$$\pi(\cdot \mid \psi)$$
$$\pi(\psi)$$

$$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=1}^{n_1}$$

$$\{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=n_1+1}^{N}$$

$$\pi(\psi \mid X) \propto$$

$$\mathrm{Dec}\left[k_s, \prod_{i=1}^{N} \pi(\mathbf{x}_i^\star \mid \mathrm{Enc}(k_p, \psi)) \times \right.$$

$$\left. \mathrm{Enc}(k_p, \pi(\psi)) \right]$$

$$\mathbf{x}_i^\star = \mathrm{Enc}(k_p, \mathbf{x}_i)$$

✗ Likelihood restricted to low degree polynomials

✗ Can only handle very small N due to multiplicative depth

✗ MAP/posterior? How? MCMC?

✗ ~~Who holds secret key?~~

Approximate Bayesian Computation

1. Sample $\psi_j \sim \pi(\psi),\ j \in \{1, \ldots, m\}$
2. For each ψ_j, simulate a dataset Y_j from $\pi(\cdot \mid \psi_j)$ of the same size, N, as X.
3. Accept ψ_j if $d(S(X), S(Y_j)) < \varepsilon$.

Where $S(\cdot)$ is some (vector) of summary statistics; $d(\cdot, \cdot)$ is a distance metric; and ε is a user defined threshold.

When $S(\cdot)$ is sufficient and $\varepsilon \to 0$, this procedure will converge to the usual Bayesian posterior.

Approximate Bayesian Computation

1. Sample $\psi_j \sim \pi(\psi),\ j \in \{1, \ldots, m\}$
2. For each ψ_j, simulate a dataset Y_j from $\pi(\cdot \mid \psi_j)$ of the same size, N, as X.
3. Accept ψ_j if $d(S(X), S(Y_j)) < \varepsilon$.

Where $S(\cdot)$ is some (vector) of summary statistics; $d(\cdot, \cdot)$ is a distance metric; and ε is a user defined threshold.

When $S(\cdot)$ is sufficient and $\varepsilon \to 0$, this procedure will converge to the usual Bayesian posterior.

Benefit: Eve can do steps 1 & 2 and encrypt her simulated data, eliminating need for function privacy.

Approximate Bayesian Computation

1. Sample $\psi_j \sim \pi(\psi),\ j \in \{1, \ldots, m\}$
2. For each ψ_j, simulate a dataset Y_j from $\pi(\cdot \mid \psi_j)$ of the same size, N, as X.
3. Accept ψ_j if $d(S(X), S(Y_j)) < \varepsilon$.

Where $S(\cdot)$ is some (vector) of summary statistics; $d(\cdot, \cdot)$ is a distance metric; and ε is a user defined threshold.

When $S(\cdot)$ is sufficient and $\varepsilon \to 0$, this procedure will converge to the usual Bayesian posterior.

Benefit: Eve can do steps 1 & 2 and encrypt her simulated data, eliminating need for function privacy.

Problems: $d(\cdot, \cdot)$ can only be low degree polynomials; Must compute $S(\cdot)$ secretly for Cain and Abel's pooled data; Naïve ABC performs poorly & choosing ε blindfolded.

Naïve encrypted ABC (I) – Eve & data owners $1, \ldots, P$

1. Eve samples $\psi_j \sim \pi(\psi),\ j \in \{1, \ldots, m\}$; simulates datasets Y_j of size N from $\pi(\cdot \mid \psi_j)$; and computes $S(Y_j)$.

Naïve encrypted ABC (I) – Eve & data owners $1, \ldots, P$

1. Eve samples $\psi_j \sim \pi(\psi),\ j \in \{1, \ldots, m\}$; simulates datasets Y_j of size N from $\pi(\cdot \mid \psi_j)$; and computes $S(Y_j)$.
2. Eve computes HSS shares $S^{\star p}(Y_j)$, $p \in \{1, \ldots, P+1\}$;
   - send $S^{\star p}(Y_j)$ to data owner p
   - retain $S^{\star P+1}(Y_j)$

Naïve encrypted ABC (I) – Eve & data owners $1, \ldots, P$

1. Eve samples $\psi_j \sim \pi(\psi),\ j \in \{1, \ldots, m\}$; simulates datasets Y_j of size N from $\pi(\cdot \mid \psi_j)$; and computes $S(Y_j)$.
2. Eve computes HSS shares $S^{\star p}(Y_j)$, $p \in \{1, \ldots, P+1\}$;
   - send $S^{\star p}(Y_j)$ to data owner p
   - retain $S^{\star P+1}(Y_j)$
3. Data owners $k \in \{1, \ldots, P\}$ create HSS shares $S^{\star p}(X_k)$, $p \in \{1, \ldots, P+1\}$
   - send $S^{\star p}(X_k)$ to data owner p (retaining when $p = k$)
   - send $S^{\star P+1}(X_k)$ to Eve

Naïve encrypted ABC (I) – Eve & data owners $1, \ldots, P$

1. Eve samples $\psi_j \sim \pi(\psi),\ j \in \{1, \ldots, m\}$; simulates datasets Y_j of size N from $\pi(\cdot \mid \psi_j)$; and computes $S(Y_j)$.
2. Eve computes HSS shares $S^{\star p}(Y_j)$, $p \in \{1, \ldots, P+1\}$;
   - send $S^{\star p}(Y_j)$ to data owner p
   - retain $S^{\star P+1}(Y_j)$
3. Data owners $k \in \{1, \ldots, P\}$ create HSS shares $S^{\star p}(X_k)$, $p \in \{1, \ldots, P+1\}$
   - send $S^{\star p}(X_k)$ to data owner p (retaining when $p = k$)
   - send $S^{\star P+1}(X_k)$ to Eve
4. All compute $S^{\star p}(X) = \tilde{S}\left(\bigcup_k S^{\star p}(X_k)\right)$, where $\tilde{S}(\cdot)$ is a **homomorphically computable pooling function**.

Naïve encrypted ABC (I) – Eve & data owners $1, \ldots, P$

1. Eve samples $\psi_j \sim \pi(\psi),\ j \in \{1, \ldots, m\}$; simulates datasets Y_j of size N from $\pi(\cdot \mid \psi_j)$; and computes $S(Y_j)$.
2. Eve computes HSS shares $S^{\star p}(Y_j)$, $p \in \{1, \ldots, P+1\}$;
   - send $S^{\star p}(Y_j)$ to data owner p
   - retain $S^{\star P+1}(Y_j)$
3. Data owners $k \in \{1, \ldots, P\}$ create HSS shares $S^{\star p}(X_k)$, $p \in \{1, \ldots, P+1\}$
   - send $S^{\star p}(X_k)$ to data owner p (retaining when $p = k$)
   - send $S^{\star P+1}(X_k)$ to Eve
4. All compute $S^{\star p}(X) = \tilde{S}\left(\bigcup_k S^{\star p}(X_k)\right)$, where $\tilde{S}(\cdot)$ is a **homomorphically computable pooling function**.
5. All compute $d_j^{\star p} = d\left(S^{\star p}(X), S^{\star p}(Y_j)\right)$, where $d(\cdot)$ is a **homomorphically computable distance metric**.

Naïve encrypted ABC (II) – Eve & data owners $1, \ldots, P$

6. All send their shares, $d_j^{\star p}$, to a randomly chosen data owner $k \in 1, \ldots, P$

Naïve encrypted ABC (II) – Eve & data owners $1, \ldots, P$

6. All send their shares, $d_j^{\star p}$, to a randomly chosen data owner $k \in 1, \ldots, P$
7. Data owner k reconstructs $d_j = \text{Dec}(d_j^{\star 1}, \ldots, d_j^{\star P+1})$

Naïve encrypted ABC (II) – Eve & data owners $1, \ldots, P$

6. All send their shares, $d_j^{\star p}$, to a randomly chosen data owner $k \in 1, \ldots, P$
7. Data owner k reconstructs $d_j = \text{Dec}(d_j^{\star 1}, \ldots, d_j^{\star P+1})$
8. Data owner k sends to Eve a list of those indices j such that $d_j < \varepsilon$.

Naïve encrypted ABC (III) – in pictures

$\pi(\psi) \longrightarrow \{\psi_j\}_{j=1}^m$

$\pi(\cdot \mid \psi)$

$\{S^\star(Y_j)\}_{j=1}^m$

$X_1 = \{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=1}^{n_1}$

$X_2 = \{\mathbf{x}_i = (x_{i1}, \ldots, x_{id})\}_{i=n_1+1}^{N}$

$S^\star(X) = \tilde{S}\left(X_1^\star, X_2^\star, S^\star(X_1), S^\star(X_2)\right)$

$d_j^\star = d(S^\star(Y_j), S^\star(X))$

$d_j = \mathrm{Dec}(d_j^{\star\mathrm{Eve}}, d_j^{\star\mathrm{Cain}}, d_j^{\star\mathrm{Abel}})$

$\mathcal{J} = \{j \,:\, d_j < \varepsilon\}$

Accept $\{\psi_j \,:\, j \in \mathcal{J}\}$

Points to note

- Samples ψ_j are never seen by Cain and Abel
- Eve learns only an accept/reject
  - Final distances between summary statistics decrypted by Cain or Abel
- Cain and Abel do not learn about each other's data
  - only see composite distance between pooled summary stats and Eve's simulation
  - can make distances information theoretically secure by adding random values generated by Cain, Abel and Eve
- **BUT**, Cain and Abel do have to know $S(\cdot)$, which in most ABC settings is model dependent $\implies$ risk to Eve

Obstacles to cryptographic ABC

- Homomorphically computable pooling of summary statistics
- Summary statistics that don't reveal model
- Homomorphically computable distance metric
- Blindfold selection of ε

Obstacles to cryptographic ABC

- Homomorphically computable pooling of summary statistics
- Summary statistics that don't reveal model
- Homomorphically computable distance metric
- Blindfold selection of ε
  - Propose using ABC-PMC/SMC, with distance chosen to retain $\alpha\%$ of samples instead. Eve then uses accepted ψ_j on step t to propose step $t+1$ and repeat algorithm.
  - Standard idea — details omited.

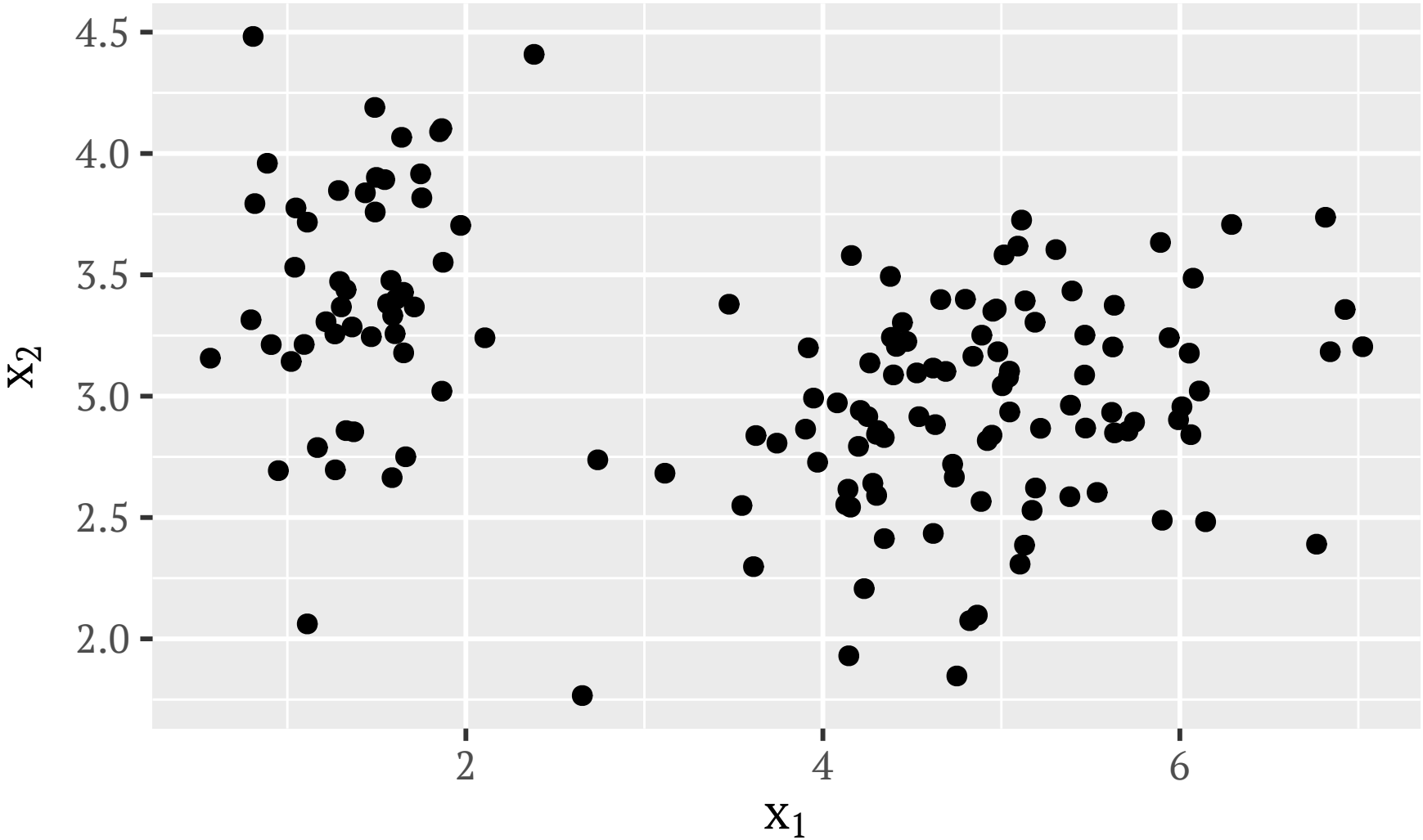
Collection of Coarse Random Marginals (CCRM)

Construct in the manner of a decision forest:

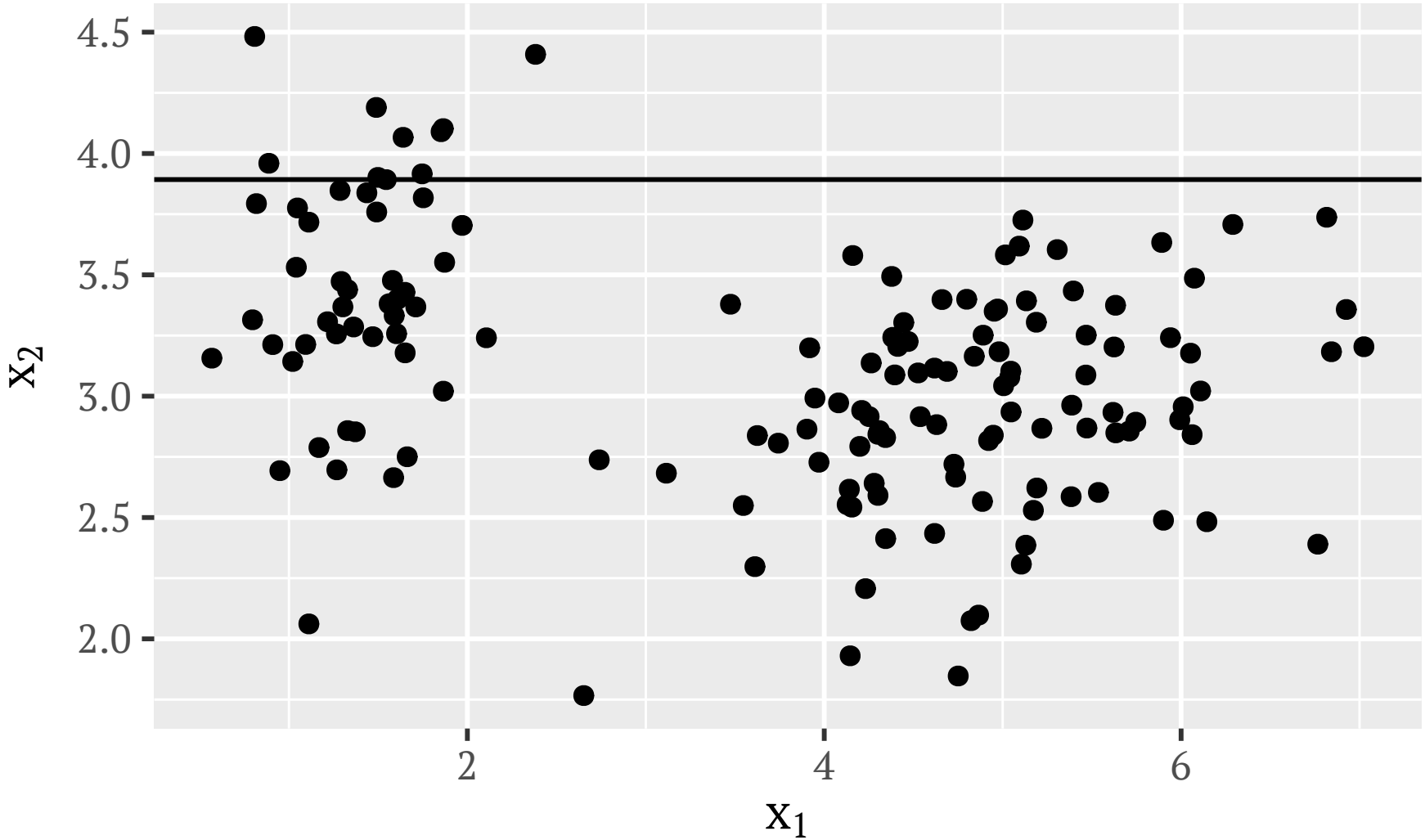
- Grow T trees, each to predetermined fixed depth L
- Choose variable $v \in \{1, \ldots, d\}$ uniformly at random
- Each split point uniformly at random in range of $x_{\cdot v}$
  - Thus Cain and Abel must provide range of each variable in the data, though this range need not be tight
  - e.g. release $(\min_i x_{iv} + \eta, \max_i x_{iv} + \eta)$ for $\eta \sim N(0, \sigma^2)$ with σ^2 chosen not to exclude too large a range
- $\mathbf{s} = S(\cdot)$ is then the counts of observations in each terminal leaf
  - vector of $T2^L$ counts
  - $\tilde{S}(\cdot)$ is then simply vector addition
- Define

$$d(S(X), S(Y_j)) = \sum_{i=1}^{T2^L} \left(s_i^X - s_i^{Y_j} \right)^2$$

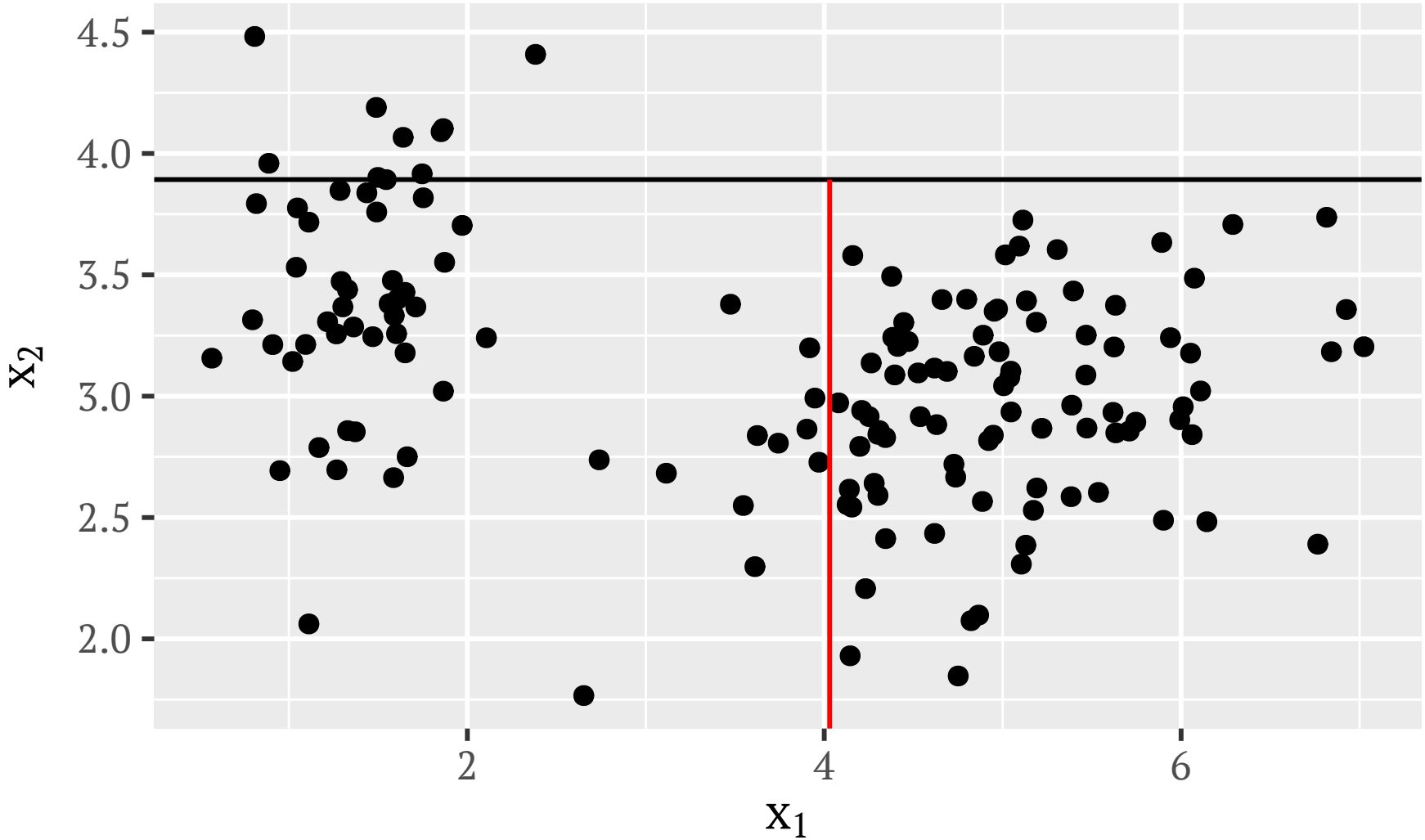
Collection of Coarse Random Marginals (CCRM)



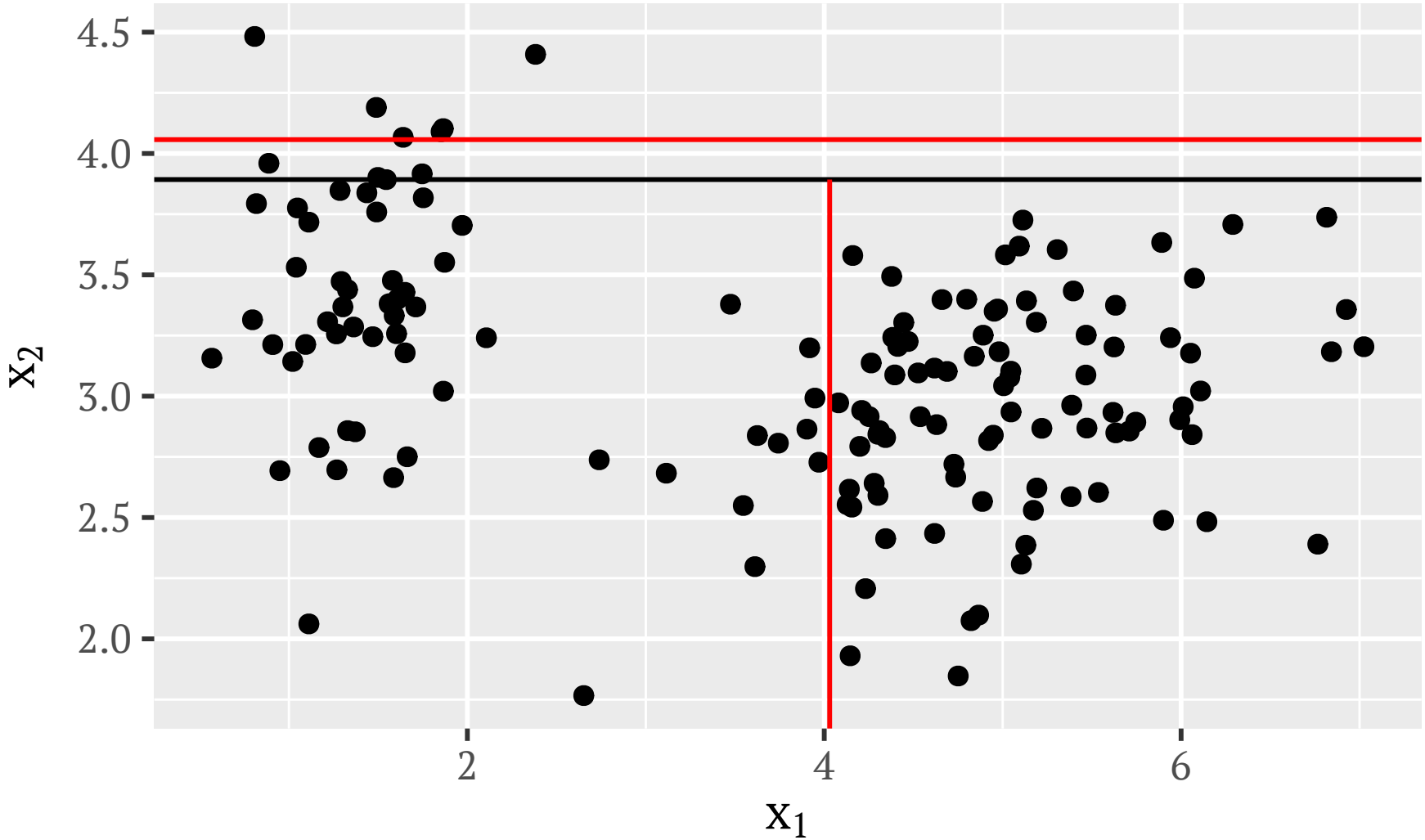
Collection of Coarse Random Marginals (CCRM)



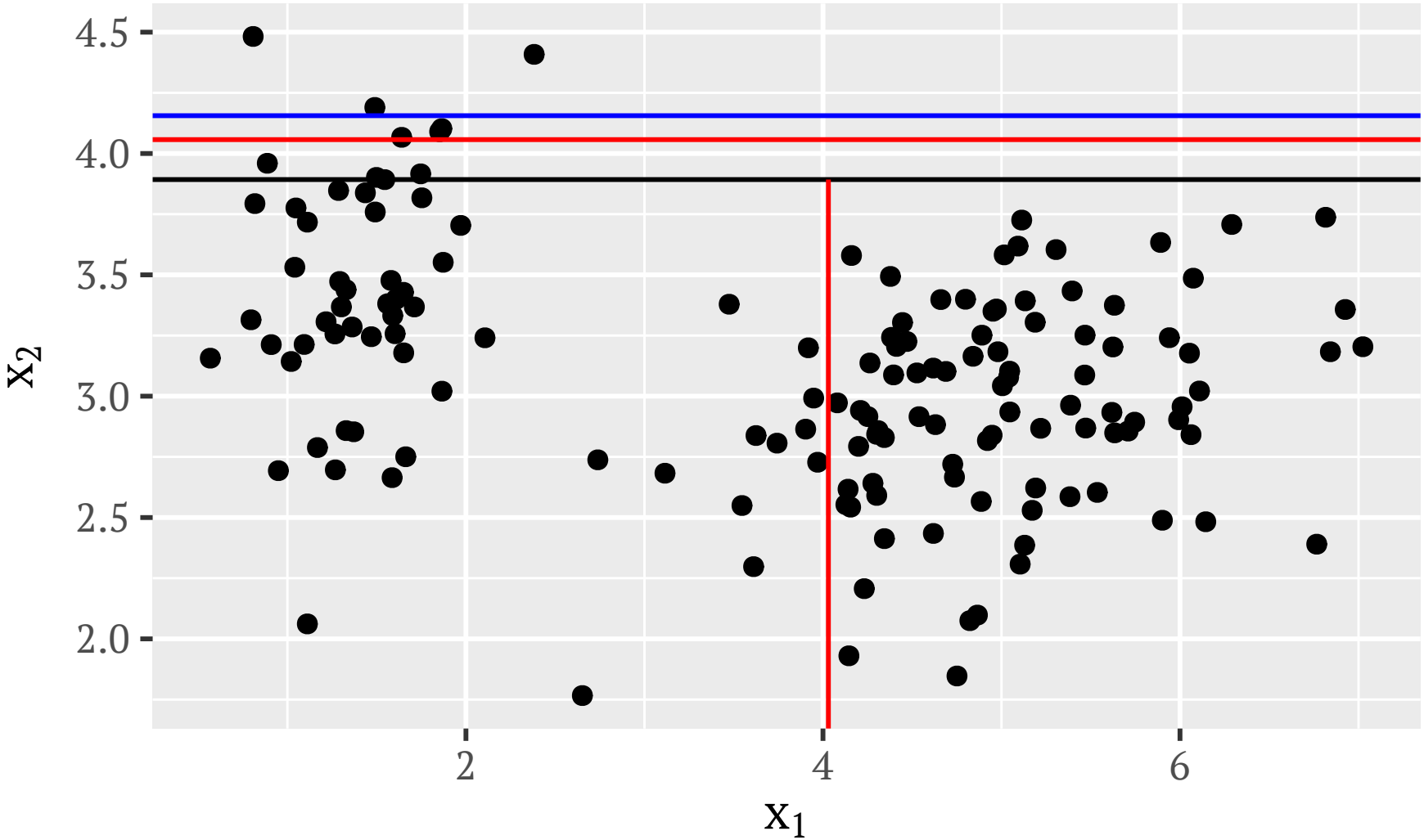
Collection of Coarse Random Marginals (CCRM)



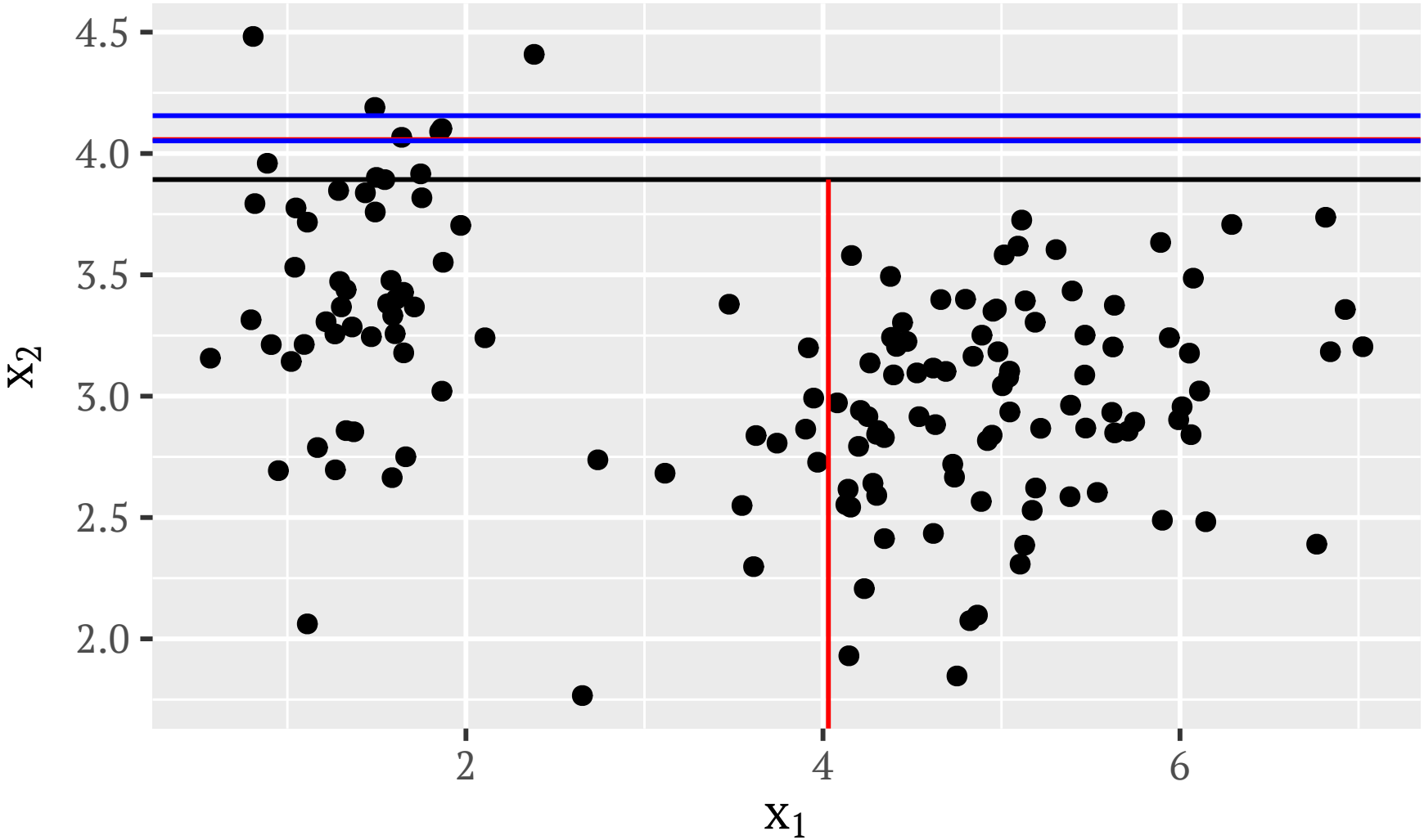
Collection of Coarse Random Marginals (CCRM)



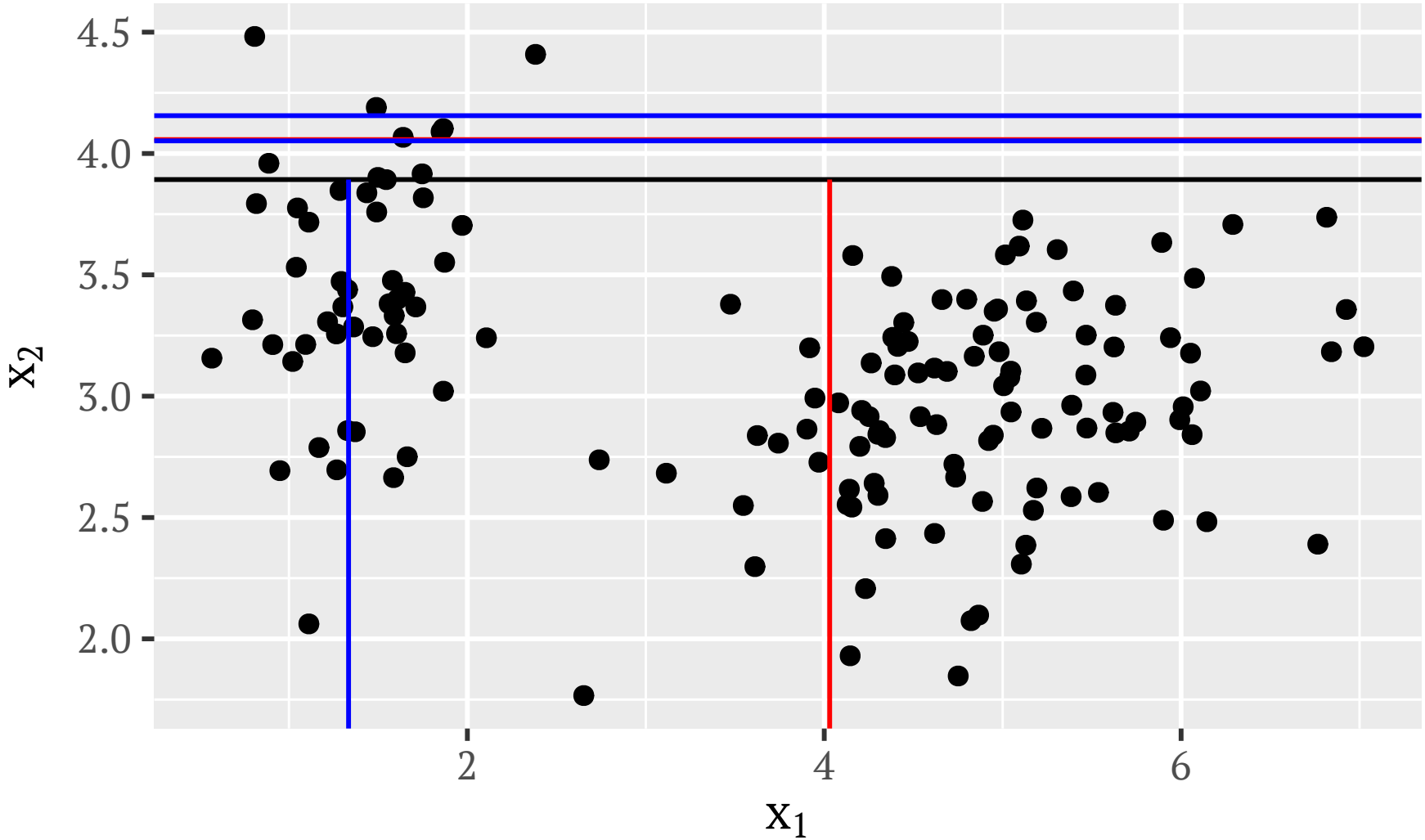
Collection of Coarse Random Marginals (CCRM)



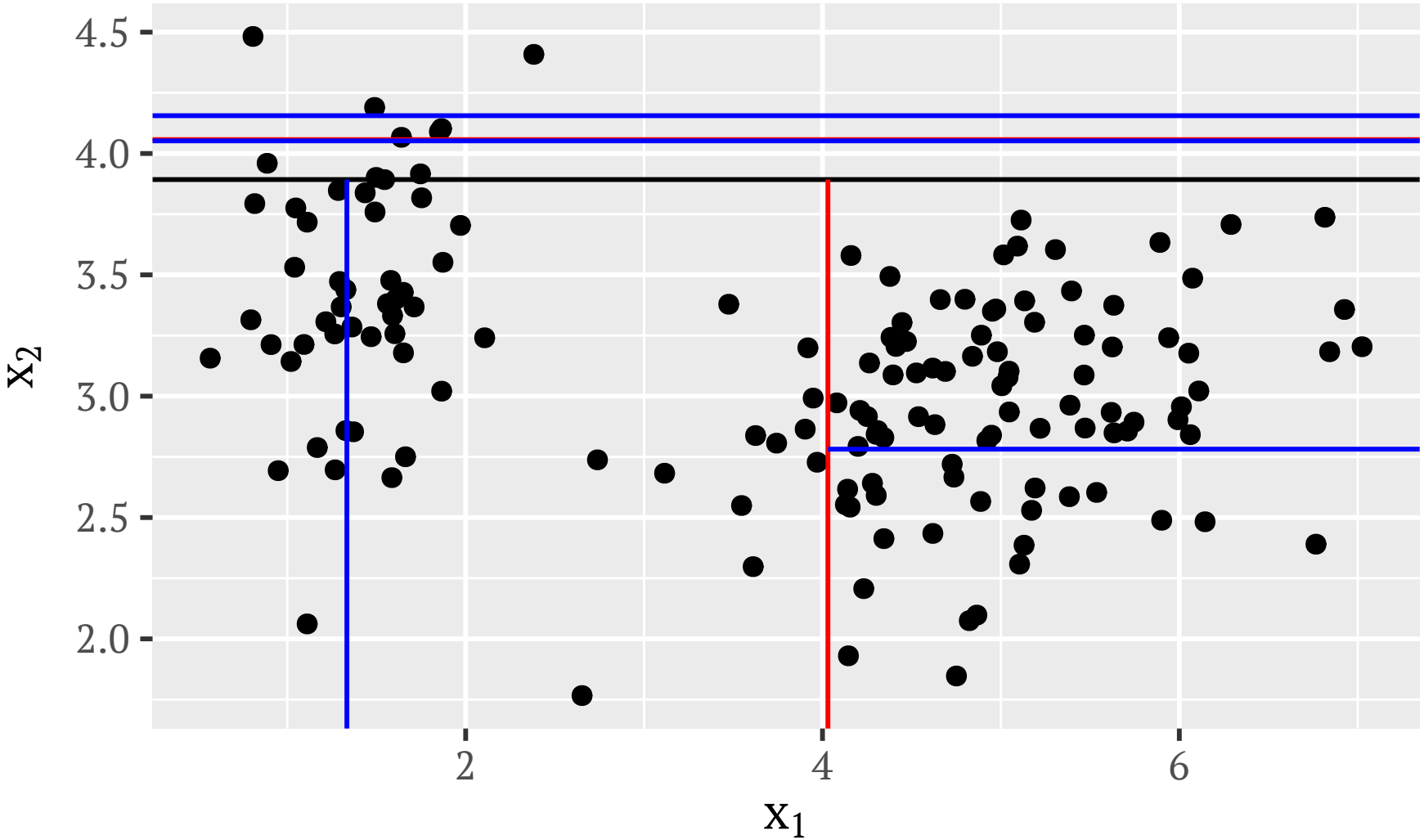
Collection of Coarse Random Marginals (CCRM)



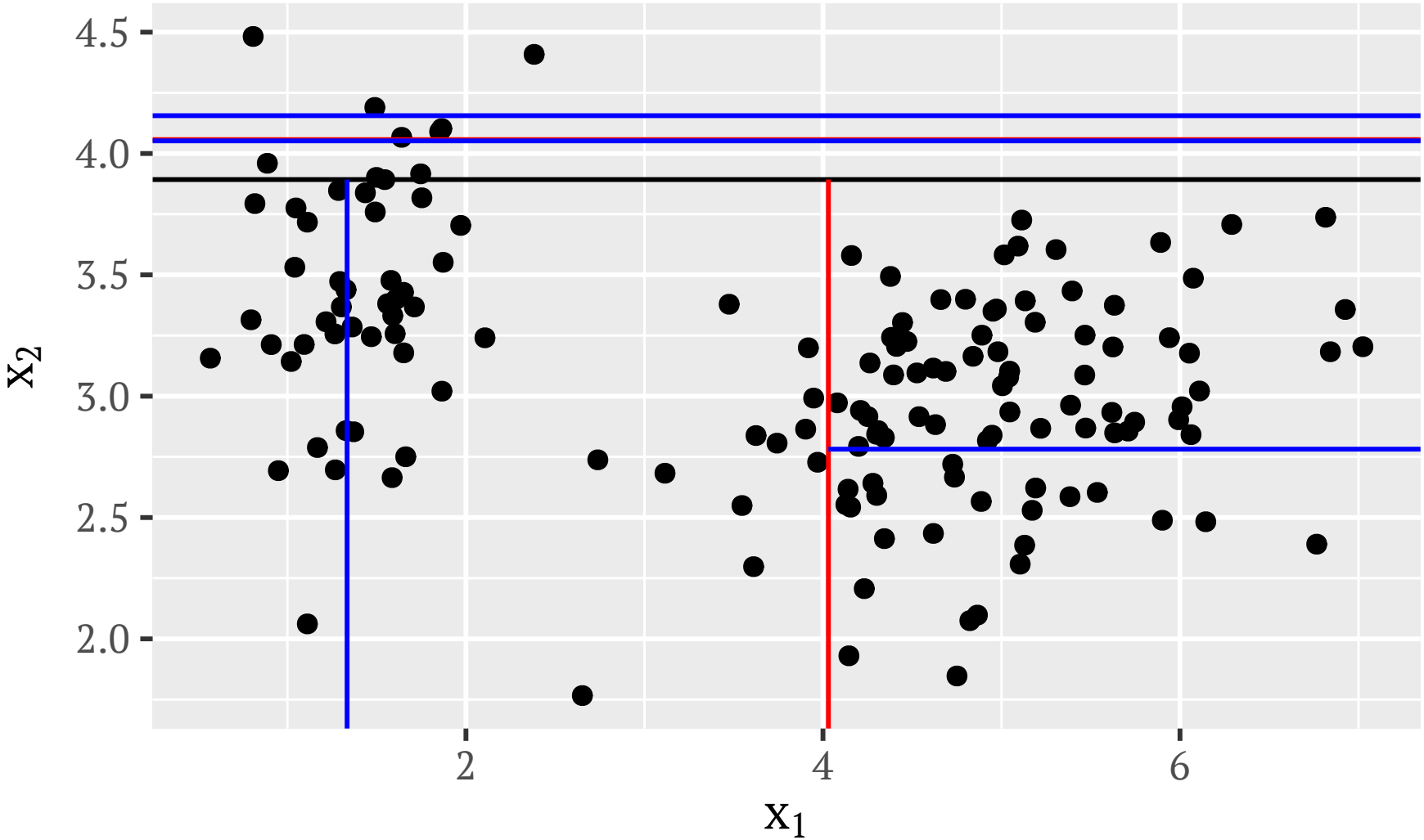
Collection of Coarse Random Marginals (CCRM)



Collection of Coarse Random Marginals (CCRM)



Collection of Coarse Random Marginals (CCRM)



$$S(X) = (\dots, 3, 3, 0, 3, 43, 33, 64, 24, \dots)$$

CCRM solutions

- Homomorphically computable pooling of summary statistics
  - **simple vector addition**
- Summary statistics that don't reveal model
  - **CCRM is completely random, grown the same way for all models and data sets. Only weak information about range of each variable leaked.**
- Homomorphically computable distance metric
  - **sum of squared differences**

Variance of distance metric per CRM

Lemma *Let the random variable V be multinomially distributed with success probabilities $p = (p_1, \ldots, p_k)$ for n trials. Then,*

$$
\begin{aligned}
&\operatorname{Var}\left(\sum_{i=1}^{k}(V_i - c_i)^2\right) \\
&= \sum_{i=1}^{k}\Big[({}^nC_{n-4} - n^2(n-1)^2)p_i^4 + (6\,{}^nC_{n-3} + 2n(n-1)(4c_i - n))p_i^3 \\
&\quad + (7n(n-1) - n^2 - 4c_i n(2n-3)(1+c_i))p_i^2 + (n + 4c_i n(c_i - 1))p_i \\
&\quad + \sum_{\substack{j=1 \\ i\neq j}}^{k}\Big[- n(2c_i - 1)(2c_j - 1)p_i p_j + 2n(n-1)(2c_j - 1)p_i^2 p_j \\
&\quad + 2n(n-1)(2c_i - 1)p_i p_j^2 - 2n(n-1)(2n-3)p_i^2 p_j^2\Big]\Big]
\end{aligned}
$$

$\Longrightarrow$ can be used to weight random marginals differently.

ABCDE: Approximate Bayesian Computation Done Encrypted

Tying it all together:

- ABC-PMC/SMC
- Homomorphic Secret Sharing with data pooling
- CCRM summary statistic protecting model/prior privacy
- Pooled $S(\cdot)$ computable encrypted from multiple data owners
- Distance computable encrypted and not learned by modeller
- Variance of each CRM computable encrypted for weighting

Selected connections in ABC literature

- Bernton, E., Jacob, P. E., Gerber, M., & Robert, C. P. (2017). Inference in generative models using the Wasserstein distance. *arXiv:1701.05146*.

- Gutmann, M. U., Dutta, R., Kaski, S., & Corander, J. (2017). Likelihood-free inference via classification. *Statistics and Computing*, 1-15.

- Fearnhead, P., & Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society: Series B*, 74(**3**), 419-474.

Toy example

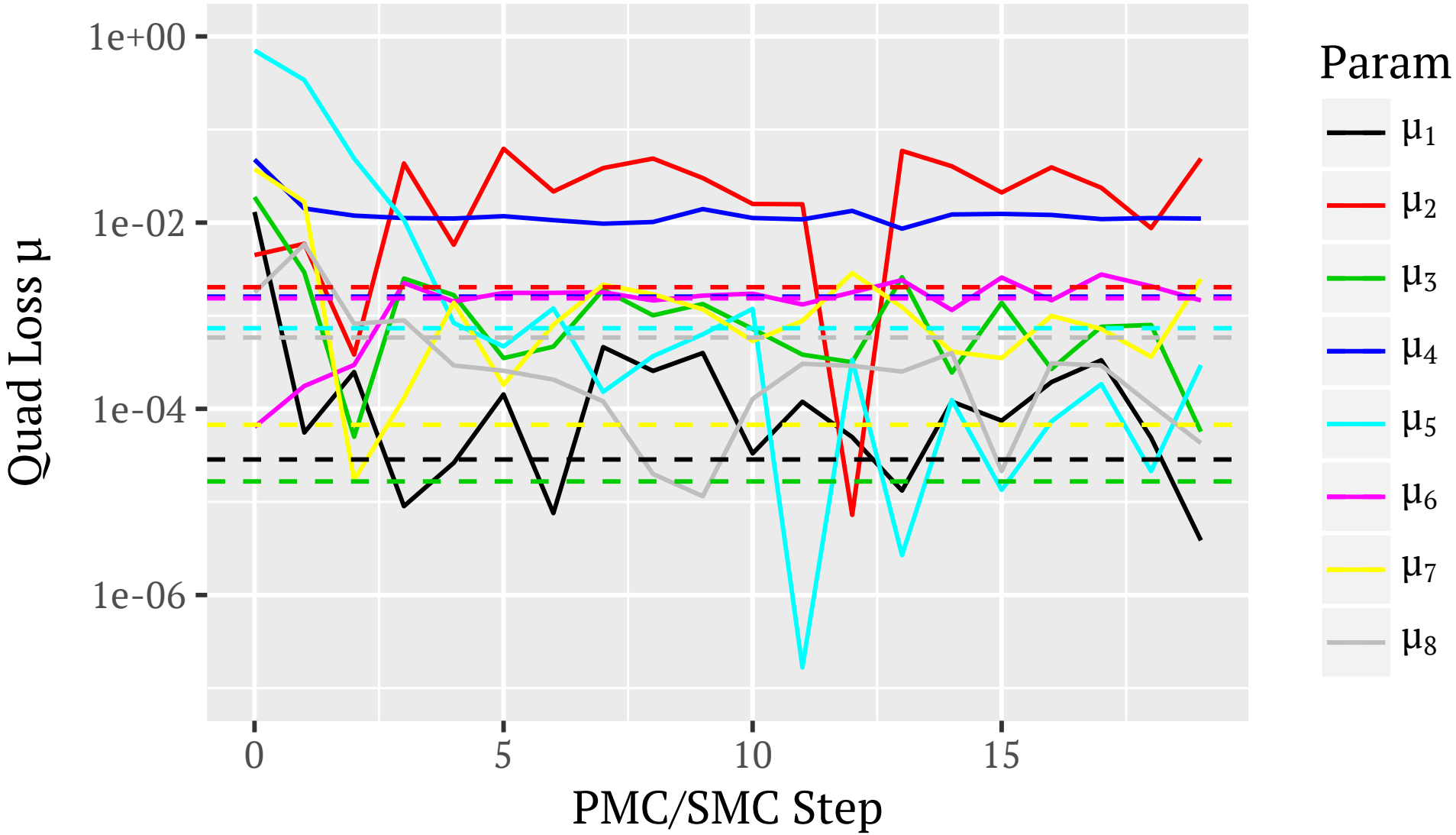
Super simple first example, 8-dimensional multivariate Normal.

$$X \sim \mathrm{N}(\boldsymbol{\mu} = \mathbf{0}, \Sigma = I)$$
$$\mu_i \sim \mathrm{N}(\eta_i, \sigma = 2)$$

where η_i chosen independently uniformly at random on the interval $[-1, 1]$ for repeated experiments.

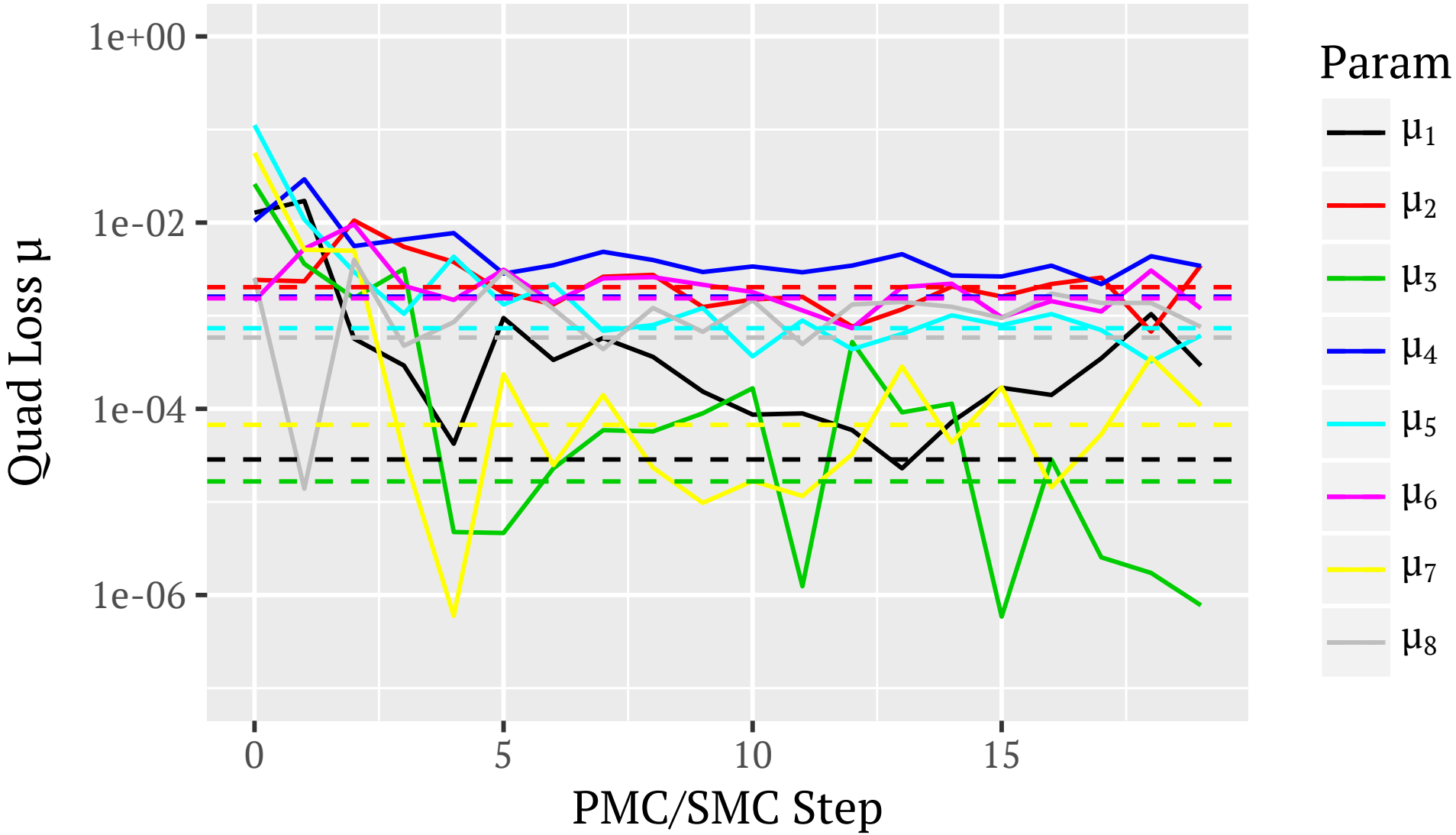
- Simulate $n = 1000$ observations
- Range of all dimensions taken to be $[-4, 4]$ for construction of CCRM, without checking true range of X
- Standard ABC used $S(X) = (\bar{x}_1, \ldots, \bar{x}_8)$

Toy example: 8D Normal, marginal quadratic loss



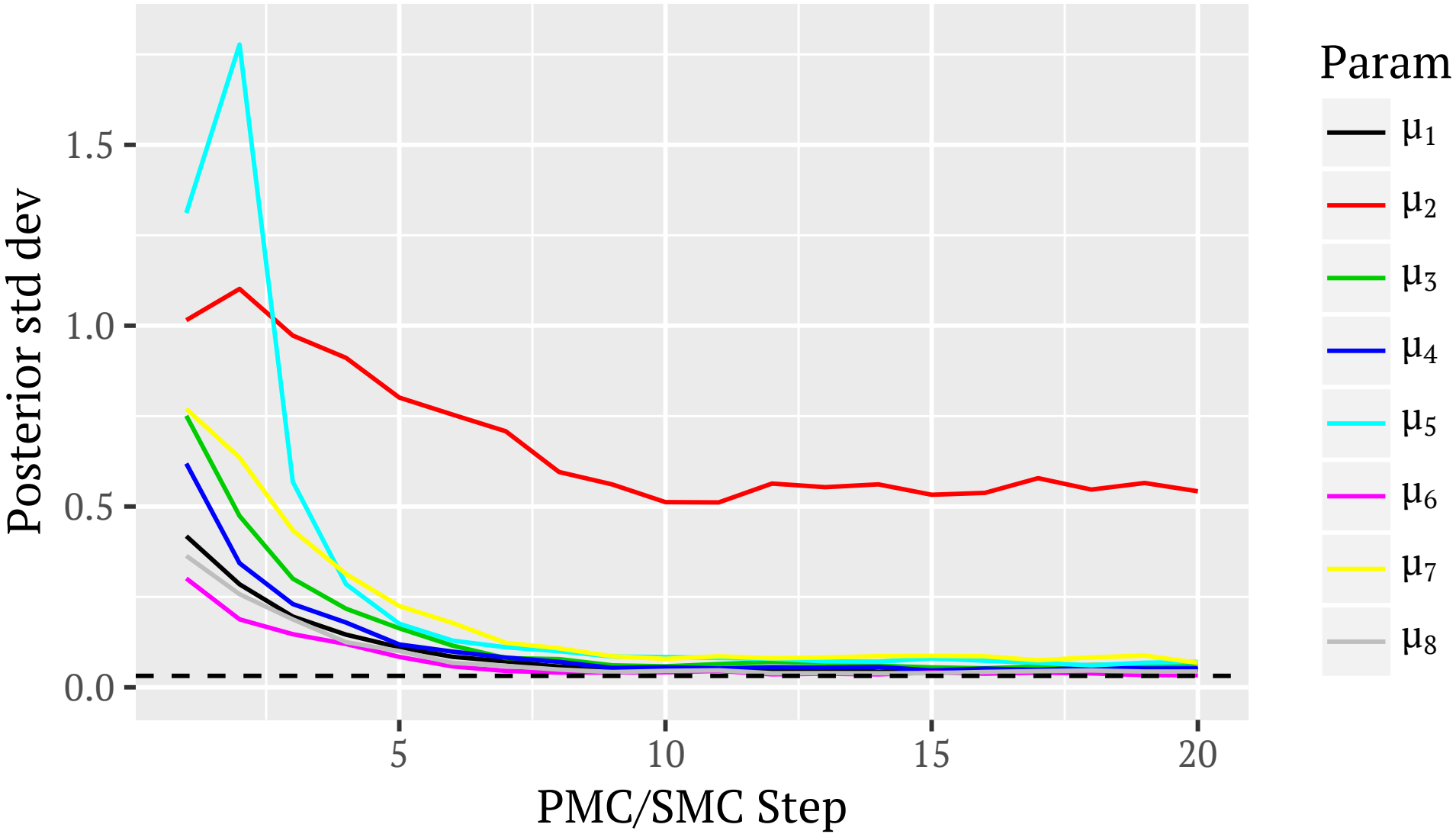
$$n = 10^3, T = 20, L = 2, m = 10^4, \alpha = 0.01$$

Toy example: 8D Normal, marginal quadratic loss



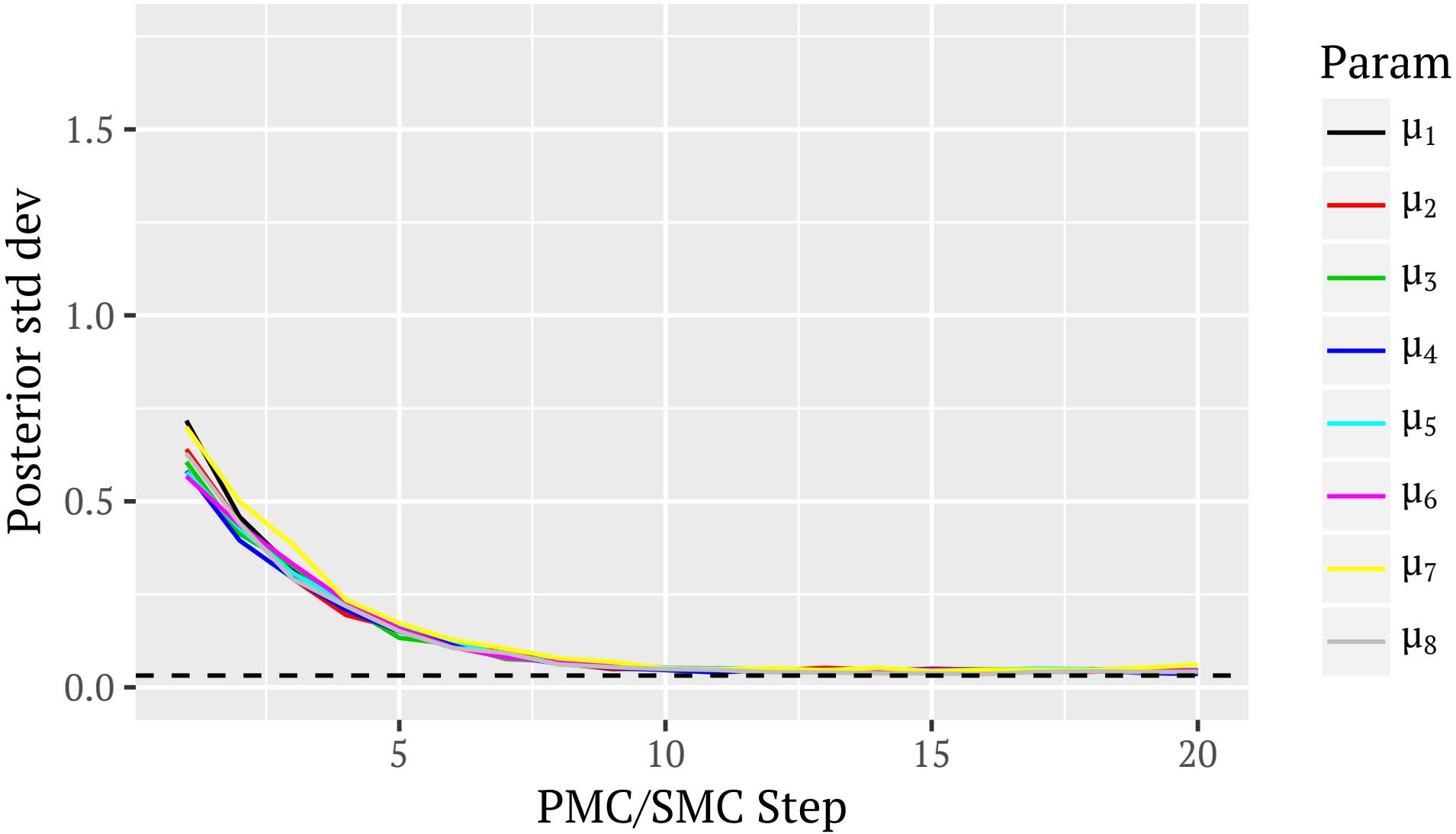
$$n = 10^3, T = 1000, L = 2, m = 10^4, \alpha = 0.01$$

Toy example: 8D Normal, marginal posterior σ



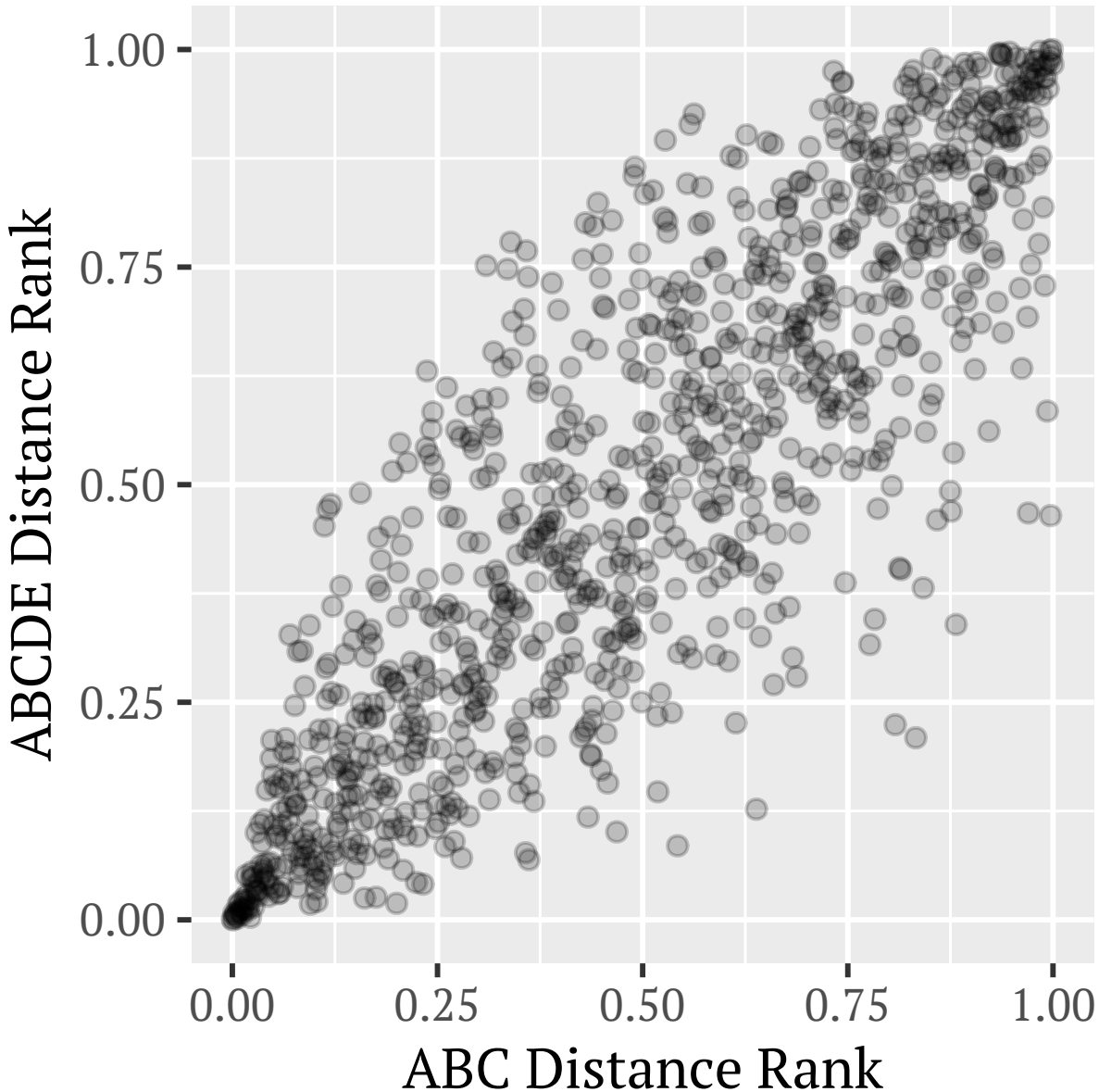
$$n = 10^3, T = 20, L = 2, m = 10^4, \alpha = 0.01$$

Toy example: 8D Normal, marginal posterior σ



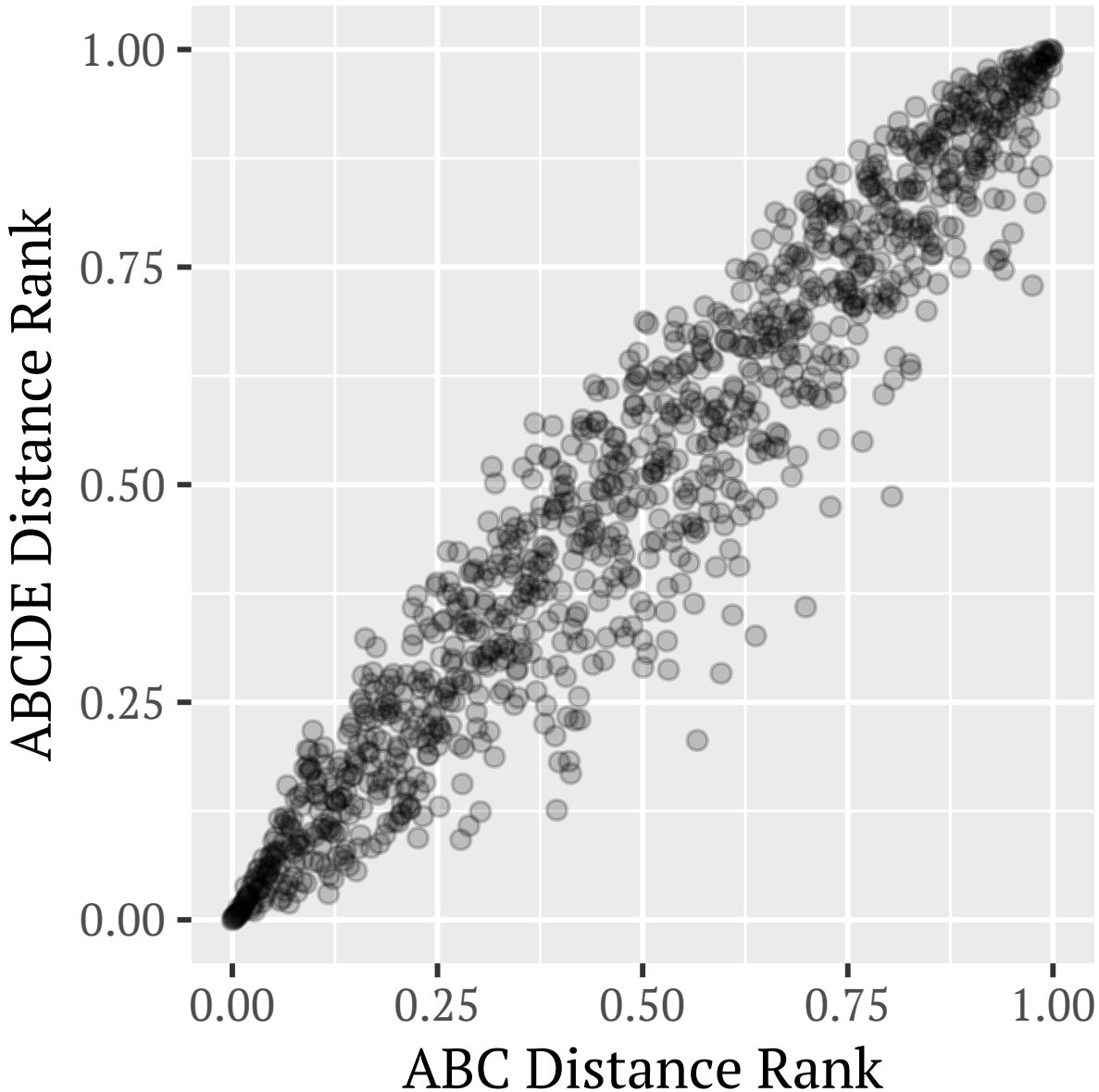
$$n = 10^3, T = 1000, L = 2, m = 10^4, \alpha = 0.01$$

Toy example: distance concordance



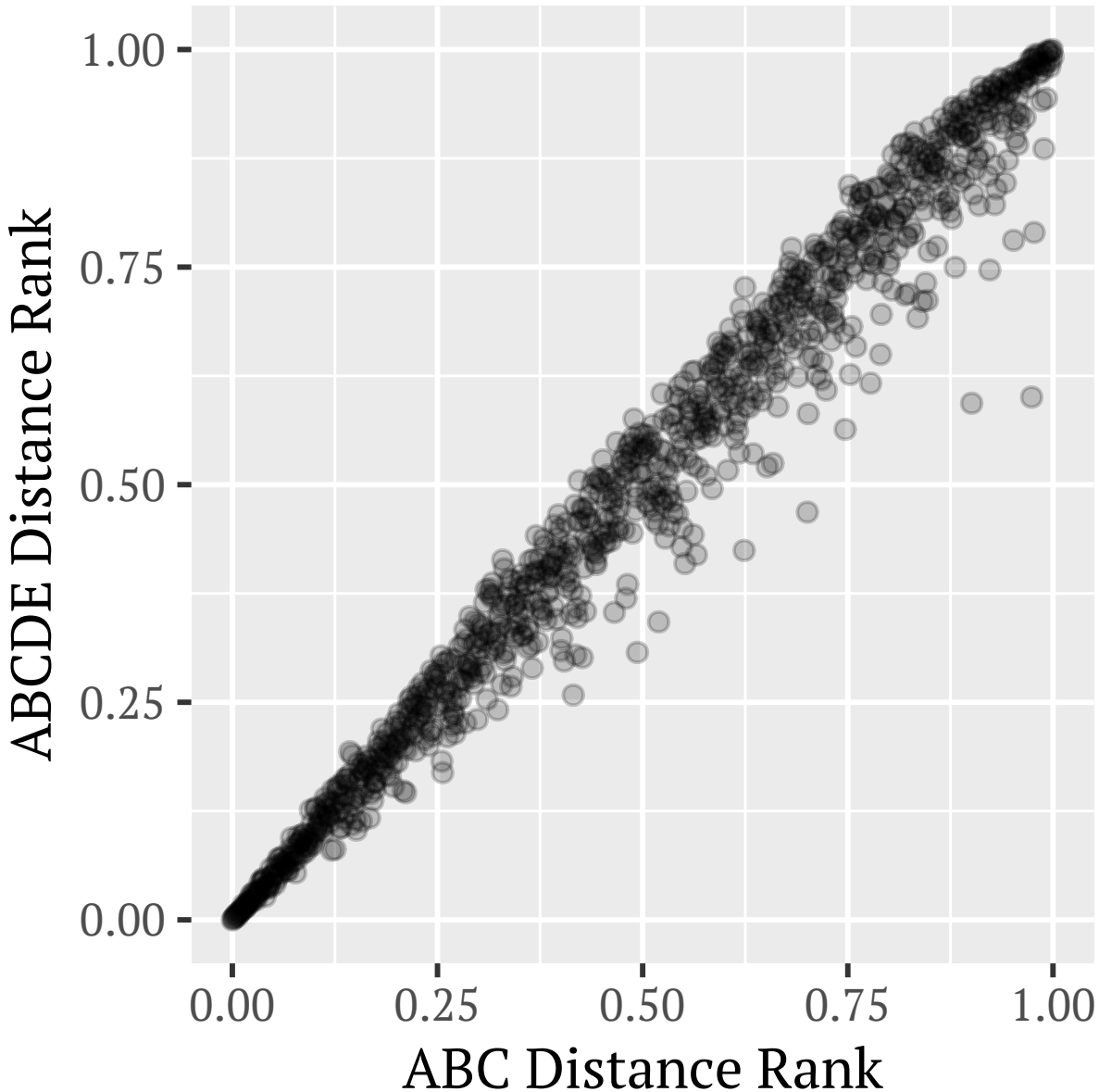
$T = 20$

Toy example: distance concordance



$T = 100$

Toy example: distance concordance



$T = 1000$

Expected quadratic loss

Can understand lowest ABC error achievable without Monte Carlo error:

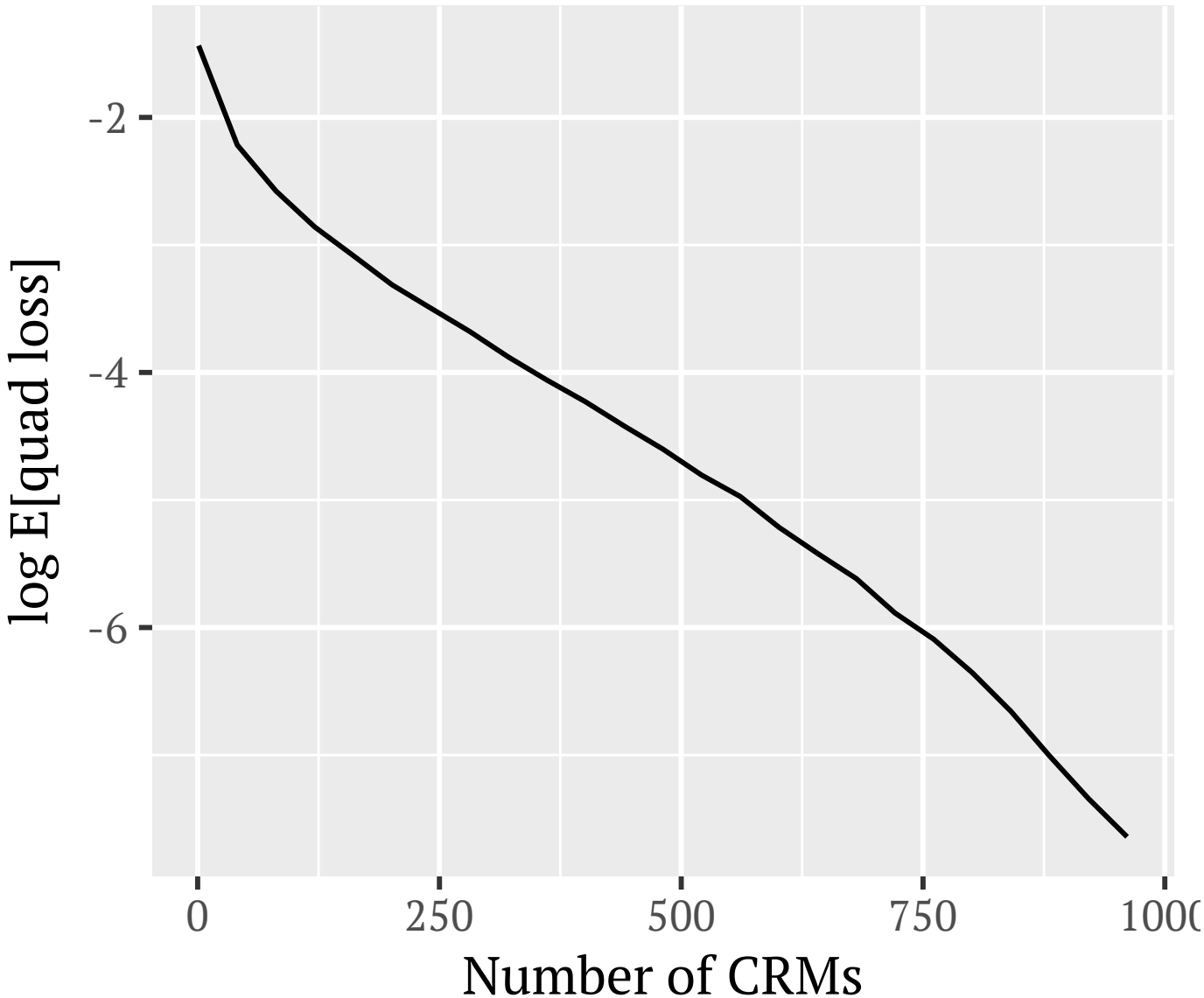
$$\mathbb{E}\left[(\mu - \hat{\mu})^2 \,|\, T = t\right]$$
$$= \frac{1}{|\mathcal{A}^t|} \int_{\mathcal{A}^t} \left(\mu - \int_{-\infty}^{\infty} \theta \, \mathbb{P}\left(S(x) = S(x^{\text{obs}}) \,|\, da_1, \ldots, da_t\right) \, \pi(d\theta)\right)^2$$

because for 1-level CRMs:

$$\mathbb{P}\left(S(x) = S(x^{\text{obs}}) \,|\, da_1, \ldots, da_t\right)$$
$$= \prod_{k=1}^{t} \binom{n}{m_k} F_{v_k}(X < a_k)^{m_k} (1 - F_{v_k}(X < a_k))^{n - m_k}$$

where $m_k = \#\{i : x_i^{\text{obs}} < a_k\}$.

Expected quadratic loss



g-and-k distribution (Haynes et al. 1997)

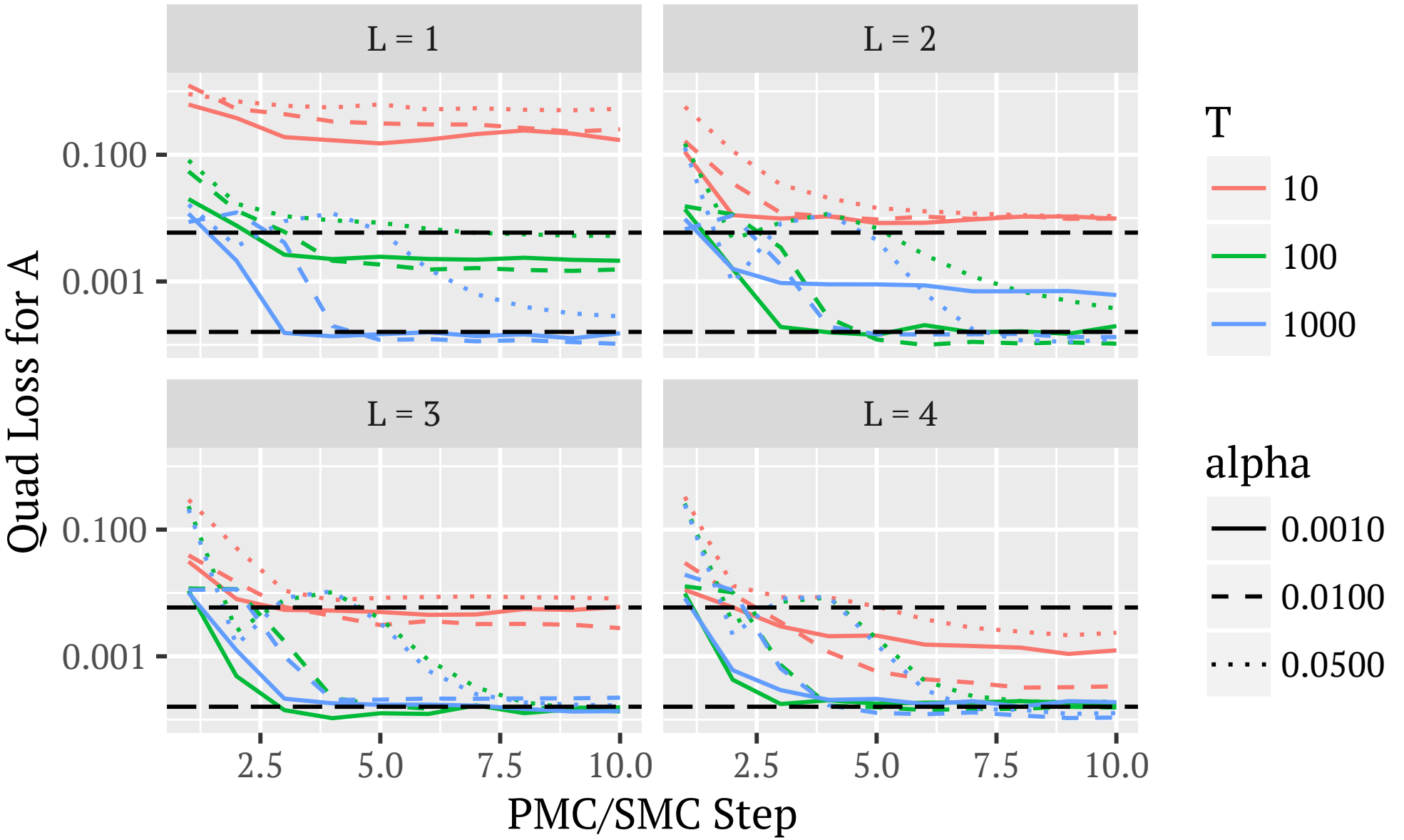
Defined via inverse distribution function

$$F^{-1}(x \mid A, B, g, k) = A + B \left[1 + 0.8 \frac{1 - \exp\left(-g\Phi^{-1}(x)\right)}{1 + \exp\left(-g\Phi^{-1}(x)\right)} \right] \left(1 + \Phi^{-1}(x)^2\right)^k \Phi^{-1}(x)$$

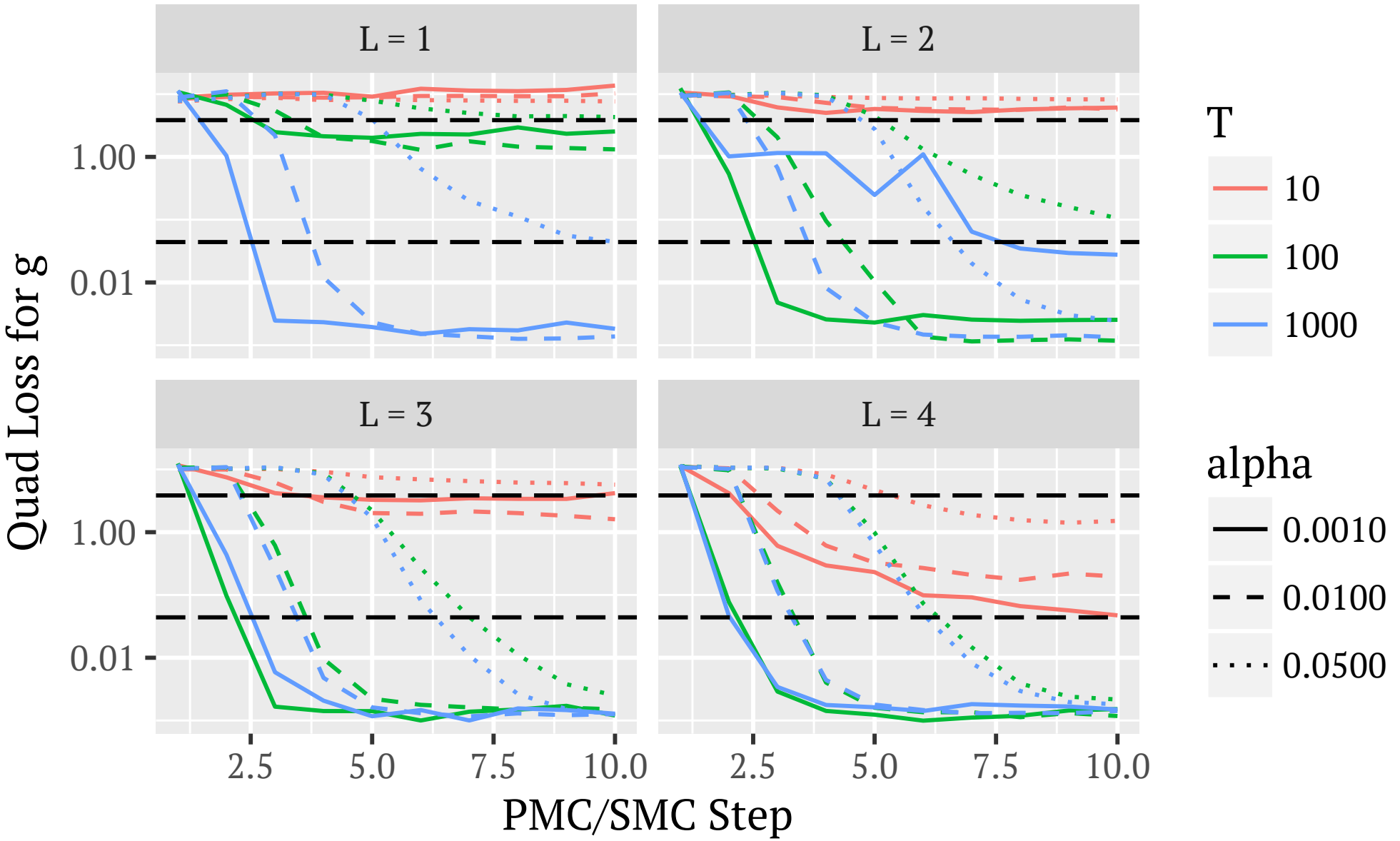
Following Allingham et al. (2009) and Fearnhead & Prangle (2012), take:

- $A = 3, B = 1, g = 2, k = \frac{1}{2}$
- simulate $n = 10000$ observations
- standard ABC uses the order statistics, $S(X) = (x_{(1)}, \ldots, x_{(n)})$

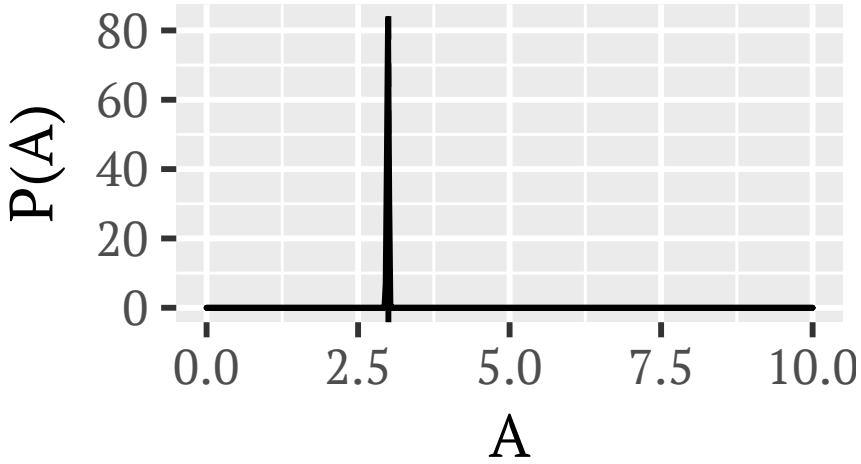
g-and-k: quadratic loss

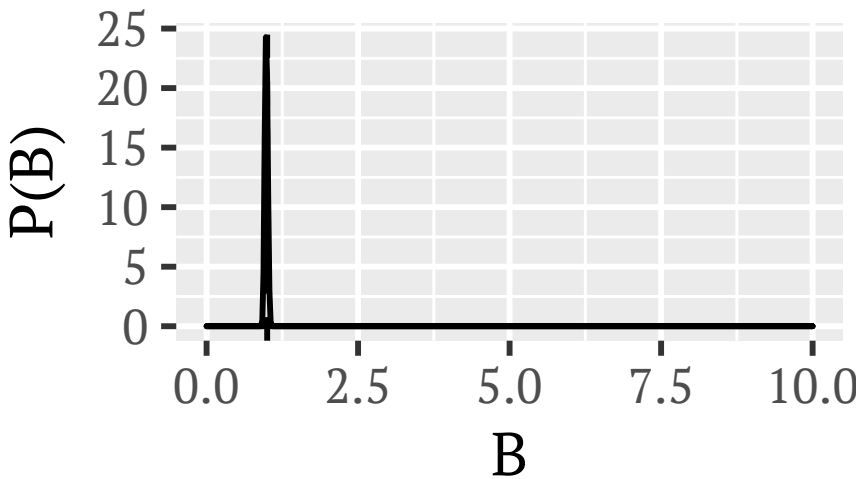


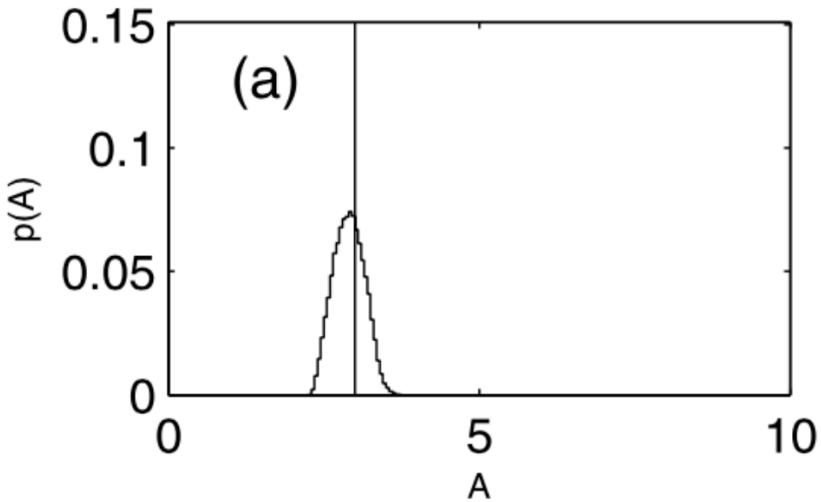
g-and-k: quadratic loss

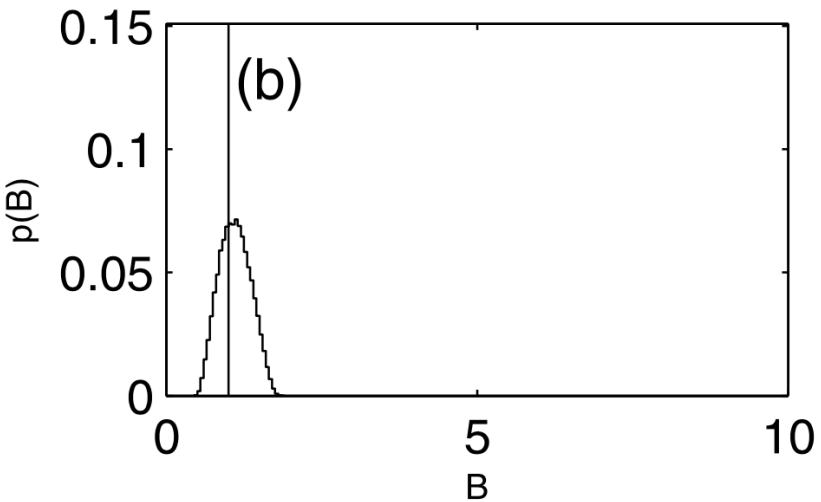


g-and-k: density plots





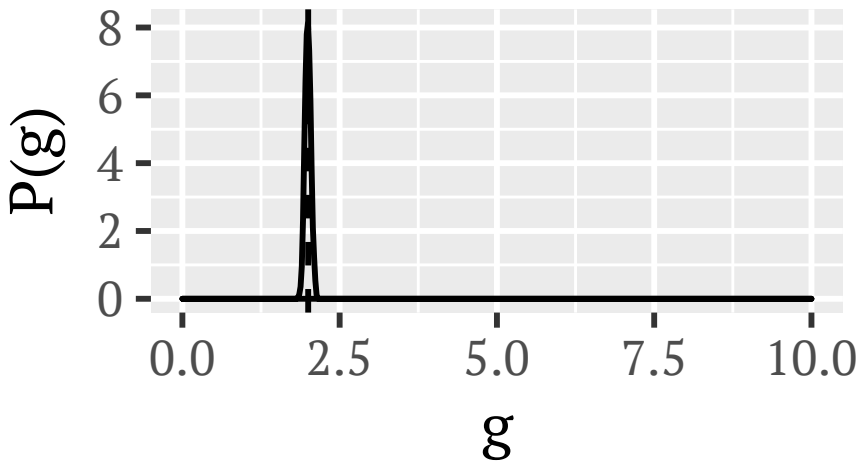


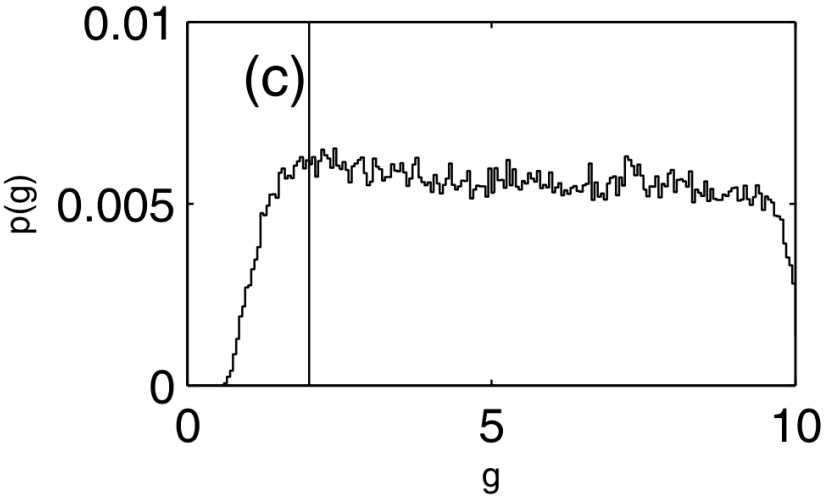


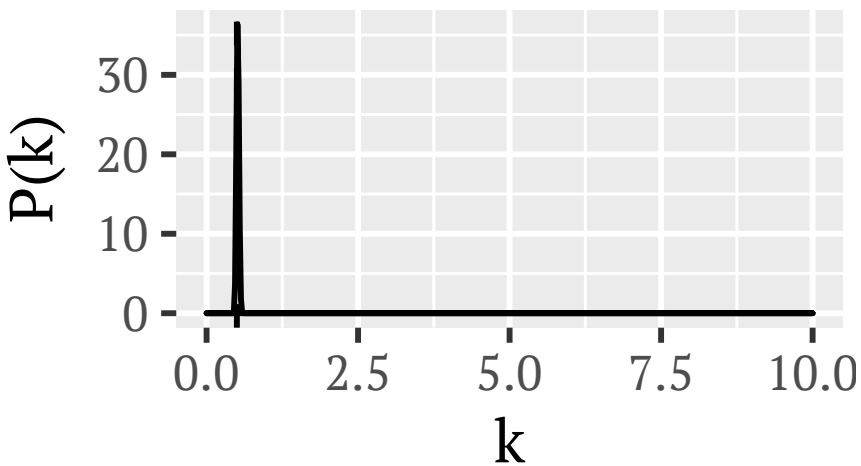
$T = 1000, L = 3, m = 10^5, \alpha = 0.01$

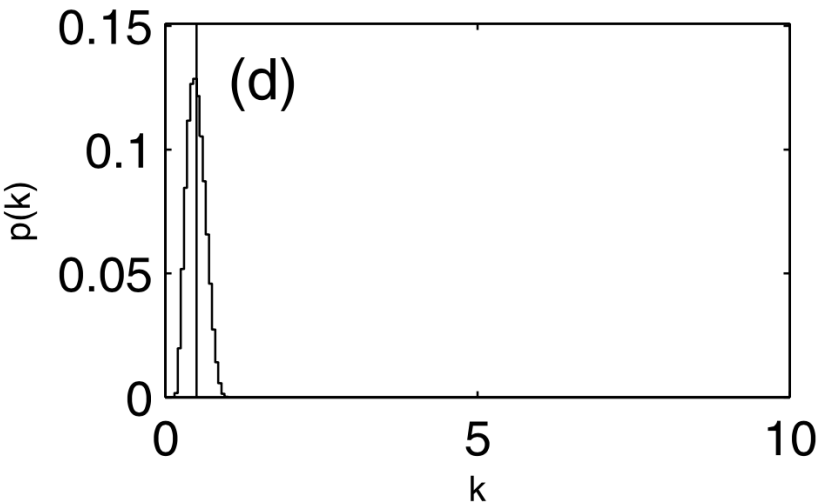
Allingham et al (2009)

g-and-k: density plots









$T = 1000, L = 3, m = 10^5, \alpha = 0.01$ Allingham et al (2009)

Tuberculosis Transmission (Tanaka et al. 2006)

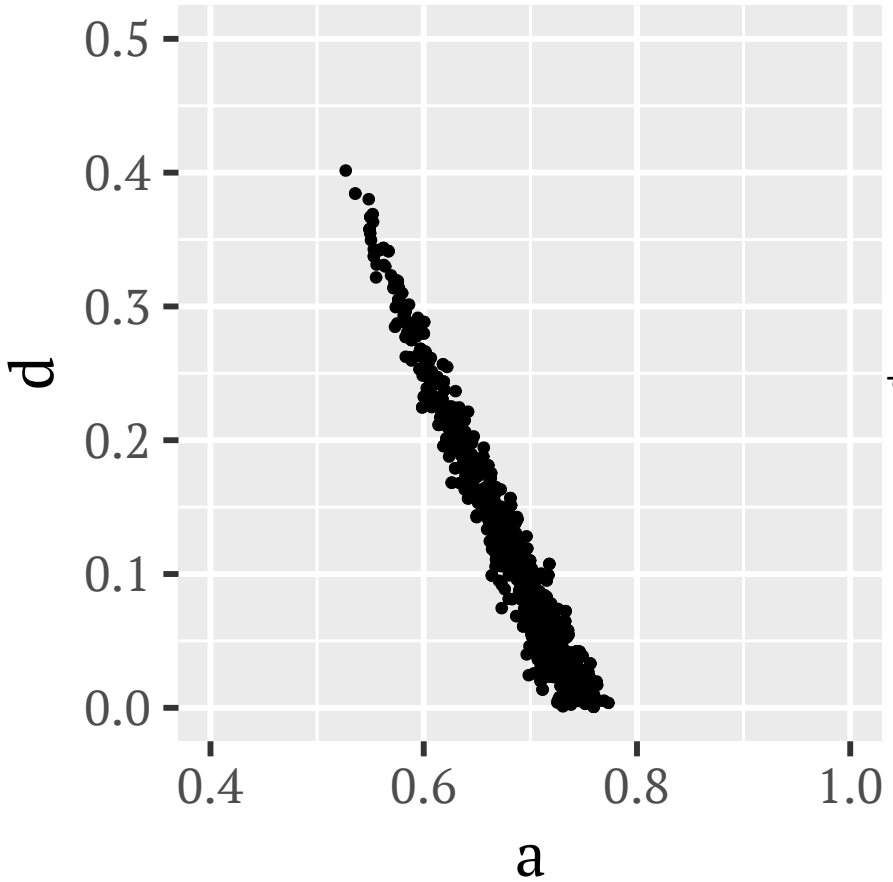
Model of transmission of disease,

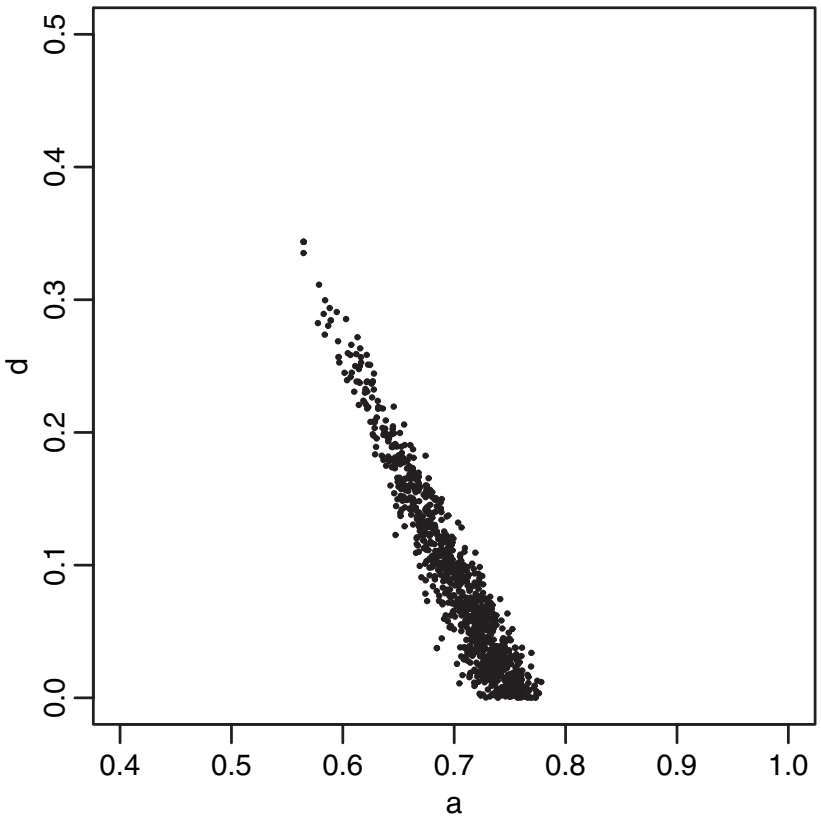
- 'birth' of new infections, rate α
- 'death' recovery or mortality of carrier, rate δ
- 'mutation' genotype of bacterium mutates within carrier, rate θ (infinite-alleles assumption)

$X_i(t)$ num infections type i at time t; $G(t)$ num unique genotypes.

- San Francisco tuberculosis data 1991/2, 473 samples (no time)
- Fearnhead & Prangle (2012) transform $(\alpha/(\alpha+\delta+\theta), \delta/(\alpha+\delta+\theta))$
- $S(X) = (G(t_{\text{end}})/473, 1 - \sum_i (X(t_{\text{end}})/473)^2)$

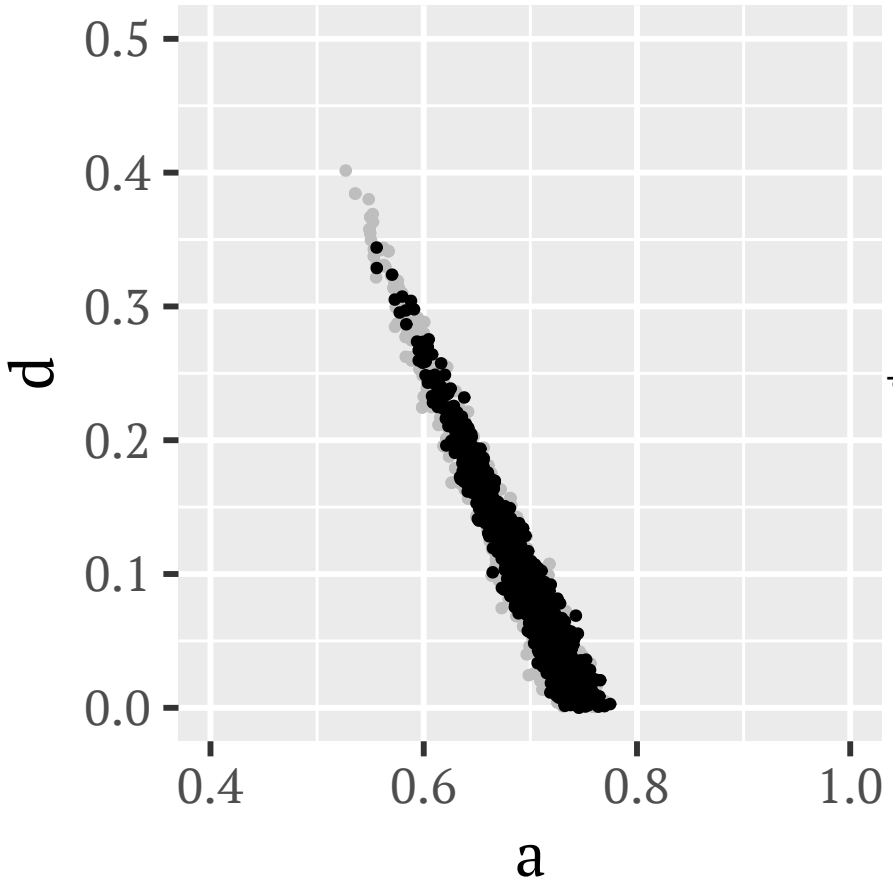
Posterior samples

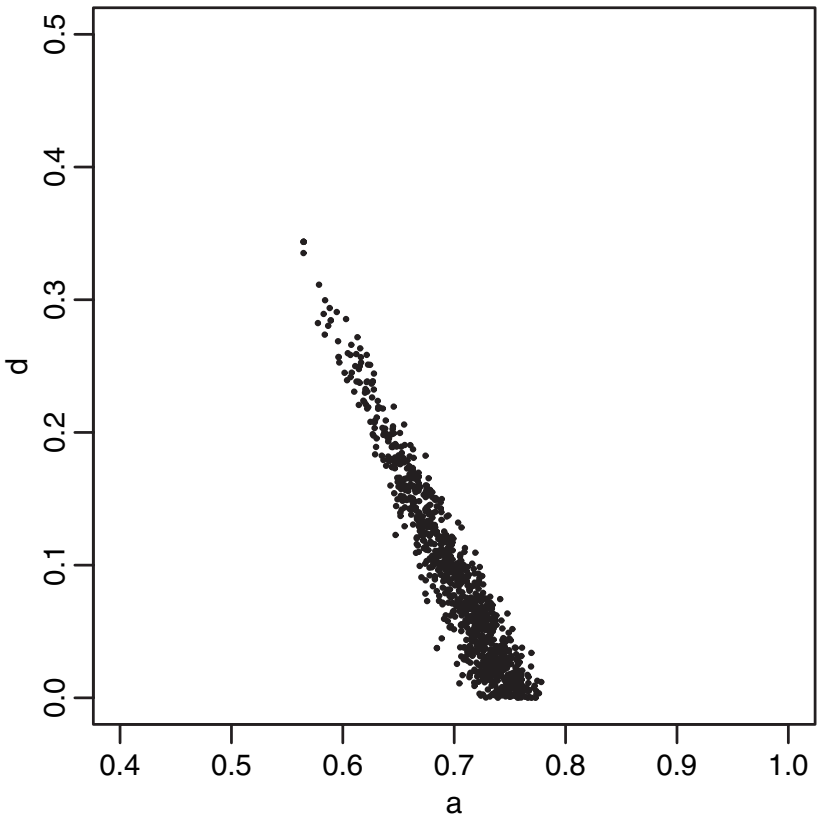




Semi-automatic ABC

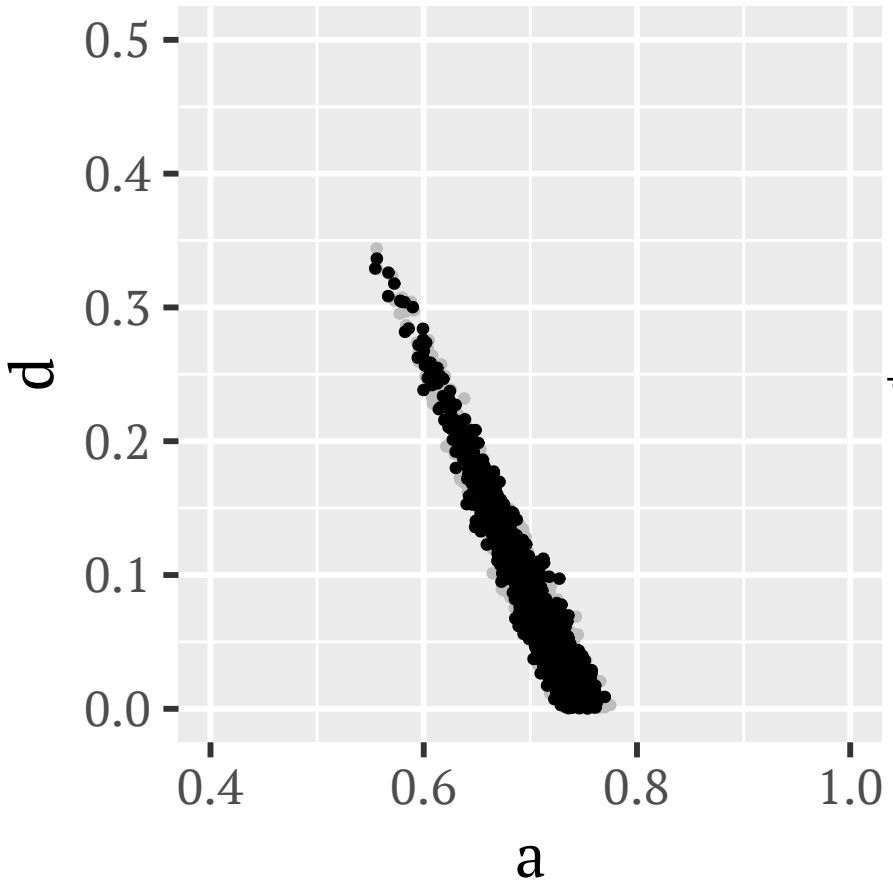
Posterior samples

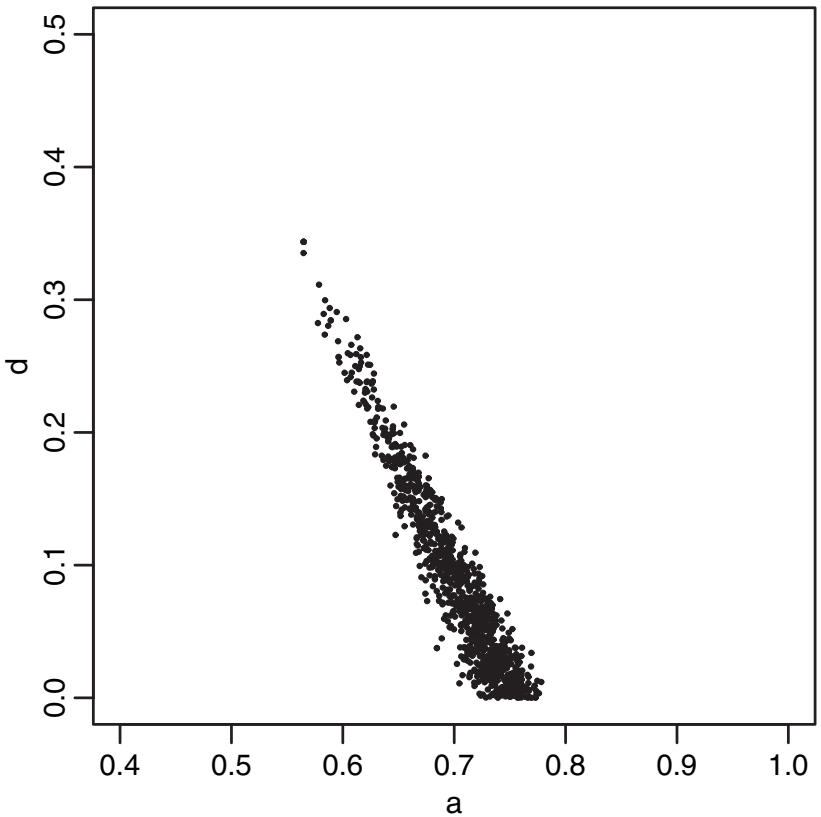




Semi-automatic ABC

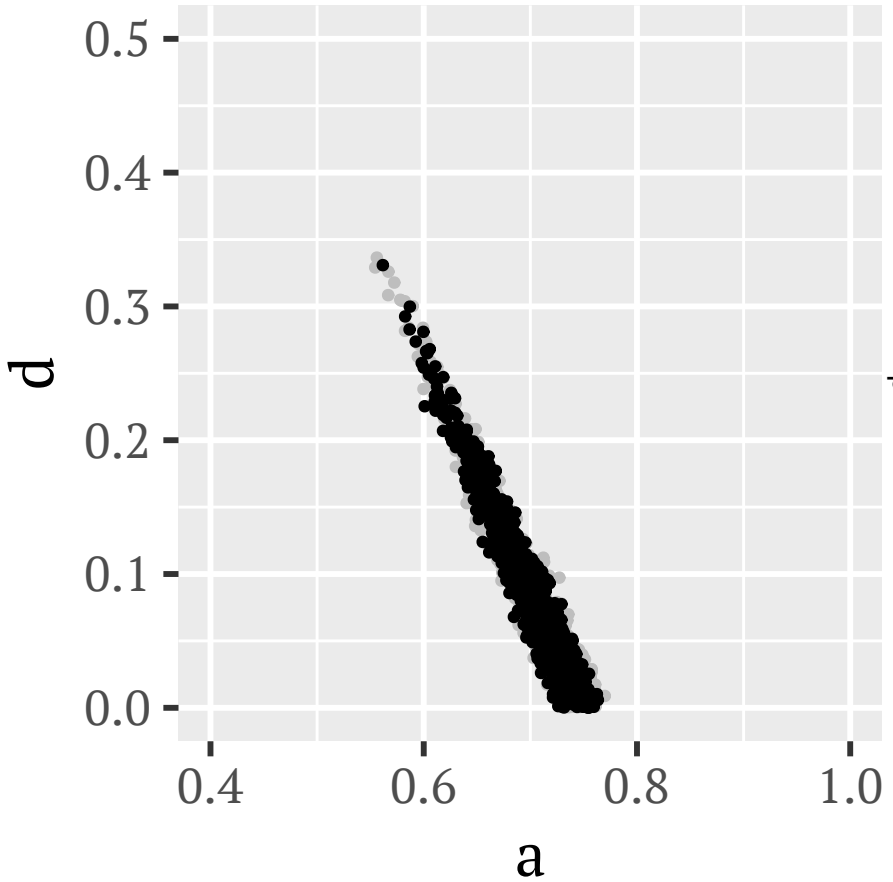
Posterior samples

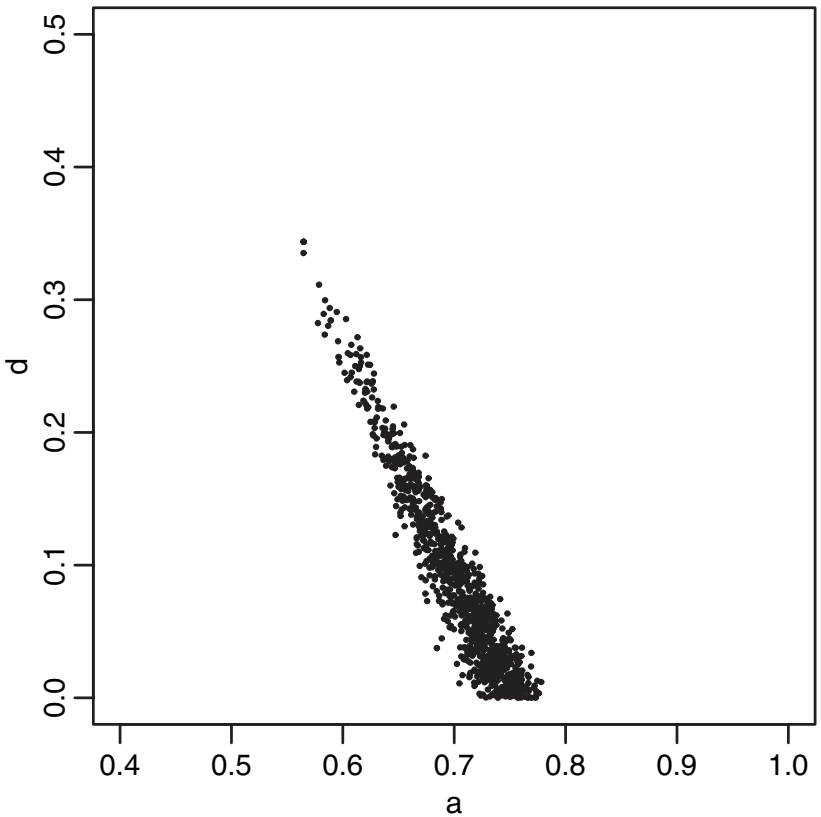




Semi-automatic ABC

Posterior samples





Semi-automatic ABC

Theory (Sam Livingstone, UCL)

Proposition 1:

When $d = 1$, if $\rho_T(S(x), S(y)) := \sum_{k=1}^{T} \rho(S_k(x), S_k(y))$ for some discrepency $\rho : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ then

$$\lim_{T\to\infty} \frac{\rho_T(S(x), S(y))}{T} \xrightarrow{a.s.} \int_{-\infty}^{\infty} \rho(F_X(z), F_Y(z))dz,$$

where F_X and F_Y are the empirical cumulative distribution functions for the data sets $x_{1:n}$ and $y_{1:n}$ respectively. In particular

1. If $\rho_T(S(x), S(y)) := \|S(x) - S(y)\|_1$, then $T^{-1}\rho_T(S(x), S(y)) \xrightarrow{a.s.} W_1(x_{1:n}, y_{1:n})$
2. If $\rho_T(S(x), S(y)) := \|S(x) - S(y)\|_2^2$, then $T^{-1}\rho_T(S(x), S(y)) \xrightarrow{a.s.} \int_{-\infty}^{\infty}(F_X(z) - F_Y(z))^2 dz$.

Conclusions

- So far, this ...
  - Provides encrypted inference whilst preserving model, prior and data privacy
  - Enables pooling of multiple data owners
  - Theoretically arbitrary low-dimensional models
- ... but this is work-in-progress! Currently in progress:
  - Method of ensuring differential privacy
  - Encrypted software implementation of this scheme
  - Best use of weights
  - Fuller understanding of accuracy for CCRM choices
  - Data as a service
- Perhaps also useful as a model independent summary statistic for unencrypted ABC too?
- Questions, comments and discussion welcome!

Conclusions

- So far, this ...
  - Provides encrypted inference whilst preserving model, prior and data privacy
  - Enables pooling of multiple data owners
  - Theoretically arbitrary low-dimensional models
- ... but this is work-in-progress! Currently in progress:
  - Method of ensuring differential privacy
  - Encrypted software implementation of this scheme
  - Best use of weights
  - Fuller understanding of accuracy for CCRM choices
  - Data as a service
- Perhaps also useful as a model independent summary statistic for unencrypted ABC too?
- Questions, comments and discussion welcome!

Thank you!

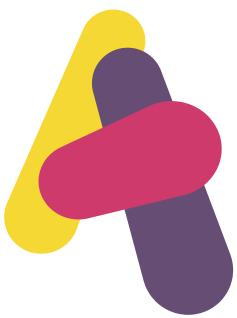
Shameless plug! Knowledge Transfer Partnership

Forthcoming KTP associate job, based at Atom Bank working with me and Camila Caiado at Durham University.

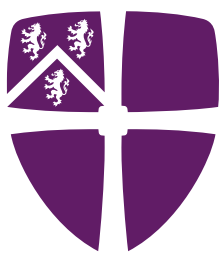
Jointly working with Computer Science KTP associate based at Atom and working with Newcastle University.

Statistical modelling and encrypted statistics for mortgage books.

Expected to advertise for an August – October 2018 start.



Atom bank



Durham University

References I

Bost, R., Popa, R. A., Tu, S., & Goldwasser, S. (2015). Machine learning classification over encrypted data. *NDSS*.

Esperança, P. M., Aslett, L. J. M., & Holmes, C. C. (2017). Encrypted accelerated least squares regression. Singh A. & Zhu J. (eds) *Proceedings of the 20th international conference on artificial intelligence and statistics*, Proceedings of machine learning research, Vol. 54, pp. 334–43. Fort Lauderdale, FL, USA: PMLR.

Gascón, A., Schoppmann, P., Balla, B., Raykova, M., Doerner, J., Zahur, S., & Evans, D. (2017). Privacy-preserving distributed linear regression on high-dimensional data. *Proceedings on privacy enhancing technologies*, Vol. 4, pp. 345–64. DOI: 10.1515/popets-2017-0053

Goldwasser, S., & Micali, S. (1982). Probabilistic encryption & how to play mental poker keeping secret all partial information. *Proceedings of the fourteenth annual acm symposium on theory of computing*, pp. 365–77.

Graepel, T., Lauter, K., & Naehrig, M. (2012). ML confidential: Machine learning on encrypted data. *Proceedings of the 15th international conference on information security and cryptology*, Lecture notes in computer science, Vol.

References II

7839, pp. 1–21. Springer-Verlag. DOI: 10/bdxc

Lauter, K., López-Alt, A., & Naehrig, M. (2014). Private computation on encrypted genomic data. *International conference on cryptology and information security in latin america*, pp. 3–27.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. *International conference on the theory and applications of cryptographic techniques*, pp. 223–38.

Wu, D., & Haven, J. (2012). *Using homomorphic encryption for large scale statistical analysis*. Stanford University.