Blessings of Dimensionality: Finding the optimal k for nearest-neighbor projected-distance feature selection (for detecting interaction and main effects in high dimensional data)

Bryan A. Dawkins[1], Trang T. Le[2] and Brett A. McKinney[1,3,*]
[1]Department of Mathematics, University of Tulsa, Tulsa, OK 74104, USA
[2]Department of Biostatistics, Epidemiology and Informatics, University of Pennsylvania, Philadelphia, PA 19104
[3]Tandy School of Computer Science, University of Tulsa, Tulsa, OK 74104, USA.

Abstract

It is commonly known that high-throughput data has many inherent statistical challenges, such as multiple testing, sparsity and over fitting. Collectively, these challenges are known as the Curse of Dimensionality. Here we highlight an important Blessing of Dimensionality: the ability to identify interactions with neighborhoods of instances. We review the nearest-neighbor concept for finding interactions among attributes. We present a novel simulation method for generating interactions from random networks with fine-tuned control over interaction effect size. Using our new simulation strategy, we determine optimal fixed-k for neighborhood computation in nearest-neighbor distance-based feature selection for different combinations of data dimensions m (number of instances) and p (number of attributes), different ratios of main/interaction effect among functional attributes, and different combinations of main and interaction effect sizes. We discuss ways to maximize the blessings and minimize the curses of dimensionality to reliably identify interactions. Our results will show how optimal fixed-k changes under a variety of conditions that we see in real data, which will serve as a guide for other researchers using nearest-neighbor distance-based feature selection.

Author summary

Introduction

Relief-based methods identify interacting attributes as important by using nearest-neighbor information in higher dimensions (the "blessings of dimensionality"). Myopic methods that do not account for information from higher dimensions such as univariate tests are susceptible to false negatives when there are interactions. For example, in the plot of variable A versus C in a three-variable simulation (Fig. 1-I), variable A appears to show no difference between cases and controls (the marginal group means are the same). However, A is actually simulated to have a strong differential correlation with B, conditioned on the outcome variable (Fig. 1-II). Current Relief-based methods determine the importance of an attribute by computing the average difference of attribute A value between a target instance (X) and its nearest instance from the opposite class (Miss), $d_{\mathrm{X,M}}(A)$, subtracted from the similarly projected difference of target X and its nearest instance from the same class (Hit), $d_{\mathrm{X,H}}(A)$. A positive value from this calculation, i.e., $d_{\mathrm{X,M}}(A) - d_{\mathrm{X,H}}(A) > 0$, suggests that attribute A is useful for discriminating between cases and controls.

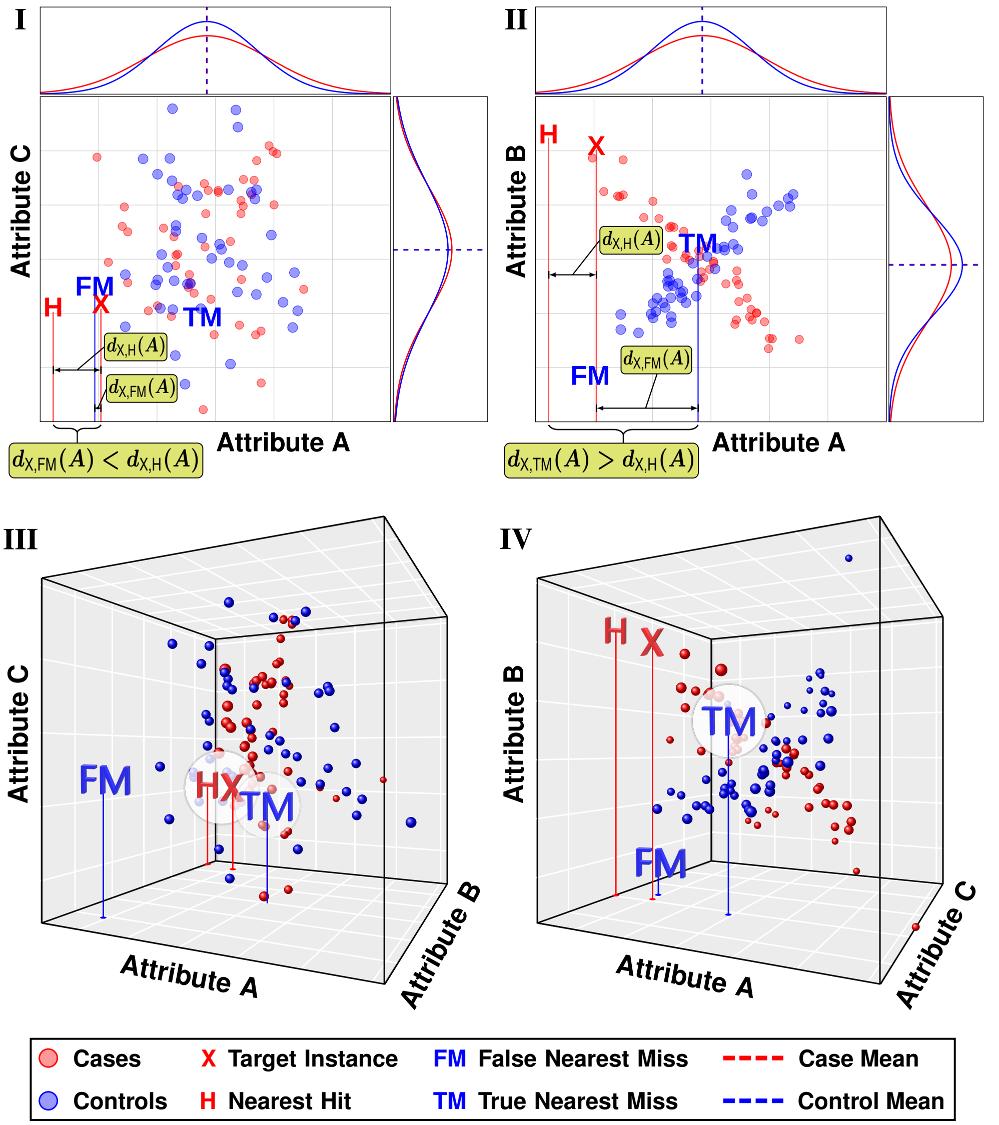

Fig 1. Imposters vs true neighbors in the presence of interactions with three variables. Attributes A, B, and C have no main effect. (**I**) Scatter plot of simulated irrelevant Attribute C with a functional Attribute A. (**II**) Scatter plot of Attributes A and B, which interact through differential correlation. Computing nearest neighbors with irrelevant attributes (**I**) or lower dimensions leads to imposter nearest neighbors and degrades the ability of Relief-based methods to identify interaction effects. Computing distances in only these two dimensions leads to an imposter false miss (FM) for the nearest neighbor from the opposite outcome class for target instance X. This imposter leads to Attribute A predicting closer projected distances for misses than hits (H), which incorrectly indicates that A is a poor discriminator (yellow boxes in **I**). (**III-IV**) Computing nearest neighbors in higher dimensions or with the correct interaction partner leads to imposter nearest neighbor (FM) being replaced by the true nearest miss neighbor (TM) for target instance X, which correctly leads to Attribute A predicting closer projected distances for hits (H) than misses, which is an indication that Attribute A is a good discriminator (yellow boxes **II**).

Relief-based methods use information from all available attributes (omnigenic) to estimate an attribute's importance. However, if relevant higher-dimensional information is not used to establish the neighborhoods of instances, these methods will miss the effect of A because "imposter" neighbors will be used in the attribute estimate (False Miss (FM) in Fig. 1-I, where $d_{\mathrm{X,FM}}(A) < d_{\mathrm{X,H}}(A)$). If one were to compute nearest neighbors in the A-C plane (ignoring the B dimension), the nearest miss would be an imposter (FM), which leads to a negative contribution to the importance score for A. One might call this C attribute a type-I confounding attribute because it increases the chances of interacting attributes to be false negatives. When nearest neighbors are calculated based on higher dimensions with relevant information (Fig. 1-III), it is clear that TM is closer to X than FM. The imposter (FM) is replaced by the true nearest miss (TM) and attribute A correctly shows a greater projected difference between misses than hits (Fig. 1-II $d_{\mathrm{X,TM}}(A) > d_{\mathrm{X,H}}(A)$), which is the signature of an important attribute. Univariate methods still cannot find the importance of A unless the interaction is explicitly modeled, but as long as functional variables A and B are in the space for nearest neighbor calculations (Fig. 1-III - IV), imposters can be excluded and Relief-based methods will find that A (and B) are important discriminators.

Using same interaction, increase background noise genes to see degrading of A and B Relief importance because of curse of dimensionality (sparseness).

1 Neighborhood methods

Nearest-neighbor projected distance (NPD) methods rely on a neighborhood algorithm for feature selection. One may specify a fixed-k number of neighbors, a fixed radius, an average radius SURF, a MultiSURF radius that adapts for each instance [1], or a gene-wise adaptive-k [2]. The definitive difference between fixed-k and either fixed or adaptive radius methods is that there is always a deterministic number of neighbors in fixed-k neighborhoods, but there is a variable neighborhood order in each fixed or adaptive radius method. In nearest-neighbor feature selection, neighborhood order is directly related to selected feature quality. Correlation and interaction effects change the distribution of pairwise distances between instances, which ultimately changes the probability of neighborhood inclusion for fixed or adaptive radius methods. In order to select the best set of features from data, it is important to know how pairwise feature correlation changes the optimal neighborhood order. We approach this problem by first determining the functional relationship between fixed-k and MultiSURF radius in null data with no effects.

2 Derivation of expected k for multiSURF neighborhoods

The MultiSURF radius for an instance is the mean of its distances to all other instances subtracted by $\alpha = 1/2$ of the standard deviation of this mean. Previously we showed empirically for balanced case-control datasets that a good constant-k approximation to the expected number of neighbors within the multiSURF radii is $k = m/6$ [3], where m is the number of samples. Here we derive a more exact theoretical mean that shows the mathematical connection between neighbor-finding methods. This fixed-k approximation to MultiSURF is independent of the type of data and the particular radii of each instance in the data.

The distance between instances ($i, j \in \mathcal{I}$, $|\mathcal{I}| = m$) in the data set $X^{m \times p}$ of m instances and p attributes is calculated in the space of all attributes ($a \in \mathcal{A}$, $|\mathcal{A}| = p$)

using a metric such as

$$D_{ij}^{(q)} = \left(\sum_{a \in \mathcal{A}} |\mathrm{d}_{ij}(a)|^q \right)^{1/q}, \tag{1}$$

which is typically Manhattan ($q = 1$) but may also be Euclidean ($q = 2$). The quantity $\mathrm{d}_{ij}(a)$, known as a "diff" in Relief literature, is the projection of the distance between instances i and j onto the attribute a dimension. The function $\mathrm{d}_{ij}(a)$ supports any type of attributes (e.g., numeric and categorical). For example, the projected difference between two instances i and j for a continuous numeric ($\mathrm{d}^{\mathrm{num}}$) attribute a may be

$$\begin{aligned} \mathrm{d}_{ij}^{\mathrm{num}}(a) &= \mathrm{diff}(a, (i,j)) \\ &= |\hat{X}_{ia} - \hat{X}_{ja}|, \end{aligned} \tag{2}$$

where $\hat{X}$ represents the standardized data matrix X. We use a simplified $\mathrm{d}_{ij}(a)$ notation in place of the $\mathrm{diff}(a, (i,j))$ notation that is customary in Relief-based methods. We omit the division by $\max(a) - \min(a)$ used by Relief to constrain scores to the interval from -1 to 1. As we show in subsequent sections, NPDR scores are standardized regression coefficients with corresponding P values, so any scaling operation at this stage is unnecessary for comparing attribute scores. The numeric $\mathrm{d}_{ij}^{\mathrm{num}}(a)$ projection is simply the absolute difference between row elements i and j of the data matrix $X^{m \times p}$ for the attribute column a.

We define the NPDR neighborhood set $\mathcal{N}$ of ordered pair indices as follows. Instance i is a point in p dimensions, and we designate the topological neighborhood of i as N_i. This neighborhood is a set of other instances trained on the data $X^{m \times p}$ and depends on the type of Relief neighborhood method (e.g., fixed-k or adaptive radius) and the type of metric (e.g., Manhattan or Euclidean). If instance j is in the neighborhood of i ($j \in N_i$), then the ordered pair $(i,j) \in \mathcal{N}$ for the projected-distance regression analysis. The ordered pairs constituting the neighborhood can then be represented as nested sets:

$$\mathcal{N} = \{\{(i,j)\}_{i=1}^{m}\}_{\{j \neq i : j \in N_i\}}. \tag{3}$$

The cardinality of the set $\{j \neq i : j \in N_i\}$ is k_i, the number of nearest neighbors for subject i.

2.1 Predicted number of neighbors in the MultiSURF alpha neighborhood

Regardless of the predictor data type (numeric or categorical), the distribution of the p predictors (uniform, Gaussian, or binomial), or the metric used to compute distances (Manhattan or Euclidean), the $m(m-1)/2$ pairwise distances in the p-dimensional space are well approximated by a normal distribution. An instance j is in the adaptive α-radius neighborhood of i ($j \in N_i^{\alpha}$) under the condition

$$D_{ij} \leq R_i^{\alpha} \implies j \in N_i^{\alpha}, \tag{4}$$

where the threshold radius for instance i is

$$R_i^{\alpha} = \bar{D}_i - \alpha\, \sigma_{\bar{D}_i} \tag{5}$$

and

$$\bar{D}_i = \frac{1}{m-1} \sum_{j \neq i} D_{ij}^{(\cdot)} \tag{6}$$

is the average of instance i's pairwise distances (Eq. 1) with standard deviation $\sigma_{\bar{D}_i}$. MultiSURF implements $\alpha = 1/2$ [1].

The probability of the remaining $m-1$ instances being inside the α-radius of instance i (R_i^α) can be viewed as $m-1$ Bernoulli trials each with a probability of success q_α. Then the average average number of neighbors is given by

$$\bar{k}_\alpha = (m-1)q_\alpha, \tag{7}$$

from the mean of a binomial random variable. To calculate q_α, we assume the distribution of distances $\{D_{ij}\}_{j\neq i}$ of neighbors of instance i is normal $N(\bar{D}_i, \sigma_{\bar{D}_i})$. Our empirical studies confirm a normal distribution and that it is robust to data type and metric. Extreme violations of independence of attributes (extreme correlations or interactions) will cause the distribution to be right skewed, but this effect is difficult to observe in real data. Thus, for a Gaussian pairwise distance distribution, the probability q_α for one instance $j \neq i$ to be in the neighborhood of i ($j \in N_i^\alpha$) is given by the area under the mean-centered ($\bar{D}_i$) Gaussian from $-\infty$ to R_i^α. An illustration of the area computed to estimate q_α is given by Fig. 2. This integral can be written in terms of the error function (erf):

$$q_\alpha = \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{\alpha}{\sqrt{2}}\right)\right). \tag{8}$$

And finally using Eqs. (7 and 8) we find

$$\bar{k}_\alpha = \left\lfloor \frac{m-1}{2}\left(1 - \operatorname{erf}\left(\frac{\alpha}{\sqrt{2}}\right)\right)\right\rfloor, \tag{9}$$

where we apply the floor function to ensure the number of neighbors is integer. For data with balanced hits and misses in standard fixed-k Relief, one divides this formula by 2. For multiSURF ($\alpha = 1/2$), this formula gives $\bar{k}_\alpha^{\text{hit/miss}} = \bar{k}_{1/2}^{\text{hit/miss}} = \frac{1}{2}\bar{k}_{1/2} = .154(m-1)$, which is very close to our previous empirical estimate $m/6$. When we compare multiSURF neighborhood methods with fixed-k neighborhoods, we use $\bar{k}_{1/2}$. Using this $\alpha = 1/2$ value has been shown to give good performance for simulated data sets. However, the best value for α is likely data-specific and may be determined through nested cross-validation and other parameter tuning methods.

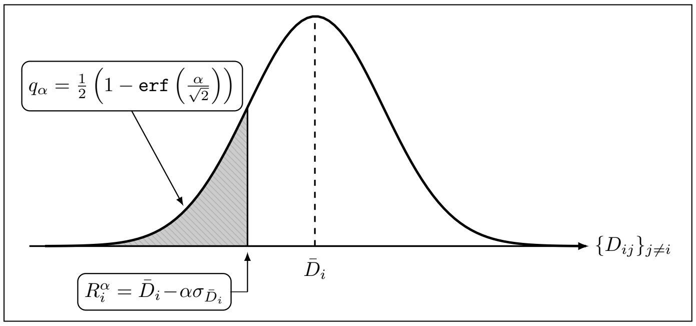
$q_\alpha = \frac{1}{2}\left(1 - \texttt{erf}\left(\frac{\alpha}{\sqrt{2}}\right)\right)$
$\bar{D}_i$
$\{D_{ij}\}_{j\neq i}$
$R_i^\alpha = \bar{D}_i - \alpha\sigma_{\bar{D}_i}$

Fig 2. Illustration of the expected probability of a fixed instance j being in the fixed radius neighborhood of another instance i. The fixed radius is parameterized by a fraction α of the standard deviation of all pairwise distances measured from instance i to all possible neighbors.

3 Optimal neighborhood parameters for detecting effects

k or α. Balancing blessing and curse of dimensionality.

4 ICA?

Using same interaction, increase background noise genes to see degrading of A and B Relief importance because of curse of dimensionality (sparseness).

5 Data simulations

Each simulated data set $X^{(m\times p)}$ contains m instances and p features, where

$$m \in \{100, 250, 500\} \text{ and } \tag{10}$$

$$p \in \{1000, 2500, 5000\}. \tag{11}$$

All combinations of m and p were explored in order to determine how neighborhood selection parameters change with dimensionality. We considered a balanced binary outcome only so that there were exactly $m/2$ cases and $m/2$ controls. In each simulation, 10% of the total number of features p were functionally related to the outcome variable while the remaining 90% were simply background features with no effect. Functional features were given either main or interaction effect in order to create a random mixed effects data set for which optimal neighborhood method parameters could be calculated.

5.1 Interaction effects

We extend the interaction effect simulation introduced in [4]. This method first generates a random graph from either Erdős-Rényi or Scale-free degree distribution. In the control group, functional features are given large pairwise correlations with all other features. Differential correlations (interaction effects) are created by randomly permuting functional feature data entries within the case group only, which destroys and preserves the correlation in case and control groups, respectively. This method creates large effect sizes, which are easily detected by nearest-neighbor distance based methods. The reason for this ease of detection is the uniformity in low and high correlations in case and control groups, respectively. In order to establish more influence over the number of differential pairwise correlations, we simulated correlation matrices for case and control groups directly. We allowed only functional connections to be given differential correlation between case and control groups, where a functional connection is simply the presence of an edge (or link) from one feature to a functional feature in the random network that is generated.

Analogous to the interaction simulation method given in [4], we give the control group high and low positive pairwise correlation between network connected and non-connected features, respectively. The average magnitudes of the high and low correlations in the control group are determined by fixed parameters ρ^{hi} and ρ^{lo}, respectively. For all simulations, we fixed the low correlation parameter ρ^{lo}. In order to examine how interaction effect size changes with functional connection pairwise correlation, we let $\rho^{\text{hi}} \in \{0.2, 0.5, 0.8\}$. Interaction effect size increases and decreases monotonically as ρ^{hi} increases and decreases, respectively. Interaction effect size is highly related to how connected a functional feature is in the random network. We control network connectivity by the probability of edge inclusion for Erdős-Rényi networks and fixed node degree for Scale-free networks. Similar to ρ^{hi}, interaction effect size increases and decreases monotonically as functional network connectivity increases and decreases, respectively. Furthermore, we determine interaction effect sizes by giving functional connection pairwise correlations in the case group values determined by the parameter b^{int}, which is defined as

$$b^{\text{int}} = -t\rho^{\text{hi}} + (1-t)\rho^{\text{hi}}, \text{ where } t \in [0, 1]. \tag{12}$$

As $t \to 0$, the effect size decreases monotonically. On the other hand, the effect size increases monotonically as $t \to 1$. By controlling ρ^{hi}, b^{int}, and the level of network connectivity, we have the ability to more finely control the interaction effect size than the method presented in [4].

After correlation matrices are created for cases and controls, we compute the upper triangular Cholesky factors (U^{case} and U^{ctrl}) for each correlation matrix. We simulate null data matrices X^{case} and X^{ctrl} for cases and controls, respectively, such that

$$x_{ij}^{\text{case}}, x_{ij}^{\text{ctrl}} \sim \mathcal{N}(0,1) \quad \forall i, j. \tag{13}$$

Multiplication of X^{case} and X^{ctrl} by the Cholesky factors U^{case} and U^{ctrl}, respectively, produce case and control sub-matrices Y^{case} and Y^{ctrl} with the correlation structure described previously. These sub-matrices are then combined into a single data $m \times p$ matrix given by

$$X = \begin{bmatrix} Y^{\text{ctrl}} \\ \hdashline Y^{\text{case}} \end{bmatrix}. \tag{14}$$

A diagram outlining the interaction simulation algorithm for a data set consisting of 7 features is shown in Fig. 3. Boxes 1 and 2 display the random network and its characteristics, such as, adjacencies and node degrees. In particular, boxes 1 and 2 show the features that are selected to be functional (highlighted in green). Box 3 shows the case and control correlation matrices generated using input parameters ρ^{hi}, ρ^{lo}, and b^{int}. In the control group, high correlation (red) is assigned to all connected pairs from the network. That is,

$$P_{ij}^{\text{ctrl}} = \rho^{\text{hi}} + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0,1). \tag{15}$$

In both case and control groups, low correlation (red) is assigned to non-connected features from the network. These low correlations are given by

$$P^{\text{case}} = P^{\text{ctrl}} = \rho^{\text{lo}} + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0,1). \tag{16}$$

In the case group, pairwise correlations associated with functional connections (blue) from the network are assigned a correlation that is functionally related (see Eq. 12) to high correlations in the control group. These correlations are given by

$$P_{ij}^{\text{case}} = b^{\text{int}} + \varepsilon_{ij}, \quad \varepsilon_{ij} \in \mathcal{N}(0,1). \tag{17}$$

Other than entries associated with functional connections, the case and control correlation matrices are identical. Box 4 shows the Cholesky decompositions for P^{ctrl} and P^{case}, which are given by

$$\begin{aligned} P^{\text{ctrl}} &= U^{\text{ctrl}} \left(U^{\text{ctrl}}\right)^{\text{T}} \quad \text{and} \\ P^{\text{case}} &= U^{\text{case}} \left(U^{\text{case}}\right)^{\text{T}}. \end{aligned} \tag{18}$$

Random case and control data with correlation structure determined by P^{case} and P^{ctrl}, respectively, are created as mentioned previously (box 5). These sub-matrices are given by

$$\begin{aligned} Y^{\text{ctrl}} &= X^{\text{ctrl}} \left(U^{\text{ctrl}}\right)^{\text{T}}, \quad x_{ij}^{\text{ctrl}} \sim \mathcal{N}(0,1) \quad \text{and} \\ Y^{\text{case}} &= X^{\text{case}} \left(U^{\text{case}}\right)^{\text{T}}, \quad x_{ij}^{\text{case}} \sim \mathcal{N}(0,1). \end{aligned} \tag{19}$$

The full data set, given previously by Eq. 14, concludes the generation of the full $m \times p$ data set X with interaction effects.

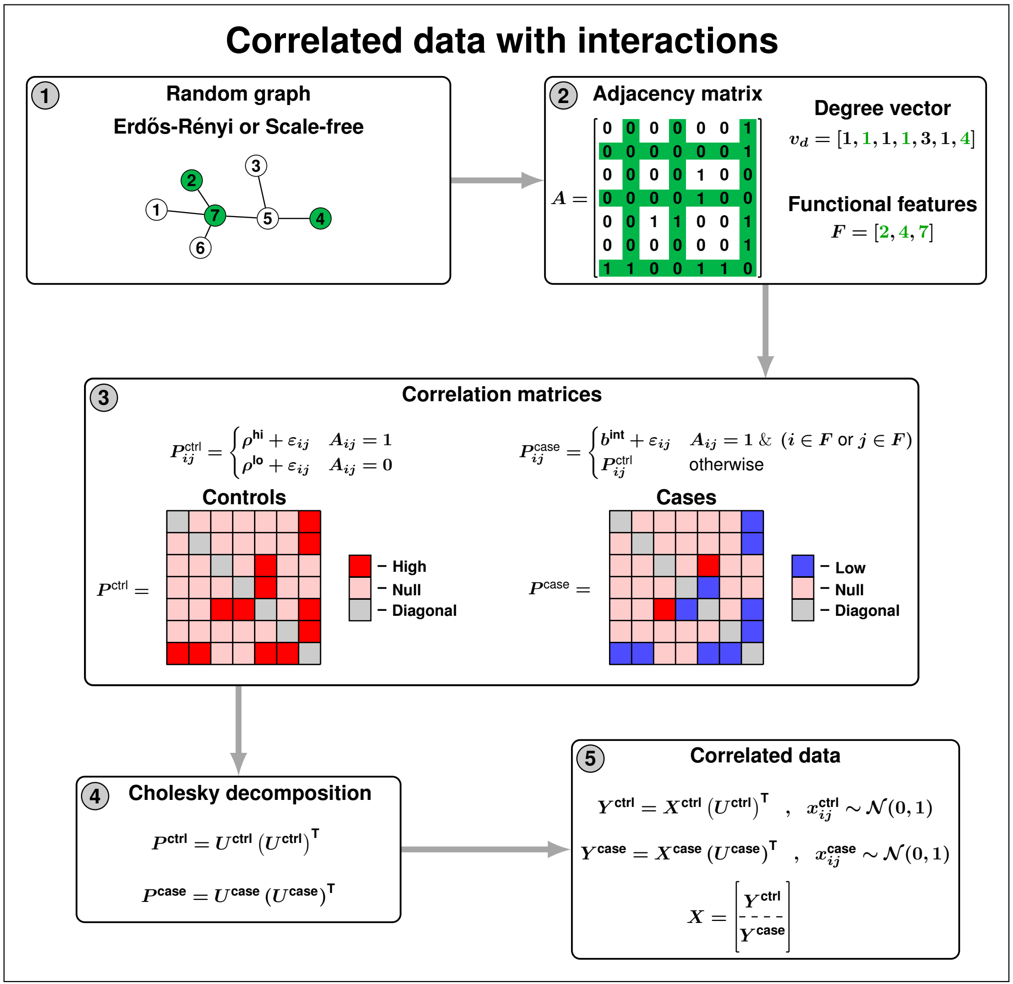
Correlated data with interactions
1 Random graph
Erdős-Rényi or Scale-free
1
2
3
4
5
6
7
2 Adjacency matrix
$A = \begin{bmatrix} 0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&1&1&0&0&1 \\ 0&0&0&0&0&0&1 \\ 1&1&0&0&1&1&0 \end{bmatrix}$
Degree vector
$v_d = [1, 1, 1, 1, 3, 1, 4]$
Functional features
$F = [2, 4, 7]$
3 Correlation matrices
$P^{\text{ctrl}}_{ij} = \begin{cases} \rho^{\text{hi}} + \varepsilon_{ij} & A_{ij} = 1 \\ \rho^{\text{lo}} + \varepsilon_{ij} & A_{ij} = 0 \end{cases}$
$P^{\text{case}}_{ij} = \begin{cases} b^{\text{int}} + \varepsilon_{ij} & A_{ij} = 1 \;\&\; (i \in F \text{ or } j \in F) \\ P^{\text{ctrl}}_{ij} & \text{otherwise} \end{cases}$
Controls
$P^{\text{ctrl}} =$
High
Null
Diagonal
Cases
$P^{\text{case}} =$
Low
Null
Diagonal
4 Cholesky decomposition
$P^{\text{ctrl}} = U^{\text{ctrl}} (U^{\text{ctrl}})^{\mathsf{T}}$
$P^{\text{case}} = U^{\text{case}} (U^{\text{case}})^{\mathsf{T}}$
5 Correlated data
$Y^{\text{ctrl}} = X^{\text{ctrl}} (U^{\text{ctrl}})^{\mathsf{T}} \quad , \quad x^{\text{ctrl}}_{ij} \sim \mathcal{N}(0, 1)$
$Y^{\text{case}} = X^{\text{case}} (U^{\text{case}})^{\mathsf{T}} \quad , \quad x^{\text{case}}_{ij} \sim \mathcal{N}(0, 1)$
$X = \begin{bmatrix} Y^{\text{ctrl}} \\ \hdashline Y^{\text{case}} \end{bmatrix}$

Fig 3. Algorithm for interaction simulations from a random undirected network with seven nodes (or features). **Box 1:** A random network is generated, whose degree distribution is either Erdős-Rényi or Scale-free. **Box 2:** Adjacency matrix (A) and degree vector (v_d) corresponding to the random network are computed and functional features (F) are randomly selected from those with positive degree. **Box 3:** Two correlation matrices are generated for cases and controls. In the control group, high (ρ^{hi}) and low (ρ^{lo}) correlations are assigned to connected ($A_{ij} = 1$) and non-connected ($A_{ij} = 0$) feature pairs, respectively. In the case group, differential correlation (b^{int}) is applied to functional connections. **Box 4:** Upper triangular Cholesky factors are computed for case/control correlation matrices. **Box 5:** Standard normal random data matrices (X^{ctrl} and X^{case}) are given correlation structure associated with case and control groups and combined into full data matrix with interaction effects (X).

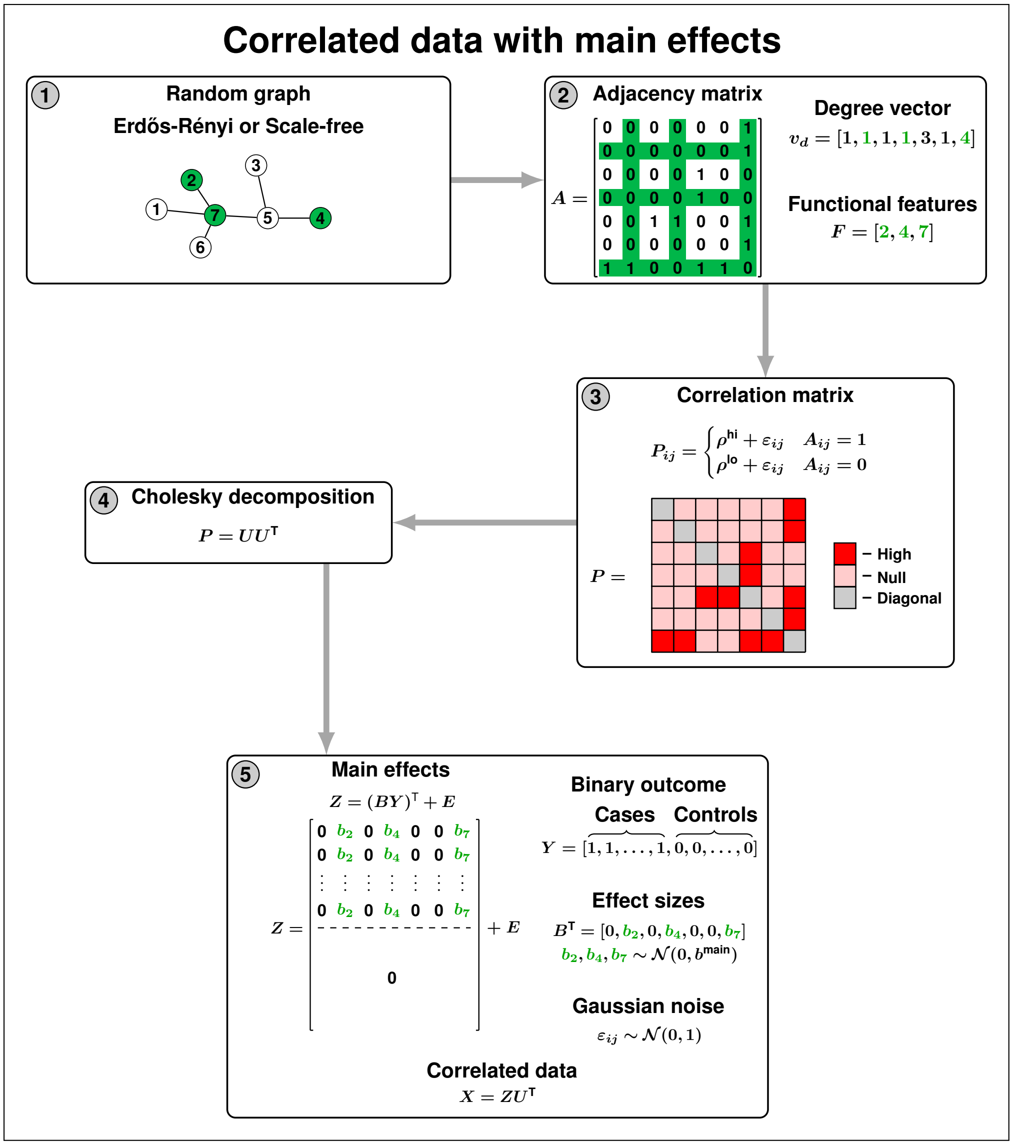
Correlated data with main effects
1
Random graph
Erdős-Rényi or Scale-free
1
2
3
4
5
6
7
2
Adjacency matrix
$A =$
0 0 0 0 0 0 1
0 0 0 0 0 0 1
0 0 0 0 1 0 0
0 0 0 0 1 0 0
0 0 1 1 0 0 1
0 0 0 0 0 0 1
1 1 0 0 1 1 0
Degree vector
$v_d = [1, 1, 1, 1, 3, 1, 4]$
Functional features
$F = [2, 4, 7]$
3
Correlation matrix
$P_{ij} = \begin{cases} \rho^{\text{hi}} + \varepsilon_{ij} & A_{ij} = 1 \\ \rho^{\text{lo}} + \varepsilon_{ij} & A_{ij} = 0 \end{cases}$
$P =$
– High
– Null
– Diagonal
4
Cholesky decomposition
$P = UU^{\mathsf{T}}$
5
Main effects
$Z = (BY)^{\mathsf{T}} + E$
$Z =$
0 b_2 0 b_4 0 0 b_7
0 b_2 0 b_4 0 0 b_7
0 b_2 0 b_4 0 0 b_7
0
$+ E$
Binary outcome
Cases Controls
$Y = [1, 1, \ldots, 1, 0, 0, \ldots, 0]$
Effect sizes
$B^{\mathsf{T}} = [0, b_2, 0, b_4, 0, 0, b_7]$
$b_2, b_4, b_7 \sim \mathcal{N}(0, b^{\text{main}})$
Gaussian noise
$\varepsilon_{ij} \sim \mathcal{N}(0, 1)$
Correlated data
$X = ZU^{\mathsf{T}}$

Fig 4. Algorithm for simulating main effects with correlation from a random undirected network with seven nodes (or features). **Box 1:** A random network is generated, whose degree distribution is either Erdős-Rényi or Scale-free. **Box 2:** Adjacency matrix (A) and degree vector (v_d) corresponding to the random network are computed and functional features (F) are randomly selected from those with positive degree. **Box 3:** Correlation matrix generated with high (ρ^{hi}) and low (ρ^{lo}) correlations assigned to connected ($A_{ij} = 1$) and non-connected ($A_{ij} = 0$) features, respectively. **Box 4:** Upper triangular Cholesky factors are computed for correlation matrix. **Box 5:** Main effects are simulated for functional variables (F) with effect sizes determined by b^{main}. Data is given correlation by multiplying main effects (Z) by Cholesky factor (U).

References

1. Ryan J. Urbanowicz, Randal S. Olson, Peter Schmitt, Melissa Meeker, and Jason H. Moore. Benchmarking relief-based feature selection methods for bioinformatics data mining. *Journal of Biomedical Informatics*, 85:168–188, 2018.

2. Brett A McKinney, Bill C White, Diane E Grill, Peter W Li, Richard B Kennedy, Gregory A Poland, and Ann L Oberg. ReliefSeq: a gene-wise adaptive-K nearest-neighbor feature selection tool for finding gene-gene interactions and main effects in mRNA-Seq gene expression data. *PloS one*, 8(12):e81527, 2013.

3. Trang T Le, Ryan J Urbanowicz, Jason H Moore, and Brett A McKinney. Statistical inference relief (stir) feature selection. *Bioinformatics*, page bty788, 2018.

4. Caleb A Lareau, Bill C White, Ann L Oberg, and Brett A McKinney. Differential co-expression network centrality and machine learning feature selection for identifying susceptibility hubs in networks with scale-free structure. *BioData mining*, 8(1):5, 2015.