PGM Programming Assignment: Bayes Nets for Genetic Inheritance

1 Overview

Because of your success in modeling credit-worthiness, your fame as an expert in graphical models has spread, and now genetic counselors are seeking your assistance. Genetic counselors want your help in advising couples with a family history of a genetic disease. Specifically, they want to help such couples decide whether to have a biological child or to adopt by assessing the probability that their un-born child will have the disease. Through this assignment, you will see how Bayesian networks can be used to determine such probabilities through modeling the mechanism of genetic inheritance.

During this assignment, you will use genetic information to predict the probability that a person will have a physical trait. Many human physical traits, such as freckles, hair color, and many diseases[1] are regulated by one or more proteins. These proteins are coded for by genes, which are sequences of DNA that are found in every cell and passed down from generation to generation. Each gene can have multiple alleles, which are different versions of the gene. For example, a gene involved in hair color might have an allele that makes a person's hair brown and another allele that makes the person's hair red. Genetic inheritance patterns[2] are generally consistent from generation to generation, so **template models** are a natural way to model them. If you do not know much about genetics, you may find the Khan Academy Introduction to Heredity Video helpful `http://www.khanacademy.org/video/introduction-to-heredity?playlist=Biology`. There is also an on-line Appendix, and you will be directed to it during the assignment so that you have the appropriate background to complete the different sections.

Genetic counselors will be giving you pedigrees (family trees), allele frequencies for different alleles, and some information on the effects of having different alleles. You will construct Bayesian networks from this information and use them to determine the probabilities of having different genetic traits. You will then use your network to answer questions that a genetic counselor might encounter. These questions can be found in the **companion quiz**. **Some questions in the quiz require that you have done most of the programming assignment.**

This assignment will cover some facets of Bayesian networks that were not covered in the earlier programming assignment. **You will implement a complete Bayesian network**, writing code to construct the entire network from scratch! You will experience how using templates allows you to copy CPDs. You will also be exposed to some of modeling choices in constructing Bayesian networks and see how adding nodes sometimes allows you to dramatically decrease the number of necessary parameters. In addition, you will get to work with both table and sigmoid CPDs. Don't worry – while the description of this assignment may seem long, *little coding is required*.

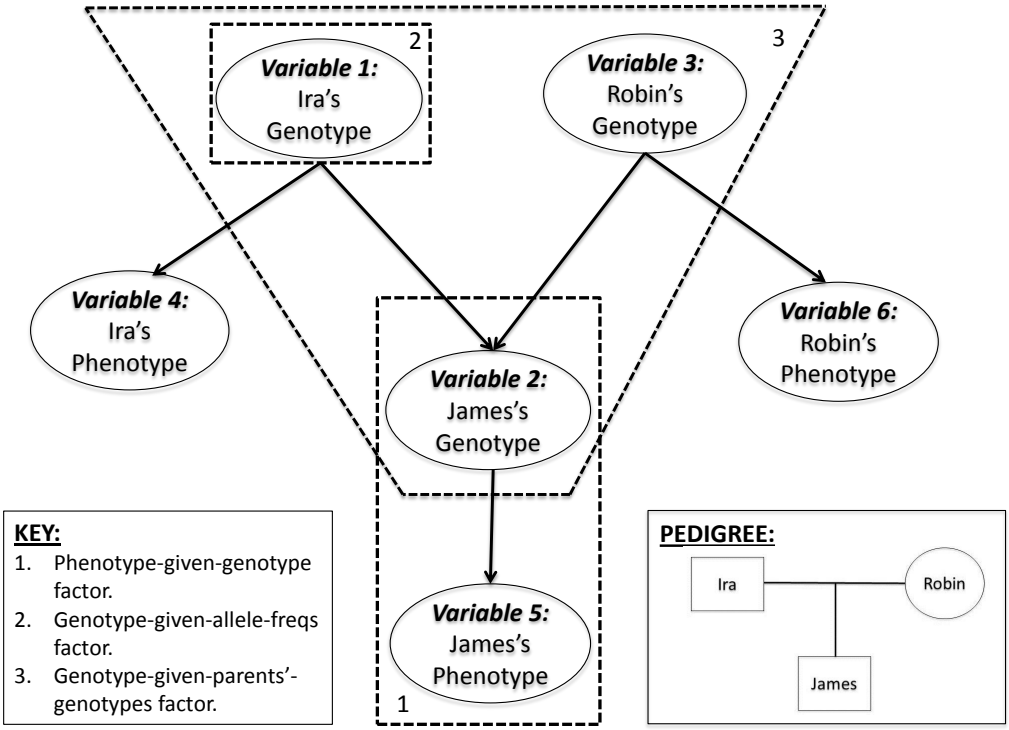
[1] Information on the traits described in this assignment can be found at http://ghr.nlm.nih.gov/condition/.

[2] To learn more about genetic inheritance, look through the on-line Appendix and the links to video lectures on genetics.

2 Constructing Bayesian Networks for Genetic Inheritance

2.1 Modeling Genetic Inheritance Using Bayesian Networks

Suppose a genetic counselor comes to you for assistance and gives you a pedigree. From this pedigree, you can construct a Bayesian network to model the process of genetic inheritance. An example of a pedigree and the corresponding Bayesian network is shown below.



This network for a three person pedigree has six nodes; there are two nodes for each person: one for the person's genotype and one for the person's phenotype. The networks you will construct will have three types of factors, like the network shown above. For the purposes of this assignment, we will assume that every person's phenotype is influenced by only his/her genotype. So, for each person, there is a factor (factor type 1) for P(person's phenotype | person's genotype). Each person's genotype is determined by that person's parents' genotypes, so, for each person, there is a factor (factor type 3) for P(person's genotype | genotype of person's first parent, genotype of person's second parent). The genotype factor for a person whose parents are not specified is just the prior P(person's genotype) (factor type 2), and its values are based on the allele frequencies in the population. You can also see that each variable has been given a number; these numbers will be used to identify the variables in the code. For example, P(James's Phenotype | James's Genotype) is equivalent to P(Variable 5 | Variable 2).

In this assignment, you will construct factors like the ones in this network. You will then construct ground Bayesian networks with these factors as templates. You will also query the networks to see how probabilities of having genotypes and phenotypes change when some of them are observed.

2.2 Constructing Genetic Network Factors

2.2.1 Constructing Phenotype Factors for a Mendelian Model

To start out, you will model simple Mendelian inheritance[3]. An example of a disease that is approximately inherited in this pattern is Tay-Sachs disease. For the purposes of this exercise, assume that there are two alleles for Tay-Sachs disease, dominant allele **T** and recessive allele **t**. Tay-Sachs is a recessive trait, so **t** is the allele that causes the disease. You will now construct a factor for the probability distribution of having a simple Mendelian trait like Tay-Sachs disease given the genotype. Fill in **phenotypeGivenGenotypeMendelianFactor.m**.

- **phenotypeGivenGenotypeMendelianFactor.m (5 points)** – This function computes the probability of each phenotype given the different genotypes for a trait with Mendelian inheritance, meaning that each allele combination causes a specific phenotype with probability 1. For example, in the network in Figure 1, there is a factor for P(JamesPhenotype | JamesGenotype) whose values are the entries of the CPD for JamesPhenotype. As in PA1, factors have the following components:
  - **var** – These are the variables in the factors.
  - **card** – These are the cardinalities (numbers of possible values) of each variable.
  - **val** – These are the values of the factor for each possible assignment of variables to values.

We have also provided the accessor functions Get/SetValueOfAssignment and convenience functions IndexToAssignment/AssignmentToIndex for you to manipulate these factors.

Note that, in this factor and in all other factors, the variable to the left of the conditioning bar must be the first variable in the .var field of the factor struct. For example, if the factor F is P(Variable 5 | Variable 2), then F.var(1) = 5. You may test this and all other functions by following the instructions in **sampleGeneticNetworks.m**. You can submit this function and all other functions in this assignment by running **submit**, and a unit test will be run.

2.2.2 Constructing Phenotype Factors for a Non-Mendelian Model

However, many traits are not inherited in a strictly Mendelian way. For instance, there are genotypes that, instead of causing or preventing someone from having a physical trait, just make someone more or less likely to have a physical trait. You will now construct factors with CPDs for phenotypes for which the inheritance pattern is not Mendelian[4]. We will consider a non-Mendelian inheritance model where each genotype i leads to some probability α_i of having a physical trait. We can apply this model to cystic fibrosis[5], in which the allele **F** is involved in increasing the risk of getting cystic fibrosis. If people who are **FF** (genotype 1) have an 80% chance of having cystic fibrosis, people who are **Ff** (genotype 2) have a 60% chance of having the disease, and people who are **ff** (genotype 3) have a 10% chance of having the disease, then $\alpha_1 = 0.8$, $\alpha_2 = 0.6$, and $\alpha_3 = 0.1$.

[3] See Appendix, Section 2.

[4] See Appendix, Section 3

[5] Note that this is not exactly how the inheritance of cystic fibrosis works. In reality, there are more than 2 alleles for the gene for cystic fibrosis, different alleles lead to different forms fo the disease, and the alpha values are different from those given here. We made the simplifying assumption of there being 2 alleles and 1 form of the disease for the purposes of this assignment.

This inheritance model is more general than the Mendelian inheritance model. For example, for the trait in Figure 1, the CPD entries are not restricted to 0 or 1. In addition, this type of model allows for having more than two alleles per gene. To have a sense of what the Bayesian network for the inheritance of such a gene would look like, imagine the CPDs in Figure 1 being extended to have more columns for additional genotypes. In this more general case, a person has $\binom{n}{2} + n$ possible genotypes, where n is the number of alleles, because there are $\binom{n}{2}$ unique combinations of pairs of alleles with no repeats and n combinations of pairs of alleles in which both alleles are the same. You will now construct a P(person's phenotype | person's genotype) factor for this model. Fill in **phenotypeGivenGenotypeFactor.m**.

- **phenotypeGivenGenotypeFactor.m (5 points)** – This function computes the probability of each phenotype given the different genotypes for a trait. For example, in the generalized version of the picture from Section 2.1, there is a factor for P(JamesPhenotype | JamesGenotype) whose values are the CPD for JamesPhenotype.

2.2.3 Constructing Genotype Factors for Ancestral Nodes

Now that you have constructed factors for the probabilities of having different phenotypes, you will construct factors for the probabilities of having different genotypes. There are two types of factors that you will need to construct. We start with the simpler case – the factors for family members who have no specified parents. The values in these factors are based on allele frequencies in the population[6]. An example of such a factor in Figure 1 is P(IraGenotype). We will assume that the frequencies of different alleles are independent and that each person has an equal probability of having each allele for each copy of a gene. To create these factors, fill in **genotypeGivenAlleleFreqsFactor.m**.

- **genotypeGivenAlleleFreqsFactor.m (5 points)** – This function computes the probability of each genotype given the allele frequencies in the population. You may assume that the alleles assort independently in the population, so that the probability of having a specific genotype is simply the product of the frequencies of its constituent alleles in the population.

2.2.4 Constructing Genotype Factors for Nodes with Parents

You will now consider the more complex case – factors for the probabilities of a child having different genotypes given the parents' genotypes. An example of such a factor is P(JamesGenotype | IraGenotype, RobinGenotype) in Figure 1. To construct this type of factor, fill in **genotypeGivenParentsGenotypesFactor.m**.

- **genotypeGivenParentsGenotypesFactor.m (8 points)** – This function creates a factor in which the values are the probability of each genotype for a child given each possible combination of the parents' genotypes.

2.3 Constructing a Complete Bayesian Network

Having constructed all of the necessary factors, you will build a Bayesian network for the inheritance of an autosomal trait with one gene. You will use this network to study the cystic fibrosis example. To construct the network, fill in **constructGeneticNetwork.m**.

[6] See Appendix, Section 4. Also, note that the allele frequencies given for this assignment are not reflective of true allele frequencies.

- **constructGeneticNetwork.m (25 points)** – This function constructs a Bayesian network for genetic inheritance. The input will be a pedigree and a vector of allele frequencies, and the output will be a struct array of factors. Look in the code for information about variable numbering. The pedigree data structure has the following components:
  - **names** – This is an $n \times 1$ cell array with the names of the n people in the family.
  - **parents** – This is an $n \times 2$ array with the indices of the parents for each person in the family. Each row i has two parents for the person i, where we define person i to be the person whose name is specified in names{i}. The entries are numbers corresponding to the indices in **names** of the parents' name. If no parents are specified for person i, the parent entries in row i are both 0. (For the purposes of this assignment, we will assume that everyone has either both parents or neither parent specified.) For example, let names = {'Ira', 'James', 'Robin'}, meaning that the names of the people in the family are Ira, James, and Robin. If parents(2, 1) = 1 and parents(2, 2) = 3, then Ira and Robin are the parents of James. The same would be true if parents(2, 1) = 3 and parents (2, 2) = 1.

Note that the network is an instance of a template model; many of the factors have the same CPDs. Your method for computing the phenotype factors does not depend on any characteristic of the person whose phenotype it was, so all of the phenotype CPDs are the same. This is also true for your other methods for computing factors. To see a sample output, look at **sampleFactorList.mat** (you can load this by running **sampleGeneticNetwork**).

2.4 Applying Your Bayesian Network to Make Inheritance Predictions

Congratulations! You have constructed a Bayesian network for genetic inheritance from scratch! In practice, a genetic counselor often knows whether certain members of a family have a disease and might even have genetic testing results for some family members. In other words, some of the phenotype and genotype information is often observed (in PA1, we would run ObserveEvidence to account for this information). **sendToSamiamInfo.m** is a script that has the information and calls to the appropriate functions that you need in order to call your code to construct a network and then convert the network into a file that can be viewed in SamIam so that you can query your network. Run the script, open your network in SamIam, and try observing different genotypes and phenotypes to see how different observations affect various probabilities. **Answer companion quiz questions 1-2**. **Note:** When answering questions in the companion quiz that involve networks in SamIam, make sure to use the **hugin** inference method (choose hugin from the drop-down menu in the center).

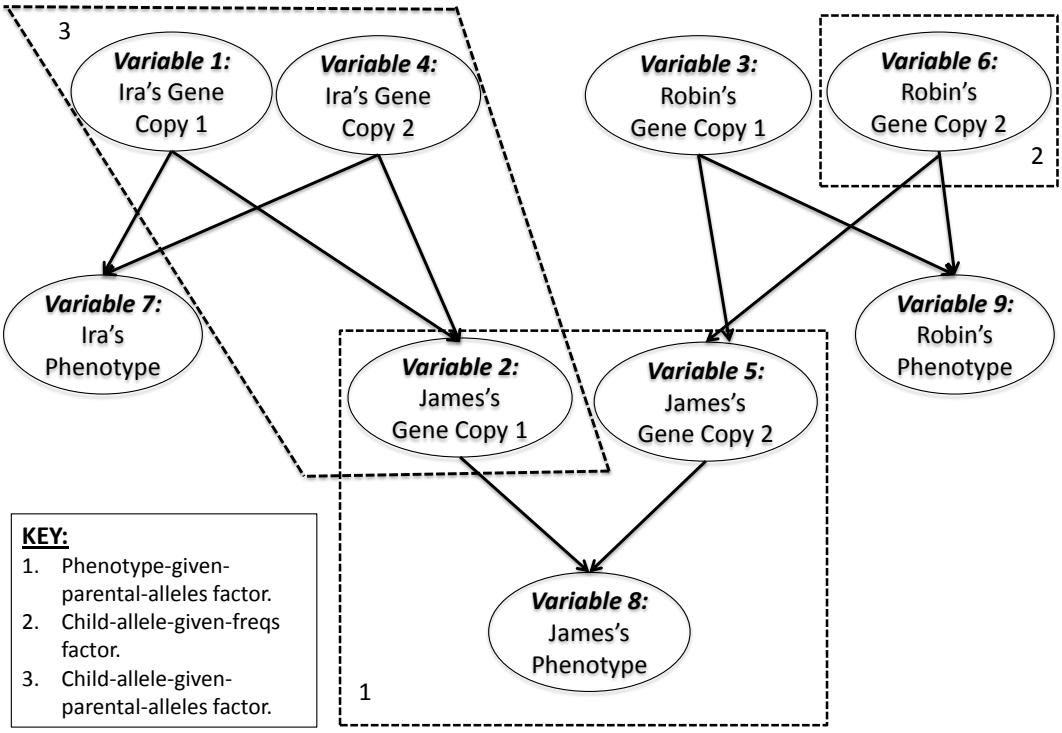
3 Constructing a Decoupled Bayesian Network

3.1 Constructing New Factors for Decoupled Networks

You will now construct an alternate Bayesian network that represents the same genetic inheritance pattern but which is simpler in some ways (and more easily extended to model more complex inheritance patterns). To motivate this, recall that we made the simplifying assumption earlier in the assignment that the gene for cystic fibrosis has only two alleles. In reality, this gene has more than two alleles, so we will now introduce a third allele **n** for this gene into our model. Think about what the Bayesian network is like now that we are accounting for three alleles for the cystic fibrosis gene. Think about how many entries would be in the CPDs. **Answer**

companion quiz question 3. Recall that each person has two copies of each autosomal gene, one from his/her mother, and one from his/her father. In other words, a person will have a copy of a gene that is identical to one of his/her mother's copies of the gene and another copy of the same gene that is identical to one of his/her father's copies of the gene.[7] Think about how you might be able to add some additional nodes and substantially reduce the number of entries in some of the CPDs.

In this alternate network, for each person, the genotype node from the previous network is replaced by a node for each gene copy (one inherited from each parent), so now there are two nodes that together represent the genotype instead of one node in the previous network. Thus, the node for the copy of the gene inherited from the mother is now separate from the node for the copy of the gene inherited from the father. If the parent of the person is not specified, the probabilities for the alleles for the copy of the gene are determined based on allele frequencies, as before. Below is an example of such an alternate network, which we will call a "decoupled" network. (Do you see why this is "decoupled?")



Since the network structure has changed, you need to re-define the factors. This new network has three types of factors. We have provided you with the code for creating the non-phenotype factors. These functions are **childCopyGivenParentalsFactor.m**, which creates a factor whose values are the probabilities of a child inheriting each allele from a parent given each possibility for each copy of the gene (factor type 3), and **childCopyGivenFreqsFactor.m**, which creates a factor whose values are the frequencies of each allele in the population (factor type 2). To create the remaining type of factor, you need to fill in **phenotypeGivenCopiesFactor.m** (factor type 1).

[7] In reality, this is not always the case because mutations and recombination events can occur.

- **phenotypeGivenCopiesFactor.m (5 points)** – This function computes the probability that a child has a phenotype given the alleles for the child's maternal and paternal copies of the gene. An example of such a factor is P(James's Phenotype | James's Gene Copy 1, James's Gene Copy 2) in the figure. In this factor, the values are the entries in the CPD for James's Phenotype.

3.2 Constructing a Decoupled Bayesian Network

Now you are ready to put the factors together to construct the decoupled Bayesian network. You will first use this network to study the modified example of cystic fibrosis with three alleles: **F**, **f**, and **n**. Allele **F** makes cystic fibrosis more likely to occur, allele **f** makes cystic fibrosis somewhat less likely to occur, and allele **n** makes cystic fibrosis very unlikely to occur. (Note that there are new alphas for the genotypes with 3 alleles in **sampleGeneticNetwork.m**.) Fill in **constructDecoupledGeneticNetwork.m**.

- **constructDecoupledGeneticNetwork.m (15 points)** – This function constructs a decoupled Bayesian network for genetic inheritance; the output should be a struct array of factors. Look in the code for information about variable numbering.

To see a sample output, look at **sampleFactorListDecoupled.mat**.

3.3 Applying Your Decoupled Bayesian Network to Make Inheritance Predictions

Congratulations! You have constructed a decoupled Bayesian network for genetic inheritance! This network allows you to answer some additional queries, for example, computing the probabilities of phenotypes if the allele for only one chromosome of a person is known. Run **sendToSamiamInfoDecoupled**, open your network in SamIam, and try observing different gene copies and phenotypes to see how the probabilities change. Look at the CPDs, and compare them to the CPDs that you would get from running **constructGeneticNetwork.m**. **Answer companion quiz questions 4-6**.

4 Constructing a Bayesian Network for Traits Caused by Multiple Genes

4.1 Overview of Sigmoid CPDs

Impressed by your assistance for single-gene traits, the genetic counselor wants your help with a more complex problem: predicting the inheritance of traits controlled by multiple genes. You will now construct a Bayesian network that works for traits that are controlled by multiple genes[8]. For this part of the assignment, you will consider the example of spinal muscular atrophy (SMA). Think about how incorporating multiple genes changes the network.

The genetic counselor thinks that there is an additive effect that depends on the alleles that a person has. Say there are m genes and n_j alleles for each gene j. We define

$$f(X_1^1, \ldots, X_{n_1}^1 \ldots, X_{n_m}^m, Y_1^1, \ldots, Y_{n_m}^m) = \sum_{j=1}^{m} \sum_{i=1}^{n_j} w_i^j (X_i^j + Y_i^j),$$

[8] See Appendix, Section 6.

where each $X_i^j = 1\{\text{person inherits allele } i \text{ for gene } j \text{ from parent 1}\}$ (binary variable that is 1 if the person inherits allele i for gene j from parent 1 and 0 otherwise), each $Y_i^j = 1\{\text{person inherits allele } i \text{ for gene } j \text{ from parent 2}\}$, and each w_i^j is a weight indicating how relevant the allele i for gene j is to the phenotype. Thus, there is a different weight for each allele of each gene, and the alleles that are most involved in causing a person to have a trait have the highest corresponding weights. Note that we are assuming that the weight for the copy of the gene from the mother is the same as the weight for the copy of the gene from the father. The larger the value of $f(X_1^1, \ldots, X_{n_1}^1 \ldots, X_{n_m}^m, Y_1^1, \ldots, Y_{n_m}^m)$, the larger the likelihood that a person has the trait. We need to restrict our likelihood to be a valid probability, so we pass f through a sigmoid function to obtain values that are in the range [0, 1]; the sigmoid function is defined as

$$\text{sigmoid}(z) = \frac{e^z}{1+e^z}.$$

Thus, we define the probability of having a trait controlled by n genes to be

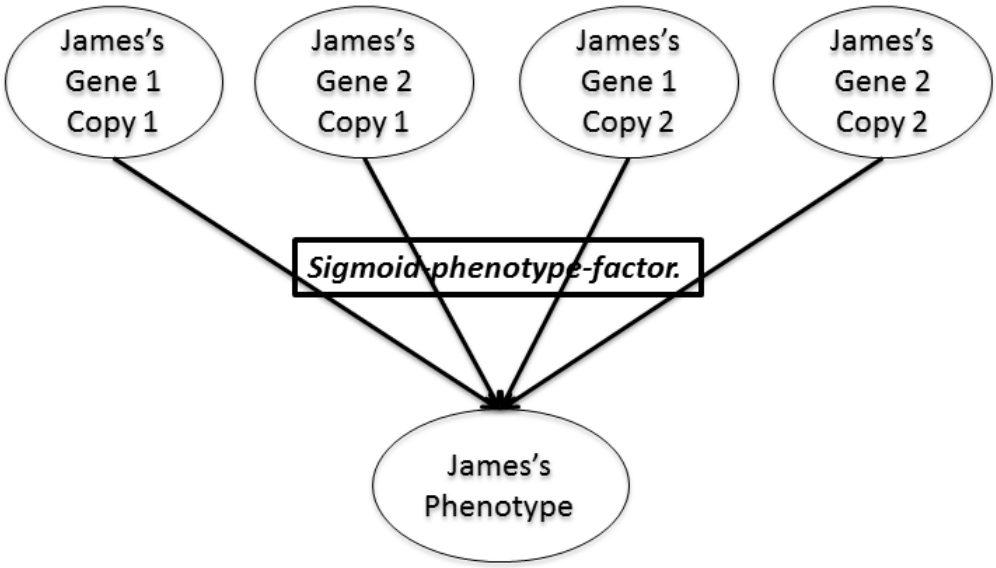
$$\text{sigmoid}(f(X_1^1, \ldots, X_{n_1}^1 \ldots, X_{n_m}^m, Y_1^1, \ldots, Y_{n_m}^m)).$$

Note that the phenotype variable is dependent (through the indicator features X_i^j, Y_i^j) on the gene copy variables, meaning each allele that a person inherits can influence the person's phenotype, just like it can when using table CPDs. As you will see, this allows us to express sigmoid CPDs as table CPDs.

We will now illustrate the concept of a sigmoid CPD with an example. Say there are two genes for SMA. The first gene has alleles A and a, and the second gene has alleles B and b. Let allele A have weight -2, allele a have weight 1, allele B have weight -0.5, and allele b have weight 0.1. This means that the allele most involved in SMA is allele a from the first gene, and the next most involved allele is allele b from the second gene. Allele A from the first gene is most involved in preventing SMA, and allele B from the second gene is the next most involved in preventing SMA. If a person has genotype AabB, then the probability that the person has SMA is $\text{sigmoid}((-2(1+0)+1(0+1))+(-0.5(0+1)+0.1(1+0))) = \text{sigmoid}(-1.4) \approx 0.198$.

4.2 Constructing Factors with Sigmoid CPDs

Now that we understand how we could model genetic inheritance for traits with multiple genes, we will construct factors that represent sigmoid CPDs. In order to do this, we need to be able to compute the value of the sigmoid function. Below is a picture of a part of a factor for a sigmoid CPD for a trait that is controlled by multiple genes.



You will now construct a table factor in which the values are defined by a sigmoid CPD. Fill in **constructSigmoidPhenotypeFactor.m**.

- **constructSigmoidPhenotypeFactor.m (10 points)** – This function constructs a factor for a phenotype variable given the 4 variables for both copies of 2 genes. You will implement the sigmoid CPD using the existing factor data structure; in effect, you will be expressing a sigmoid CPD as an equivalent table CPD. An example of this factor is P(James's Phenotype | James's Gene 1 Copy 1, James's Gene 2 Copy 1, James's Gene 1 Copy 2, James's Gene 2 Copy 2) in the figure above. The values of this factor are the entries in the CPD for JamesPhenotype. **You may assume that there are only 2 phenotypes and only 2 genes. Note:** There are many ways to implement this function, and some allow for the more general case of having more than 2 genes (though you are **not required** to use one of these implementations). Also, **computeSigmoid.m** will be useful for this function.

Answer companion quiz questions 7-9.

4.3 Making Observations in a Bayesian Network for the Inheritance of Multi-Gene Traits

The genetic counselor now wants your help in guiding a family with a history of spinal muscular atrophy, which is controlled by multiple genes. The phenotype you need to consider is the presence or absence of the disease. We have created a network for multi-gene traits and provided it for you in **spinalMuscularAtrophyBayesNet.net**. This network was made with a simpler pedigree than was used for the previous networks so that it would be easier to visualize. Open the network in SamIam. Try observing different phenotypes to see how the probabilities of having different alleles for copies of genes and other people's phenotypes change. **Answer companion quiz questions 10-11**.

5 Conclusion

Congratulatons! You have now constructed Bayesian networks that model multiple genetic inheritance patterns. You have used these networks to help a genetic counselor predict the probabilities that different family members will have a disease given different genotype and phenotype information. These networks are often used in practice[9] and can be extended to model more complex genetic inheritance patterns and incorporate environmental factors. Feel free to extend your Bayesian networks and try out additional queries in your networks. Hope you had fun!

[9] Lauritzen, S.L. and Sheehan, N.A. (2003). Graphical Models for Genetic Analsysis. *Statistical Science* **18** 489-514.