Biologically Plausible Feature Selection Through Relative Correlation

Patrick Connor and Thomas Trappenberg

Abstract—Biological organisms need to accurately infer which features of their environment predict future rewards and punishments for survival sake. This problem resembles linear regression, which finds parameter values expressing the linear relationship between features and an outcome. The least squares regression method generalizes well when there is little system noise and at least as many training data points (experiences) as input features. When this is not the case, feature selection may be applied to eliminate irrelevant features and improve generalization. Here, we show a biologically plausible approach to feature selection that computes the maximum likelihood estimate of Pearl’s “Noisy OR” model. We show that this results in highlighting the features that are most correlated with the outcome at the expense of the least correlated. We extend this “relative correlation” approach to represent global inhibitory features and show that as additive noise and the number of irrelevant features are increased, relative correlation leads to substantially less prediction error on test data than least means squares in a simple linear regression task. We demonstrate how relative correlation can be implemented in a dual pathway neural network and discuss some similarities between it and the basal ganglia.

I. INTRODUCTION

Biological systems have the problem of discerning good predictors of important real-world outcomes. When there are lots of potential predictors (i.e., a high-dimensional space) and little experience, this becomes a challenging task. A good starting point would be to use a model of low complexity – a simple linear model. Linear regression determines the linear relationship between input features and an outcome and accounts for a broad spectrum of real-world relationships. One limitation, however, is that linear regression tends to effectively reflect the training data but is less able to generalize to test data when the training data is noisy or the features outnumber the data points.

Common ways of overcoming this include substantially increasing the number of training data points or reducing the number of input features by selecting only the most relevant ones. In some circumstances, acquiring additional data is costly or not feasible. Automatic feature selection is a broad and useful area of research [1], [2], but has been less well studied in a biological context. One common approach is to compute the mutual information between each feature and the outcome, choosing the features with the highest values. The simplest measure of mutual information is the Pearson correlation coefficient, which seems biologically plausible. One problem with using this approach, however, is that it is data hungry. We will show that with few training data points being a constraint (i.e., little experience in a real-world environment), there is a good possibility that the computed correlation of some relevant features falls comfortably in the same range as truly irrelevant features and therefore might be missed.

Linear regression answers the question, “How can we best fit an n-dimensional hyperplane to a set of data?” Perhaps the better question is, “What subset of those n-dimensions can most fully explain the data?” By favouring explanations that use fewer features, an approach is more likely to generalize well, not being influenced by “unnecessary” (and probably irrelevant) features. Pearl [3] described a model that computes the probability of an outcome from a set of binary input features. His “Noisy OR” model maintains an individual probability for each feature and the total probability of an outcome occurring becomes the probabilistic union of all the present features’ probabilities. In the present work, we adapt this model to linear regression, albeit with several limitations. Using maximum likelihood estimation to infer the individual probabilities, this model appears to naturally accentuate the most highly correlated features and substantially reduce the influence of the least correlated features. Here, we also extend the Noisy OR model to include global inhibitory features, those which negatively influence a prediction. The Dual Noisy OR model achieves effective generalization in the presence of significant system noise and many irrelevant features. The final result is lower prediction error than standard linear regression under these circumstances.

Could such an approach be at work in the brain? Although the Noisy OR and Dual Noisy OR models are effective, they are designed for binary inputs and outputs, whereas the brain is probably not likewise constrained. We propose a neural network model that incorporates the principle of relative correlation by implementing the effects of inhibition differently than in linear regression, taking a major cue from the probabilistic models. Our neural network model shares some similarities with the learning rules and structure of the basal ganglia, which is commonly believed to be involved in reinforcement learning, that is, learning about rewards and punishments in the world.

II. BACKGROUND

In linear regression, we assume that the training data is generated according to,

\[ y = \phi^T x + \eta \]  

(1)
where \( x \) is an input vector, \( x_0 = 1 \) is the “always-on” input associated with the bias, \( y \) is the output, and \( \eta \) is an additive noise. Linear regression infers the parameter values, \( \phi \), from training data, which can be accomplished through maximum likelihood estimation. We define the probability that a data point is generated by a Gaussian random variable with a linear mean (i.e., Equation 1) as follows

\[
p(y|x; \phi) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \phi^T x)^2}{2\sigma^2}\right) \tag{2}
\]

where \( \sigma \) is the standard deviation of the Gaussian random variable. The probability or likelihood that a training set is generated from this distribution becomes

\[
L(\phi) = p(y^{(1)}, \ldots, y^{(m)}|x^{(1)}, \ldots, x^{(m)}; \phi) = \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}; \phi) \tag{3}
\]

where \( m \) is the number of training data points. Our goal is to maximize this likelihood and thereby find the n-dimensional linear function (where \( n \) is the number of features in \( X \)) that most likely generated the training data. We can maximize this convex function by taking its log and ascending its gradient,

\[
\log L(\phi) = \log \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}; \phi) = \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)}; \phi)
\]

\[
= -\frac{m}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y^{(i)} - \phi^T x^{(i)})^2 \tag{4}
\]

Taking the gradient of this function with respect to each \( \phi \) gives

\[
\frac{\partial \log L(\phi)}{\partial \phi_{j}} = -\frac{m}{\sigma^2} \sum_{i=1}^{m} (y^{(i)} - \phi^T x^{(i)}) x_{j}^{(i)} \tag{5}
\]

The gradient can be ascended by iteratively updating the model parameters,

\[
\phi_{j} = \phi_{j} + \alpha \frac{\partial \log L(\phi)}{\partial \phi_{j}} \tag{6}
\]

where the learning rate, \( \alpha = \frac{1}{n+2} \) absorbs the \( \frac{1}{\sigma^2} \) factor in Equation 5 and is optimal [4] for Gaussian inputs that have a zero mean and a variance of one. The true linear equation is parameterized by a vector, \( \psi \), each value falling between -0.5 and 0.5.

In Figure 1 we fix the number of \( \psi \) values at 50 (49 inputs plus the bias) and vary the variance of the additive noise between 0 and 0.25. For each setting in this and all other simulations, we repeat the experiment 25 times to provide the mean and standard deviation statistics displayed in the figures (error regions indicate one standard deviation unless otherwise specified). We compare the prediction error for the training data with the prediction error for a test set of (50) data points. A third curve labeled as “Diff from mean Y” serves as a simple baseline. This “model” always predicts that the output is the mean of the training data outputs \( y \). The curve represents the mean difference between the actual outputs of the test data and the mean output of the training data. Note that in the test set no Gaussian noise is added to more clearly show the differences in the models’ performance. In this scenario, we see that prediction error increases far more rapidly with increasing additive noise for the test data than for the training data.

In Figure 2 we fix the noise at zero and vary the number of inputs between 2 and 100 while maintaining a training set with only 50 data points. With zero noise, all data points will be exactly aligned and finding the line of best fit using the LMS algorithm will solve this linear system of equations exactly, when there are as many data points as unknowns. At 49 input features this threshold is crossed and the prediction error in the test set begins to climb steeply, heading toward the asymptote of our simple baseline. However, the training set error remains low. The lower prediction error for the training data seen in both simulations is known as overfitting, though this may seem unusual at first for linear regression. Regression fits a line to the training data, which is ideal when there is no additive noise and as many data points as parameters. When either of these cases is violated, the training data does not perfectly or completely represent the distribution from which the test set is derived. Thus, the line of best fit is crooked to some degree relative to the true linear model which generates the data.

An important aspect of these two simulations is that only three values of \( \psi \) (associated with the bias and two input features) were randomly chosen (uniform distribution between -0.5 and 0.5), while the other values of \( \psi \) were set to zero. Thus only the inputs associated with two variables actually affect the output. In principle, then, it would require very few data points to learn the true parameter values for these relevant inputs if the irrelevant features could be confidently detected (and dropped) somehow. As a result, many feature selection or feature selection methods have been devised. In a common and relatively plausible approach, relevant features are selected based on how highly correlated they are with the output. This can be done using the Pearson product-moment correlation coefficient or rather the sample
As the variance of the additive Gaussian noise is increased, LMS prediction error in the training and test sets increases. The test set’s prediction error quickly rises above a simple baseline labeled “Diff from mean Y”, which represents the average absolute difference between the test data outputs and the mean of training data outputs.

The correlation coefficient for computing directly from data,

$$r_{xj} = \frac{\sum_{i=1}^{m} (x_j^{(i)} - \mu_x)(y^{(i)} - \mu_y)}{\sqrt{\sum_{i=1}^{m} (x_j^{(i)} - \mu_x)^2 \sum_{i=1}^{m} (y^{(i)} - \mu_y)^2}}$$  \hspace{1cm} (7)

When this coefficient gives +1, the feature perfectly correlates with the result. A value of −1 represents a perfectly anti-correlated feature and a 0 value represents an uncorrelated feature. This approach identifies only linear features, although such model-independent feature selection can be generalized to higher order feature combinations (i.e., mutual information [5]).

As we will show, one problem with feature selection by individual feature correlations is that some features may get overlooked or be drowned out by additive noise, especially with small datasets. It would seem best to find a way to maintain the full set of features and determine their relative relevance for prediction, highlighting the most relevant features and setting the linear model’s parameters near zero for irrelevant features. As mentioned, using maximum likelihood estimation for the exact model that generated the data is the optimal way to discover its underlying parameters. In the event, however, that some of the input features are irrelevant, a model that is able to exploit this would be better equipped.

III. THE NOISY OR MODEL (PEARL, 1988)

For the moment, let us consider the case where we are given binary inputs representing features of the world (1 for present, 0 for absent) and binary outputs representing the occurrence or not of an outcome. This is the input and output arrangement used by the Noisy OR model [3] shown in Figure 3. The Noisy OR model, like LMS, makes predictions based on parameters inferred from data. The important difference is that its prediction is probabilistic, that is, it considers that an outcome may or may not occur, rather than always expecting an outcome with some value.

The Noisy OR model is intended to capture the probability of an outcome occurring ($y = 1$), where an individual probability indicates its feature’s contribution to this likelihood. To get the probability of the outcome occurring, we simply union the individual probabilities of the present features. Using an equivalent but slightly different formulation of the Noisy OR model than Pearl [3], we write the probability or likelihood of an outcome occurring as

$$P(y = 1|x; \phi) = 1 - \prod_j \phi_j^{x_j}$$  \hspace{1cm} (8)

where $x$ is a binary input vector and $\phi$ is here a vector of probabilities (each represents 1 minus the probability that the associated input will be followed by an outcome). The probability of getting no outcome ($y = 0$) becomes $1 - P(y = 1|x, \phi)$. Thus the probability that a certain output ($y$) occurs given an input ($x$) is written as

$$P(y|x; \phi) = (1 - \prod_j \phi_j^{x_j})^y (\prod_j \phi_j^{x_j})^{1-y}$$  \hspace{1cm} (9)

As in linear regression, the main issue is that we do not know the individual probabilities a priori and therefore
must determine them. For this, we will also use maximum likelihood estimation. Given a set of training data, we would like to find the individual probabilities that maximize the likelihood of the Noisy OR model generating that training data or

\[
L(\phi) = \prod_{i=1}^{m} (1 - \prod_{j} \phi_{j}^{(i)})^{y_{i}} (\prod_{j} \phi_{j}^{(i)})^{1-y_{i}}
\]  

(10)

As was done for the linear regression case, we could take the derivative of this convex function and perform gradient descent. However, for computational interests, we have chosen to employ a simple numerical gradient, where we compute the change in likelihood for a minuscule change in each probability. These changes in likelihood are normalized by the sum total of proposed changes in likelihood and then used to update the individual feature probabilities, \( \phi_{j} \).

Although, the model was originally designed for representing the probabilities of outcomes, it can also be used for linear regression to a limited degree, where \( P(y = 1|x; \phi) \) becomes the output prediction. One minor required change is that an “always-on” input be provided for setting up the linear function bias term. The model’s binary output can simply be relaxed to include positive real values without any reformulation. Negative output values are inappropriate since the output represents a probability. Because the model’s binary input represents the presence or absence of a feature, negative values do not have a natural interpretation and are thus not permitted. Using real-valued inputs can also be problematic, so we will continue to use binary inputs in our regression simulations. Even with these concessions, however, there is reason to expect it will perform worse than LMS under ideal conditions because the Noisy OR model is not linear. Instead of a weighted sum, the Noisy OR model provides a union of probabilities, which subtracts out the “common area” covered by both inputs. The larger the individual probabilities, the worse this non-linearity is.

One fundamental problem remains. The Noisy OR model has no way of representing negative linear parameters since values of \( 1 - \phi \) principally represent probabilities. In the following section, we propose a solution.

IV. THE DUAL NOISY OR MODEL

As shown in Figure 4, our proposed Dual Noisy OR model extends the Noisy OR model to represent the negative parameters of a linear function (i.e., global inhibitory influences) by having both a “positive” Noisy OR model that represents the probability that an outcome will occur (\( P_{+} \)) and a “negative” Noisy OR model that represents the probability that the outcome will be inhibited or canceled (\( P_{-} \)). The probability of the outcome occurring then becomes the probability of the outcome multiplied by the probability of it not being canceled or,

\[
P(y|x; \phi) = P_{+} (1 - P_{-})
\]  

(11)

where \( P_{+} \) and \( P_{-} \) are expressed by Equation 8, where each model has its own distinct set of \( \phi \) values. The associated likelihood function can be drawn from this, just as was done for Equation 8. The likelihood becomes

\[
L(\phi) = \prod_{i=1}^{m} ((1 - \prod_{j} \phi_{+j}^{(i)}) \prod_{j} \phi_{-j}^{(i)})^{y_{i}}
\]

\[
(1 - (1 - \prod_{j} \phi_{+j}^{(i)}) \prod_{j} \phi_{-j}^{(i)})^{1-y_{i}}
\]  

(12)

and the same numerical derivative approach is used to maximize the likelihood as is done for the Noisy OR model.

![Fig. 4. The Dual Noisy OR model is an extension of the Noisy OR model that is able to incorporate global inhibitory influences and thereby represent the negative parameters of a linear function.](image)

V. COMPARING LMS TO THE DUAL NOISY OR MODEL

In the real world, where we try to predict the occurrence of rewards and punishments, a positive parameter in a linear regression model can represent a feature of the environment that predicts such an outcome. A negative parameter, then, can be seen as representing a feature of the environment that partially cancels an otherwise positive prediction. For example, the ringing of a bell may come to signify that a food reward will shortly be given. However, the ringing of a bell and the flashing of a light together can come to signify that no food will follow, where the flashing light serves to cancel the otherwise predicted outcome. Importantly, the flashing light in this example does not signal a negatively valued outcome regardless of how potent it may be, but rather the absence of a positive outcome. Embodying this type of scenario in terms of linear regression would lead to setting negative outputs in the training and test data to zero. Naturally, this truncation would introduce some degree of noise, which makes prediction more difficult. On the other hand, the exact prediction of negative values would be replaced by the easier task of predicting an output of zero instead.

For the comparisons provided here, we will limit ourselves to binary input values in all models. For the Dual Noisy OR model, we also zero any negative output values (in both the training and test sets), since this model cannot express negative outcomes. We keep the same learning rate for LMS as before. Although this is no longer optimal, the training error in the variable number of features case remains extremely low.

We run LMS in two configurations: one that
Fig. 5. Test set prediction error of LMS (with and without negative output values) and the Dual Noisy OR model as the variance of the additive output noise is varied and the number of input features remains fixed at 49. LMS performs poorly in the presence of noise, while the Dual Noisy OR model manages it effectively. Dual Pathway Regression (DPR), which is discussed later, performs comparably with the Dual Noisy OR model from which it is derived.

has both positive and negative output values and another that has only positive output values. In these tests, there will always be two relevant features. In an attempt to encourage positive output values, the relevant features may be either both positive or one positive and one negative, and each scenario will be equally likely. The bias will be set to zero, mimicking the fact that rarely does a motivational outcome occur without some predictive feature attached. There will always be 50 data points in each of the training and test sets.

Figure 5 compares the test set prediction error of LMS and the Dual Noisy OR model as the variance of the additive output noise is varied and the number of input features remains fixed at 49. LMS quickly crosses the simple baseline regardless of whether or not it sets all negative outputs to zero, as is done for the Dual Noisy OR model. The Dual Noisy OR model appears to have a relatively linear prediction error growth as the additive Gaussian noise increases, but stays below the baseline. Figure 6 compares the test set prediction error of LMS and the Dual Noisy OR model as the number of irrelevant features is varied but the additive output noise is fixed at zero. Here, the Dual Noisy OR model maintains a relatively flat mean prediction error curve. The error is extremely low for LMS (with negative outputs permitted) while the number of input features is less than the number of data points but climbs beyond this point. Dual Pathway Regression (DPR), which is discussed later, performs comparably with the Dual Noisy OR model from which it is derived.

VI. INDIVIDUAL VERSUS RELATIVE CORRELATION

The common use of the sample correlation coefficient to perform feature selection can be problematic. To illustrate this, we again simulate a linear system with two relevant features, training with 50 data points and zero additive noise. Figure 7 shows the correlation (computed from Equation 7) between two relevant features and the output as the ratio of their true parameter values is varied. We fix the first parameter value at 0.5 and vary the second parameter between 0.0 and 0.5 making the ratio span between 0 and 1. As the ratio descends to 0.5, the less correlated relevant feature’s range begins to overlap with the range of the average irrelevant feature, given that this range encompasses two standard deviations (25 different datasets). This means that in a data set generated under these ideal circumstances (no noise and at least as many data points as model parameters), it is possible that some relevant features will masquerade as an irrelevant ones. Here, the number of irrelevant features, which do not affect the output, does not affect these correlation values. Increasing the number of

In these simulations, the binary input vectors are randomly generated and so the presence/absence of relevant features are not correlated in any way. If features were correlated, we would expect to see them share the weight of the prediction. If two features were perfectly correlated (always present/absent at the same time) with one another and correlated with the outcome 90% of the time, they would equally share the probability of predicting the outcome, each having \( \phi \) values of around 0.684 (since 0.684 \(*\ 0.684 = 0.900\).)

In linear regression, maximum likelihood estimation is performed on a probability density function so that the likelihood increases as the prediction error for the training set decreases. The reason behind the Dual Noisy OR model’s success is that instead of maximizing the likelihood of generating the data by minimizing error, the Noisy OR and Dual Noisy OR models appear to do so by maximizing the relative correlation between input features and the outcome.
data points will improve the separation between relevant and irrelevant feature correlations. However, additive noise and increasing the number of relevant features tends to decrease the separation between relevant and irrelevant features as well. In short, the correlation technique’s effectiveness is limited to conditions of low noise and substantially more data points than the number of relevant features.

Unlike this correlation measure, the Noisy OR and Dual Noisy OR models compute a relative correlation. In the Noisy OR model, \( P(y|x; \phi) \) increases by increasing values of \( 1 - \phi_j \) for \( x_j = 1 \) when \( y = 1 \), that is, by increasing the influence of features that correlate with the output. \( P(y|x; \phi) \) also increases by decreasing values \( 1 - \phi_j \) for \( x_j = 1 \) when \( y = 0 \), that is, by decreasing the influence of the lesser correlated and uncorrelated features. Thus, maximizing \( L(\phi) \) in the Noisy OR model maximizes the likelihood of the input generating the output by maximizing the influence of the features that correlate most with the output and by minimizing the influence of features that least correlate with the output. In some sense, it conducts a race between features to explain the data, and the more consistently correlated features win and suppress the much less correlated features. Imagine a combination of present (\( x=1 \)) features, some which are highly correlated with the output and some which are much less correlated. With enough training epochs, the parameters \( (1 - \phi_j) \) associated with this combination grow to accurately predict the outcome (i.e., there is no error). However, in subsequent training trials, the less correlated features’ parameters will be partly extinguished when they are presented but no outcome occurs. When the original combination is presented again in the next epoch, the previous reduction in the less correlated features’ parameters makes room again for the original combination’s parameters to grow. Since the highly correlated features are extinguished less often than the less correlated features, the highly correlated features’ parameters slowly grow from the continual weight loss of the less correlated features. In the end, the highly correlated features become accentuated relative to the less correlated features, thus computing a relative correlation.

To be more concrete, it helps to demonstrate this process by using LMS but hard-limiting parameters at zero to prevent any parameters from becoming negative. In doing so, we begin to approximate the Noisy OR model, whose individual probabilities are only positive. In the same way as the Noisy OR model, features that co-occur with the outcome most frequently will gain parameter strength and eventually steal away strength from the features which co-occur less frequently with the outcome (and more frequently with no outcome, \( y = 0 \)).

In Figure 8, we replicate the simulation in Figure 6 except that the relevant features’ true linear parameters are always set to be positive to suit the models under test. Here we see that the hard-limited LMS and the Noisy OR model give steady test set prediction errors throughout, both of which fare better than normal LMS. The hard-limited LMS actually performs the best, maintaining extremely low prediction error throughout, which makes sense given that it is a linear model whereas the Noisy OR model is not. Figure 9 compares LMS parameters when hard-limiting is used and when it is not. This shows that features least correlated with the output (i.e., the irrelevant features) are extinguished in the hard-limiting case. Thus stripping LMS of its negative parameters can perform this relative correlation computation when the true linear parameters are always positive. An initial thought may be that to detect the negatively correlated features, one might simply reuse this approach but with \( 1 - y \) as the output. However, there are two reasons why a certain feature may encourage a low output value: it may either be inhibitory or irrelevant. The simplist approach would not make this distinction. We now turn to modifying LMS in a way that allows it to model negative linear parameters and yet employ relative correlation.

VII. RECONFIGURING LMS TO PERFORM RELATIVE CORRELATION

There are a few reasons why it seems appropriate to develop an LMS-based computation of relative correlation. First, it has previously been related to the single neuron (i.e., the Perceptron). Secondly, the probabilistic models are limited to using binary inputs. Finally, the process may help us better understand what makes the probabilistic approach generalize better.

From the last section, we saw that by eliminating inhibitory parameters, relative correlation was achieved using LMS. Thus we propose that the mechanism behind overcoming overfiting here is to change the way in which inhibition is integrated. In principle, there are at least two ways in which this can be done. One is to sum negatively weighted inputs with positively weighted inputs (i.e., LMS). The other, put forward by the Dual Noisy OR model, is to multiply...
the positive expectation of the outcome by some number between 0 and 1, where more inhibitory influence reduces this number. If we are to integrate this into LMS, it will involve creating a positive and negative model as in the Dual Noisy OR model. Formally, we write this as

$$y = 2\phi^T x (1 - 1 + e^{-\phi^T x})$$

(13)

where $\phi_+$ and $\phi_-$ are the positive model and negative model parameters respectively (though both always have positive values), which each receive the same set of inputs $x$. The sigmoid function ensures the range of values for the inhibitory influence is contained between 0 and 1, where the slope is 1 when all parameters are 0. The parameter values can be trained using gradient descent, as in the other methods evaluated here. This is the natural outcome of performing maximum likelihood estimation on a Gaussian random variable with Equation 13 at its mean. The gradient of a parameter depends on its associated pathway. For the positive pathway parameters,

$$\frac{\partial y}{\partial \phi_{+,j}} = 2x_j \frac{1}{1 + e^{\phi_j^T x}}$$

(14)

and the negative pathway parameters,

$$\frac{\partial y}{\partial \phi_{-,j}} = -2x_j e^{\phi_j^T x} \frac{e^{\phi_j^T x}}{(1 + e^{\phi_j^T x})^2}$$

(15)

where a positive or negative parameter is updated by

$$\phi_j := \phi_j - 2y_0 \frac{\partial y}{\partial \phi_j}$$

(16)

A few interesting properties can be seen in these learning rules. For one, the pathways learn in opposing directions. Also, because initial parameters are set very near zero, initial parameter changes are almost identical to LMS for the positive pathway. An important difference is that the negative pathway parameters are updated in proportion to the existing positive prediction (which will correlate with prediction error). This is how the approach distinguishes between inhibitory and irrelevant features: inhibitory features tend to reduce existing positive predictions whereas irrelevant features do not.

Figures 5 and 6 demonstrate the effectiveness of this approach, which we call Dual Pathway Regression. When it is tested in these tasks, it gives results comparable to the Dual Noisy OR model. Thus, this reconfigured perceptron shows that relative correlation for even inhibitory features can be implemented in terms of an LMS-like approach. It also shows that the Dual Noisy OR model’s secret to improving generalization is largely because it models global inhibitory features differently from LMS.

VIII. DISCUSSION

The dual pathway nature of the Dual Noisy OR model and Dual Pathway Regression resembles the dual pathways in the basal ganglia brain region. Within this structure there is a “direct pathway” that facilitates (the positive pathway) actions and an “indirect” pathway that inhibits them (the negative pathway), both pathways converging in the output nuclei of the region. Learning rules that invoke opposite signs for these pathways have been derived from neuroscientific studies of cortico-striatal synaptic plasticity [6], [7]. It also appears that the basal ganglia (more specifically, the substantia nigra pars compacta) generates a signal that encodes a prediction error-like signal [8], which influences learning in these pathways. Having two opposing pathways that simply perform oppositely and sum their effects in the output is not necessary, since a single pathway could accomplish this (given a learning rate boost to compensate for not having a second pathway). The present work suggests that the second
pathway may be specifically designed to represent inhibitory features, as occurs in the Dual Noisy OR and Dual Pathway Regression models. The indirect pathway of the basal ganglia makes a stop at an intermediate nucleus (the globus pallidus externa or GPe). A feature of the direct (positive) pathway is that it sends collateral connections to the GPe as well [9], for which there is no clear interpretation as yet (see [10] for one possibility). If we expand Equation 11, we get 

$$P(y|x; \phi) = P_+ - P_+ P_-.$$ 

If the collateral connections to the GPe have a multiplicative effect, they would appear to implement this equation.

There is evidence of relative correlation in classical conditioning studies. Simultaneous Feature Positive Discrimination (SFPD) [11] is a conditioning phenomenon where there are two stimuli which are correlated with reinforcement. One stimulus is followed by reinforcement every time it is presented, whereas the other stimulus is followed by reinforcement when it is paired with the first stimulus and not reinforced when presented in other trials. After training, the animal learns that the highly correlated stimulus predicts the reinforcement and that the other stimulus does not, even though it is correlated with reinforcement half of the time. In contrast, partial reinforcement [12] involves the presentation of a stimulus that is followed by reinforcement with some probability less than one. We submit that the reason why this can be learned by the animal is that there are not any more highly correlated features combined with these partial predictors in the experiment. Otherwise, a relative correlation would take place as seems to occur in SFPD.

One criticism of the present work regarding biologically plausibility is that learning continues until the training set prediction error reaches a minimum or some low error threshold (i.e., batch learning). Based on our previous work and work in progress, we argue that some batch processes can be approximated in an online learner. Elsewhere, we have shown how several retrospective revaluation (classical conditioning) phenomena can be simulated using a specialized online associative learning approach [13], which to be achieved by LMS would require batch learning.

A beneficial trait of the probabilistic and Dual Pathway Regression models is that the individual probabilities or parameters clearly indicate which features are relevant/irrelevant. These models may then also serve as standalone feature selection methods.

Regularization is a recognized technique to improve generalization in LMS that also seems biologically plausible. Regularization (i.e., weight decay, ridge regression, and the LASSO) tends to suppress irrelevant features and additive noise [14]. The least correlated features succumb to regularization whereas the highly correlated features overcome it. Regularization thus somewhat mimics the relative correlation computation. Future work will involve making comparisons between regularization and the present approach.

We have shown how relative correlation approaches are effective in feature selection for linear regression. Yet it is known that organisms also face non-linear predictive situations, which could benefit from feature selection. In recent work [13], we demonstrated a robust approach to learning non-linear relationships in a dual pathway framework operating on similar learning rules as in Dual Pathway Regression. Thus a natural next step is to combine this mechanism with the new way of integrating inhibition described herein.

**IX. Conclusions**

LMS is effective when the additive noise is zero and there are at least as many data points as features. However, when this is not the case, being able to discern the most highly correlated features is an effective strategy for reducing test set prediction error. This strategy falls out of maximum likelihood estimation for the Noisy OR model. We have shown that models that achieve feature selection by computing the relative correlation have lower prediction errors than LMS as system noise and the number of irrelevant features increases (when there are relatively few data points).

We have identified that integrating inhibition by multiplying a positive prediction by a number between 0 and 1 (smaller for larger inhibition) causes an LMS-like model to employ relative correlation. Such a solution may be implemented in the biology in the structure of the basal ganglia.

**References**


