ftp://psyche.mit.edu/pub/jordan/uai.ps

Why the logistic function? A tutorial discussion on probabilities and neural networks

Michael I. Jordan
Massachusetts Institute of Technology

Computational Cognitive Science
Technical Report 9503

August 13, 1995

This paper presents a tutorial introduction to the logistic function as a statistical object. Beyond the discussion of the whys and wherefores of the logistic function, I also hope to illuminate the general distinction between the "generative/causal/class-conditional" and the "discriminative/diagnostic/predictive" directions for the modeling of data. Crudely put, the belief network community has tended to focus on the former while the neural network community has tended to focus on the latter (although there are numerous papers in both communities going against their respective grains). It is the author's view that these two directions are two sides of the same coin, a corollary of which is that the two network-based communities are in closer contact than one might otherwise think. To illustrate some of the issues involved, I discuss the simplest nonlinear neural network—a logistic function of a linear combination of the input variables (also known in statistics as a logistic regression).

The logistic function has had a lengthy history in classical statistics and in neural networks. In statistics it plays a leading role in the methodology of logistic regression, where it makes an important contribution to the literature on classification. The logistic function has also appeared in many guises in neural network research. In early work, in which continuous time formalisms tended to dominate, it was justified via its being the solution to a particular differential equation. In later work, with the emphasis on discrete time, it was generally used more heuristically as one of the many possible smooth, monotonic "squashing" functions that map real values into a bounded interval. More recently, however, with the increasing focus on learning, the probabilistic properties of the logistic function have begun to

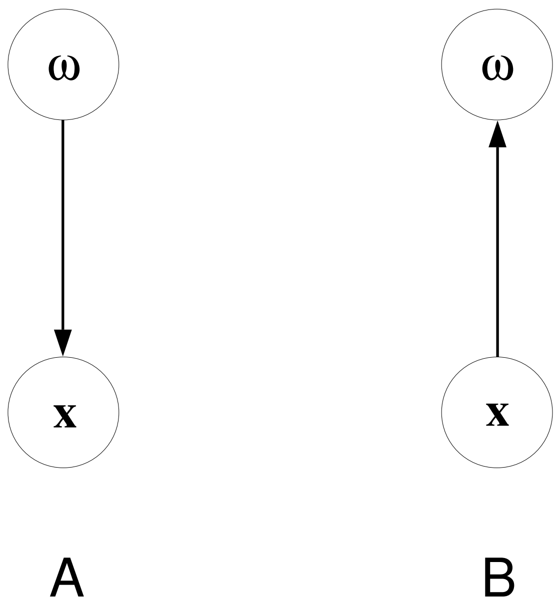

Figure 1: Two possible belief networks for solving a binary classification problem. In the remainder of the paper, we refer to network A as being oriented in the "generative" direction, and network B as oriented in the "diagnostic" direction.

be emphasized. This emphasis has led to better learning methods and has helped to strengthen the links between neural networks and statistics.

Binary classification

We consider a simple classification problem in which data are labeled by a random variable ω, which takes its values from a discrete set, $\omega \in \{\omega_0, \omega_1\}$. The measured data are in the form of a d-dimensional random vector, $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$. A belief network model for this problem is shown in Figure 1(A), where we show a binary node ω and a directed link from ω to the measurement vector $\mathbf{x}$. In this network model, we must specify marginal probabilities $P(\omega)$ and conditional probabilities $P(\mathbf{x}|\omega)$.

Given a particular vector $\mathbf{x}$, we wish to assign it to one of the two classes. To this end we use Bayes' rule and calculate the relevant posterior probability:

$$P(\omega_0|\mathbf{x}) = \frac{P(\mathbf{x}|\omega_0)P(\omega_0)}{P(\mathbf{x})} \tag{1}$$

Continuing the derivation, we expand the denominator and (with some foresight) introduce an exponential:

$$\begin{aligned} P(\omega_0|\mathbf{x}) &= \frac{P(\mathbf{x}|\omega_0)P(\omega_0)}{P(\mathbf{x}|\omega_0)P(\omega_0) + P(\mathbf{x}|\omega_1)P(\omega_1)} \\ &= \frac{1}{1 + \exp\left\{-\log\left[\frac{P(\mathbf{x}|\omega_0)}{P(\mathbf{x}|\omega_1)}\right] - \log\left[\frac{P(\omega_0)}{P(\omega_1)}\right]\right\}} \end{aligned} \tag{2}$$

We see that the posterior probability can be written in the form of the *logistic function*:

$$y = \frac{1}{1 + e^{-\xi}}, \tag{3}$$

where ξ is a function of the likelihood ratio $P(\mathbf{x}|\omega_0)/P(\mathbf{x}|\omega_1)$ and the prior ratio $P(\omega_0)/P(\omega_1)$. Of course we haven't necessarily achieved anything in general with this bit of mathematical maneuvering, but we may have achieved something in particular cases if the likelihood ratio turns out to be a simple function of $\mathbf{x}$.

In belief network A in Figure 1, we must choose a particular form for the conditional probabilities $P(\mathbf{x}|\omega)$ (the *class-conditional densities*). Let us assume that they are multivariate Gaussians with identical covariance matrices Σ:

$$P(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma^{-1}(\mathbf{x}-\mu_i)} \tag{4}$$

where μ_i is the mean vector for class i.

Expanding the exponent of the Gaussians, and substituting into the formula for the posterior, we obtain:

$$P(\omega_0|\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}}, \tag{5}$$

where

$$\mathbf{w} = \Sigma^{-1}(\mu_0 - \mu_1) \tag{6}$$

and

$$b = \frac{1}{2}(\mu_1 + \mu_0)^T\Sigma^{-1}(\mu_1 - \mu_0) + \log\frac{P(\omega_0)}{P(\omega_1)}. \tag{7}$$

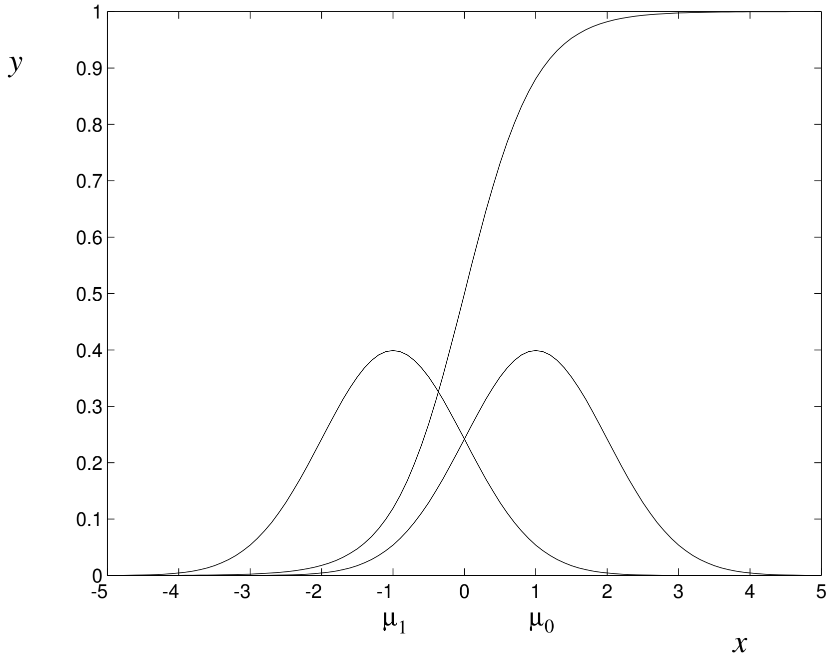

Figure 2: A binary classification problem. The class-conditional densities are Gaussians with unit variance. The posterior probability is the logistic function $y = 1/(1 + \exp(-2x))$.

This function is a ramp varying from zero to one along the direction from μ_1 to μ_0. The geometry is summarized in 1-D in Figure 2.

Now let us now consider an alternative belief network representation of our problem. In belief network B in Figure 1, the conditional probability $P(\omega|\mathbf{x})$ must be specified. We already know what form this conditional distribution should take if we wish to be consistent with our earlier assumption that the class-conditional densities are Gaussian: The conditional should be a logistic-linear function of $\mathbf{x}$. Our belief network requires $n+1$ numbers on the link from $\mathbf{x}$ to ω to parameterize the linear part of the function (cf. Equation 5).

To complete the quantitative specification of network B, we must also specify the marginal $P(\mathbf{x})$. In using the network for classification, however, we will be providing it with $\mathbf{x}$ values as evidence, thus we will not bother with modeling $P(\mathbf{x})$.[1]

[1] $P(\mathbf{x})$ is a mixture density and if we did wish to model it we would use the EM algo-

What are the advantages and disadvantages of the two networks? Clearly belief network A is more "modular" than the second. The class-conditional densities are likely to be local, characteristic functions of the objects being classified, invariant to the nature and number of the other classes (cf. Pearl). (Note in particular that network A separates out $P(\omega)$ from the conditional; in network B $P(\omega)$ is tangled up with the conditional.) Indeed, we can more easily envision Nature generating objects by first deciding what kind of object to generate and then generating it from a recipe. Going the other direction seems less Natural. Are there *any* advantages to the $\mathbf{x} \rightarrow \omega$ direction?

Suppose that we investigate class-conditional densities other than Gaussians. For example, $P(\mathbf{x}|\omega)$ might be a gamma density, or, if we consider discrete measurements, binomial or Poisson. We can redo our calculation of the posterior probability for each of these cases, or alternatively, we can note that all of these densities (and many others besides, including the negative binomial, the hypergeometric and the exponential), are special cases of a general family of distributions known as the exponential family. The exponential family densities are all of the following form:

$$P(\mathbf{x}|\theta, \phi) = \exp\{(\theta^T \mathbf{x} - b(\theta))/a(\phi) + c(\mathbf{x}, \phi)\}, \tag{8}$$

where θ and ϕ are the parameters (θ is known as the *location parameter*, and ϕ is the *dispersion parameter*).

If we now assume that each of our class-conditional densities are members of the (same) exponential family distribution, with equal dispersion parameters, and substitute this general form into Eq. 2, we find that we once again obtain a linear form for the discriminant ξ. Thus in all cases the posterior probability is a logistic function of a linear combination of the components of $\mathbf{x}$.

This shows that there is an advantage to specifying our classification problem in the form of belief network B. This network representation is invariant to a *family* of classification problems; those in which the class-conditional densities are in the exponential family (with equal dispersion parameters). In other words, we don't require a particular distribution to be specified when we use network B. Belief network A, on the other hand,

rithm. This might be useful if there were missing components in the $\mathbf{x}$ vector. However, modeling the mixture density requires making specific assumptions about the form of the mixture components and, as will become clear, avoiding having to make such specific assumptions is one of the advantages of modeling in the diagnostic direction. Missing input data, therefore, tempers the advantages associated with diagnostic modeling.

does require such a specification. If the specification isn't a good match to the data set, performance will suffer. In statistical language, network B is more *robust* than network A.

Another advantage of network B is that its parameterization is simpler than that of network A. In the Gaussian case, network A requires knowledge of $\mathcal{O}(n^2)$ parameters (the covariance matrix and mean vectors), whereas network B requires only $\mathcal{O}(n)$ numbers (the coefficients of the linear combination). This is often the case for classification problems—generative models typically require many more parameters than are required of diagnostic models.

Lest the reader begin to feel the pendulum swinging too rapidly in the other direction, let us state our view: neither direction is inherently better, rather the generative and diagnostic directions are two sides of the same coin and have complementary advantages and disadvantages. From a purely probabilistic point of view the two networks in Figure 1 are equivalent parameterizations of the joint density. They are *not* equivalent, however, from a statistical point of view, and in particular situations statistical considerations, in conjunction with considerations such as modularity, stability, prototypicality, causality, etc., may lead one to prefer one over the other. The major statistical advantage of network A is that if the model specification is nearly correct, then estimation in that network structure will be more *efficient* than estimation in B (will require fewer data points to obtain a given level of statistical accuracy). The statistical advantage of network B, on the other hand, is that it is more *robust* to uncertainty about the data generation process. For certain simple models quantitative measures of efficiency and robustness are available (see, e.g., McLachlan, 1992); however, for more complex models the theory is less developed. It may be that the tradeoffs are different for different kinds of nonlinear models. This is a topic that needs increased research attention.

Each diagnostic architecture will have an accompanying family of generative architectures and vice versa. It is important to have some appreciation of the family that is being assumed on the other side of the coin while working with a particular architecture. For example, to understand the robustness that one might expect of a particular diagnostic architecture, it is important to know what generative architectures will yield the diagnostic law being proposed. When studying a generative architecture, it is important to understand what form the corresponding diagnostic law will take; this can help the researcher to focus on the important parts of the generative law (e.g., the generative parameters to which the diagnostic law is the most sensitive).

It may also be worth considering architectures that explicitly parameterize both a generative and a diagnostic law rather than obtaining one from the other. For a recent proposal of this kind see Hinton, Dayan, Frey, and Neal (1995).

Parameter estimation

We now press on and consider the problem of parameter estimation. Let us assume that we have collected a set of data in the form of $\{\mathbf{x}, \omega\}$ pairs, and assume that we wish to adjust the parameters in the two networks in Figure 1. For simplicity we will assume uniform priors on all of the parameters and focus on maximizing the likelihood. We also ignore the possibility of associating different costs with the different classification decisions.

We assume that we have a training set $\mathcal{X} = \{\mathbf{x}^{(p)}, \omega^{(p)}\}_{p=1}^{N}$ and that the goal is to maximize the log likelihood

$$\log P(\mathcal{X}|\Theta) = \sum_{p=1}^{N} \log P(\mathbf{x}^{(p)}, \omega^{(p)}) \tag{9}$$

with respect to the parameters Θ.

For network A, we need not trouble ourselves to do the derivation—it is clear how to set the parameters from the data. We split the data set into those pairs for which $\omega = \omega_0$ and those pairs for which $\omega = \omega_1$. We then estimate the parameters of $P(\mathbf{x}|\omega)$ separately in the two cases. For example, if $P(\mathbf{x}|\omega)$ is a Gaussian model for the two classes, we estimate a mean vector and covariance matrix separately for each of the classes. The maximum likelihood estimate for the parameters in our model are the corresponding sample mean vectors and a pooled sample covariance matrix that combines the sample covariance matrices for the two classes. These estimates maximize the joint likelihood in Equation 9. We also must estimate $P(\omega)$, but this is clearly achieved by calculating the proportion of samples in the two classes.

Estimation for network A will always reduce to separate estimation for each of the two classes. Since we have specified the class-conditional densities, we simply use standard algorithms for estimation of this density.

Estimation for network B requires a bit more work. We factor the log likelihood into an unconditional term $P(\mathbf{x})$ that we will not bother with and a conditional term $P(\omega|\mathbf{x})$ that will be our focus. Given that ω is a binary random variable, this density is a Bernoulli density. To simplify the notation, let us define an indicator random variable z_p, which is one when $\omega^{(p)} = \omega_0$, and zero otherwise. Let us also define

$$y_p \equiv P(\omega^{(p)} = \omega_0 | \mathbf{x}^{(p)}),$$

which, as we have seen earlier, is a logistic-linear function of $\mathbf{x}^{(p)}$. Using L to denote the conditional density of the data set, we write:

$$L = \prod_{p=1}^{N} y_p^{z_p} (1 - y_p)^{(1 - z_p)}. \tag{10}$$

This then yields the following log likelihood:

$$\log L = \sum_{p=1}^{N} [z_p \log y_p + (1 - z_p) \log(1 - y_p)], \tag{11}$$

which is the function that we wish to maximize.

The negative of the log-likelihood in Equation 11 is a *cross entropy* between the indicator z variables and the posterior probabilities y. Our derivation shows that the cross entropy is a natural cost function for the binary classification problem.

We now appeal to standard optimization methods to minimize the cross entropy. Simple gradient descent yields the following update rule for $\mathbf{w}$:

$$\Delta \mathbf{w} = \mu \sum_{p=1}^{N} (z_p - y_p) \mathbf{x}^{(p)}, \tag{12}$$

where μ is the step size. Alternatively, taking second derivatives of $\log L$ yields the Newton-Raphson update:

$$\Delta \mathbf{w} = (X^T V_r X)^{-1} X^T V_r \mathbf{z}^*, \tag{13}$$

where r is the iteration number, X is the matrix whose rows are the vectors $\mathbf{x}^{(p)}$, V_r is a diagonal matrix whose elements are $y_p(1 - y_p)$, and $\mathbf{z}^*$ is a vector whose elements are $(z_p - y_p)/\{y_p(1 - y_p)\}$. This update rule is referred to as Iteratively Reweighted Least Squares (IRLS).[2] Newton-Raphson takes this

[2]The name comes from noting that Equation 13 is in the form of a set of normal equations for a weighted least squares regression of $\mathbf{z}^*$ on X. The weights in this regression (the matrix V_r) change at each iteration of the least squares equations; thus the regression is "iteratively reweighted."

form not only for logistic regression problems, but for a family of statistical models known as Generalized Linear Models (McCullagh & Nelder, 1984).[3]

Note that both of the fitting algorithms for Network B are based on the "errors" $(z_p - y_p)$. This is generally the case—error-based updates generally arise for fitting algorithms that directly adjust the parameters of diagnostic models.

Nonlinear discriminant functions

The derivation that we have presented shows how the logistic function arises as a natural representation for the posterior probability in a binary classification problem. The particular derivation also yielded a linear form for the discriminant surface ξ. In generalizing to architectures in which the discriminant surface is nonlinear, there are two possibilities. We can retain the logistic function, retain the cross entropy cost function based on the Bernoulli distribution, and focus on nonlinear representations for the discriminant surface within the "shell" provided by the logistic regression model. Or we can drop the logistic function altogether and focus on methods that alter the discriminant surface directly to minimize some cost function related to classification error. The tendency within the neural network field has been to retain the logistic function and the associated cross entropy function. This approach also characterizes related methodologies such as "generalized additive models" (Hastie & Tibshirani, 1990), where the binary response is modeled as the logistic function of the additive combination of nonlinear functions of the coordinates of $\mathbf{x}$.

Methods that retain the logistic function have one natural advantage in the context of the current essay—they provide a probability. We can therefore easily imagine utilizing a simple neural network (or generalized additive model, or some other "flexible" approximator within a logistic shell) to replace the probabilistic tables or linear regressions often used in the belief network literature. Using a linear network with a logistic function—the focus of this essay—is closely related to the use of the "noisy-OR" approximation (cf. Neal, 1992). But a much wider variety of nonlinear approximations for ξ are available and can provide a more flexible approximation to a probability

[3] One important instance of a Generalized Linear Model is the generalization of logistic regression to $m > 2$ classes. The probabilistic model for the multi-class problem is the multinomial model, and the logistic function is replaced with a normalized exponential function known in the neural network literature as the "softmax" function. See Jordan and Jacobs (1994) and Rumelhart, Durbin, Golden and Chauvin (1995) for further discussion.

table.

One final issue bears commenting upon. Classical neural networks utilize logistic functions not only for the output units of the network but also for the "hidden units" of the network. What does our analysis have to say about the cascades of logistic functions that arise in layered networks? In fact, within the framework of function approximation theory (which is often invoked to justify neural network architectures; see, e.g., Hornik, Stinchcombe, and White, 1989), there is no compelling reason to use the logistic function as compared to any other smooth bounded monotonic nonlinearity (the asymptotic approximation error will scale similarly for essentially any such nonlinearity). It is also possible, however—and increasingly a topic of research interest—to take a more thoroughgoing probabilistic stance and treat layered neural networks as "sigmoid belief networks," in which the logistic functions are viewed as probabilities associated with latent binary random variables. Such a network is a neural network with belief network semantics. For recent work in this direction, see Hinton et al. (1995) and Saul, Jaakkola, and Jordan (1995).

A large number of more advanced topics take off from where we are ending, including the topic of "discriminative training" (a hybrid methodology that uses a generative model, but alters the training procedure of the generative model so as to obtain better discriminative behavior), the use of the EM algorithm with binary latent variables, connections to chain graphs and CG distributions, and connections to information theoretic methods. The development of these topics and others are left as exercises for the reader!

Acknowledgments

Thanks go to Wray Buntine, Peter Dayan, and David Heckerman for their helpful comments on the manuscript.

Bibliography

Most of the material discussed here is available in modern statistics books. A good advanced book on classification is:

McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition.* New York: John Wiley.

See, for example, pages 276-279 of McLachlan for discussion of the efficiency and robustness of logistic regression vs. linear discriminant analysis.

Classification ideas have also been developed somewhat separately in the pattern recognition literature. The standard reference is:

Duda, R. O. & Hart, P. E. (1973). *Pattern Classification and Scene Analysis.* New York: John Wiley.

and it is still well worth reading. The two directions (generative and diagnostic) are quite explicit in Duda and Hart, and they have much to say about the relationships. Interestingly, however, the logistic function is nowhere to be found. It is a major missing component in an otherwise comprehensive textbook. Duda and Hart do present a theorem that effectively shows that the terms in the Taylor series expansion for the posterior probability can be estimated by a learning algorithm, but they fail to notice that the Taylor series in question is the Taylor series of the logistic function!

Additional references from the text:

Hastie, T. & Tibshirani, R. J. (1990). *Generalized Additive Models.* London: Chapman and Hall.

Hinton, G. E., Dayan, P., Frey, B. J., & Neal, R. M. (1995). The wake-sleep algorithm for unsupervised neural networks. *Science.*

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*, 359-366.

Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, *6*, 181-214.

Neal, R. M. (1992). Connectionist learning of belief networks. *Artificial Intelligence*, *56*, 71-113.

Rumelhart, D. E., Durbin, R., Golden, R., & Chauvin, Y. (1995). Backpropagation: The basic theory. In Chauvin, Y., & Rumelhart, D. E. (Eds.), *Backpropagation: Theory, Architectures, and Applications.* Hillsdale, NJ: Lawrence Erlbaum.

Saul, L. K., Jaakkola, T., & Jordan, M. I. (1995). Mean field learning theory for sigmoid belief networks. Computational Cognitive Science Tech. Rep. 9501, MIT, Cambridge, MA. `[ftp://psyche.mit.edu/pub/lksaul/belief.ps.Z]`

Appendix

The exponential family of distributions has a number of useful properties. In this section, we briefly examine one of these properties in order to provide some additional insight into the role of the logistic function in classification problems.

Recall the form of the Bernoulli distribution for a binary random variable x:

$$P(x) = \mu^x (1-\mu)^{1-x},$$

with parameter μ (the probability of "success").

Recall also the general form of the exponential family:

$$P(x|\theta, \phi) = \exp\{(\theta x - b(\theta))/a(\phi) + c(x, \phi)\},$$

where we have specialized to a scalar random variable x.

The Bernoulli distribution can be rewritten in the exponential family form as follows:

$$P(x) = \exp\left\{\left[\log\left(\frac{\mu}{1-\mu}\right)\right] x + \log(1-\mu)\right\}.$$

We see that μ and θ are alternative parameterizations of the Bernoulli distribution. The relationship between these parameterizations can be extracted as follows:

$$\theta = \log\left(\frac{\mu}{1-\mu}\right),$$

which is the log odds of "success." Inverting the log odds, we obtain:

$$\mu = \frac{1}{1+e^{-\theta}},$$

which is the logistic function.

In the terminology of Generalized Linear Models (McCullagh & Nelder, 1984), we have shown that the logistic function is the (inverse) "canonical link" for the Bernoulli distribution. Conceptually, this makes the decision of using the logistic function in our classification model somewhat subordinate to our decision to use the Bernoulli distribution. Assuming that we have decided that the Bernoulli distribution is appropriate for our problem (a sensible assumption for binary classification), we derive the logistic function as the inverse canonical link, showing that the logistic function is a sensible choice for the nonlinear function linking the predictor (θ) to the mean (μ)

(see McCullagh & Nelder, 1984, for discussion of the properties of canonical links).

This approach can be pursued to provide link functions for the other members of the exponential family. One writes the distribution under consideration in the exponential family form and reads off the canonical link. This approach provides natural nonlinear models for many different types of data formats, including counts, rates, time intervals, etc.