Exploiting generative models in discriminative classifiers

Tommi S. Jaakkola and David Haussler
{tommi,haussler}@cse.ucsc.edu
Department of Computer Science
University of California
Santa Cruz, CA 95064
and
Isaac Newton Institute for Mathematical Sciences
University of Cambridge
20 Clarkson Road
Cambridge, CB3 0EH, U.K.

Abstract

Generative probability models such as hidden Markov models provide a principled way of treating missing information and dealing with variable length sequences. On the other hand, discriminative methods such as support vector machines enable us to construct flexible decision boundaries and often result in classification performance superior to that of the model based approaches. An ideal classifier should combine these two complementary approaches. In this paper, we develop a natural way of achieving this combination by deriving kernel functions for use in discriminative methods such as support vector machines from generative probability models. We provide a theoretical justification for this combination as well as demonstrate a substantial improvement in the classification performance in the context of DNA and protein sequence analysis.

1 Introduction

Speech, vision, text and biosequence data can be difficult to deal with in the context of simple statistical classification problems. Because the examples to be classified are often sequences or arrays of variable size that may have been distorted in particular ways, it is common to estimate a generative model for such data, and then use Bayes rule to obtain a classifier from this model. However, many discriminative methods, which directly estimate a posterior probability for a class label (as in Gaussian process classifiers [8]) or a discriminant function for the class label (as in support vector machines [9]) have in other areas proven to be superior to generative models for classification problems. The problem is that there has been no systematic way to extract features or metric relations between examples for use with discriminative methods in the context of difficult data types such as those listed above. Here we propose a general method for extracting these discriminatory features using a generative model. The features we propose are generally applicable but are most naturally suited to kernel methods for discriminative classification.

2 Kernel methods

Here we provide a brief introduction to kernel methods; see, e.g., [9] [8] for more details. Suppose now that we have a training set of examples X_i and corresponding binary labels S_i (± 1). In kernel methods, as we define them, the label for a new example X is obtained from a weighted sum of the training labels. The weighting of each training label S_i consists of two parts: 1) the overall importance of the example X_i as summarized with a coefficient λ_i and 2) a measure of pairwise "similarity" between between X_i and X, expressed in terms of a kernel function $K(X_i, X)$. The predicted label $\hat{S}$ for the new example X is derived from the following rule:

$$\hat{S} = \text{sign}\left(\sum_i S_i \, \lambda_i K(X_i, X) \right) \tag{1}$$

We note that this class of kernel methods also includes probabilistic classifiers, in which case the above rule refers to the label with the maximum probability. The free parameters in the classification rule are the coefficients λ_i and to some degree also the kernel function K. To pin down a particular kernel method, two things need to be clarified. First, we must define a classification loss, or equivalently, the optimization problem to solve to determine appropriate values for the coefficients λ_i. Slight variations in the optimization problem can take us from support vector machines to generalized linear models. The second and the more important issue is the choice of the kernel function – the main topic of this paper. We begin with a brief illustration of generalized linear models as kernel methods.

2.1 Generalized linear models

For concreteness we consider here only logistic regression models, while emphasizing that the ideas are applicable to a larger class of models[1]. In logistic regression models, the probability of the label S given the example X and a parameter vector θ is given by[2]

$$P(S|X, \theta) = \sigma\left(S\theta^T X \right) \tag{2}$$

[1]Specifically, it applies to all generalized linear models whose transfer functions are log-concave.

[2]Here we assume that the constant $+1$ is appended to every feature vector X so that an adjustible bias term is included in the inner product $\theta^T X$.

where $\sigma(z) = (1+e^{-z})^{-1}$ is the logistic function. To control the complexity of the model when the number of training examples is small we can assign a prior distribution $P(\theta)$ over the parameters. We assume here that the prior is a zero mean Gaussian with a possibly full covariance matrix Σ. The maximum *a posteriori* (MAP) estimate for the parameters θ given a training set of examples is found by maximizing the following penalized log-likelihood:

$$\sum_i \log P(S_i|X_i,\theta) + \log P(\theta) \quad = \quad \sum_i \log \sigma\left(\, S_i \theta^T X_i \,\right) - \frac{1}{2}\theta^T \Sigma^{-1} \theta + c \tag{3}$$

where the constant c does not depend on θ. It is straightforward to show, simply by taking the gradient with respect to the parameters, that the solution to this (concave) maximization problem can be written as[3]

$$\hat{\theta} \;=\; \sum_i S_i \lambda_i \,\Sigma X_i, \quad \text{where} \tag{4}$$

$$\lambda_i \;=\; \frac{\partial}{\partial z} \log \sigma(z) \Big|_{z = S_i \hat{\theta}^T X_i} = \sigma\left(-S_i \hat{\theta}^T X_i\right) \tag{5}$$

Note that the coefficients λ_i appear as weights on the training examples as in the definition of the kernel methods. Indeed, inserting the above solution back into the conditional probability model gives

$$P(S|X,\hat{\theta}) = \sigma\left(\, S \sum_i S_i \lambda_i \,(X_i^T \Sigma X)\right) \tag{6}$$

By identifying $K(X_i, X) = X_i^T \Sigma X$ and noting that the label with the maximum probability is the one that has the same sign as the sum in the argument, this gives the decision rule (1).

Trough the above derivation, we have written the primal parameters θ in terms of the dual coefficients λ_i[4]. Consequently, the penalized log-likelihood function can be also written entirely in terms of λ_i; the resulting likelihood function specifies how the coefficients are to be optimized. This optimization problem has a unique solution and can be put into a generic form (see [5] for details). Also, the form of the kernel function that establishes the connection between the logistic regression model and a kernel classifier is rather specific, i.e., has the inner product form $K(X_i, X) = X_i^T \Sigma X$. However, as long as the examples here can be replaced with feature vectors derived from the examples, this form of the kernel function is the most general. We discuss this further in the next section.

3 The kernel function

The inner product form $K(X_i, X) = X_i^T \Sigma X$ for the kernel above is actually quite general. For a general kernel function to be valid, roughly speaking it only needs to be positive semi-definite (see e.g. [10]). According to the Mercer's theorem,

[3] This corresponds more generally to a Legendre transformation of the loss functions $\log \sigma(z)$.

[4] This is possible for all those θ that could arise as solutions to the maximum penalized likelihood problem; in other words, for all relevant θ.

any such valid kernel function admits a representation as a simple inner product between suitably defined feature vectors, i.e.,

$$K(X_i, X_j) = \phi_{X_i}^T \phi_{X_j} \tag{7}$$

where the feature vectors come from some fixed mapping $X \to \phi_X$. For example, in the previous section the kernel function had the form $X_i^T \Sigma X_j$, which is a simple inner product for the transformed feature vector $\phi_X = \Sigma^{\frac{1}{2}} X$.

Specifying a simple inner product in the feature space defines a Euclidean metric space. Consequently, the Euclidean distances between the feature vectors are obtained directly from the kernel function:

$$\| \phi_{X_i} - \phi_{X_j} \|^2 = K(X_i, X_i) - 2K(X_i, X_j) + K(X_j, X_j) \tag{8}$$

In addition to defining the metric structure in the feature space, the kernel defines a pseudo metric in the original example space through $D(X_i, X_j) = \| \phi_{X_i} - \phi_{X_j} \|$. Thus the kernel embodies prior assumptions about the metric relations between the original examples. No systematic procedure has been proposed for finding kernel functions, let alone finding ones that naturally handle variable length examples etc. This is the topic of the next section.

4 Kernels from generative probability models: the Fisher kernel

The key idea here is to derive the kernel function from a generative probability model. While the need for such a kernel function is clear, it is less apparent how it can be derived from the generative model in a general way. We arrive at the same kernel function from two different perspectives, that of enhancing the discriminative power of the model (discussed in Appendix A) and from an attempt to find a natural comparison between examples induced by the generative model. The latter perspective is presented in this section.

We have seen in the previous section that defining the kernel function automatically implies assumptions about metric relations between the examples. We argue that these metric relations should be defined directly from a generative probability model $P(X|\theta)$. When our goal is classification, this generative model may include the classification variable S as a latent variable. However, of interest here is the difference in the generative process between a pair of examples X_i and X_j, rather than say the differences in the posterior probabilities for the labels S computed separately for each of the example, which is all that is used for discrimination in the purely generative approach.

To capture the generative process in a metric between examples we use the gradient space of the generative model. The gradient of the loglikelihood with respect to a parameter describes how that parameter contributes to the process of generating a particular example. For the exponential family of distributions, under the natural parameterization θ, these gradients, less a normalization constant that depends on θ, form sufficient statistics for the example. This gradient space also naturally preserves all the structural assumptions that the model encodes about the generation process.

To develop this idea more generally, consider a parametric class of models $P(X|\theta)$, $\theta \in \Theta$. This class of probability models defines a Riemannian manifold M_Θ with a local metric given by the Fisher information matrix I, where $I = E_X\{U_X U_X^T\}$, $U_X = \nabla_\theta \log P(X|\theta)$, and the expectation is over $P(X|\theta)$ (see e.g. [1]). For simplicity we have suppressed the dependence of I and U_X on the parameter setting θ, or equivalently, on the position in the manifold. The gradient of the log-likelihood, U_X, is called the *Fisher score*, and plays a fundamental role in our development.

The local metric on M_Θ defines a distance between the current model $P(X|\theta)$ and a nearby model $P(X|\theta+\delta)$. This distance is given by $D(\theta, \theta+\delta) = \frac{1}{2}\delta^T I \delta$, which also approximates the KL-divergence between the two models for a sufficiently small δ.

The Fisher score $U_X = \nabla_\theta \log P(X|\theta)$ maps an example X into a feature vector that is a point in the gradient space of the manifold M_Θ. We call this the *Fisher score mapping*. This gradient U_X can be used to define the direction of steepest ascent in $\log P(X|\theta)$ for the example X *along the manifold*, i.e., the gradient in the direction δ that maximizes $\log P(X|\theta)$ while traversing the minimum distance in the manifold as defined by $D(\theta, \theta+\delta)$. This latter gradient is known as the natural gradient (see e.g. [1]) and is obtained from the ordinary gradient via $\phi_X = I^{-1}U_X$. We will call the mapping $X \to \phi_X$ the *natural mapping* of examples into feature vectors. Again, we have supressed dependence on the parameter setting θ here. The natural kernel of this mapping is the inner product between these feature vectors relative to the local Riemannian metric:

$$K(X_i, X_j) \propto \phi_{X_i}^T I \phi_{X_j} = U_{X_i}^T I^{-1} U_{X_j} \tag{9}$$

We call this the Fisher kernel owing to the fundamental role played by the Fisher scores in its definition. The role of the information matrix is less significant; indeed, in the context of logistic regression models, the matrix appearing in the middle of the feature vectors relates to the covariance matrix of a Gaussian prior, as show above. Thus, asymptotically, the information matrix is immaterial, and the simpler kernel $K_U(X_i, X_j) \propto U_{X_i}^T U_{X_j}$ provides a suitable substitute for the Fisher kernel.

While from the metric point of view a scaled/translated kernel $\tilde{K}(X_i, X_j) = cK(X_i, X_j) + c_0$ (where $c, c_0 \geq 0$) is an equivalent kernel, such operations may be important in practice. For example, in the logistic regression context, the additive constant in the kernel has to do with the prior variance of the bias term[5]. The multiplicative factor c relates to the overall prior variance of the remaining parameters, i.e., to the natural gradients, and can be of order 1 whenever the information matrix is included.

More generally, we emphasize that the Fisher kernel defined above provides only the basic comparison between the examples, defining what is meant by an "inner product" between examples when the examples are objects of various types (e.g. variable length sequences). The way such a kernel function is used in a discriminative classifier is not specified here. Using the Fisher kernel directly in a kernel classifier, for example, amounts to finding a linear separating hyperplane in the natural gradient (or Fisher score) feature space. The examples may not be linearly separable in this feature space although the the natural metric structure is given by the Fisher kernel. It may be advantageous to search in the space of quadratic (or

[5]We have tacitly assumed the that the original examples were given in homogenous coordinates.

higher order) decision boundaries, which is equivalent to transforming the Fisher kernel according to $\tilde{K}(X_i, X_j) = (1 + K(X_i, X_j))^m$ and using the resulting kernel $\tilde{K}$ in the classifier.

We are now ready to state a few properties of this kernel function in the form of a theorem:

Theorem 1 *For any (suitably regular[6]) probability model $P(X|\theta)$ with parameters θ, the Fisher kernel*

$$K(X_i, X_j) = U_{X_i}^T \, I^{-1} \, U_{X_j} \tag{10}$$

where $U_X = \nabla_\theta \log P(X|\theta)$ has the following properties:

a) it is a valid kernel function

b) it is invariant to any invertible (and differentiable) transformation of the parameters

c) a kernel classifier employing the Fisher kernel derived from a model that contains the label as a latent variable is, asymptotically, at least as good a classifier as the MAP labeling based on the model

The first property is immediate since I is positive definite. The second property follows from the fact that the kernel was derived with reference only to the manifold M_Θ, which is intrinsic to the class of probability models and not dependent on the parameterization. [7] The third property can be established on the basis of the discriminative derivation of this kernel (see Appendix A). We omit the details for brevity.

To summarize, we have defined a generic procedure for obtaining kernel functions from generative probability models. Consequently the benefits of generative models are immediately available to the discriminative classifier employing this kernel function. We now turn the experimental demonstration of the effectiveness of such a combined classifier.

5 Experimental results

Here we consider two relevant examples from biosequence analysis and compare the performance of the combined classifier to the best generative models used in these problems. We start with a DNA splice site classification problem, where the objective is to recognize true splice sites, i.e., the boundaries between expressed regions (exons) in a gene and the intermediate regions (introns). The data set used in our experiments consisted of 9350 DNA fragments from *C. elegans* collected by Jonathon King of the Sanger Centre and Arun Jagota at UCSC. Each of the 2029 true examples is a sequence X over the DNA alphabet $\{A, G, T, C\}$ of length 25 centered on the consensus 'GT' at the 5' splice boundary. The 7321 false examples are similar sequences centered at instances of 'GT' that occur near but not at 5'

[6] We must have a twice differentiable likelihood so that the Fisher information I exists, and I must be positive definite at the chosen θ.

[7] A mechanical proof involving Jacobians is also straightforward.

splice sites. All recognition rates we report on this data set are averages from 7-fold cross-validation.

To use the combined classifier in this setting requires us to choose a generative model for the purpose of deriving the kernel function. In order to test how much the performance of the combined classifer depends on the quality of the underlying generative model, we chose the poorest model possible. This is the model where the DNA residue in each position in the fragment is chosen independently of others, i.e.,

$$P(X|\theta) = \prod_{i=1}^{25} P(X_i|\theta_i) \tag{11}$$

and, furthermore, the parameters θ_i are set such that $P(X_i|\theta_i) = 1/4$ for all i and all $X_i \in \{A, G, T, C\}$. This model assigns the same probability to all examples X. As poor a model as this is, we can still derive the Fisher kernel from such a model and use it in a discriminative classfier. In this case we used a logistic regression model as in (6) with a quadratic Fisher kernel $\tilde{K}(X_i, X_j) = (1 + K(X_i, X_j))^2$. Figure 1 shows the recognition performance of this kernel method, using the poor generative model, in comparison to the recognition performance of a naive Bayes model or a hierarchical mixture model. The comparison is summarized in ROC style curves plotting false positive errors (the errors of accepting false examples) as a function of false negative errors (the errors of missing true examples) when we vary the classification bias for the labels. The curves show that even with such a poor underlying generative model, the combined classifier is consistently better than either of the better generative models alone.

In the second and more serious application of the combined classifier, we consider the well-known problem of recognizing remote homologies (evolutionary/structural similarities) between protein sequences[8] that have low residue identity. Considerable recent work has been done in refining hidden Markov models for this purpose [7, 2, 3, 6], and such models current achieve the best performance[9]. We use these state-of-the-art HMMs as comparison cases and also as sources for deriving the kernel function. Here we used logistic regression with the simple kernel $K_U(X_i, X_j)$, as the number of parameters in the HMMs was several thousand.

The experiment was set up as follows. We picked a particular superfamily (glycosyl-transferases) from the TIM-barrel fold in the SCOP protein struture classification [4], and left out one of the four major families in this superfamily for testing while training the HMM as well as the combined classifier on sequences corresponding to the remaining three families. The false training examples for the discriminative method came from those sequences in the same fold but not in the same superfamily. The test sequences consisted of the left-out family (true examples) and proteins outside the TIM barrel fold (false examples). This gave us a scheme for four-fold cross validation. The number of training examples varied around 100 depending on the left-out family. As the sequences among the four glycosyltransferase families are extremely different, this is a challenging discrimination problem. Figure 2 shows the recognition performance curves for the HMM and the corresponding

[8] These are variable length sequences thus rendering many discriminative methods inapplicable.

[9] See `http://cyrah.med.harvard.edu/~jong/assess_final.html`

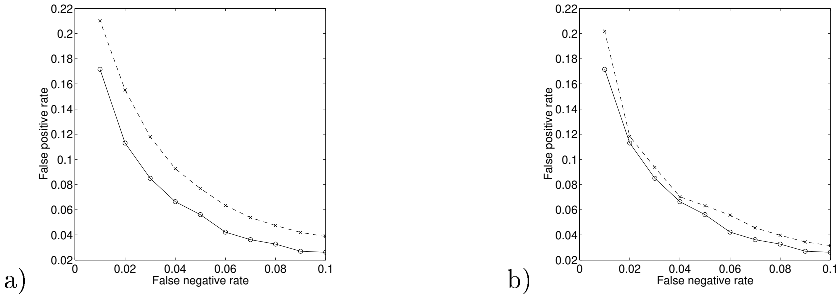

Figure 1: Comparison of classification performance between a kernel classifier from the uniform model (solid line) and a mixture model (dashed line). In a) the mixture model is a naive Bayes model and in b) it has three components in each class.

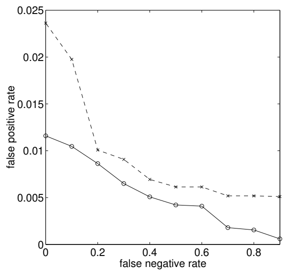

Figure 2: Comparison of homology recognition performance between a hidden Markov model (dashed line) and the corresponding kernel classifier (solid line).

kernel method, averaged over the four-way cross validation. The combined classifier yields a substantial improvement in performance over the HMM alone.

6 Discussion

The model based kernel function derived in this paper provides a generic mechanism for incorporating generative models into discriminative classifiers. For discrimination, the resulting combined classifier is guaranteed to be superior to the generative model alone with little additional computational cost. We note that the power of the new classifier arises to a large extent from the use of Fisher scores as features in place of original examples. It is possible to use these features with any classifier, e.g. a feedforward neural net, but kernel methods are most naturally suited for incorporating them.

Finally we note that while we have used classification to guide the development of the kernel function, the results are directly applicable to regression, clustering, or even interpolation problems, all of which can easily exploit metric relations among the examples defined by the Fisher kernel.

Acknowledgments

Tommi Jaakkola and David Haussler both gratefully acknowledge support from the Isaac Newton Institute for Mathematical Sciences, T.J. acknowledges support from a DOE/Sloan Foundation postdoctoral fellowship and D.H. the support of DOE grant DE-FG03-95ER62112 and NSF grant IRI-9123692.

References

[1] S.-I. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

[2] P. Baldi, Y. Chauvin, T. Hunkapillar, and M. McClure. Hidden Markov models of biological primary sequence information. *PNAS*, 91:1059–1063, 1994.

[3] S. Eddy. Multiple alignment using hidden Markov models. In C. Rallings et al., editors, *ISMB-95*, pages 114–120, Menlo Park, CA, July 1995. AAAI/MIT Press.

[4] T. Hubbard, A. Murzin, S. Brenner, and C. Chothia. Scop: a structural classification of proteins database. *NAR*, 25(1):236–9, Jan. 1997.

[5] T. S. Jaakkola and D. Haussler. Probabilistic kernel methods. 1998. Manuscript in preparation.

[6] K. Karplus, Kimmen Sjölander, C. Barrett, M. Cline, D. Haussler, R. Hughey, L. Holm, and C. Sander. Predicting protein structure using hidden Markov models. *Proteins: Structure, Function, and Genetics*, Supplement 1(1):134–139, 1997.

[7] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *JMB*, 235:1501–1531, Feb. 1994.

[8] D. J. C. MacKay. Introduction to gaussian processes. 1997. Available from `http://wol.ra.phy.cam.ac.uk/mackay/`.

[9] V. Vapnik. *The nature of statistical learning theory.* Springer-Verlag, 1995.

[10] G. Wahba. *Spline models for observational data.* CBMS-NSF Regional Conference Series in Applied Mathematics, 1990.

A A discriminative derivation of the model based kernel

Consider a class of generative probability models $P(X|\theta)$ where the label S appears as a latent variable, i.e., $P(X|\theta) = \sum_S P(X|S,\theta)P(S|\theta)$, and where the marginal probability over the label $P(S|\theta)$ is functionally independent of the parameters in the conditional probabilities $P(X|S,\theta)$. We can use this model to classify any new example based on the posterior probability $P(S|X,\theta)$ that it assigns to the labels.

Our goal now is to derive a kernel function from this model in such a way that that a kernel classifier employing this kernel would be in most cases more powerful and, in all cases, at least as powerful as the original model. We start by defining what we call a *differential extension* of the original model and show that the maximum posterior probability decisions from the original model can be obtained as a special case from the extended model. We then turn the decision rule from the extended model into a kernel classifier.

A.1 Differential extension

Consider two models, $P(X|\theta_1)$ and $P(X|\theta_{-1})$, where their parameters are differentially close to θ, i.e., $\theta_1 \approx \theta_{-1} \approx \theta$. Below we define a differential extension of the original model $P(X|\theta)$ by associating the above two models with the ± 1 values of the label S and defining the extended joint probability[10] as $P_{ext}(X, S) = P(X|\theta_S)P_{ext}(S)$ for $S \in \{\pm 1\}$. We claim that the posterior probability from this extended model can be reduced to the same classifier as the original model. To this end, we define

$$\Delta_S = P(S|\theta) - P(\bar{S}|\theta) \tag{12}$$
$$\Delta_{S|X} = P(S|X,\theta) - P(\bar{S}|X,\theta) \tag{13}$$

where $\bar{S}$ denotes the binary complement of S. We can now impose the following differential change to the marginal $P(S|\theta)$ (possible because of the functional independence assumption made earlier)

$$P(X|\theta_S) = \sum_{S'} P(X|S',\theta)\left[P(S'|\theta)\frac{1+\epsilon SS'}{1+\epsilon\Delta_S}\right] = P(X|\theta)\frac{1+\epsilon\Delta_{S|X}}{1+\epsilon\Delta_S} \tag{14}$$

where the second equality is straightforward to verify. If we also choose $P_{ext}(S) = (1+\epsilon\Delta_S)/2$, the extended posterior probability becomes

$$P_{ext}(S|X) = \frac{P(X|\theta_S)P_{ext}(S)}{P_{ext}(X)} = \frac{1}{2}(1+\epsilon\Delta_{S|X}) \tag{15}$$

where the second equality is again straightforward to obtain (details omitted here for brevity). Since $\Delta_{S|X}$ is positive for the label with the maximum posterior probability relative to $P(S|X,\theta)$, it is evident that this is also the most likely label according to $P_{ext}(S|X)$. Thus the decision rule for the extended model is the same as that of the original model.

A.2 Model based kernel from the differential extension

Let us now return to the general form of the extended posterior probability and rewrite it in terms of the logistic function

$$P_{ext}(S|X) = \sigma\left(\log\frac{P(X|\theta_S)}{P(X|\theta_{\bar{S}})} + \log\frac{P_{ext}(S)}{P_{ext}(\bar{S})}\right) \tag{16}$$

where $\sigma(\cdot)$ is the logistic function as before. Since the parameters θ_1 and θ_{-1} are differentially close to θ we can expect the Taylor expansions

$$\log P(X|\theta_S) \approx \log P(X|\theta) + (\theta_S - \theta)^T \nabla_\theta \log P(X|\theta) \tag{17}$$
$$= \log P(X|\theta) + (\theta_S - \theta)^T U_X \tag{18}$$

[10]The differential assumption applies only to the parameters θ_S; there are no constraints on $P_{ext}(S)$.

to be accurate. Here $U_X = \nabla_\theta \log P(X|\theta)$ is the Fisher score. Inserting these back into the posterior probability formulation gives

$$\begin{aligned} P_{ext}(S|X) &= \sigma\left((\theta_S - \theta_{\bar{S}})^T U_X + \log \frac{P_{ext}(S)}{P_{ext}(\bar{S})} \right) & (19) \\ &= \sigma\left(S(\theta_1 - \theta_{-1})^T U_X \right) & (20) \end{aligned}$$

where we have assumed for clarity only that $P_{ext}(S) = P_{ext}(\bar{S})$. The resulting decision rule is a logistic regression model with the parameter vector $\tilde{\theta} = (\theta_1 - \theta_{-1})$ and "examples" U_X. To complete this derivation, we assign a natural prior distribution over the new parameters $\tilde{\theta}$ that assigns low probability to values that would arise when there are large differences in the probability models corresponding to θ_1 and θ_{-1}:

$$\begin{aligned} P(\tilde{\theta}) &\propto \exp\{-KL(\theta_1 \,\|\, \theta_{-1})\} = \exp\left\{ -\int_X P(X|\theta_1) \log \frac{P(X|\theta_1)}{P(X|\theta_{-1})} \, dX \right\} & (21) \\ &\approx \exp\left\{ -\frac{1}{2}(\theta_1 - \theta_{-1})^T I \, (\theta_1 - \theta_{-1}) \right\} = \exp\left\{ -\frac{1}{2} \tilde{\theta}^T I \tilde{\theta} \right\} & (22) \end{aligned}$$

where I is the Fisher information matrix at θ. The validity of the approximation used above follows again from the differential assumption that $\theta_1 \approx \theta_{-1}$.

Finally, equation (6) above shows that in the dual variables, the logistic regression model (20) with the above prior yields the model

$$P_{ext}(S|X) = \sigma\left(S \sum_i S_i \lambda_i \, (U_{X_i}^T \, I^{-1} \, U_X) \right), \tag{23}$$

which is the general form for logistic regression using the Fisher score function as the feature vector and the Fisher kernel

$$K(X_i, X) = U_{X_i}^T \, I^{-1} \, U_X. \tag{24}$$

This shows that logistic regression with the Fisher kernel is always at least as powerful a classification method as the underlying generative model, and motivates the choice of the Fisher kernel as a natural choice from the perspective of differential discrimination with respect to the generative model.