Probabilistic kernel regression models

Tommi S. Jaakkola and David Haussler
Department of Computer Science
University of California
Santa Cruz, CA 95064
{tommi,haussler}@cse.ucsc.edu

Abstract

We introduce a class of flexible conditional probability models and techniques for classification/regression problems. Many existing methods such as generalized linear models and support vector machines are subsumed under this class. The flexibility of this class of techniques comes from the use of kernel functions as in support vector machines, and the generality from dual formulations of standard regression models.

1 Introduction

Support vector machines [10] are linear maximum margin classifiers exploiting the idea of a kernel function. A kernel function defines an embedding of examples into (high or infinite dimensional) feature vectors and allows the classification to be carried out in the feature space without ever explicitly representing it. While support vector machines are non-probabilistic classifiers they can be extended and formalized for probabilistic settings[12] (recently also [8]), which is the topic of this paper. We can also identify the new formulations with other statistical methods such as Gaussian processes[11, 13, 4].

We begin by defining the class of kernel regression techniques for binary classification, establish the connection to other methods, and provide a practical measure for assessing the generalization performance of these methods. Subsequently, we extend some of these results to sequential Bayesian estimation. Finally, we will provide a theorem governing general kernel reformulation of probabilistic regression models.

2 Binary classification

We start by considering Gaussian process classifiers[13, 4] that are fully Bayesian methods. To this end, define a set of zero mean jointly Gaussian random variables $\{Z_i\}$, one corresponding to each example X_i to be classified. Assume that the covariance $Cov(Z_i, Z_j)$ between any two such variables is given by a kernel function $K(X_i, X_j)$ of the corresponding examples (we need the kernel function to be strictly positive definite in this case). Assume further that the binary ± 1 labels $\{S_i\}$ are generated with probabilities $P(S_i|Z_i) = \sigma(S_i Z_i)$ where, for example, $\sigma(z) = (1 + e^{-z})^{-1}$ is the logistic function. The example vectors $\{X_i\}$ thus specify the Gaussian variables $\{Z_i\}$ that are subsequently passed through transfer functions to yield probabilities for the labels. Similar Z's (and hence similar labels) are assigned to input vectors that are "close" in the sense of the kernel. Given now a training set of labels $\{S_i\}$ and example vectors $\{X_i\}$ we can, in principle, compute the posterior distribution of the latent Gaussian variables $\{Z_i\}$ and use it in assigning labels for yet unknown examples; the Gaussian variable Z_t corresponding to the new example X_t is correlated with $\{Z_i\}$ constrained by the fixed training labels. The calculations involved in this procedure are, however, typically infeasible.

Instead of trying to maintain a full posterior distribution over the latent Gaussian variables, we may settle for the MAP configuration $\{\hat{Z}_i\}$, and assign the label to a new example according to the probability $P(S_t|\hat{Z}_t) = \sigma(S_t \hat{Z}_t)$. Definition 1 below gives a generic formulation of this procedure in terms of dual parameters. The dual parameters arise from Legendre transformations of concave functions (see e.g. [7]); the concave functions in this case are the classification losses $\log P(S_t|\hat{Z}_t)$. We consider such transformations in more detail later in the paper.

Definition 1 *We define a kernel regression classifier to be any classification technique with the following properties: 1) given any example vector X, the method predicts the (maximum probability) label $\hat{S}$ for X ac-*

cording to the rule

$$\hat{S} = \text{sign}\left(\sum_{i=1}^{T} \lambda_i S_i K(X, X_i)\right) \quad (1)$$

where $(S_1, X_1), \ldots, (S_T, X_T)$ are labeled training examples, the λ_i are non-negative coefficients, and the kernel function $K(X_i, X_j)$ is positive (semi-) definite. 2) The coefficients $\{\lambda_t\}$ weighting the training examples in the classification rule are obtained by maximizing

$$J(\lambda) = -\frac{1}{2}\sum_{i,j} \lambda_i \lambda_j S_i S_j K(X_i, X_j) + \sum_i F(\lambda_i) \quad (2)$$

subject to $0 \leq \lambda_i \leq 1$, where the potential function *$F(\cdot)$ is continuous and concave (strictly concave whenever the kernel is only positive semi-definite).*

The assumptions of positive (semi-)definite kernel function and (strictly) concave and continuous potential functions $F(\cdot)$ are introduced primarily to ensure that the solution to the maximization problem is unique. In practice this solution can be achieved monotonically by successively updating each individual coefficient λ_i from

$$\frac{\partial}{\partial \lambda_i} J(\lambda) = -\sum_j \lambda_j S_i S_j K(X_i, X_j) + \frac{\partial}{\partial \lambda_i} F(\lambda_i) = 0 \quad (3)$$

while holding the other λ_j, for $j \neq i$, fixed. The solutions to these one dimensional equations are relatively easy to find for any kernel method in our class. The optimal solution is characterized by these fixed point equations, reminiscent of mean field equations.

Let us now consider a few examples to gain more insight into the nature and generality of this class. Support vector machines[1], for example, can be seen as realizations of this class simply by setting $F(\lambda_t) = \lambda_t$ (see e.g. [10]). Generalized linear models[6] can be also seen as members of this class. For example, consider a logistic regression model, where the probabilities for the labels are given by $P(S_i|X_i, \theta) = \sigma(S_i \theta^T X)$ and the prior distribution $P(\theta)$ over the parameters is a zero mean Gaussian. With each input vector X_i we can associate a new variable $Z_i = \theta^T X_i$ that defines the conditional probability for the label S_i through $P(S_i|X_i, \theta) = \sigma(S_i Z_i)$ as above. Under the Gaussian prior assumption, these new variables $\{Z_i\}$ are jointly Gaussian random variables with a covariance matrix given by

$$Cov(Z_i, Z_j) = E\{\,(\theta^T X_i)(\theta^T X_j)\,\} = X_i^T \Sigma X_j \quad (4)$$

where Σ is the prior covariance of θ. The logistic regression problem is thus equivalent to a particular Gaussian process classifier. Consequently, MAP estimation of the parameters in the logistic regression models corresponds exactly to the MAP Gaussian process formulation (Definition 1). The relation between MAP estimation and Definition 1 is presented more generally later in the paper (Theorem 1). We note here only that the potential function $F(\lambda)$ in the logistic regression case is the binary entropy function $F(\lambda) = -\lambda \log(\lambda) - (1-\lambda)\log(1-\lambda)$ (see Appendix E) and the kernel function is the covariance function $Cov(Z_i, Z_j)$ given by Eq. (4).

2.1 The kernel function

Here we discuss a few properties of the kernel function as given in Eq. (4). First, interpreting the kernel function as a covariance function for a Gaussian process classifier suggests treating it as a similarity function. In this sense, examples are similar when their associated labels would be a priori (positively) correlated. A simple inner product is not necessarily a good measure of similarity since, for example, it is possible for an example to be more similar to another example than to itself. Typically, however, the kernel function is not a simple inner product between the examples but an inner product between feature vectors corresponding to the examples. For example, in Eq. (4) the feature vectors are $\phi_X = \Sigma^{\frac{1}{2}} X$. Any valid kernel function can be reduced to a simple inner product between (possibly infinite dimensional) feature vectors [10, 11]. When the feature mapping is non-linear, the kernel can define a reasonable similarity measure in the original example space even though this property doesn't hold in the feature space.

In going from logistic regression models to Gaussian process classifiers the prior covariance matrix Σ over the original parameters θ plays a special role in specifying the inner product in Eq. (4). In other words, the prior covariance matrix directly changes the metric in the example space. This metric is, however, the inverse of what is natural in the parameter space, i.e., Σ^{-1}. This inverse relation follows from a more general property of Riemannian metrics in dual coordinate systems.

If we change the kernel function, our assumptions concerning the examples (similarity, metric properties)

[1] Note that in support vector machines a bias term is added explicitly into the classification rule and treated separately in the optimization problem. In our formulation the bias term is realized indirectly through an additive constant in the kernel, where the magnitude of this constant specifies the prior variance over the bias term. Put into our setting, support vector machines assume a flat prior and consequently the two definitions agree in so far as the constant term in the kernel is appropriately large.

will change. This suggests that the modeling effort in these classifiers should go into finding an appropriate kernel function. We can, for example, derive kernels from generative probability models [3] or directly encode invariances into the kernel function [1].

2.2 A measure of generalization error

Definition 1 provides us with a large class of techniques with relatively few restrictions on e.g. the choice of the kernel function. To compensate this flexibility we must provide means for assessing their generalization performance in order to be able to limit the complexity of the final classifier to an appropriate level. Our emphasis here is on practical measures.

Support vector machines attain sparse solutions in the sense that most of the coefficients λ_i are set to zero as a result of the optimization. This computationally attractive property also yields a direct assessment of generalization [10]: the expected ratio of the number of non-zero coefficients to the number of training examples bounds the true generalization error. The applicability of this measure is limited to support vector machines, however, since the probabilistic classifiers generally do not attain sparse solutions (making the sparsity measure vacuous). The lemma below provides a more general cross-validation measure that applies to all kernel classifiers under Definition 1:

Lemma 1 *For any training set $\mathcal{D} = \{S_t, X_t\}_{t=1}^T$ of examples and labels and for any kernel regression classifier from Definition 1 the leave-one-out cross-validation error estimate of the classifier is bounded by*

$$\frac{1}{T}\sum_{t=1}^{T} \text{step}\left(-S_t \sum_{i \neq t} \lambda_i S_i K(X, X_i) \right) \qquad (5)$$

where $\{\lambda_t\}$ are the coefficients optimized in the presense of all the training examples.

The step functions in the lemma count the number of times the sign of the training label S_t disagrees with the sign of the prediction based on the other examples. If we include the missing t^{th} terms in the predictions, the error estimate would reduce to the training error (cf. the prediction rule in Definition 1). The cross-validation error bound is thus no more costly to evaluate than the training error and obviously requires no retraining. As for the accuracy of this bound we note that in case of support vector machines, it can be shown that the result above provides a slightly better estimate than the sparsity bound[2]. The proof of the lemma is given in Appendix A.

[2] The sparsity bound can be, in principle, defined in terms of "essential" support vectors rather than just those with non-zero coefficients. This would improve the estimate but would also make it much more difficult to evaluate in practice.

3 Bayesian formulation

In the above MAP formulation the kernel function itself remains fixed, regardless of the nature or the number of training examples. This is in contrast with a full Bayesian approach where the kernel function would have to be modified based on observations. More precisely, in the above formulation it is the prior distribution over the parameters θ that specifies the (simple) inner product between the examples; in a Bayesian setting, roughly speaking, this inner product would be defined in terms of the posterior distribution. While the full Bayesian approach is unfortunately not feasible in most cases, it is nevertheless possible to employ approximate methods for updating the kernel function through observations.

Several approaches have already been proposed for this purpose, including the use of Laplace approximation in the context of multi-class regression [13] and the use of variational methods [5]. Our approach is rather complementary in the sense that we provide a recursive variational approach that avoids the need for simultaneously optimizing a large number of variational parameters as discussed in [5].

3.1 Bayesian logistic regression

Here we consider a Bayesian formulation of the logistic regression models. We start by briefly reviewing the variational approximation technique [2] that enables us to estimate the posterior distribution over the parameters in these models. We subsequently extend this approximate solution for use with kernel functions.

In Bayesian estimation we can, in principle, update the parameter distribution sequentially, one example at a time:

$$\begin{aligned} P(\theta|\mathcal{D}_t) &\propto P(S_t|X_t, \theta) P(\theta|\mathcal{D}_{t-1}) &\quad (6)\\ &= \sigma(\, S_t \theta^T X_t \,)\, P(\theta|\mathcal{D}_{t-1}) &\quad (7) \end{aligned}$$

where $\mathcal{D}_t = \{(S_1, X_1), \ldots, (S_t, X_t)\}$ is the set of examples observed up to the time t. We constrain the above general formulation a bit by assuming that the prior distribution $P(\theta) = P(\theta|\mathcal{D}_0)$ over the parameters is a multivariate Gaussian with possibly arbitrary covariance structure. While such assumption does not by itself make the sequential updating feasible in terms being able to represent the true posterior distribution, it nevertheless opens the way to a closed form approximate solution. To this end we employ a variational transformation of the logistic function as given by[3].

$$\sigma(z) \geq \sigma(\xi)\,\exp\left\{(z-\xi)/2+\lambda(\xi)(z^2-\xi^2)\right\} \quad (8)$$
$$\equiv \sigma_\xi(z) \quad (9)$$

where ξ is an adjustable parameter known as the variational parameter. Inserting the approximation $\sigma_\xi(z)$ back into the sequential update equation Eq. (7) we obtain

$$P(\theta|\mathcal{D}_t) \propto \sigma_\xi(\,S_t\theta^T X_t\,)\,P(\theta|\mathcal{D}_{t-1}) \quad (10)$$

Since the transformed logistic function $\sigma_\xi(\cdot)$ is a quadratic function of its argument in the exponent, it follows that any Gaussian prior $P(\theta|\mathcal{D}_{t-1})$ will result in a Gaussian posterior in this approximation. The mean and the covariance of this Gaussian can be related to the mean and the covariance of the prior through

$$\Sigma_t = \Sigma_{t-1} - c_t\,\Sigma_{t-1}\,X_t\,X_t^T\,\Sigma_{t-1} \quad (11)$$
$$\mu_t = \Sigma_t(\,\Sigma_{t-1}^{-1}\mu_{t-1} + \frac{1}{2}S_t X_t\,) \quad (12)$$

where the subindex t referes to the set of examples $\mathcal{D}_t = \{(S_1, X_1), \ldots, (S_t, X_t)\}$ observed so far. The variational parameter defines the extent to which the covariance matrix is updated, i.e., it defines c_t:

$$c_t = \frac{2\lambda_t}{1+2\lambda_t X_t^T \Sigma_{t-1} X_t} \quad (13)$$

where $\lambda_t = \tanh(\xi_t/2)/(4\xi_t)$. We would like to set the variational parameter ξ_t so as to improve the accuracy of the approximation. A suitable error measure for the approximation can be derived from the fact that the variational transformation introduces a *lower* bound. The approximation in fact yields a lower bound on the likelihood of the conditional observation $S_t|X_t$

$$P(S_t|X_t,\mathcal{D}_{t-1}) = \int \sigma(\,S_t\theta^T X_t\,)P(\theta|\mathcal{D}_{t-1})d\theta$$
$$\geq \int \sigma_\xi(\,S_t\theta^T X_t\,)\,P(\theta|\mathcal{D}_{t-1})d\theta \quad (14)$$

where the last integral can be computed in closed form. The maximization of the bound yields a fixed point equation for ξ_t [2]:

$$\xi_t^2 = X_t^T\Sigma_t X_t + (\mu_t^T X_t)^2 \quad (15)$$

that can be solved iteratively (note that both Σ_t and μ_t depend on ξ_t through the equations above).

[3] Other approximations are possible as well such as the Laplace approximation [9].

3.2 Kernel extension

In order to be able to employ kernels in the context of these Bayesian calculations we have to reduce all the calculations with the input examples X_k to appropriate inner products. Such inner products can be then replaced with arbitrary positive semi-definite kernels. As before, we define the a priori inner product as $X_k^T\Sigma_0 X_{k'}$ which is valid since the prior covariance is positive definite. For simplicity, we assume that the mean of the Gaussian prior over the parameters is zero. Consequently, it remains to show that the sequential updating scheme can be carried out by only referring to the value and not the form of the inner products $K(X_k, X_{k'}) = X_k^T\Sigma_0 X_{k'}$.

We adopt the following compact representations of the posterior mean and the time dependent kernel:

$$K_t(k,k') = X_k^T\Sigma_t X_{k'} \quad (16)$$
$$M_t(k) = \mu_t^T X_k \quad (17)$$

It will become clear later that it is necessary to consider only predictive quantities, i.e. those for which $k, k' > t$. We would like to now express the update equations for the mean and the covariance in terms of these new quantities. Consider first Eq. (11), the covariance update formula. Prior and post multiplying the update formula with X_k^T and $X_{k'}$, respectively, and using the definition for $K_t(k,k')$ given above, we obtain:

$$K_t(k,k') = K_{t-1}(k,k') - c_t\,K_{t-1}(k,t)K_{t-1}(t,k') \quad (18)$$

Thus the $K_t(k,k')$ satisfy a simple recurrence relation that connects them back to the a priori kernel $K_0(k,k') = X_k^T\Sigma_0 X_{k'}$. We can also derive a recurrence relation for $M_t(k)$ in a similar way (see Appendix B) giving

$$M_t(k) = M_{t-1}(k) + c_t\left[\frac{S_t}{4\lambda_t} - M_{t-1}(t)\right]K_{t-1}(t,k) \quad (19)$$

Since $M_0(k) = 0$ by assumption (i.e. the prior mean is zero), the values $M_t(k)$ can be rooted in the kernels and the observed labels S_t.

Finally, both K_t and M_t iterations make use of the coefficients c_t,

$$c_t = \frac{2\lambda_t}{1+2\lambda_t K_{t-1}(t,t)} \quad (20)$$

and $\lambda_t = \tanh(\xi_t/2)/(4\xi_t)$ which need to be specified. In other words, we need to be able to optimize the variational parameter ξ_t in terms of K_{t-1}, M_{t-1}, and

S_t alone in order to preserve the recurrence relations. Starting from the fixed point equation Eq. (15) we get (the details can be found in Appendix C)

$$\begin{aligned} \xi_t^2 &= X_t^T \Sigma_t X_t + (\mu_t^T X_t)^2 && (21) \\ &= K_t(t,t) + M_t(t)^2 && (22) \\ &= \left(\frac{c_t}{2\lambda_t}\right) K_{t-1}(t,t) + \\ & \left(\frac{c_t}{2\lambda_t}\right)^2 \left(M_{t-1}(t) + \frac{1}{2} S_t K_{t-1}(t,t)\right)^2 && (23) \end{aligned}$$

Note that ξ_t appears on the right hand side only in the expressions $c_t/(2\lambda_t)$. It follows that to optimize ξ_t in the process of absorbing a new observation we only need to know $M_{t-1}(t)$, $K_{t-1}(t,t)$ and S_t. How these values can be computed and stored efficiently is illustrated in the next section.

3.3 Efficient implementation

Due to the form of the dependencies in the recurrence relations for K_t and M_t, we can carry out the computations in the sequential estimation procedure efficiently and compactly. To show this we proceed inductively. Assume therefore that we have a lower diagonal matrix $\mathcal{K}_t$ and a vector $\mathcal{M}_t$ of the form

$$\begin{aligned} \mathcal{K}_t &= \begin{bmatrix} K_0(1,1) & & & \\ K_0(2,1) & K_1(2,2) & & \\ K_0(3,1) & K_1(3,2) & & \\ \vdots & \vdots & \ldots & \\ K_0(t,1) & K_1(t,2) & \ldots & K_{t-1}(t,t) \end{bmatrix} \\ \mathcal{M}_t &= \begin{bmatrix} M_0(1) & M_1(2) & \ldots & M_{t-1}(t) \end{bmatrix}^T \quad (24) \end{aligned}$$

which we have constructed from the already observed examples up to (S_t, X_t). To absorb a new training example (S_{t+1}, X_{t+1}) or to evaluate the predictive probability of S_{t+1} given X_{t+1}, we need to be able to optimize the variational parameter ξ_{t+1} associated with this example. Consider therefore the fixed point equation (23). The required quantities are $K_t(t+1,t+1)$ and $M_t(t+1)$ corresponding to the next diagonal component of the $\mathcal{K}$ matrix and the next component of the $\mathcal{M}$ vector, respectively. We start computing these quantities by filling in the next row of $\mathcal{K}$ with the kernels $K_0(t+1,k)$, $k = \{1,\ldots,t+1\}$. Consequently, we can apply the recurrence relation Eq. (18) to replace these values (except the first one) with $K_1(t+1,k)$, $k = \{2,\ldots,t+1\}$. Note, however, that we must replace these values in the reverse order, from $k = t+1$ down to $k = 2$, due to the dependence structure in the recurrence relation. Following in this manner we can fill in $K_{t'}(t+1,k)$ for $k = \{t'+1,\ldots,t+1\}$ and ultimately get $K_t(t+1,t+1)$ in time $\mathcal{O}(t^2/2)$. $M_t(t+1)$ can be computed even more directly by starting from $M_0(t+1) = 0$ and using the recurrence relation Eq. (19) to give $M_1(t+1), M_2(t+1), \ldots, M_t(t+1)$ in time $\mathcal{O}(t)$.

4 Generic kernel regression

Definition 1 and some of the discussion in the previous sections can be generalized to a multi-class or to a continuous response setting.

Theorem 1 *Let $P(Y|X,\theta)$ be a conditional probability model over a discrete or continuous variable Y, where X is a finite real vector of inputs and $\theta = \{\theta_1, \theta_2, \ldots, \theta_m\}$ denotes the parameters. Assume that 1) $P(Y|X,\theta) = P(Y|Z_1, Z_2, \ldots, Z_m)$ where $Z_i = \theta_i^T X$; 2) For all values y of Y, $\log P(y|Z_1, Z_2, \ldots, Z_m)$ is a jointly concave continuously differentiable function of $Z_1, Z_2, \ldots, Z_m$; 3) The prior distribution over the parameter vectors $\{\theta_i\}$ is a zero mean multivariate Gaussian with a block diagonal covariance matrix $\Sigma = diag(\Sigma_1, \Sigma_2, \ldots, \Sigma_m)$. Then, given a training set $\mathcal{D} = \{Y_t, X_t\}_{t=1}^T$, the conditional probability model corresponding to the maximum a posteriori (MAP) set of parameters has the form*

$$P(Y|X, \theta_{MAP}) = P(Y|\hat{Z}_1, \hat{Z}_2, \ldots, \hat{Z}_m) \quad (25)$$

where $\hat{Z}_i = \sum_{t=1}^T \lambda_{t,i} (X_t^T \Sigma_i X)$, the coefficients $\lambda = \{\lambda_{t,i}\}$ attain the unique maximum value of

$$\begin{aligned} J(\lambda) = & -\frac{1}{2} \sum_{i,t,t'} \lambda_{t,i} \lambda_{t',i} (X_t^T \Sigma_i X_{t'}) \\ & + \sum_{t=1}^T F_t(\lambda_{t,1}, \lambda_{t,2}, \ldots, \lambda_{t,m}), \quad (26) \end{aligned}$$

and the potential functions $F_t(\lambda_{t,1}, \lambda_{t,2}, \ldots, \lambda_{t,m})$ are the Legendre transformations of the classification loss functions $\log P(Y_t|Z_1, Z_2, \ldots, Z_m)$:

$$\begin{aligned} & F_t(\lambda_{t,1}, \lambda_{t,2}, \ldots, \lambda_{t,m}) = \\ & \quad \min_{Z_1, \ldots, Z_m} \left\{ \sum_i \lambda_{t,i} Z_i - \log P(Y_t|Z_1, Z_2, \ldots, Z_m) \right\} \\ & \quad (27) \end{aligned}$$

The rather strong assumption of continuous differentiability can be easily relaxed to piece-wise differentiability[4]. The proof is given in Appendix D.

The Legendre transformations in the theorem are easy to compute in typical cases, e.g., when the conditional

[4]Piece-wise continuously differentiable functions can be obtained as limits of continosly differentiable functions.

probabilities $P(Y|Z_1, Z_2, \ldots, Z_m)$ are softmax functions (i.e., $e^{Z_Y} / \sum_y e^{Z_y}$). We have listed a few examples in Appendix E. The inner products $(X_t^T \Sigma_i X)$ appearing in the dual formulation can be replaced with any valid kernel function $K_i(X_t, X)$ such as the Gaussian kernel[5]. Note that Definition 1 makes stronger assumptions about the Legendre transformations and/or the positive definiteness of the kernel function so as to end up with a unique solution in terms of the new parameters λ. For the purpose of prediction the possible non-uniqueness is immaterial since the resulting predictive distribution remains unique. Concavity of the objective function also assures us that the solution is relatively easy to find in all cases.

5 Discussion

In any classification/regression problem it is necessary to select an appropriate representation of examples as well as the model and parameter estimation method. In this paper we have focused on the latter, deriving a generic class of probabilistic regression models and a parameter estimation technique that can make use of arbitrary kernel functions. This allows greater flexibility in specifying probabilistic regression models of various complexity levels without fear of local minima. We can also obtain quick assessments of their generalization performance. The issue concerning the choice of the kernel function or, equivalently, the representation of examples, has been addressed elsewhere [3, 1] and

References

[1] Burges C. (1998). Geometry and invariance in kernel based methods. To appear in *Advances in kernel methods – Support vector learning.* MIT press.

[2] T. Jaakkola and M. Jordan (1996). A variational approach to Bayesian logistic regression problems and their extensions. In *Proceedings of the sixth international workshop on artificial intelligence and statistics.*

[3] Jaakkola T. and Haussler D. (1998). Exploiting generative models in discriminative classifiers. Available at `http://www.cse.ucsc.edu/research/ml/publications.html`.

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

[5]The Gaussian kernel is given by $K_i(X_t, X) = e^{-\beta(X_t-X)^T\Sigma_i(X_t-X)}$.

[5] Gibbs M. MacKay D. (1997). Variational Gaussian process classifiers. Draft manuscript, available at `ftp://wol.ra.phy.cam.ac.uk/mackay`.

[6] McCullagh P. and Nelder J. (1983). *Generalized linear models.* London: Chapman and Hall.

[7] Rockafellar R. (1970). *Convex Analysis.* Princeton Univ. Press.

[8] Smola A., Schlkopf B., Mlller K., (1998). General Cost Functions for Support Vector Regression. *ACNN'98, Australian Congress on Neural Networks.*

[9] D. Spiegelhalter and S. Lauritzen (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks* **20**: 579-605.

[10] Vapnik V. (1995). *The nature of statistical learning theory.* Springer-Verlag.

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

[12] Wahba, G. (1997). Support Vector Machines, Reproducing Kernel Hilbert Spaces and the Randomized GACV. University of Wisconsin - Madison technical report TR984rr.

[13] Williams C. and Barber D. (1997). Bayesian Classification with Gaussian Processes. Manuscript in preparation.

A Proof of the cross-validation bound

Consider the case where the t^{th} example is removed from the training set $\mathcal{D}$. In this case we would optimize the remaining coefficients $\Lambda_{\mathcal{D}-t} = \{\lambda_i\}_{i \neq t}$ by maximizing

$$J_{\mathcal{D}-t}(\Lambda_{\mathcal{D}-t}) = -\frac{1}{2} \sum_{i,j \neq t} \lambda_i \lambda_j S_i S_j K(X_i, X_j) + \sum_{i \neq t} F(\lambda_i)$$

By our assumptions, the solution is unique and we denote it by $\Lambda^t_{\mathcal{D}-t}$. Consider now adding the t^{th} example back into the training set; the optimal setting of the coefficients $\Lambda_{\mathcal{D}-t}$ will naturally change as they need to be optimized jointly with λ_t. To assess the effect of adding the t^{th} example, we perform the joint optimization as follows. First, we fix λ_t to the value that would have resulted from the joint optimization, call it λ_t^*. The remaining coefficients $\Lambda_{\mathcal{D}-t}$ can be obtained from a reduced objective function where all the terms depedending solely on λ_t are omitted. Thus when the t^{th}

example is included, the remaining coefficients $\Lambda_{\mathcal{D}-t}$ are obtained by maximizing

$$J_{\mathcal{D}}(\Lambda_{\mathcal{D}-t}) = J_{\mathcal{D}-t}(\Lambda_{\mathcal{D}-t}) - \lambda_t^* S_t \sum_{i \neq t} \lambda_i S_i K(X_t, X_i) \quad (28)$$

Let $\Lambda_{\mathcal{D}-t}^*$ be the maximizing coefficients. Clearly

$$J_{\mathcal{D}}(\Lambda_{\mathcal{D}-t}^*) \geq J_{\mathcal{D}}(\Lambda_{\mathcal{D}-t}^t) \quad (29)$$

Expanding each side according to eq. (28) and rearranging terms we get

$$\begin{aligned} &\lambda_t^* S_t \sum_{i \neq t} \lambda_i^t S_i K(X_t, X_i) \\ \geq\ & \lambda_t^* S_t \sum_{i \neq t} \lambda_i^* S_i K(X_t, X_i) \\ & + J_{\mathcal{D}-t}(\Lambda_{\mathcal{D}-t}^t) - J_{\mathcal{D}-t}(\Lambda_{\mathcal{D}-t}^*) \\ \geq\ & \lambda_t^* S_t \sum_{i \neq t} \lambda_i^* S_i K(X_t, X_i) \quad (30) \end{aligned}$$

where we have used the fact that $J_{\mathcal{D}-t}(\Lambda_{\mathcal{D}-t}^t) \geq J_{\mathcal{D}-t}(\Lambda_{\mathcal{D}-t}^*)$ as the coefficients $\Lambda_{\mathcal{D}-t}^t$ maximize $J_{\mathcal{D}-t}(\cdot)$. Whenever $\lambda_t^* > 0$, we can divide the first and the last term of Eq. (30) by λ_t^* and get the desired result: the sign of the first term indicates the correctness of the cross-validation prediction (positive is correct); its lower bound, the last term, is the one that appears in the lemma and uses only the coefficients optimized in the presense of all the training examples.

Note finally that when $\lambda_t^* = 0$ the t^{th} example is not used in the classifier and the lemma holds trivially.

B Recurrence relation for $M_t(k)$

Let us start by simplifying the posterior mean update:

$$\begin{aligned} \mu_t &= (\Sigma_{t-1} - c_t \Sigma_{t-1} X_t X_t^T \Sigma_{t-1}) \times \\ & \quad \times (\Sigma_{t-1}^{-1} \mu_{t-1} + \frac{1}{2} S_t X_t) \quad (31) \\ &= \mu_{t-1} - c_t (X_t^T \mu_{t-1}) \Sigma_{t-1} X_t \\ & \quad + \frac{1}{2} S_t (1 - c_t X_t^T \Sigma_{t-1} X_t) \Sigma_{t-1} X_t \quad (32) \\ &= \mu_{t-1} - c_t (X_t^T \mu_{t-1}) \Sigma_{t-1} X_t \\ & \quad + \frac{1}{2} S_t \frac{c_t}{2\lambda_t} \Sigma_{t-1} X_t \quad (33) \\ &= \mu_{t-1} + c_t \left[\frac{S_t}{4\lambda_t} - X_t^T \mu_{t-1} \right] \Sigma_{t-1} X_t \quad (34) \end{aligned}$$

where we have used the fact that $(1 - c_t X_t^T \Sigma_{t-1} X_t) = c_t/(2\lambda_t)$ (see the definition of c_t given in the text). In terms of $M_t(k) = \mu_t^T X_k$, we can write the above result as

$$M_t(k) = M_{t-1}(k) + c_t \left[\frac{S_t}{4\lambda_t} - M_{t-1}(t) \right] K_{t-1}(t, k) \quad (35)$$

C Fixed point equation for ξ_t

The objective here is to transform the fixed point equation

$$\xi_t^2 = K_t(t,t) + M_t(t)^2 \quad (36)$$

into the form that explicates the dependence of the right hand side on ξ_t. Applying the recurrence relation for $K_t(t,t)$ we find

$$\begin{aligned} K_t(t,t) &= K_{t-1}(t,t) - c_t K_{t-1}(t,t)^2 \quad (37) \\ &= K_{t-1}(t,t)(1 - c_t K_{t-1}(t,t)) \quad (38) \\ &= \frac{c_t}{2\lambda_t} K_{t-1}(t,t) \quad (39) \end{aligned}$$

where $K_{t-1}(t,t)$ is independent of ξ_t (depends only on $\xi_1, \ldots, \xi_{t-1}$). Similarly, we expand $M_t(t)$:

$$\begin{aligned} M_t(t) &= M_{t-1}(t) + c_t \left[\frac{S_t}{4\lambda_t} - M_{t-1}(t) \right] K_{t-1}(t,t) \quad (40) \\ &= M_{t-1}(t)(1 - c_t K_{t-1}(t,t)) \\ & \quad + \frac{c_t S_t}{4\lambda_t} K_{t-1}(t,t) \quad (41) \\ &= \frac{c_t}{2\lambda_t} M_{t-1}(t) + \frac{c_t}{2\lambda_t} \frac{S_t}{2} K_{t-1}(t,t) \quad (42) \\ &= \frac{c_t}{2\lambda_t} \left(M_{t-1}(t) + \frac{1}{2} S_t K_{t-1}(t,t) \right) \quad (43) \end{aligned}$$

where the only dependence on ξ_t is now in $c_t/(2\lambda_t)$. Combining these two results gives

$$\begin{aligned} \xi_t^2 &= \left(\frac{c_t}{2\lambda_t} \right) K_{t-1}(t,t) + \\ & \left(\frac{c_t}{2\lambda_t} \right)^2 \left(M_{t-1}(t) + \frac{1}{2} S_t K_{t-1}(t,t) \right)^2 \quad (44) \end{aligned}$$

D Proof of Theorem 1

Given a training set of examples $\mathcal{D} = \{Y_t, X_t\}_{t=1}^T$, the MAP parameter solution is obtained by maximizing the following penalized likelihood function

$$J(\theta) = \sum_{t=1}^{T} \log P(Y_t | Z_t) - \frac{1}{2} \sum_{i=1}^{m} \theta_i^T \Sigma_i^{-1} \theta_i \quad (45)$$

where the first term is the log-probability of the observed labels and the second comes from the log of the block-diagonal Gaussian prior distribution. We have omitted the terms that do not depend on the parameters θ and overloaded our previous notation in the sense that Z_t now refers to the vector $\{Z_{t,1}, \ldots, Z_{t,m}\}$. The solution θ_{MAP} is unique since $J(\theta)$ is strictly concave in θ (owing to the log-prior term).

Now, by our assumptions $\log P(Y_t|Z_t)$ is jointly concave continuously differentiable function of Z_t and thus by convex duality (see e.g. [7]) we get: there exists a function F_t with the same properties such that

$$\log P(Y_t|Z_t) = \min_{\lambda_t} \left\{ \sum_{i=1}^{m} \lambda_{t,i} Z_{t,i} - F_t(\lambda_t) \right\} \quad (46)$$

$$F_t(\lambda_t) = \min_{Z_t} \left\{ \sum_{i=1}^{m} \lambda_{t,i} Z_{t,i} - \log P(Y_t|Z_t) \right\} \quad (47)$$

where $\lambda_t = \{\lambda_{t,1}, \ldots, \lambda_{t,m}\}$. These transformations are also known as Legendre transformations and the function F_t is known as the dual or conjugate function. Note that the conjugate function F_t of $\log P(Y_t|Z_t)$ is in general different for each distinct Y_t, hence the additional subindex.

Let us now introduce these transformations into the objective function $J(\theta)$ and define $J(\theta, \lambda)$ as

$$J(\theta,\lambda) = \sum_{t=1}^{T} \left[\sum_i \lambda_{t,i} Z_{t,i} - F_t(\lambda_t) \right] - \frac{1}{2} \sum_{i=1}^{m} \theta_i^T \Sigma_i^{-1} \theta_i \quad (48)$$

where we have dropped the associated mimizations with respect to the λ coefficients. Clearly, $J(\theta) = \min_\lambda J(\theta, \lambda)$. The lemma below establishes the connection to Theorem 1:

Lemma 2 *The objective function in Theorem 1 is concave and is given by the negative of*

$$J(\lambda) = \max_\theta J(\theta, \lambda) \quad (49)$$

This result implies that maximizing the objective function in Theorem 1 is equivalent to computing $\min_\lambda J(\lambda)$ in our notation here.

Proof: Recall that $Z_{t,i} = \theta_i^T X_t$ which implies that for any fixed setting of λ, $J(\theta, \lambda)$ is a quadratic function of the parameters θ. We can therefore solve for the maximizing θ:

$$\theta_i^\lambda = \sum_t \lambda_{t,i} \Sigma_i X_t \quad (50)$$

and substitute this back into $J(\theta, \lambda)$ giving

$$J(\lambda) = \frac{1}{2} \sum_{i,t,t'} \lambda_{t,i} \lambda_{t',i} (X_t^T \Sigma_i X_{t'}) - \sum_t F_t(\lambda_t) \quad (51)$$

which is indeed the negative of the objective function appearing in the theorem as desired. It remains to show that $J(\lambda)$ is convex (or that $-J(\lambda)$ is concave). Note first that the conjugate functions F_t are concave and thus all $-F_t$ terms are convex. Also the first term in Eq. (51) corresponds to

$$\frac{1}{2} \sum_i (\theta_i^\lambda)^T \Sigma_i^{-1} (\theta_i^\lambda) \quad (52)$$

where each θ_i^λ is a linear function of λ, and hence the above term is also convex. Without additional assumptions $J(\lambda)$ may not be strictly convex and thus the solution in terms of λ may not be unique. $\min_\lambda J(\lambda)$ and the associated θ^λ remain unique, however. □

Since, by definition, maximizing $J(\theta)$ means evaluating $\max_\theta \min_\lambda J(\theta, \lambda)$ and because we have just shown that $\min_\lambda \max_\theta J(\theta, \lambda)$ corresponds to maximizing the objective function in Theorem 1, it remains to show that "$\max_\theta \min_\lambda = \min_\lambda \max_\theta$" for $J(\theta, \lambda)$. We state this as a lemma:

Lemma 3 *For $J(\theta, \lambda)$ given by Eq. (48)*

$$\max_\theta \min_\lambda J(\theta, \lambda) = \min_\lambda \max_\theta J(\theta, \lambda) \quad (53)$$

Proof: Let θ^* be the unique maximum of $J(\theta)$ (the left hand side above). θ^* is also finite[6]. Since in general for any fixed λ^*

$$\max_\theta \min_\lambda J(\theta, \lambda) \leq \min_\lambda \max_\theta J(\theta, \lambda) \quad (54)$$

$$\leq \max_\theta J(\theta, \lambda^*) \quad (55)$$

it suffices to show that there exists λ^* such that $J(\theta^*) = \max_\theta J(\theta, \lambda^*)$.

To this end, the finiteness of θ^* together with the continuous differentiability assumption guarantees that

$$\lambda_t^* = \nabla_{Z_t} \log P(Y_t|Z_t) \mid_{\theta=\theta^*} \quad (56)$$

exists for all t. It can be shown that these are also the minimizing coefficients in our Legendre transformations. Consequently, the minimum is attained:

$$J(\theta^*) = \min_\lambda J(\theta^*, \lambda) = J(\theta^*, \lambda^*) \quad (57)$$

At this minimum $\nabla_\lambda J(\theta^*, \lambda) \mid_{\lambda=\lambda^*}$ vanishes and thus

$$\nabla_\theta J(\theta) \mid_{\theta=\theta^*} = \nabla_\theta J(\theta, \lambda^*) \mid_{\theta=\theta^*} = 0 \quad (58)$$

The last equality gives sufficient guarantees that for our choice of λ^*

$$\max_\theta J(\theta, \lambda^*) = J(\theta^*, \lambda^*) \quad (59)$$

Comparing this with Eq. (57) completes the proof. □

[6] The concavity of the log-conditional probabilities implies that they have sublinear asymptotics. Thus in the maximization the quadratic prior term will eventually dominate.

E Examples

Here we provide a few examples of how to compute the Legendre transformations. Consider first the logistic regression case:

$$\log P(S_t|Z) = \log \sigma(S_t Z) \tag{60}$$

where $Z = \theta^T X$. Treating this log-probability as a function of Z, we can find its Legendre transformation:

$$F_t(\lambda) = \max_Z \{ \lambda Z - \log \sigma(S_t Z) \} \tag{61}$$

To perform this maximization, we take the derivative with respect to Z and set it to zero:

$$\lambda - S_t \sigma(-S_t Z) = 0 \tag{62}$$

which implies that $Z = -S_t \log(S_t\lambda)/(1 - S_t\lambda)$. Substituting this Z back into Eq. (61), we get

$$\begin{aligned} F_t(\lambda) &= -\lambda S_t \log \frac{S_t\lambda}{1 - S_t\lambda} - \log(1 - S_t\lambda) && (63) \\ &= H(S_t\lambda) && (64) \end{aligned}$$

where $H(\cdot)$ is the binary entropy function. If, in addition, we make a change of variables $\lambda_t = S_t\lambda$, then $F_t(\lambda_t) = H(\lambda_t)$ and is no longer a function of t. If the objective function in Theorem 1 is expressed in terms of these new λ_t, it reduces to the form given in Definition 1.

These calculations can be generalized to the multi-class setting where the probability model is the soft-max:

$$\log P(Y_t|Z_1, \ldots, Z_m) = Z_{Y_t} - \log \sum_{i=1}^{m} e^{Z_i} \tag{65}$$

with $Z_i = \theta_i^T X$. The Legendre transformation is obtained from

$$F_t(\lambda) = \max_{Z_1,\ldots,Z_m} \left\{ \sum_i \lambda_i Z_i - Z_{Y_t} + \log \sum_{i=1}^{m} e^{Z_i} \right\} \tag{66}$$

Similarly to the two class case we find the maximum by setting the derivaties with respect to Z_i to zero:

$$\lambda_i - \delta_{Y_t,i} + \frac{e^{Z_i}}{\sum_j e^{Z_j}} = 0 \tag{67}$$

(note that this implies that $\sum_i \lambda_i = 0$). The solution for the Z variables is unique up to a constant:

$$Z_i = \log(\delta_{Y_t,i} - \lambda_i) + constant. \tag{68}$$

Substituting these back into the transformation gives

$$F_t(\lambda) = -\sum_{i=1}^{m} (\delta_{Y_t,i} - \lambda_i) \log(\delta_{Y_t,i} - \lambda_i) \tag{69}$$

which is, not surprisingly, the entropy function. A change of variables $\lambda_{t,i} = \delta_{Y_t,i} - \lambda_i$ simplifies the transformation: $F_t(\lambda_t) = H(\lambda_t)$ and F no longer depends on t. In the new variables $\lambda_t = \{\lambda_{t,1}, \ldots, \lambda_{t,m}\}$, the objective function in Theorem 1 reduces to

$$\begin{aligned} J(\lambda) = -\frac{1}{2} \sum_{i,t,t'} (\delta_{Y_t,i} - \lambda_{t,i})(\delta_{Y_{t'},i} - \lambda_{t',i})(X_t^T \Sigma_i X_{t'}) \\ + \sum_t H(\lambda_t) && (70) \end{aligned}$$

We can also rewrite the predictive probability model in Theorem 1 in terms of the new $\lambda_{t,i}$ and get

$$P(Y|\hat{Z}_1, \ldots, \hat{Z}_m) = \frac{e^{\hat{Z}_Y}}{\sum_i e^{\hat{Z}_i}} \tag{71}$$

where $\hat{Z}_i = \sum_t (\delta_{Y_t,i} - \lambda_{t,i})(X_t^T \Sigma_i X)$.