Published in final edited form as:
Neuroimage. 2011 September 15; 58(2): 560–571. doi:10.1016/j.neuroimage.2011.06.053.

Utilizing Temporal Information in fMRI Decoding: Classifier Using Kernel Regression Methods

Carlton Chu[1,2], **Janaina Mourão-Miranda**[3,4], **Yu-Chin Chiu**[5], **Nikolaus Kriegeskorte**[6], **Geoffrey Tan**[2], and **John Ashburner**[2]

[1] Section on Functional Imaging Methods, Laboratory of Brain and Cognition, National Institute of Mental Health, NIH, USA

[2] Wellcome Trust Centre for Neuroimaging, University College London, United Kingdom

[3] Centre for Computational Statistics and Machine Learning, Department of, Computer Science, UCL, UK

[4] Department of Neuroimaging, Institute of Psychiatry, KCL, UK

[5] Department of Psychology, University of California, San Diego, CA

[6] Medical Research Council, Cognition and Brain Sciences Unit, 15 Chaucer Road, Cambridge CB2 7EF, UK

Abstract

This paper describes a general kernel regression approach to predict experimental conditions from activity patterns acquired with functional magnetic resonance image (fMRI). The standard approach is to use classifiers that predict conditions from activity patterns. Our approach involves training different regression machines for each experimental condition, so that a predicted temporal profile is computed for each condition. A decision function is then used to classify the responses from the testing volumes into the corresponding category, by comparing the predicted temporal profile elicited by each event, against a canonical haemodynamic response function. This approach utilizes the temporal information in the fMRI signal and maintains more training samples in order to improve the classification accuracy over an existing strategy. This paper also introduces efficient techniques of temporal compaction, which operate directly on kernel matrices for kernel classification algorithms such as the support vector machine (SVM). Temporal compacting can convert the kernel computed from each fMRI volume directly into the kernel computed from beta-maps, average of volumes or spatial-temporal kernel. The proposed method was applied to three different datasets. The first one is a block-design experiment with three conditions of image stimuli. The method outperformed the SVM classifiers of three different types of temporal compaction in single-subject leave-one-block-out cross-validation. Our method achieved 100% classification accuracy for six of the subjects and an average of 94% accuracy across all 16 subjects, exceeding the best SVM classification result, which was 83% accuracy (p=0.008). The second dataset is also a block-design experiment with two conditions of visual attention (left or right). Our method yielded 96% accuracy and SVM yielded 92% (p=0.005). The third dataset is from a fast event-related experiment with two categories of visual objects. Our method achieved 77% accuracy, compared with 72% using SVM (p=0.0006).

Address for Correspondence Carlton CHU Section on Functional Imaging Methods, Room 1D80, Building 10 National Institute of Health 9000 Rockville Pike Bethesda, Maryland 20892, USA *Tel* (+1) 301 402 1379 *Fax* (+1) 301 4021370 carltonchu1@gmail.com.

Keywords

Kernel Methods; Machine Learning; Kernel Ridge Regression (KRR); fMRI Prediction; Support Vector Machine (SVM); Relevance vector machines (RVM); Multi-class; Temporal compression; Temporal compacting

Introduction

There has been increasing interest in pattern classification of fMRI data. In contrast with the conventional mass univariate approach, which aims to find the brain regions that correlate with the experimental conditions, the goal of the pattern recognition approach, also known as the decoding framework, is to predict the experimental stimuli and cognitive state from the brain activation patterns. Many studies have shown high accuracies of decoding fMRI patterns with machine learning methods (Chu et al., 2010b; Friston et al., 2008; Haxby et al., 2001; Haynes and Rees, 2006; Mitchell et al., 2004; Strother et al., 2004).

Many of the benefits of neuroimaging may come from its potential to make predictions, rather than from the estimates of statistical parameters. Attempts had been made try to apply fMRI decoding into practical and clinical applications (Craddock et al., 2009; Davatzikos et al., 2005; Fu et al., 2008). Unlike some decoding works that used the classification accuracies as the surrogates of hypothesis testing (Hassabis et al., 2009; Haynes et al., 2007), improving prediction accuracies is always important in practical applications, for example brain-computer interfaces (BCI). Higher prediction accuracy also plays a major role in the design of real-time fMRI and neural feedback experiments (Weiskopf et al., 2007). If the experimental stimulus depends on the feedback of the prediction, near-perfect prediction may be necessary.

One common technique to improve predictive accuracy is to apply feature selection over the spatial domain (Chu et al., 2010a; De Martino et al., 2008; Mourao-Miranda et al., 2006). Common methods include filtering out non-informative voxels with univariate statistical maps e.g. SPM maps, recursive feature elimination (RFE) (Guyon and Elisseeff, 2003), and knowledge-based spatial priors. In the PBAIC 2007 competition (Chu et al., 2010b), we found that feature selection, based on prior knowledge of brain function, significantly improved performance in predicting two of the ratings (barking and inside/outside). In this manuscript, we introduce a different method that improves classification accuracy by utilizing the temporal information. Temporal and spatial information are orthogonal to each other, so one can apply both to obtain the best performance. However, the current work only focuses on optimizing the use of temporal information.

Several studies of fMRI decoding have employed the support vector machine (SVM) or other binary classifiers (LaConte et al., 2005; Mourao-Miranda et al., 2005). In these approaches, fMRI volumes are treated as the input features and the patterns are the strength of Blood Oxygenation Level Dependent (BOLD) signal. However, there is strong temporal correlation in the fMRI time series, especially due to the delay and smoothing from the hemodynamic response (HRF). For a block design, temporal shift is often applied to account for the hemodynamic delay and the volumes are averaged over each block (Cox and Savoy, 2003). Such strategies ignore the temporal profiles caused by the hemodynamic response. An alternative method, which preserves the HRF information, involves applying the regression to obtain parameter maps, sometimes referred to as “Beta maps” (Eger et al., 2008; Kriegeskorte et al., 2008). However, all these methods involving temporal compaction greatly reduce the number of training samples from the number of time points to the number of stimulus trials, hence hinder the training process, especially when the repeated conditions

are few. Inspired by the Pittsburgh Brain activity Interpretation Competition (PBAIC) (Chu et al., 2010b), we propose a novel approach that treats the problem as one of regression. We used a matching function to compare the predicted time series with canonical time series patterns for each target class, selecting the closest match as our prediction. The matching function can also be considered as a simple classifier. This new method greatly improves the prediction accuracy of experimental conditions within a single subject, over the conventional SVM with different temporal compression, especially for small numbers of training trials. We applied the proposed method to three different datasets, with multi-class classification and binary classification. The first and second dataset are from experiments using block designs, whereas the third dataset is from a fast event-related design experiment.

In addition, we also introduced a convenient way to derive a temporally compacted kernel directly from the original input kernel. Conventionally, temporal compaction is often applied before generating the kernel matrix from input features. Because this is a linear operation, we show how to compact the input kernel matrix into three common forms: boxcar averaging, beta-map, and spatial-temporal (temporal concatenation).

Methods

Kernel methods

Kernel methods are a collection of algorithms that, instead of evaluating the parameters in the space of input features, transform the problem into its dual representation. Therefore, solutions are sought in the kernel space, and the complexity of the algorithm is bounded by the number of training samples. A simple linear kernel can be calculated from the dot products of pair-wise vectors with equal elements. We define the input matrix as $\mathbf{X}$, where $\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2\cdots\mathbf{x}_n]^T$, each row of $\mathbf{X}$ is one fMRI volume with d voxels. The linear kernel matrix is calculated by $\mathbf{K} = \mathbf{X}\mathbf{X}^\mathbf{T}$. An advantage of kernel methods is the use of the so called “kernel trick”. Because the algorithm only requires the similarity measures described by the kernel matrix, implicit feature mapping into higher or even infinite dimensional space is possible, such as when using the radial basis function (RBF) kernel (Shawe-Taylor and Cristianini, 2004). Therefore it is possible that a nonlinear pattern in the original input space may appear linear in the higher dimensional feature space. Practically, since the images are already in a high-dimensional space, we did not apply any non-linear kernel in this work. Typical kernel algorithms include the Support Vector Machine (SVM), Relevance Vector Machine (RVM) and the Gaussian Process models (GP) (Cristianini and Shawe-Taylor, 2000; Rasmussen and Williams, 2006; Tipping, 2001; Vapnik, 1998).

Intuitively, the kernel matrix can be conceptualised as a matrix of similarity measures between each pair of input features. It contains all the information available about the relative positions of the inputs in the feature space. In other words, if we rotate and translate the data points in feature space, the information contained in the kernel matrix will not change, although the values of the kernel matrix may change. Because the information is encoded in the relative similarity, kernel methods require the training data when making predictions. If a sparse kernel method is used, at least the signature samples (e.g. support vectors or relevance vectors) would be required for the predicting phase. An exception is when the linear kernel is used, when it is possible to project the training results back into the original feature space. The general equation for making predictions with kernel methods is

$$t_* = \sum_{i=1}^{N} a_i K(\mathbf{x}_i, \mathbf{x}_*) + b \quad (1)$$

Here, t_* is the predicted score for regression, and if it is a classification algorithm, then it is the distance to the decision boundary. N is the number of training samples, x_i is a feature vector of the training sample, and is the x_* is the feature vector of the testing sample. Each kernel weight is encoded in a_i and b is a constant offset, both of which are learnt from the training samples. $K(.,.)$ is the kernel function, which contains dot-products for a linear kernel.

Detrend using residual forming matrix

Simple linear regression is often used to remove low frequency components in the fMRI time series at each voxel. This linear procedure can be reformulated as applying a residual forming matrix (Chu et al., 2010b) $\mathbf{R} = (\mathbf{I} = \mathbf{C}\mathbf{C}^{+})$ to the fMRI time series, where $\mathbf{C}$ is a set of any basis functions that model the low frequency components. For example, a linear basis set could model a constant term as well as a linear drift by $\mathbf{C}=\begin{pmatrix} 1 & 1 \\ \vdots & \vdots \\ N & 1 \end{pmatrix}$. $\mathbf{C}^{+} = (\mathbf{C}^{\mathbf{T}}\mathbf{C})^{-1}\mathbf{C}^{\mathbf{T}}$ denotes the pseudo-inverse of $\mathbf{C}$. For the same input matrix $\mathbf{X}$ defined previously, the detrended data can be computed by $\mathbf{X}_{\text{detrend}} = \mathbf{R}\mathbf{X}$. Recall that the linear kernel is calculated as $\mathbf{K}=\mathbf{X}\mathbf{X}^{\mathbf{T}}$. To compute the linear kernel from data with drifts removed:

$$\mathbf{K}_{\text{detrend}}=\mathrm{RX}(\mathrm{RX})^{\mathbf{T}}=\mathrm{RX}\mathbf{X}^{\mathbf{T}}\mathbf{R}^{\mathbf{T}}=\mathrm{R}K\mathrm{R}^{\mathbf{T}} \quad (2)$$

This operation is simpler than detrending each voxel separately. In this work, we only use simple linear detrending.

Temporal compaction of kernel matrices

Because the parameter maps, i.e. beta maps, are obtained from linear regression, it is possible to formulate some forms of temporal compaction as matrix operations. In fact, both “average of volumes” and “beta maps” are a weighted linear combination of the images in the time series. So far, a square residual forming matrix has been described for removing uninteresting signal from the kernel, but other forms of the matrix may also be used. Instead, it can be a matrix for converting a kernel matrix generated from the original data, into a kernel that is obtained by generating dot products from the parameter images.

Mathematically, we can define a vector of weighting coefficients, $\mathbf{p}$, which has the same number of elements as the number of images in the time series. This weighting vector is generated by taking the pseudo-inverse of the regressor in the design matrix of the corresponding block. Usually, the regressor is the HRF convolved block (see Figure 2) or a boxcar function with six seconds of delay (two TRs in our experiment) after the onset of the stimulus, which comes from the delay of the peak of the HRF. If every block has the same length, we can use the Kronecker product to generate the “average forming matrix” or “beta map forming matrix” (temporal compressing matrix) by $\mathbf{P} = \mathbf{I}\otimes\mathbf{p}^{\mathbf{T}}$, where $\mathbf{I}$ is the number of blocks by number of blocks identity matrix. This approach can be extended to event related fMRI as well. If each event is modeled as a separate regressor in the design matrix, the temporal compaction matrix, $\mathbf{P}$, is simply the pseudo inverse of the design matrix. The new data matrix can be evaluated by $\tilde{\mathbf{X}} = \mathbf{P}\mathbf{X}$ and the compressed kernel can also be evaluated directly from the original linear kernel generated from all image volumes, $\tilde{\mathbf{K}}=\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathbf{T}} = \mathbf{P}\mathbf{X}\mathbf{X}^{\mathbf{T}}\mathbf{P}^{\mathbf{T}}=\mathbf{P}\mathbf{K}\mathbf{P}^{\mathbf{T}}$. The dimension of this new kernel will be the number of blocks or events, rather than number of fMRI volumes in the series (see Figure 2).

There is also another formulation called “spatial-temporal” (Mourao-Miranda et al., 2007). In this formulation, images in each block are concatenated into one long vector, hence the

input features contain both spatial and temporal information, i.e. the temporal information is not averaged. Unfortunately, this formulation cannot be arranged into the same matrix operation. Therefore, we express each element in the condensed kernel as a sum of weighted kernel elements in the original kernel (Figure 3).

$$\tilde{k}_{ol}=\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}k_{n(o-1)+i,n(l-1)+j} \tag{3}$$

where $\tilde{k}_{ol}$ is the element at row o and column l in the compressed kernel, n is the number of image volumes in each block. **D** is the n by n weighting matrix containing the coefficients d_{ol} for each of the elements in the original kernel. Because the kernel matrix **K** is symmetric, the weighting matrix is also symmetric. The weighting matrix for the 'beta map' and 'average of volumes' can be computed directly from their weighting vector **p**, by $\mathbf{D}=\mathbf{pp}^{\mathbf{T}}$. For the spatial-temporal operation, the weighting matrix is a partial diagonal matrix, such that $\mathbf{D}_{i,i}=\begin{cases}0 & i\notin S\\ 1 & i\in S\end{cases}$, where S is the set of images concatenated in the block, and it is often selected to be the same set as the averaging operation. Generally speaking, the full kernel matrix from the entire time series is often utilized in the kernel regression framework, whereas the compacted kernel matrix is used in classification problems, where the objective is to categorise events.

Support Vector Machine (SVM)

In this work, SVM specifically refers to support vector classification (SVC). SVC is also known as the maximum margin classifier (Cristianini and Shawe-Taylor, 2000), and has shown superior performance in practical applications with high-dimensional data. Motivated by statistical learning theory (Vapnik, 1995), the decision boundary is chosen so that it achieves the maximum separation between both classes. The standard formulation of optimizing the hard-margin SVC is

$$\begin{aligned}&\text{minimize} && 1/2\,\mathbf{w}^{\mathbf{T}}\mathbf{w}\\ &\text{subject to} && t_i\left(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i=b\right)\geq 1, i=1\ldots,N\end{aligned} \tag{4}$$

This is often known as the primal form of the SVC optimization. Here, **w** is a vector of feature weights, $t\in\{-1,1\}$ is the label for the classes, **x** is a vector of input features, and b is an offset. Although it is possible to solve this optimization in the primal form (Chapelle, 2007), the optimization is often solved in the dual formulation by introducing Lagrange multipliers.

$$\begin{aligned}&\text{maximize} && -\tfrac{1}{2}\mathbf{a}^{\mathbf{T}}\,\mathbf{Ha}+\sum_{i=1}^{n}a_i\\ &\text{subject to} && \sum_{i=1}^{n}t_ia_i=0\\ & && a_i\geq 0,\qquad i=1,\ldots,N\end{aligned} \tag{5}$$

Here, a_i is a vector of Lagrange multipliers, and **H** is a N by N matrix defined by $h_{i,j}=(t_it_j\mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j : i,j=1,\ldots,N)$. More generally, we can replace $\mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j$ by the kernel $K(\mathbf{x}_i,\mathbf{x}_j)$. This formulation makes SVM a kernel algorithm, and can be optimized by standard quadratic programming, but is also often solved by more efficient algorithms e.g. Sequential Minimal Optimization (SMO) (Platt, 1999). The computational complexity of SMO is between linear and quadratic in the training size, and depends on the sparsity of the data. Once the

Lagrange multipliers a_i and the offset parameter b are computed, the prediction can be made based on equation (1) for a new testing sample, and the decision boundary is at the line where equation (1) yields zero. In this study, we used the implementation of hard margin SVC in LIBSVM (Chang and Lin, 2001).

Multi-class SVC

The original SVC is a binary classifier. To perform multi-class classification, the common approach in the neuroimaging field is to combinine a number of binary classifiers (Hassabis et al., 2009; Mourao-Miranda et al., 2006). For situations with H classes, there are two commonly used approaches; one is the "one versus the rest" classifier. This works by training H classifiers, each of them trained with one class versus the other H-1 classes. The classification for a testing point is determined by the classifier that achieves the highest classification scores i.e. furthest away from the decision boundary toward the particular class. Another approach is called the "one versus one" classifier, which works by introducing $H(H\text{-}1)/2$ or C^K_2 classifiers. Each of the classifiers is trained with one class versus another class only. The assigning of a testing point is achieved by majority vote. In other words, it is assigned to the most frequent class to which it is classified by all the classifiers. Ambiguous cases may occur in this approach. For example if we have three classes 1, 2, and 3, then we will have three classifiers (1vs2, 1vs3, 2vs3). The testing point may be classified into class 1, class 3, and class 2 from the three classifiers respectively.

Alternatively, one can use error correcting output codes to improve the performance (Dietterich and Bakiri, 1995). In the example of three classes, the error codes can be generated from six binary classifiers (1 vs 2, 1 vs 3, 2 vs 3, 1 vs (2+3), 2 vs(1+3), 3 vs (1+2)). This approaches can avoid ambiguous cases and aggregate more information. There are also classifiers that are capable of doing multi-class classification, for example multiclass Fisher's discriminant, multiclass logistic regression (Bishop, 2006; Girolami and Rogers, 2006), and decision tree methods (Quinlan, 1993).

Multi-class classifier using kernel regression (MCKR)

We design a multi-class classifier that is very similar to one versus the rest classification for fMRI experiments. This method utilizes the temporal information without compressing it into a reduced kernel. Our approach breaks the classification into three stages: 1. Training K regression models; 2. Predicting the temporal profiles for a testing block; 3. Matching the predicted K profiles with the canonical profile, which can be computed by convolving one block with the canonical HRF (Friston et al., 2007) (see Figure 4). This approach was originally inspired by the PBAIC (Chu et al., 2010b), therefore we took a similar approach in the training phase. That is, we only changed the target variable, but used the same input features. In our case, the experiment had three conditions in the design. We trained three different regression machines, where each of the machines took the same kernel generated from the fMRI volumes as input features, but the target variables were the corresponding regressors (columns) in the design matrix. In the predicting phase, temporal profiles of the test block (multiple fMRI volumes) were predicted from all three regression machines. To assign class membership, we compared all the predicted profiles with the canonical profile. We tried both covariance and correlation as the metric to measure similarities. Both measures ignore the constant offset, and covariance considers the magnitude of the prediction, while correlation ignores the information of magnitude. In practice, the results using correlation and covariance are not statistically different. The class was assigned to the condition for which the machine achieved the highest similarity between the predicted profile and the canonical profile. Although this technique is introduced as a multi-class classifier, it works the same for binary classification. This method was applied to data from block designs (dataset 1 & dataset 2) and fast event related designs (dataset 3).

Kernel ridge regression (KRR)

Kernel ridge regression is the kernel formulation of the ridge regression. Ridge regression is the standard least square regression with regularization that penalizes sum of squares of the weights (parameters) $\bar{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} \left(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i - t_i\right)^2 + \lambda \mathbf{w}^{\mathbf{T}}\mathbf{w}$, where $\lambda \geq 0$ is the regularization parameter. It has been shown that kernel ridge regression yields the same solution from the standard (primal) ridge regression (Shawe-Taylor and Cristianini, 2004). The kernel formulation can gain computational efficiency when the dimensions of the feature vectors are greater than the number of training samples, which is true in most neuroimaging data. The formulation of kernel ridge regression is

$$\mathbf{a} = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{t} \tag{6}$$

where **K** is the kernel matrix, and **I** is the identity matrix. Notice there is no offset parameter (b) in kernel ridge regression. Prediction can be computed from equation (1). The regularization, λ, was fixed to $1e^5$, which was based on our experience and empirical results from PBAIC (Chu et al., 2010b). In this study, the overall accuracies varied by less than 3% when λ was in the range of $1e^2$ ~$1e^5$ (i.e. the regularization parameter did not have to be very precise, as it yielded very similar results in a very wide range of values) .Because of the matrix inversion, the computational complexity of KRR is $O(N^3)$, where N is the number of examples in the training data.

Relevance vector regression (RVR)

Relevance Vector Regression (RVR) is a sparse kernel method, and is formulated in the Bayesian framework (Tipping, 2000, 2001). Strictly speaking, RVR is not a standard kernel method, because it does not have the interchangeable formulation between primal and dual. Instead, RVR treats the kernel matrix as a set of linear basis function. This implies that the input of RVR does not need to be a Mercer kernel (i.e. symmetric and positive-definite). However, the standard RVR often uses the Kernel appended with one column of ones as the input, $\mathbf{\Phi} = [1, \mathbf{K}]$, with **K** denoting the kernel matrix and **1** denoting a column of ones. The likelihood function of the observation is modelled by a Gaussian distribution, $p(\mathbf{t} \mid \mathbf{a}, \sigma^2) = N(\mathbf{t} \mid \mathbf{\Phi a}, \sigma^2\mathbf{I})$, where **t** is the vector of observed target values, **a** is the weight vector, and σ^2 is the variance of the noise. The prior of the weights, **a,** are also modelled by zero mean Gaussian, $p(\mathbf{a}|\alpha) = \prod_{i=0}^{N} N\left(a_i|0, \alpha_i^{-1}\right)$. The solution involves optimizing the marginal likelihood (type-II maximum likelihood).

$$p\left(\mathbf{t}|\alpha, \sigma^2\right) = \int p\left(\mathbf{t}|\mathbf{a}, \sigma^2\right) p(\mathbf{a}|\alpha)\, d\mathbf{a} = N(\mathbf{t}|0, \mathbf{C}) \tag{7}$$

where $\mathbf{C} = \sigma^2\mathbf{I} + \mathbf{\Phi}\Omega^{-1}\mathbf{\Phi}^{\mathbf{T}}$ is the covariance of the marginal likelihood, and $\Omega = diag(\alpha_0, \alpha_1, ..., \alpha_N)$ is the diagonal matrix of the inverse of the variance for each weight. The objective of the optimisation is to find the hyper-parameters, **α**, σ^2, which maximise the "evidence" of the data (Mackay, 1992; Tipping, 2001). This is closely related to restricted maximum likelihood (ReML) and estimation of covariance components in the statistical literature (Friston et al., 2002; Harville, 1977). The covariance matrix that maximises the marginal likelihood can be obtained by iterative re-estimation. After finding the hyper-parameters, the posterior weights can be estimated by $\mathbf{a} = (\mathbf{\Phi}^{\mathbf{T}}\mathbf{\Phi} + \sigma^2\Omega)^{-1}\mathbf{\Phi}^{\mathbf{T}}\mathbf{t}$.

When maximising the marginal likelihood, some of the α will grow very large, and effectively results in the posterior probability of those weights, a, sharply peaking at zero. This property allows irrelevant columns of the basis functions to be pruned out, and is known as automatic relevance determination (ARD) (MacKay, 1995). As with all other kernel methods, the prediction can be computed from equation (1). The computational complexity of RVR is $O(\tilde{N}^3)$ (Tipping, 2001), where $\tilde{N}$ is the averaged training size. In practice, the computation time depends on the implementation (e.g. stopping criteria).

One thing to be aware of is that all these machine learning methods (SVC, RVR, KRR) assume that the data points are independent and identically distributed (iid), whereas the actual noise for fMRI data is not i.i.d. This does not mean those algorithm will fail completely, but does imply that such mis-specified models may give sub-optimal results (Chu et al., 2010b; Rasmussen and Williams, 2006).

Weight map

Because we used a linear kernel, it was possible to project the weights back into the voxel space for both SVC and Kernel Regression. The weight map from one training of the regression machine, either KRR or RVR, was computed by $\mathbf{w} = (\mathbf{a}^\mathbf{T}\mathbf{RX})^\mathbf{T}$, where $\mathbf{X}$ is the original input matrix, $\mathbf{R}$ is either a detrending or residual forming matrix, and $\mathbf{a}$ contains the kernel weights obtained from the training. Unlike SVC or other binary classifiers, which only generate one map for binary classification, the regression machines generate one map per class. In other words, the regression machines generate two maps for a binary classification.

To generate the map for SVC using a compacted kernel matrix, we can simply add the compaction matrix and rewrite the equation as $\mathbf{w} = (\mathbf{a}^\mathbf{T}\mathbf{PRX})^\mathbf{T}$. For spatial-temporal SVC, $\mathbf{P}$ is the matrix that selects the specific fMRI volumes for each block. Each time, only one spatial map can be generated, and the temporally successive maps (i.e. the spatial map at different TR) can be computed by fixing the kernel weight, $\mathbf{a}$, and changing the matrix **P to** select different volumes.

Materials

Dataset 1

The dataset used in this study is the same as described by (Hardoon et al., 2007; Mourao-Miranda et al., 2007; Mourao-Miranda et al., 2006) And involves functional MRI scans from 16 male right-handed healthy college students (age 20–25 years). The study was performed in accordance with the local Ethics Committee of the University of North Carolina. The data were collected on a 3 T Allegra Head-only MRI system (Siemens, Erlangen, Germany). The fMRI runs were acquired using a T2* sequence with 43 axial slices (slice thickness, 3 mm; gap between slices, 0 mm; TR = 3 s; TE = 30 ms; FA = 80°; FOV = 192 mm = 192 mm; matrix, 64 = 64; voxel dimensions, 3 mm 3 mm 3 mm). In each run, 254 functional volumes were acquired.

Experimental design—The experimental stimuli were in a standard block design. It was a passive experiment with visual stimuli. The visual stimuli were categorized into three different active conditions: viewing unpleasant (dermatological diseases), neutral (people) and pleasant images (girls in bikinis). Each active condition was followed by a resting condition (fixation) with equal duration. In each run, there were six blocks of the active condition (each consisting of seven image volumes) alternating with resting (fixation) over seven image volumes. Six blocks of each of the three stimuli were presented in random order. There was only one run for each subject.

Pre-processing—The data were pre-processed using SPM5 (Wellcome Trust Centre for Neuroimaging, London, UK). All scans were first realigned and resampled to remove effects due to subject motion. This had the effect of increasing the similarity among scans, and compacting the low dimensional manifolds on which the images lay.

To further increase the signal-to-noise ratio, those voxels that were, *a priori*, considered non-informative were removed. BOLD signal change is generally believed to occur mainly in grey matter, because its major cause should be the local neuronal activity (Logothetis et al., 2001). Empirical results from the Pittsburgh Brain Activity Interpretation Competition (PBAIC) 2007 also showed that masking out non-grey matter voxels improves the classification performance (Chu et al., 2010b). Masks defining grey matter were generated for each subject by segmenting the average fMRI scan of each subject using unified segmentation implemented in SPM5 (Ashburner and Friston, 2005). This also accelerates the computation of the kernel matrices, as only about 20% of the whole image is used.

To perform multi-subject prediction, all scans were also non-linearly aligned to their population average using DARTEL (Ashburner, 2007), which is more accurate than conventional SPM spatial normalization (Bergouignan et al., 2009; Klein et al., 2009). After estimating the inter-subject alignment by matching tissue class images together, the warping parameters were used to transform each subject's fMRI volumes. An additional smoothing with 8mm FWHM Gaussian Kernel was applied. The amount of smoothing was chosen to be to the same as that in the previous studies (Mourao-Miranda et al., 2007; Mourao-Miranda et al., 2006), allowing the results to be compared (see Figure 1). Although a recent study (Op de Beeck, 2010) suggested that spatial smoothing does not affect the decoding performance, our empirical findings from PBAIC 2007 (Chu et al., 2010b) showed that spatial smoothing can improve accuracies for some subjects. Temporal normalization was also applied to each voxel for each subject, which involved dividing the voxel time courses by their standard deviations. This procedure minimized the variation of temporal scale among different subjects.

Cross-validation: To compare the accuracies of MCKR with the multi-class SVM in previous studies, we performed leave-one-block-out cross-validation (LOBOCV) within each subject separately. The fMRI volumes for LOBOCV were in the native space. In each LOBOCV trial, one block (active + rest) was removed from the dataset as the testing block i.e. one volume for SVM after compression or 14 volumes for MCKR. The training used the remaining dataset, and the label of the testing block was predicted with the parameters obtained from the training. The averaged accuracy was then calculated by averaging of the predicted accuracies from all subjects.

We also performed leave-one-subject-out cross-validation (LOSOCV) for each subject. The fMRI volumes for LOBOCV were in the population-averaged space (by DARTEL). In each LOSOCV trial, one subject was left out as the testing sample, and the data from the remaining 15 subjects were used to train the classifier. After the training, all 18 labels were predicted from the testing subject. The averaged accuracy was calculated.

Dataset 2

Details of this dataset are described fully in the thesis of (Chiu, 2010). Briefly, fifteen neurologically intact right-handed, healthy adults (four males, mean age = 21.3 yrs) were recruited from the Johns Hopkins University community. All participants completed and signed an informed consent approved by the Johns Hopkins Medicine Institution Review Board. Whole brain functional data were acquired with 40 slice echo-planar images (EPIs) in an ascending sequence, TR = 2000 ms, TE=30 ms, flip angle = 70°, scan time = 314 sec

(one run), matrix = 64 × 64, slice thickness = 3 mm, SENSE factor = 3, yielding 3-mm isotropic voxels.

Experimental design—The original experiment had 15 runs, but we only used three runs of sustained attention task, which are relevant to fMRI decoding. At the beginning of each sustained attention run, participants were instructed to start monitoring either the left or the right target stream of characters for a fixed block of 20s (10 volumes), during which they responded to the appearance of "5" in the attended stream by pressing both buttons. The initial stream (left or right) was randomly selected for each run. At the end of each block, a shift cue "X" appeared in the currently attended stream, instructing participants to shift their attention to the other side (i.e., left to right and vice versa). There were a total of 14 shift cues, creating seven blocks of sustained attention to the left and seven to the right in each run, with an additional block of attention to the initial attended location. In our decoding task, the last block was omitted to obtain an even number of conditions.

In summary, there were 15 subject, and three runs for each subject. There were two conditions (left or right) in the experiment, and each run had seven attend-left block and seven attend-right blocks. The length of each block is 10 volumes (20 s).

Pre-processing—The data were preprocessed using BrainVoyager QX v1.10 software (Brain Innovation, Maastricht, The Netherlands). Functional data were slice-time corrected (with cubic spline interpolation), motion corrected (rigid-body translation and rotation with trilinear interpolation), and then temporally high-pass filtered (3 cycles per run). No other spatial-smoothing or normalization was performed. Simple intensity-based masks were used to exclude non-brain tissues.

Cross-validation: Although, binary classification (left or right) was performed in this dataset, for the consistency of naming, we still called the proposed method MCKR in this dataset and dataset 3. Because there are multiple runs in this dataset, we performed leave-one-run-out (LORO) cross-validation, which is more conservative, for each subject. In each LOROCV trial, one run was left out as the testing sample, and the data from the remaining two runs were used to train the classifier. After the training, all 14 testing blocks in the test run were predicted. The averaged accuracy was then calculated by averaging over the predictive accuracies of all 15 subjects.

Dataset 3

This dataset was used and described by (Kriegeskorte et al., 2008; Misaki et al., 2010). Only a part of the original dataset was used to demonstrate the proposed method. Four healthy subjects participated in this event-related experiment. The data were collected on a 3T GE HDx MRI scanner with a receive only whole brain surface coil array (16 elements). Whole brain functional data were acquired with 25 slice echo-planar images (EPIs) with SENES (Acceleration factor =2), TR = 2000 ms, TE=30 ms, scan time = 544 sec (one run), matrix = 128 = 96, slice thickness = 2 mm, SENSE factor = 3), yielding 1.95 mm = 1.95 mm = 3 mm voxels. Only the occipital and temporal lobe were covered.

Experimental design—There were a total six runs in the experiment. In each run, 96 color pictures of objects on a gray background were shown. There were 48 inanimate objects and 48 animate objects. The duration of each stimulus was 300 ms. Each stimulus was shown only once in each run, in random order. The stimulus onset asynchrony was random (4s, 8s, 12s,16s). The decoding task was to discriminate animate stimuli from inanimate stimuli.

Pre-processing—The data were slice-time corrected and realigned using BrainVoyager QX. The time series in each voxel were normalized to percent signal change. No other spatial-smoothing or normalization was performed (Kriegeskorte et al., 2008). Linear detrending was applied. A mask of the human inferior temporal cortex (hIT) with 1000 voxels was computed from the overlapping voxels of an anatomical mask and a thresholded statistical map from a separate experiment (i.e. element-wise AND operation to the anatomical mask and the thresholded t-map).

Cross-validation: We also applied LOROCV in this dataset. We followed the procedure described in (Misaki et al., 2010) to perform SVC classification. In each LOROCV trial, one run was left out as the testing run, and the data from the remaining five runs were used to train the classifier. The beta-map estimates were computed from the two sets independently.

For MCKR, although dataset 3 is a fast event-related experiment, we modelled each stimulus as a small block, and performed the procedure in the same way as for a block-design experiment. In the predicting phase, we used image volumes from the onset to 14s after the onset (seven volumes) for temporal profile matching. We predicted the temporal profile within the 14s windows (based on the HRF profile) per stimulus, and then applied the matching function to make one classification for that temporal profile. In other words, we still made one prediction per stimulus per run using the proposed method, which had exactly the same number of test estimates as the SVM.

Results

Classification performance was compared between different methods, using paired t tests (Misaki et al., 2010). We found that the choice of whether covariance or correlation was used as the matching function made no significant difference. We therefore chose covariance as the matching function in all datasets and subjects because it was considered “simpler” (i.e. correlation is covariance normalized by the standard deviations, so more operations are needed to compute correlation than to compute covariance). Following Occam's razor, when a simpler procedure yields equivalent performance, we should prefer the simpler one. Although SVC with beta maps yielded higher mean accuracies than SVC with averages, predictive accuraries with either method were not significantly different from each other. MCKR using RVR performed slightly better than MCKR using KRR, but both methods were also not significantly different. There were two variants of MCKR and three variants of SVC, so to avoid multiple comparisons, we compared the worst performing MCKR (MCKR-KRR), with the best performing SVC (SVC beta map). Although this approach was very conservative, we found that MCKR performed significantly better than SVC in all three datasets ($p=0.008$, $p=0.0006$, $p=0.005$, respectively). The original cross-validation accuracies in each run and each subject are shown in the supplementary material.

Dataset 1

Within-subject classification—The averaged accuracy of LOBOCV was very high when the proposed method, MCKR, was applied. Perfect classification (100% accuracy) was obtained for 6 subjects (2 subjects by SVC beta map) and an average of 94% accuracy was achieved across 16 subjects. We achieved slightly higher accuracies for SVC compared with the previously published work (Mourao-Miranda et al., 2006). The accuracies of “one versus the rest” multi-class SVC were around 3% higher than the “one versus one” multi-class SVM, so we only present the result from “one versus the rest” SVM in this paper because we want to compare the best SVC method with the worst MCKR method. The best classification accuracy for SVC was 87.5% using the beta map, average of volumes resulted in 86% accuracy, and spatial-temporal had the worst performance with 66% accuracy (Table

1 and Figure 5). Despite using the best SVC results, the classification accuracy from multi-class SVC with beta maps was significantly lower than the worst performing MCKR (p=0.008). For multi-class classification, it is not possible to compute the receiver operating characteristic (ROC). Therefore we present the confusion matrix of MCKR with RVR, which is similar to the results from MCRK with KRR, and the confusion matrix of multi-class SVC compressed by "average of volume", which is similar to the result from using beta-maps.

In addition to performing the LOBOCV, we also performed cross-validation (CV) trials with reduced training blocks. In each of the CV trials, we randomly sampled M (a number ranging from one to five) blocks, which were used for training the classifiers. For each selected M, we repeated the sampling and CV 40 times. Surprisingly, even when we used only one training block, all methods still achieved classification accuracies above chance level (33%) (Figure 6). The results also showed that MCRK always outperformed multi-class SVC, regardless of the number of training samples (from training size of one block to five blocks, p=0.49, p=0.017, p=0.065, p=0.063, p=0.025, respectively). MCKR with only two training blocks can perform as well as SVC with four training blocks (p=0.3, no statistical difference).

Inter-subject classification—The accuracy of multi-class SVC improved greatly in LOSOCV, and the average accuracy was 95% regardless of compaction methods. However, the averaged accuracy of MCKR decreased to 91% (Figure 7 and Table 3). Despite higher accuracy in SVC, paired t tests showed no significant difference between MCKR and SVC (p=0.55). There were two subjects that had extremely low accuracy in MCKR, whereas the other 14 subjects had similar results for both methods. Also, MCKR achieved equivalent performance to multi-class SVC for classifying pleasant and unpleasant stimuli. The reduction of averaged accuracy came from a significant drop in the ability of MCKR to classify neutral stimuli (85%), especially in those two subjects (see Supplementary material). The confusion matrix of RVR (Table 4) from LOSOCV, shows a strong bias toward misclassifying a neutral stimulus as unpleasant.

We also performed the LOSOCV with reduced numbers of training subjects (Figure 8). In each of the CV trials, we randomly sampled M (1,2,3,4,5,7,10,13) training subjects. For each selected M, we repeated the sampling and CV 40 times. The accuracies of MCKR with KRR and SVC with beta-maps reached a plateau when the number of training subjects was seven or more. The accuracies of SVC with spatial-temporal reached the plateau much later, when 13 training subjects were used. SVC was only significantly better than MCKR when training with 10 and 13 subjects ($p<0.05$).

The major clusters in the weight maps generated by training all 16 subjects (Figure 9) were very prominent. Maps generated from KRR and SVC have very high correlation in both pleasant and unpleasant stimuli (r=0.76, r=0.80, respectively), but the correlation is lower (r=0.69) between maps from KRR and SVC in neutral stimuli. This may also explain the comparable performance between MCKR and SVC in classifying pleasant and unpleasant stimuli, but their different performance for classifying neutral stimuli.

Dataset 2

We only performed LOROCV for SVC using beta map and MCKR with KRR. This experiment had a very high contrast-to-noise ratio (CNR). The averaged accuracy of LOROCV was very high when the proposed method, MCKR, was applied, achieving an average accuracy of 96% (Figure 10). Perfect classification (100% accuracy) was obtained for 32 runs out of 45 (15 subjects times 3 runs per subject) comparing to 23 runs using the SVC beta map approach. SVC also performed well, achieving an average accuracy of 92%.

However, paired t tests showed MCKR to be significantly better than SVC (p=0.006). There was no bias toward predicting one particular class.

Dataset 3

This dataset had relatively lower CNR because of the fast-event related design. Only LOROCV for SVC using the beta map approach and MCKR with KRR were employed. MCKR achieved 77% accuracy and SVC achieved 72% accuracy (Figure 11). MCKR performed significantly better than SVC (p=0.0006), and resulted in the lowest p-value among the three datasets. We also found no bias toward predicting one particular stimulus category.

Discussion

We developed a novel multi-class classifier using the kernel regression method. Our MCKR method uses all the information in the fMRI volumes during training instead of compacting them into a smaller set. The results demonstrated that our MCKR method was consistently better than multi-class or binary SVC in LOBCV or LOROCV within each individual subject. The same performance from RVR and KRR implies that the superiority of our method comes mainly from the architecture of the procedure (Figure 4) rather than the specific choice of regression algorithm. Other methods, such as support vector regression or elastic net, should also perform similarly. The choice of matching function also made no statistical difference, but we used covariance in the matching function for reasons described earlier. This implies that the predicted profile contains information mainly in the shape of the profile. We suggest that MCKR performs better than SVC because of more training samples and the ability to learn the subject's individual response to each condition. With more training samples, the solutions from the kernel machines (either regression or classification) would have higher degrees of freedom, leading to more accurate estimates the noise variance. From figure 6, we can see that when the number of training samples increases, the accuracy gap between MCKR and SVC with beta-maps or averages becomes smaller. Perhaps with more repeats and a longer experiment, SVC may achieve the same performance as MCKR.

The MCKR did not show strong prediction bias, but SVC seemed to have a strong bias toward mistaking the condition of unpleasant as pleasant (Table 2). This result shows that the temporally compressed fMRI pattern under unpleasant stimuli were more similar to the pattern under pleasant stimuli than neutral stimuli. Conversely, the unbiased result from MCKR implies that the patterns between the three stimuli are equivalently distinct when more temporal information was used.

After evaluating maps generated from each individual subject and maps from training all subjects, we found that although maps generated by both regression machine and SVC are different, the main clusters in the maps are very similar. The shapes and sizes of the clusters are different, but the locations are the same. In other words, one would find the clusters in the same brain regions using either map and probably would make the same inference in brain functions using either method.

The improvement of SVC in the LOSOCV (i.e. inter-subject cross-validation) can be explained by the increase of training samples (Figure 9). We suspect that the reason why the MCKR machine did not work well between subjects, as opposed to in single subjects, was due to the inter-subject variability of activation size (different number of activated and deactivated voxels) and strength. The MCKR approach is more sensitive to inter-subject variability than is SVC with temporal compression, because regression machines have cost functions of least squares, and the inter-subject MCKR assumes all subjects have the same

magnitude of multivariate-activation. Although we normalized every voxels of each subject, some inter-subject variance still remained. For example, some subjects had uniform activation among the three stimuli, whereas some subjects had relatively low activation for the neutral stimuli. Also, some subjects have more activated voxels (bigger clusters) than others subjects. Consider a subject with slightly more activated voxels in one condition, the multivariate response of that condition for that subject would be stronger than others. This stronger response would have less impact on SVM because such effects would only result the training samples from the class/condition moving further away from the decision boundaries, and not being considered as support vectors. Such confounds would sabotage the training of MCKR with least-square objective function. If we ignored the neutral stimuli, MCKR and SVC would perform equally well. On the other hand, if we could apply feature selections to limit the number of voxels and regions, the prediction performance of the MCKR may also be as good as SVC.

The experimental design of dataset 2 was very similar to dataset 1. The duration of the block in both experiments was around 20s. The only difference was that experiment in dataset 2 had no resting condition. Because we applied LOROCV, the prediction should not be confounded by the potential temporal correlation in the training. The empirical results from dataset 2 further support the proposed method.

Although dataset 3 was from an event-related experiment, the proposed method still performed well. We applied LOROCV to prevent confounds due to overlapping events in the training. This dataset yielded the lowest p-value, probably because there was no ceiling effect. In experiments with high CNR (dataset 1 and 2), SVC could often achieve 100% classification.

In addition to introducing MCKR, we also presented an efficient method to perform temporal compaction and reformulation (Figure 3). Such a formulation can bring a different perspective to the process. In our experimental design, because of the long block design, the block operator for generating beta-maps is very similar to that for generating the average over volumes (Figure 3). This was why both compressions resulted in similar accuracy. Both beta-map and average of volumes employed off-diagonal information from the kernel, whereas the spatial-temporal only used the diagonal components. The inferior performance of spatial-temporal with low training samples may imply that the covariance between neighbouring volumes temporally contains useful information to discriminate the conditions. This kernel formulation of temporal manipulation can also be extended to frequency decomposition, for example, by applying a discrete cosine transform matrix. The effect of temporal filtering on fMRI decoding efficiency can then be studied. Also, in a fast event related experiment, it is often suggested to average over a few trials to create a beta-map (i.e. multiple trials in one column of the design matrix). The number of trials to average is often arbitrary. When a new number is selected, the general linear model needs to be re-fitted. It is more computationally efficient to apply the matrix operation on the kernel matrix directly using the formulation we proposed.

Regarding to the weight maps, we make no attempt at inference about the significance of a given region with high weights on a weight map nor do we point out any given region. This is because it is not the intent in the proposed method to draw conclusions about regionally specific effects. The proposed method was not designed to generate maps that were comparable because the final classification also depends on the matching between the template (canonical profile) and the predicted temporal profile. If the template is changed, the same “trained regressors and maps” may yield different classification accuracies. The matching function makes the inference between voxel weights and the classification results non-linear.

In summary, our proposed method, MCKR, performed significantly better than the conventional multiclass SVC and binary SVC when the number of training samples was low and the training and testing were performed within the same subject. Many fMRI studies have performed decoding only at the level of a single subject (Hassabis et al., 2009; Haynes et al., 2007). Our method is ideal for experiments that have limited repeats of stimuli, this implies one can design an experiment with fewer repeats and more conditions, or shorten the duration of experiments. Also, if the prediction system is going to be trained online, e.g., real time fMRI decoding, our MCKR should be the recommended option, especially when nearly-perfect classification is needed. Using the canonical HRF would result sub-optimal solutions and the commonly used classification approach with temporal compacting also suffers from the same disadvantage. One direction for future research is to try to learn the optimal HRF profile from the training set.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

This research was supported in part by the Intramural Research Program of the NIH, NIMH. JMM was funded by a Wellcome Trust Career Development Fellowship. JA is funded by the Wellcome Trust. The authors thank Prof. Michael Brammer for providing the first dataset. The authors thank Prof. Steven Yantis for providing the second dataset, which was supported by NIH grant R01-DA013165

References

Ashburner J. A fast diffeomorphic image registration algorithm. Neuroimage. 2007; 38:95–113. [PubMed: 17761438]

Ashburner J, Friston KJ. Unified segmentation. Neuroimage. 2005; 26:839–851. [PubMed: 15955494]

Bergouignan L, Chupin M, Czechowska Y, Kinkingnehun S, Lemogne C, Le Bastard G, Lepage M, Garnero L, Colliot O, Fossati P. Can voxel based morphometry, manual segmentation and automated segmentation equally detect hippocampal volume differences in acute depression? Neuroimage. 2009; 45:29–37. [PubMed: 19071222]

Bishop, CB. Pattern recognition and machine learning. Springer; 2006.

Chang, CC.; Lin, CJ. LIBSVM: a library for support vector machines. 2001.

Chapelle O. Training a support vector machine in the primal. Neural Computation. 2007; 19:1155–1178. [PubMed: 17381263]

Chiu, YC. Moment-by-moment tracking of attentional fluctuation in the human brain. Department of psychological & brain sciences. Johns Hopkins University; 2010.

Chu C, Bandettini P, Ashburner J, Marquand A, Kloeppel S. Classification of Neurodegenerative Diseases Using Gaussian Process Classification with Automatic Feature Determination. IEEE. 2010a:17–20.

Chu C, Ni Y, Tan G, Saunders CJ, Ashburner J. Kernel regression for fMRI pattern prediction. Neuroimage. 2010b

Cox DD, Savoy RL. Functional magnetic resonance imaging (fMRI) “brain reading”: detecting and classifying distributed patterns of fMRI activity in human visual cortex. Neuroimage. 2003; 19:261–270. [PubMed: 12814577]

Craddock RC, Holtzheimer PE 3rd, Hu XP, Mayberg HS. Disease state prediction from resting state functional connectivity. Magn Reson Med. 2009; 62:1619–1628. [PubMed: 19859933]

Cristianini, N.; Shawe-Taylor, J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press; 2000.

Davatzikos C, Ruparel K, Fan Y, Shen DG, Acharyya M, Loughead JW, Gur RC, Langleben DD. Classifying spatial patterns of brain activity with machine learning methods: application to lie detection. Neuroimage. 2005; 28:663–668. [PubMed: 16169252]

De Martino F, Valente G, Staeren N, Ashburner J, Goebel R, Formisano E. Combining multivariate voxel selection and support vector machines for mapping and classification of fMRI spatial patterns. Neuroimage. 2008; 43:44–58. [PubMed: 18672070]

Dietterich TG, Bakiri G. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research. 1995; 2:263–286.

Eger E, Ashburner J, Haynes JD, Dolan RJ, Rees G. fMRI activity patterns in human LOC carry information about object exemplars within category. J Cogn Neurosci. 2008; 20:356–370. [PubMed: 18275340]

Friston K, Chu C, Mourao-Miranda J, Hulme O, Rees G, Penny W, Ashburner J. Bayesian decoding of brain images. Neuroimage. 2008; 39:181–205. [PubMed: 17919928]

Friston, KJ.; Ashburner, J.; Kiebel, JS.; Nichols, TE.; W., WD. Statistical parametric mapping, the analysis of functional brain images. Academic press; 2007.

Friston KJ, Glaser DE, Henson RN, Kiebel S, Phillips C, Ashburner J. Classical and Bayesian inference in neuroimaging: applications. Neuroimage. 2002; 16:484–512. [PubMed: 12030833]

Fu CHY, Mourao-Miranda J, Costafreda SG, Khanna A, Marquand AF, Williams SCR, Brammer MJ. Pattern classification of sad facial processing: Toward the development of neurobiological markers in depression. Biological Psychiatry. 2008; 63:656–662. [PubMed: 17949689]

Girolami M, Rogers S. Variational bayesian multinomial probit regression with gaussian process priors. Neural Computation. 2006; 18:1790–1817.

Guyon I, Elisseeff A.e. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research. 2003; 3:1157–1182.

Hardoon DR, Mourao-Miranda J, Brammer M, Shawe-Taylor J. Unsupervised analysis of fMRI data using kernel canonical correlation. Neuroimage. 2007; 37:1250–1259. [PubMed: 17686634]

Harville DA. Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems. Journal of the American Statistical Association. 1977; 72:320–338.

Hassabis D, Chu C, Rees G, Weiskopf N, Molyneux PD, Maguire EA. Decoding Neuronal Ensembles in the Human Hippocampus. Curr Biol. 2009

Haxby JV, Gobbini MI, Furey ML, Ishai A, Schouten JL, Pietrini P. Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science. 2001; 293:2425–2430. [PubMed: 11577229]

Haynes JD, Rees G. Decoding mental states from brain activity in humans. Nat Rev Neurosci. 2006; 7:523–534. [PubMed: 16791142]

Haynes JD, Sakai K, Rees G, Gilbert S, Frith C, Passingham RE. Reading hidden intentions in the human brain. Curr Biol. 2007; 17:323–328. [PubMed: 17291759]

Klein A, Andersson J, Ardekani BA, Ashburner J, Avants B, Chiang MC, Christensen GE, Collins DL, Gee J, Hellier P, Song JH, Jenkinson M, Lepage C, Rueckert D, Thompson P, Vercauteren T, Woods RP, Mann JJ, Parsey RV. Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. Neuroimage. 2009; 46:786–802. [PubMed: 19195496]

Kriegeskorte N, Mur M, Ruff DA, Kiani R, Bodurka J, Esteky H, Tanaka K, Bandettini PA. Matching categorical object representations in inferior temporal cortex of man and monkey. Neuron. 2008; 60:1126–1141. [PubMed: 19109916]

LaConte S, Strother S, Cherkassky V, Anderson J, Hu X. Support vector machines for temporal classification of block design fMRI data. Neuroimage. 2005; 26:317–329. [PubMed: 15907293]

Logothetis NK, Pauls J, Augath M, Trinath T, Oeltermann A. Neurophysiological investigation of the basis of the fMRI signal. Nature. 2001; 412:150–157. [PubMed: 11449264]

Mackay DJC. The evidence framework applied to classification networks. Neural Computation. 1992; 4:720–736.

MacKay DJC. Probable networks and plausible predictions a review of practical Bayesian methods for supervised neural networks. Network: Computation in Neural Systems. 1995; 6:469–505.

Misaki M, Kim Y, Bandettini PA, Kriegeskorte N. Comparison of multivariate classifiers and response normalizations for pattern-information fMRI. Neuroimage. 2010; 53:103–118. [PubMed: 20580933]

Mitchell TM, Hutchinson R, Niculescu RS, Pereira F, Wang X, Just M, Newman S. Learning to decode cognitive states from brain images. Machine Learning. 2004; 57:145–175.

Mourao-Miranda J, Bokde AL, Born C, Hampel H, Stetter M. Classifying brain states and determining the discriminating activation patterns: Support Vector Machine on functional MRI data. Neuroimage. 2005; 28:980–995. [PubMed: 16275139]

Mourao-Miranda J, Friston KJ, Brammer M. Dynamic discrimination analysis: a spatial-temporal SVM. Neuroimage. 2007; 36:88–99. [PubMed: 17400479]

Mourao-Miranda J, Reynaud E, McGlone F, Calvert G, Brammer M. The impact of temporal compression and space selection on SVM analysis of single-subject and multi-subject fMRI data. Neuroimage. 2006; 33:1055–1065. [PubMed: 17010645]

Op de Beeck HP. Against hyperacuity in brain reading: spatial smoothing does not hurt multivariate fMRI analyses? Neuroimage. 2010; 49:1943–1948. [PubMed: 19285144]

Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines. Advances in Kernel Methods-Support Vector Learning. 1999; 208

Quinlan, JR. Morgan Kaufmann; San Francisco, CA: 1993. C4. 5: Programs for empirical learning..

Rasmussen, CE.; Williams, CKI. Gaussian Processes for Machine Learning. The MIT Press; 2006.

Shawe-Taylor, J.; Cristianini, N. Kernel Methods for Pattern Analysis. Cambridge University Press; 2004.

Strother S, La Conte S, Kai Hansen L, Anderson J, Zhang J, Pulapura S, Rottenberg D. Optimizing the fMRI data-processing pipeline using prediction and reproducibility performance metrics: I. A preliminary group analysis. Neuroimage. 2004; 23(Suppl 1):S196–207. [PubMed: 15501090]

Tipping ME. The Relevance Vector Machine. Advances in Neural Information Processing Systems. 2000; 12

Tipping ME. Sparse bayesian learning and the relevance vector machine. Journal of Machine Learning Research. 2001; 1:211–244.

Vapnik, V. The nature of statistical learning theory. NY Springer; 1995.

Vapnik, VN. Statistical Learning Theory. Wiley; 1998.

Weiskopf N, Sitaram R, Josephs O, Veit R, Scharnowski F, Goebel R, Birbaumer N, Deichmann R, Mathiak K. Real-time functional magnetic resonance imaging: methods and applications. Magn Reson Imaging. 2007; 25:989–1003. [PubMed: 17451904]

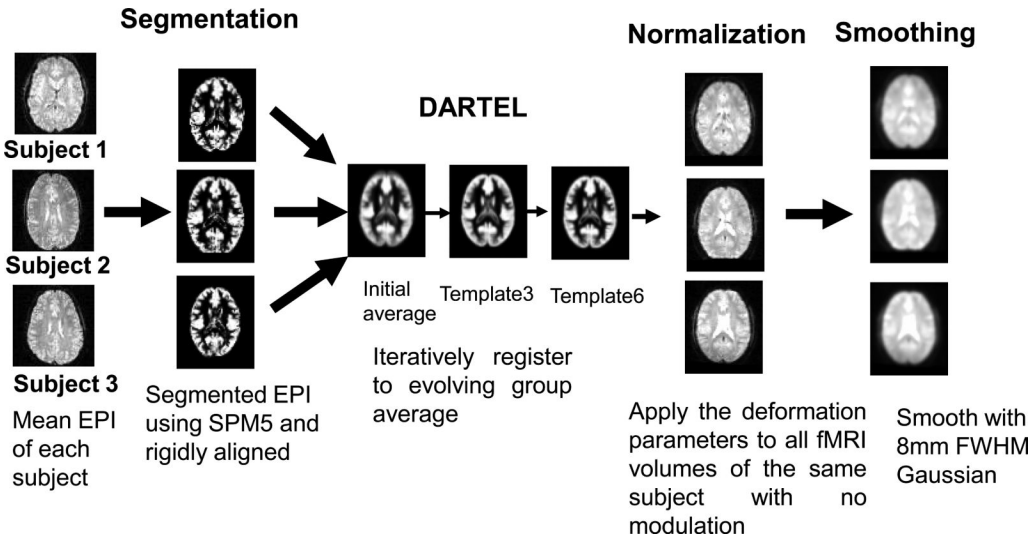

Figure 1.

The pipeline of normalizing the fMRI volumes into a common space (population average). The mean Echo Planner Images (EPI) were firstly segmented using unified segmentation. DARTEL was used to warp all 16 subjects into the evolving population average iteratively. The final deformation parameters were then applied to all the EPI of each subject without modulation.

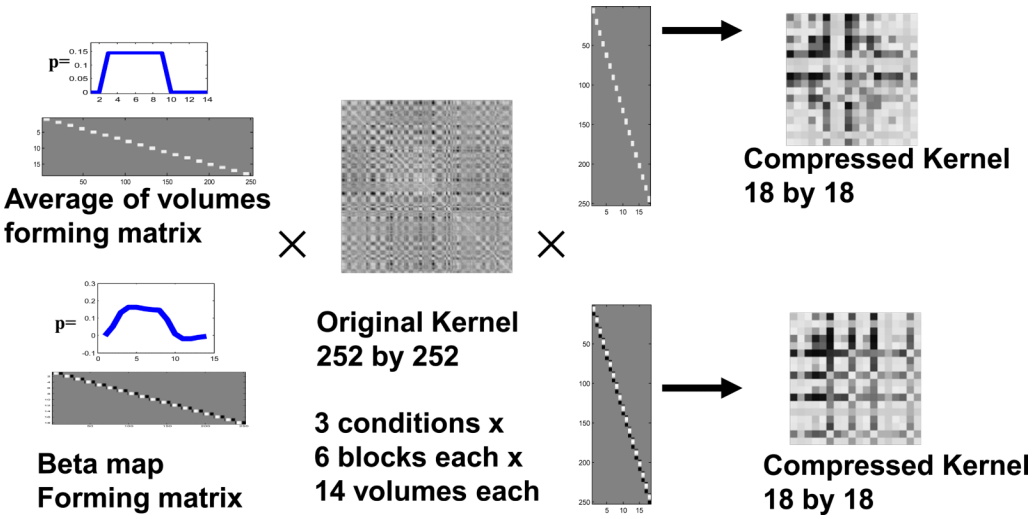

Figure 2.
Illustration of temporal compression using matrix operation.

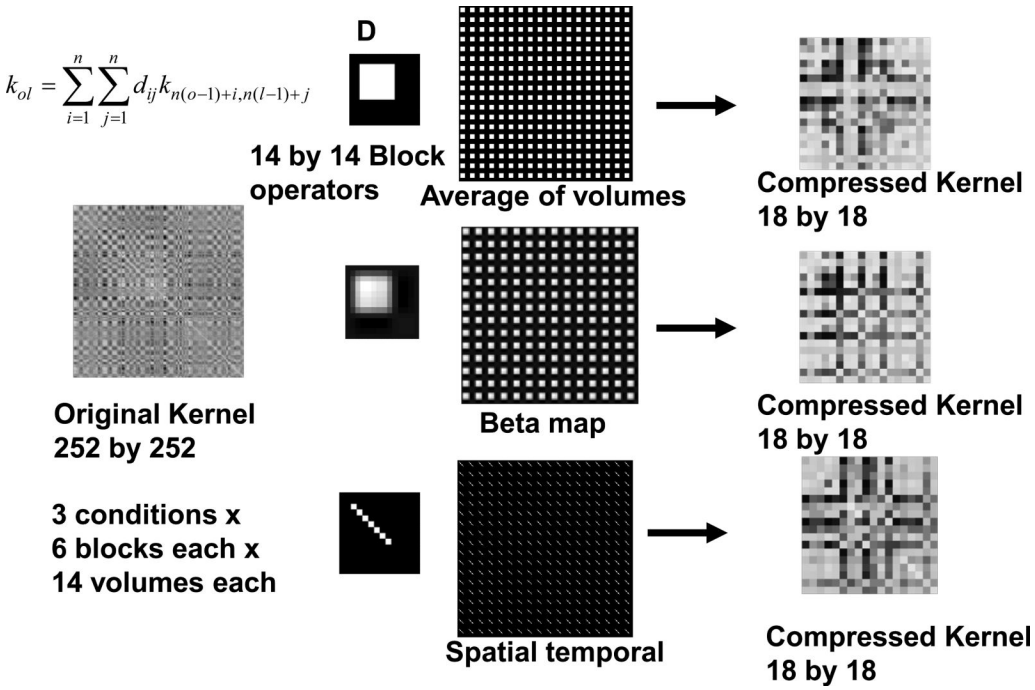

Figure 3.
Illustration of temporal compression and manipulation using a more generate operation. Spatial-temporal can only be derived using this operation.

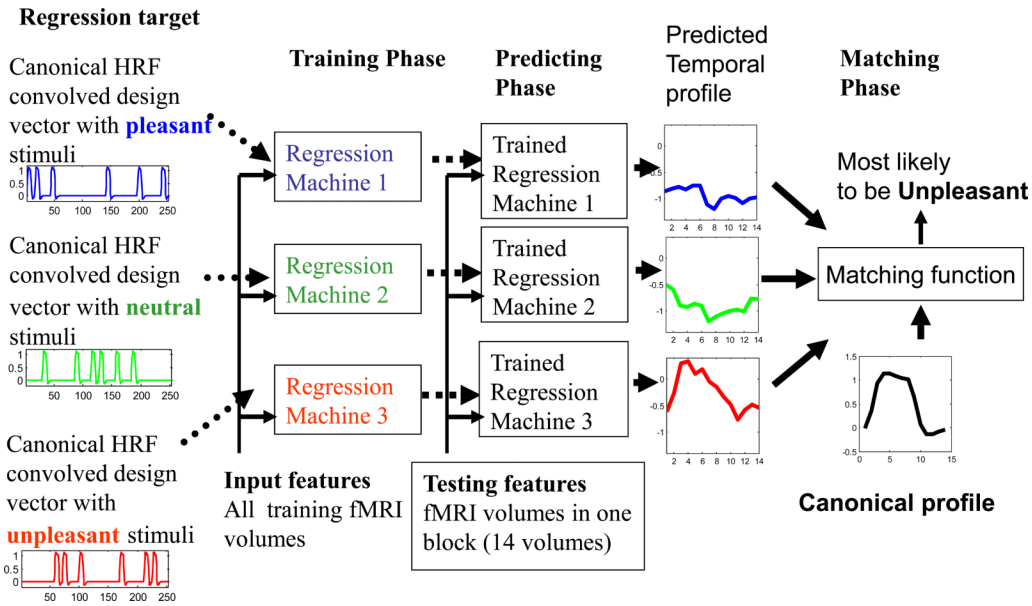

Figure 4.
Illustration of temporal compression and manipulation using a more generate operation. Spatial-temporal can only be derived using this operation.

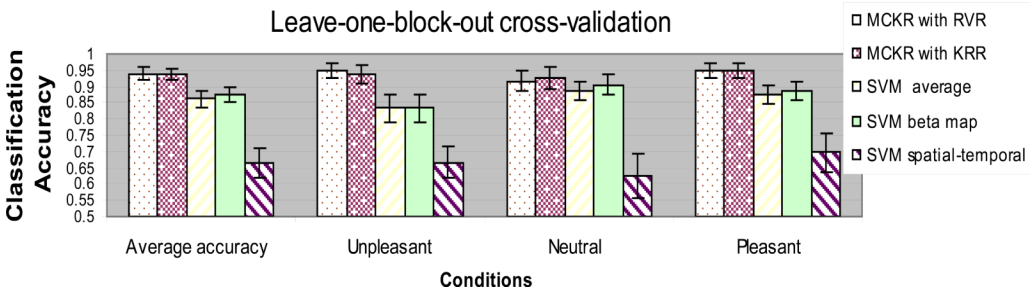

Figure 5.

Leave one block out cross-validation accuracies (within-subject cross-validation) averaged over 16 subjects with five different classifiers for different stimuli conditions. Error bars show the standard error.

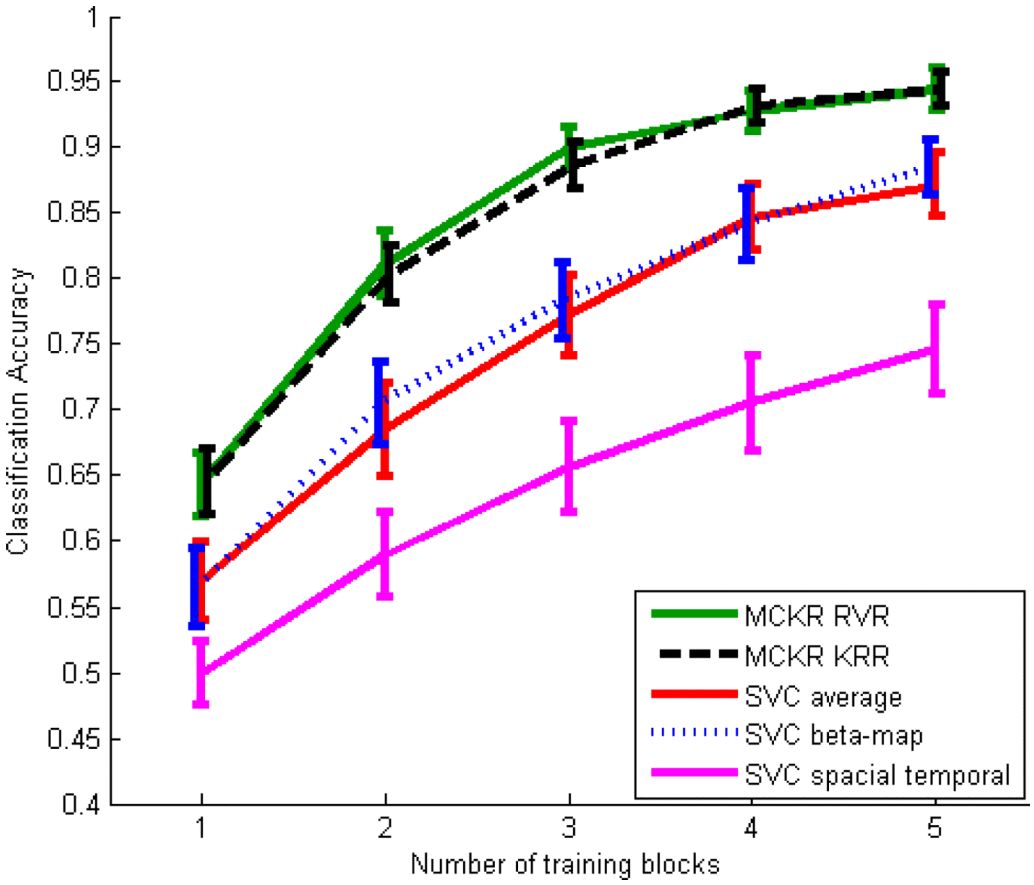

Figure 6.
Leave one block out cross-validation accuracies (within-subject cross-validation) with different size of training blocks. Each accuracy was calculated from averaging 40 cross-validation trials with randomly selected subsets. Error bars show the standard error.

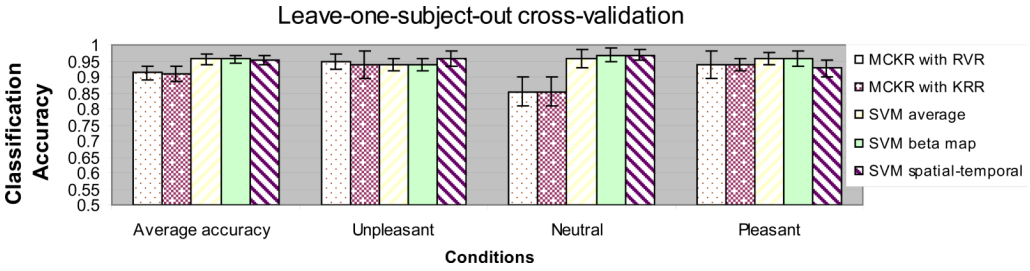

Figure 7.

Leave one subject out cross-validation (inter-subject cross-validation) accuracies averaged over 16 subjects with five different classifiers for different stimuli conditions. Error bars show the standard error.

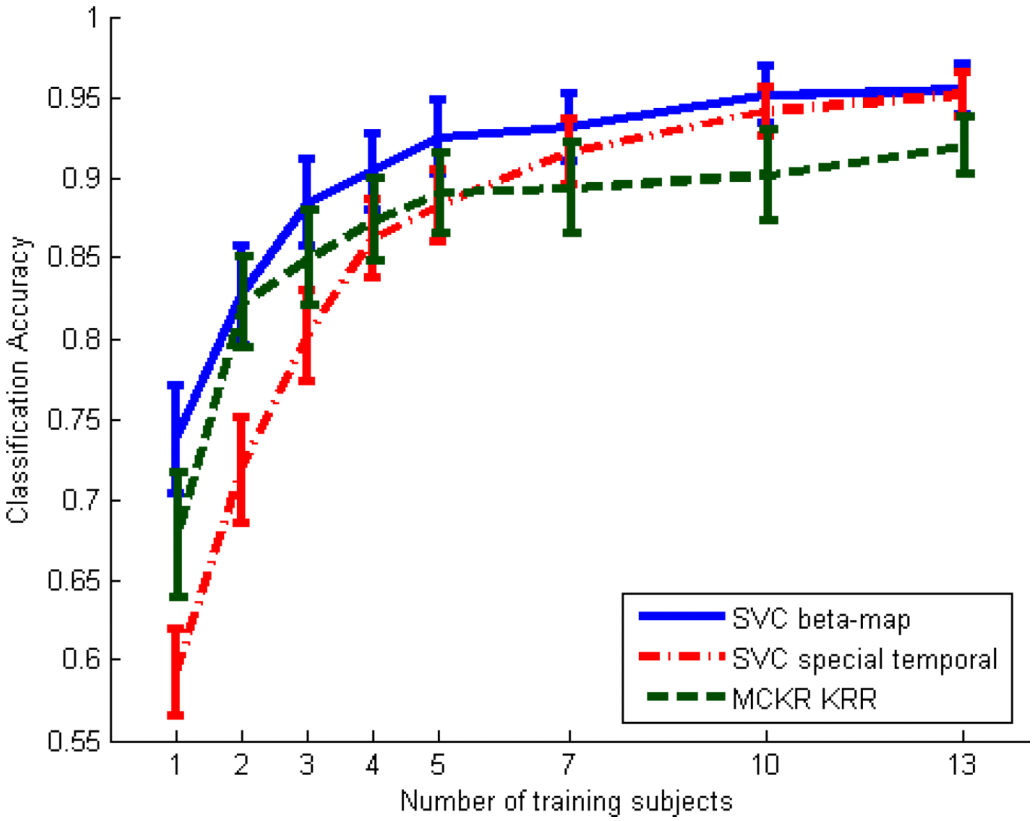

Figure 8.
Leave one subject out cross-validation accuracies with different size of training subjects. Each accuracy was calculated from averaging 40 cross-validation trials with randomly selected subsets. Error bars show the standard error.

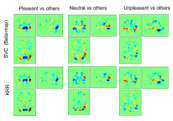

Figure 9.
The weight map (feature weights) from training all volumes of all subjects. The top row shows the feature map generated from SVC and the bottom row shows the feature map generated from KRR. From left column to right, the maps were generated form training pleasant stimuli versus the other two, neutral versus other two, and unpleasant versus other two, respectively.

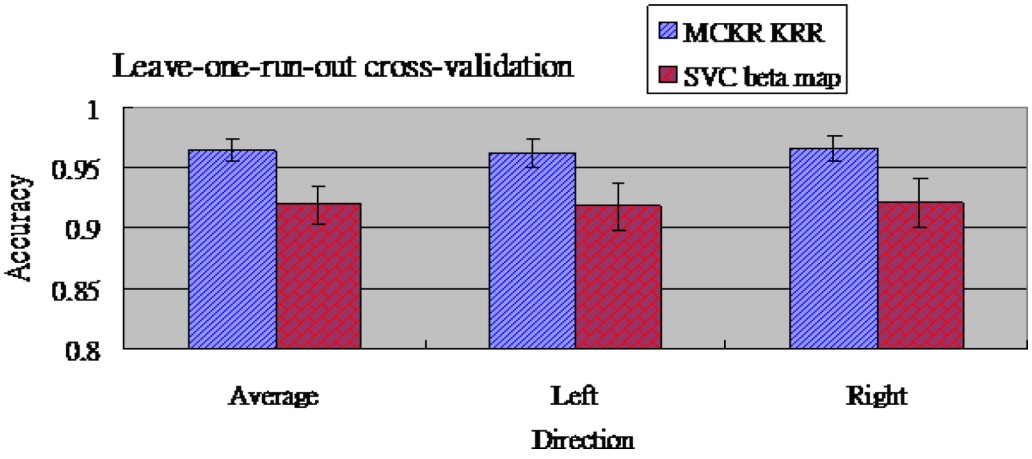

Figure 10.

Leave one run out cross-validation (within-subject cross-validation) accuracies averaged over 15 subjects (3 runs per subject) and with MCKR KRR and SVC beta map for different direction of visual attention. Error bars show the standard error.

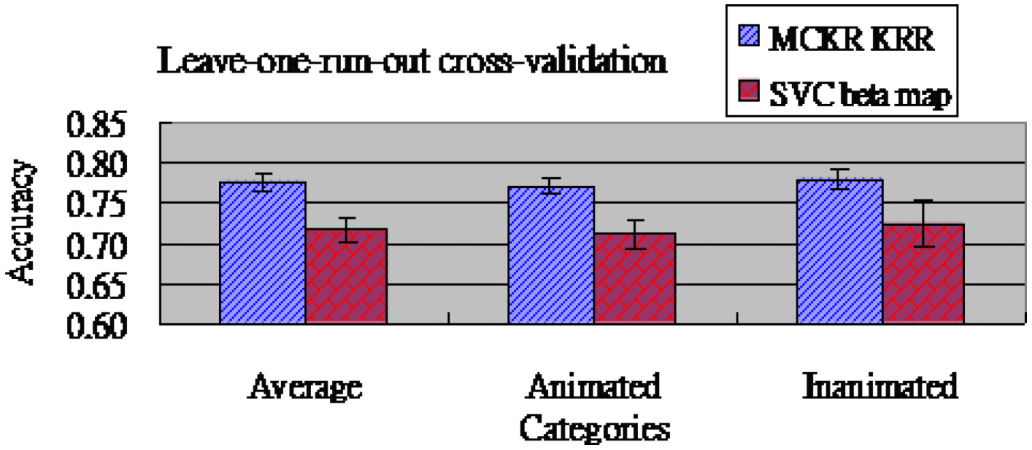

Figure 11.
Leave one run out cross-validation (within-subject cross-validation) accuracies averaged over 4 subjects (6 runs per subject) with MCKR KRR and SVC beta map for different categories of visual objects. Error bars show the standard error.

Table 1

Within-subject classification accuracy (%) of cross-validation

Multi-class classifiers	Conditions			
	Average Accuracy	Unpleasant	Neutral	Pleasant
MCKR with RVR	94.0(1.8)	94.8(2.4)	91.7(3.3)	94.8(2.4)
MCKR with KRR	93.8(1.8)	93.8(2.9)	92.7(3.3)	94.8(2.4)
SVC (average)	86.5(2.8)	83.3(2.8)	88.5(2.8)	87.5(3.0)
SVC (beta-map)	87.5(2.2)	83.3(4.2)	90.6(3.0)	88.5(2.8)
SVC (spatial temporal)	66.3(4.5)	66.7(5.0)	62.5(6.8)	69.8(6.0)

Values inside the parentheses are the standard error (%)

Table 2

Confusion matrix of MCKR with RVR and multi-class SVC compressed by “average of volume” within-subject.(LOBOCV)

Predicted class % MCKR with RVR		Actual		
		Unpleasant	**Neutral**	**Pleasant**
Predicted	Unpleasant	93.8	3.1	2.1
	Neutral	3.1	91.7	3.1
	Pleasant	3.1	5.2	94.8

Predicted class % Multi-class SVC (average of volume)		Actual		
		Unpleasant	**Neutral**	**Pleasant**
Predicted	Unpleasant	83.3	4.2	8.3
	Neutral	5.2	88.5	4.1
	Pleasant	11.5	7.3	87.5

Table 3

Inter-subject classification accuracy (%) of cross-validation

Multi-class classifiers	Conditions			
	Average Accuracy	Unpleasant	Neutral	Pleasant
MCKR with RVR	91.3(2.2)	94.8(2.4)	85.4(4.6)	93.8(4.1)
MCKR with KRR	91.0(2.2)	93.8(4.1)	85.4(4.6)	93.8(2.0)
SVC (average)	95.5(1.5)	93.8(2.0)	95.8(2.8)	95.8(1.8)
SVC (beta-map)	95.5(1.4)	93.8(2.0)	96.9(2.2)	95.8(2.3)
SVC (spatial-temporal)	95.1(1.3)	95.8(2.3)	96.9(1.6)	92.7(2.5)

Values inside the parentheses are standard error (%)

Table 4

Confusion matrix of MCKR with RVR and multi-class SVC compressed by “average of volume” inter-subject (LOSOCV).

Predicted class % MCKR with RVR		Actual		
		Unpleasant	**Neutral**	**Pleasant**
Predicted	Unpleasant	94.8	9.4	6.3
	Neutral	0	85.4	0
	Pleasant	5.2	5.2	93.7

Predicted class % Multi-class SVC (average of volume)		Actual		
		Unpleasant	**Neutral**	**Pleasant**
Predicted	Unpleasant	93.7	2.1	1.1
	Neutral	4.2	95.8	3.1
	Pleasant	2.1	3.1	95.8