Learning with Kernel Random Field and Linear SVM

Haruhisa Takahashi
The University of Electro-Communications, Choufushi Tokyo 182-8585, Japan

Keywords: Autocorrelation Kernel, MRF, Mean-field, Fisher Score, Deep Learning, SVM.

Abstract: Deep learning methods, which include feature extraction in the training process, are achieving success in pattern recognition and machine learning fields but require huge parameter setting, and need the selection from various methods. On the contrary, Support Vector Machines (SVMs) have been popularly used in these fields in light of the simple algorithm and solid reasons based on the learning theory. However, it is difficult to improve recognition performance in SVMs beyond a certain level of capacity, in that higher dimensional feature space can only assure the linear separability of data as opposed to separation of the data manifolds themselves. We propose a new framework of kernel machine that generates essentially linearly separable kernel features. Our method utilizes pretraining process based on a kernel generative model and the mean field Fisher score with a higher-order autocorrelation kernel. Thus derived features are to be separated by a liner SVM, which exhibits far better generalization performance than any kernel-based SVMs. We show the experiments on the face detection using the appearance based approach, and that our method can attain comparable results with the state-of-the-art face detection methods based on AdaBoost, SURF, and cascade despite of smaller data size and no preprocessing.

1 INTRODUCTION

A deep architecture is a recent trend in classification problems to obtain finally flexible linearly separable features against previous trend depending on rich descriptors such as SIFT, HOG, or SURF (Bengio et.al. 2012). The linear separability of data is a milestone for a good feature representation, and is realized by a higher dimensional kernel feature space in SVM as well as stacked layered representation in neural networks.

However, linear separability of training examples can not necessarily result in a good generalization if the feature extractor is not fitted in the probability distribution of instances. For example, in SVMs, kernels are required to reflect on the distribution of instances for better generalization, and yet they are hard to come by. Fisher kernel (Jaakkola & Haussler 1998) is an exceptional example, in that it is based on a generative model, but not popularly used due to its computational cost and speciality. There are other researches to combine the generative models with kernels but they take in similar problems (Lafferty Zhu & Liu 2004),(Lafferty McCallum & Pereira 2004),(Roscher 2010).

Although SVM is a general-purpose nonlinear classifier where its kernel feature space can linearly separate training data, the data manifolds themselves are not necessarily separated each other. This is a reason why the deep learning often outperforms SVM in classification capacity. In this observation we aim at a feature extractor based on data distribution, which can give linearly separable data manifolds to be combined with linear SVM classifier.

We directly use Fisher score of Markov random field (MRF) as a feature extractor that can give essentially linearly separable representation of the problem. In order to represent data manifolds effectively we need to transform MRF into a kernel based representation. Fortunately, the cliques in MRF are units of computing higher order autocorrelation, from which we can reach a definition of a novel autocorrelation kernel and the concept of kernel random fields (KRF). In this context Fisher score can be represented as a simple form in terms of the autocorrelation kernels and their differences with the mean value of KRF. Finally we propose a SVM-like classiier defined by a linear SVM applied on this feature.

We propose an efficient algorithm to compute the autocorrelation kernel in the linear order for the input length. KRF is computed by using the variational mean field that leads the Fisher score to a simple ker-

Takahashi H..
Learning with Kernel Random Field and Linear SVM.
DOI: 10.5220/0004792601670174
In *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2014), pages 167-174
ISBN: 978-989-758-018-5
Copyright © 2014 SCITEPRESS (Science and Technology Publications, Lda.)

nel difference. We show with computer experiments on the face discrimination problem that our model performs better than SVM, and can give comparable results with specifically tuned face detectors, despite the smaller training data size.

2 OUTLINE OF THE MODEL

Data manifolds tend to be flexibly linearly separated by a deep architecture, while indeed data representation in SVM or multilayer neural networks are linearly separated, but data manifolds are not necessarily separated. In order to obtain a linearly separable representation, the feature extractor should be designed so as to have a potential that takes smaller values for the object class instances than for the out-of-class instances. For example, in Auto-Encoder or Sparse Coding, difference between the input vectors and the reconstructed vectors is designed to be minimum, thus becomes a potential on the data. In a deep architecture, such feature extractors are stacked, and enabling to regularize the potential (Erhan et.al. 2010), making the data manifolds be linearly separated.

Another idea to obtain a regularized feature potential comes from the generative models. Fisher kernel is the one that reflects this idea, and defined by the inner product of Fisher score, although a problem of computational efficiency is involved. Given a model $P(x|\theta)$ Fisher score is defined by

$$\lambda_\theta(x) = \frac{\partial \log P(x|\theta)}{\partial \theta} \tag{1}$$

The vector score of Eq.(1) takes a value near 0 for an input vector x of the class, if the parameter θ is properly trained as a model of the class. This implies that Fisher kernel is an over-transformed representation in the sense that kernel values of x_1, x_2 near 0 does not necessarily mean both x_1 and x_2 belong to the class. In this reason we propose a method to directly utilize Fisher score with MRF unlike Fisher kernel method such as (Jaakkola & Haussler 1998).

MRF models objects with auto-correlative units called cliques, and have been applied for wide range of signal processing or pattern analysis fields. If we adopt MRF as a basic generative model for calculating Fisher score, we must have suffer from combinatorially huge number of model parameters corresponding to the number of cliques.

A compact expression of features can be obtained from the kernel representation of the random field. In order to take the autocorrelations of cliques into kernels, we will define a feature vector consist of cliques in section 3, and the kernel is defined by the inner product of feature vectors. Note that our definition of autocorrelation kernel does not include the second or a higher order of single variables. This reduces the computational complexity of the kernel to linear order.

The higher autocorrelation kernel introduced in section 3 is able to reflect a difference of higher order autocorrelations, while the popular Gaussian or polynomial kernel depends only on the difference of input vector values. A higher order autocorrelation kernel was examined in (Horikawa 2004), in which direct inner product of feature vectors of higher order autocorrelations was used, besides higher order moments of single variables were used in his setting. Thus it requires huge computational efforts.

If we use our definition of higher autocorrelation kernel, we can define Kernel Random Field (KRF) for n dimensional discrete states x as follwos:

$$P(x|\mu) = \frac{1}{Z}\exp\left(-\sum_{\ell=1}^{m}\mu_\ell K(\xi_\ell, x)\right) \tag{2}$$

which is equivalent to MRF, where Z is the partition function, ξ_ℓ are m training examples, and μ_ℓ are the model parameters. For the practical situations, computation of KRF of eq.(2) is hard for seeking Z. We apply the mean field approximation to derive the mean field Fisher score expression in section 5:

$$\lambda_{\ell'}(\xi_\ell) = K(\xi_{\ell'}, \xi_\ell) - K(\xi_{\ell'}, \bar{x}) \tag{3}$$

where $\bar{x}, \xi_{\ell'}, \xi_\ell$ are the mean of the states in the mean field, ℓ'th in-class training instance, and ℓth training instance, respectively. In fact, on one hand, for the in-class instances ξ_ℓ the first and the second terms in the right hand side of eq.(3) take similar (comparatively large) values, and the subtraction results in near 0. On the other hand, if ξ_ℓ is an outside-the-class instance, the first term of the right hand side of eq.(3) takes a small value, and the subtraction results in negatively large. As the result, the features of eq.(3) becomes linearly separable, because the problem is reduced to the majority voting with the negatively continuous values.

We propose a learning scheme using a linear SVM to discriminate the Fisher score features given in eq.(3). Then the training process is divided in two steps; in the first step the mean field KRF is trained for the class data, and in the second step, the linear SVM is trained on features of eq.(3) using the all training data. We will show computer experiments on the face detection problem in section 6, and show that the proposed scheme works well to get far better results than SVMs. The results are comparable to a state-of-the-art face detection system using SURF, cascade, and AdaBoost (Li & Zhang 2013).

3 HIGHER AUTOCORRELATION KERNEL

The autocorrelation kernel previously used one (Horikawa 2004) was simply computed by an inner product of feature vectors. The dimension of the feature vectors is $O(n^d)$ according to an input vector size n and a fixed order of autocorrelation $d \ll n$. In our definition of higher autocorrelation kernel, we excluded the second or a higher moment of single variables from the feature vectors. In light of this we can compute the higher autocorrelation kernel in $O(n)$ for fixed $d \ll n$, which is shown below.

Let an input vector be $x = (x_1, ..., x_n)^t$. Then we define a higher autocorrelation feature vector up to dth order as

$$\begin{aligned}\phi(x) &= (1; x_1, ...x_n; x_1x_2, x_1x_3, ..., x_{n-1}x_n; ...; \\ &\quad ..., x_{n-d+1}\cdots x_d)^t \end{aligned} \tag{4}$$

where the general term representing dth autocorrelation is $x_{i_1}x_{i_2}\cdots x_{i_d}, (i_1 < i_2, ..., < i_d)$. Then the higher autocorrelation kernel is defined by $K(x,z) = \phi(x)^t \cdot \phi(z)$. We will show an efficient computational algorithm of $K(x,z)$ in the followings.

3.1 Expression with Symmetric Polynomial

The higher autocorrelation kernel can be represented using the symmetric polynomial given by

$$\begin{aligned} S_0(x) &= 1 \\ S_1(x) &= x_1 + ... + x_n \\ S_2(x) &= x_1x_2 + x_1x_3 + ... + x_{n-1}x_n \\ S_3(x) &= x_1x_2x_3 + x_1x_2x_4 + ... + x_{n-2}x_{n-1}x_n \\ &\quad \cdots\cdots\cdots \end{aligned}$$

Let $y_i = x_iz_i$. Then

$$K(x,z) = \sum_{i=0}^{d} S_i(y),\ y = (y_1, ..., y_n)^t \tag{5}$$

If degree of the kernel should be specified, it would be written as $K_d(x,z)$. Notice that the general dth degree polynomial kernel $(< x,z > +c)^d$ takes $O(n)$, but the direct computation of eq.(5) takes $O(n^d), d \ll n$ proportional to the number of monomials.

3.2 Computational Algorithm of $O(n)$

We show the next lemma reducing the computation of eq.(5) to a recursive formula.

Lemma 1. *Let $d \leq n$ be fixed. Then S_d is computed from S_k,$(d/2 \geq k)$ in $O(n)$ as*

$$S_d = \sum_{i=[d/2]_-+1}^{n-[d/2]_++1} \hat{S}^{i-1}_{[d/2]_-} S^i_{[d/2]_+}$$

where $[\cdot]_-$, $[\cdot]_+$ represent the floor and the ceil integers, respectively, and

$$\begin{aligned} \hat{S}^i_k &= \hat{S}^{i+1}_k - x_{i+1}\hat{S}^i_{k-1}\ \ (i = n-1, ..., k),\ \ \hat{S}^n_k = S_k \\ S^i_k &= x_i \sum_{j=i+1}^{n} S^j_{k-1} \end{aligned}$$

(proof) First we show the computational complexity. Sum of $\hat{S}^{i-1}_{[d/2]_-}$ and $S^i_{[d/2]_+}$ is $O(n)$. For fixed k, both of $\hat{S}^i_k$ and S^i_k is computed in $O(n)$. The number of recursive iteration to compute these factors is less than or equal to $\log_2 d$. Thus totally the computational complexity is $O(n)$. In order to show formally the recursive formulae in the lemma, we can use the induction. However, as it makes too much complicated, we satisfied with exemplifying $d = 1, ..., 4$ in the following. Q.E.D.

Example S_1: Note that $S^i_1 = x_i,\ \ (i = 1, ..., n)$, thus $S_1 = \sum_{i=1}^n S^i_1$. Then $\hat{S}^i_1$ becomes the first degree symmetric polynomial of $x_1, ..., x_i$, and

$$\begin{aligned} \hat{S}^i_1 &= \hat{S}^{i+1}_1 - x_{i+1},\ \ (i = n-1, ..., 1), \\ &= x_1 + ... + x_i \end{aligned}$$

Thus $\hat{S}^n_1 = S_1$.

S_2: S_2 is computed using $S^i_1, \hat{S}^i_1$ as

$$S_2 = \sum_{i=2}^{n} \hat{S}^{i-1}_1 S^i_1$$

Now S^i_2 is the sum of terms containing both x_i and x_j, $(i < j)$ simultaneously in S_2. Thus

$$\begin{aligned} S^i_2 &= x_i \sum_{j=i+1}^{n} S^j_1 \\ &= x_i(x_{i+1} + .. + x_n),\ \ (i = 1, ..., n-1), \end{aligned}$$

Then $\hat{S}^i_2$ is defined as the second degree symmetric polynomial of $x_1, ..., x_i$, and determined by

$$\hat{S}^i_2 = \hat{S}^{i+1}_2 - x_{i+1}\hat{S}^i_1\ \ (i = n-1, ..., 2),\ \ \hat{S}^n_2 = S_2$$

S_3 and S_4: Since S_3 is sum of the product of S^i_2 (S^i_1) and $\hat{S}^{i-1}_1$ $(\hat{S}^{i-1}_2)$ for all i,

$$S_3 = \sum_{i=2}^{n-1} \hat{S}^{i-1}_1 S^i_2 = \sum_{i=2}^{n-1} \hat{S}^i_2 S^{i+1}_1$$

Now S^i_3 is sum of terms containing both x_i and x_j, $(i < j)$ simultaneously in S_3. Thus

$$S^i_3 = x_i \sum_{j=i+1}^{n} S^j_2$$

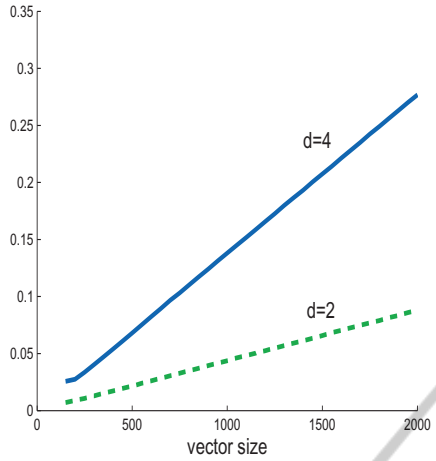

Figure 1: Computational time of higher autocorrelation kernels for size n input vectors.

Further, $\hat{S}_3^i$ is defined as the third degree symmetric polynomial of $x_1,...,x_i$, and determined by

$$\hat{S}_3^i = \hat{S}_3^{i+1} - x_{i+1}\hat{S}_2^i \ (i = n-1,...,3),\ \hat{S}_3^n = S_3$$

Similarly we can find

$$\begin{aligned} S_4 &= \sum_{i=2}^{n-2} \hat{S}_1^{i-1} S_3^i = \sum_{i=4}^{n} \hat{S}_3^{i-1} S_1^i \\ &= \sum_{i=3}^{n-1} \hat{S}_2^{i-1} S_2^i \end{aligned}$$

and

$$\begin{aligned} \hat{S}_4^i &= \hat{S}_4^{i+1} - x_{i+1}\hat{S}_3^i \ (i = n-1,...,4),\ \hat{S}_4^n = S_4 \\ S_4^i &= x_i \sum_{j=i+1}^{n} S_3^j \end{aligned}$$

In general we need $[k/2]_+$th degree S_k^i, $\hat{S}_k^i$ for seeking kth degree symmetric polynomial.

Figure 1 shows computational time of autocorrelation kernels for vectors of size n, which are randomly generated. We can confirm $O(n)$ from this experiment. It is important in applying the autocorrelation kernel that either the input vectors should be normalized with norms or the kernels are normalized as

$$\hat{K}(x,z) = \frac{K(x,z)}{\sqrt{K(x,x)K(z,z)}}$$

Autocorrelation kernel must be advantageous for pattern discrimination tasks. We compared the autocorrelation kernel with the polynomial kernel using SVM classifier for real data of face detection. We randomly chose 2000 face data and 2000 non-face data from training data of CMU+MIT dataset (see section 6). We leave out 400 face and non-face data from them, and remaining 1600 data for each class are used for training. Figure 2 compares the ROC curves for 7th degree normalized autocorrelation and polynomial kernels. The autocorrelation kernel shows much better discrimination performance than the polynomial kernel. The discrimination performance of (normalized) polynomial kernel is same or similar for the degree greater than or equal to 3.

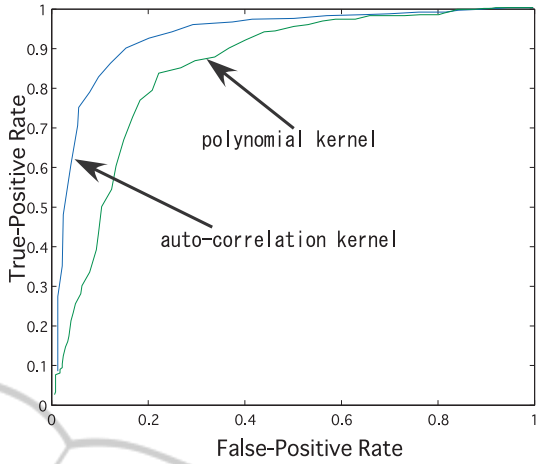

Figure 2: ROC curves of normalized autocorrelation and polynomial kernels (7th degree) with SVM.

4 MEAN FIELD KRF

In this section we introduce computationally feasible feature extraction based on MRF.

4.1 Variational Mean Field of KRF

We derive a variational mean field expression of KRF. Let the logarithmic potential function of KRF of eq.(2) be

$$V_\mu(x) = -\sum_{\ell=1}^{m} \mu_\ell K(\xi_\ell, x), \qquad (6)$$

and let states x_i take values $k \in \{0,1,2,...,r-1\}$. Let $Q(x|\mu) = \prod_{i=1}^n q_i^{x_i}$ be the mean field probability function, in which $q_i^0 = 1 - \sum_{k=1}^{r-1} q_i^k$, and $r_i^k = P(x_i = k|\mu)$ be the marginal probability function for KRF given by $P(x|\mu)$. Then the following lemma holds.

Lemma 2. *Let $\boldsymbol{E}_P\{V_\mu(x)\}$ be the mean of $V_\mu(x)$ with respect to $P(x|\mu)$. Then $Q(x|\mu)$ satisfies*

$$q_i^k = \frac{\exp\left(-\frac{\partial \mathbf{E}_P\{V_\mu(x)\}}{\partial q_i^k}\right)}{1 + \sum_{k'=1}^{r-1} \exp\left(-\frac{\partial \mathbf{E}_P\{V_\mu(x)\}}{\partial q_i^{k'}}\right)}$$

(proof) Mean field probability Q is derived minimizing the following KL divergence

$$KL(P,Q) = \sum_s P(x|\mu)\ln\frac{P(x|\mu)}{Q}$$
$$= -\sum_{i=1}^{n}\sum_{k=1}^{r-1} r_i^k \ln q_i^k - \mathbf{E}_P\{V_\mu(x)\} - \ln Z \quad (7)$$

Taking the partial derivative of the right hand side in the second equality of eq.(7) w.r.t. q_i^k, and putting 0, we can find that $r_i^k = q_i^k$. Placing this relation back into eq.(7), and taking the partial derivative q.r.t. q_i^k to put 0, the resulting equation gives the lemma.

Q.E.D.

Now let the mean of $V_\mu(x)$ on $Q(\cdot|\mu)$ be $\mathbf{E}_Q\{V_\mu(x)\}$. We will use an approximation that the partial derivative of $\mathbf{E}_P\{V_\mu(x)\}$ w.r.t. q_i^k is replaced by the partial derivative of $\mathbf{E}_Q\{V_\mu(x)\}$ w.r.t. q_i^k.

4.2 Mean Field Potential

We take advantage of the d linearity of the autocorrelation kernels to derive mean field potential of KRF.

Lemma 3. *Let the expression $\bar{\cdot}$ represent the mean by the marginal probability function $\prod_{i=1}^n r_i^{x_i}$. Then*

$$\begin{aligned}\overline{K(\xi_\ell,x)} &= K(\xi_\ell,\bar{x})\\ \overline{K(\xi_\ell,x^{(i)}[k])} &= K(\xi_\ell,\bar{x}^{(i)}[k])\end{aligned}$$

where the following notation is used

$$z^{(i)}[y] = (z_1,...,z_{i-1},y,z_{i+1},...,z_n)^t$$

(proof) From colinearity of the inner product

$$\overline{K(\xi_\ell,x)} = <\phi(\xi_\ell),\overline{\phi(x)}>$$

Taking the mean w.r.t. each entry $x_{i_1}\cdots x_{i_k}$ of the vector $\phi(x)$,

$$\overline{x_{i_1}\cdots x_{i_k}} = \bar{x}_{i_1}\cdots\bar{x}_{i_k},\ (i_1<...<i_k)$$

and the lemma is proved. Q.E.D.

Lemma 4. *For $\mu = o(\mu)$*

$$\frac{\partial r_i^k}{\partial \mu_{\ell'}} = r_i^k\left(K(\xi_{\ell'},\bar{x}^{(i)}[k]) - K(\xi_{\ell'},\bar{x})\right)$$

(proof) The marginal probability is given by

$$r_i^k = \sum_{x^{(i)}[k]} P(x|\mu) = \frac{\sum_{x^{(i)}[k]}\exp\left(\sum_{\ell=1}^m \mu_\ell K(\xi_\ell,x)\right)}{Z}$$

For $\mu = 0$ KRF becomes the independent field from eq.(2). Thus $P(x|\mu)\to\prod_{i=1}^n r_i^{x_i}$ for $\mu = o(\mu)$. Using Lemma 3

$$\begin{aligned}\frac{\partial r_i^k}{\partial \mu_{\ell'}} &= \sum_{x^{(i)}[k]} K(\xi_{\ell'},x)P(x|\mu)\\ &\quad - \sum_{x^{(i)}[k]} P(x|\mu)\sum_x K(\xi_{\ell'},x)P(x|\mu)\\ &= \sum_{x^{(i)}[k]} K(\xi_{\ell'},x)P(x|\mu) - r_i^k\overline{K(\xi_{\ell'},x)}\\ &= r_i^k\left(K(\xi_{\ell'},\bar{x}^{(i)}[k]) - K(\xi_{\ell'},\bar{x})\right)\end{aligned}$$

Q.E.D.

Lemma 5. *For $\mu = o(\mu)$*

$$\left.\frac{\partial \mathbf{E}_P\{V_\mu(x)\}}{\partial\mu_\ell}\right|_{o(\mu)} = \left.\frac{\partial \mathbf{E}_r\{V_\mu(x)\}}{\partial\mu_\ell}\right|_{o(\mu)} = -K_d(\xi_\ell,\bar{x})$$

where $\mathbf{E}_r$ is the mean by the marginal $\prod_{i=1}^n r_i^{x_i}$.

(proof) Since $P(x|\mu)\to\prod_{i=1}^n r_i^{x_i}$ for $\mu = o(\mu)$, the first equality holds in the lemma. From Lemma4 $\frac{\partial r_i^{x_i}}{\partial\mu_\ell}$ converges when $\mu\to 0$. Thus

$$\begin{aligned}&\frac{\partial \mathbf{E}_r\{V_\mu(x)\}}{\partial\mu_\ell} = -K_d(\xi_\ell,\bar{x})\\ &-\sum_{i=1}^n\sum_{\ell=1}^m\sum_x \mu_\ell K_d(\xi_\ell,x) r_1^{x_1}...\frac{\partial r_i^{x_i}}{\partial\mu_\ell}...r_n^{x_n}\\ &= -K_d(\xi_\ell,\bar{x}) + o(\mu)\end{aligned}$$

Q.E.D.

From Lemma 5, if we take the first term of Maclaurin expansion w.r.t. μ

$$\mathbf{E}_P\{V_\mu(x)\} = -\sum_{\ell=1}^m K(\xi_\ell,\bar{x})\mu_\ell \quad (8)$$

We denote the vector that is constructed by removing ith entry from a vector z as

$$z^{(i)} = (z_1,...,z_{i-1},z_{i+1},...,z_n)^t$$

Then

$$\frac{\partial K_d(\xi_\ell,\bar{x})}{\partial r_i^k} = k\xi_\ell^i K_{d-1}(\xi_\ell^{(i)},\bar{x}^{(i)}) \quad (9)$$

where ξ_ℓ^i is the ith entry of ξ_ℓ. From Lemma 2, lemma 3, and Lemma 5, we obtain the mean field equation

$$r_j^k = \frac{\exp\left(\sum_{\ell=1}^m k\xi_\ell^j K_{d-1}(\xi_\ell^{(j)},\bar{x}^{(j)})\mu_\ell\right)}{1+\sum_{k'=1}^{r-1}\exp\left(\sum_{\ell=1}^m k'\xi_\ell^{(j)}K_{d-1}(\xi_\ell^{(j)},\bar{x}^{(j)})\mu_\ell\right)} \quad (10)$$

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

The iterative method can applied to eq.(10) based on a numerical analysis of the differential equation that has an equilibrium as eq.(10). For computing eq.(10), the evaluation of $(d-1)$th degree kernel will be needed, and when n is large this takes huge computational efforts. However, if we notice that the kernels in eq.(10) should be evaluated by vectors of $z^{(i)}$ type variables, the essential computational time reduces to computing one kernel of $K_{d-1}(\xi_\ell, \overline{x})$. The method is shown in the Appendix.

5 FEATURES WITH KRF

KRF is trained by one class data, and can be applied as a feature extractor. In this section we derive the mean field expression of the maximum likelihood estimation and Fisher score as a feature extractor.

5.1 Maximum Likelihood Estimation

Given training data $\{\xi_1, ..., \xi_m\}$, we seek the parameter that maximize the empirical log likelihood

$$L(\mu) = \sum_{\ell=1}^{m} \log P(\xi_\ell|\mu)$$

If we apply the mean field $Q(x|\mu)$ instead of KRF $P(x|\mu)$,

$$\begin{aligned}\frac{\partial L(\mu)}{\partial \mu_{\ell'}} &= \sum_{\ell=1}^{m}\left(-\frac{\partial V_\mu(\xi_\ell)}{\partial \mu_{\ell'}} + \sum_x \frac{\partial V_\mu(x)}{\partial \mu_{\ell'}} P(x|\mu)\right) \\ &= \sum_{\ell=1}^{m}\left(K(\xi_{\ell'}, \xi_\ell) - \sum_x K(\xi_{\ell'}, x) Q(x|\mu)\right) \\ &= \sum_{\ell=1}^{m}\left(K(\xi_{\ell'}, \xi_\ell) - K(\xi_{\ell'}, \overline{x})\right) \qquad (11)\end{aligned}$$

where $\ell' = 1, ..., k$, k being the number of kernels used in KRF. As is shown in section 4, eq. (11) holds for a small μ. We can perform the steepest ascent method using eq.(11) for the optimum μ.

5.2 Mean Field Fisher Score

We use the mean field KRF as a feature extractor. Since the log likelihood of properly trained KRF is maximized, the Fisher (vector) score given by eq.(11) is small for a class vector ξ_ℓ, and large for an outside-the-class vector. Specifically, we choose $\{\xi_1, ..., \xi_k\}$ for $k \leq m$ as the kernel data, that is, for $\ell' = 1, ..., k$

$$\frac{\partial \log P_\mu(\xi_\ell)}{\partial \mu_{\ell'}} = K(\xi_{\ell'}, \xi_\ell) - K(\xi_{\ell'}, \overline{x}) \qquad (12)$$

are the k dimensional feature of each ξ_ℓ, where $\overline{x}$ is the mean of states with $Q(x|\mu)$. As was explained in section 2, this feature is essentially linearly separable. In general, the number of kernels k in KRF can be determined so as to let the training data of size m be mostly linearly separable. In practice this holds even for not so large k, but as will be shown in section 6, such a k is not necessarily resulted in the best generalization even if m training data are mostly linearly separated. We can use a linear SVM to discriminate one class from others on features of eq.(12).

6 EXPERIMENTS

Face detection is a practically important, typical and difficult pattern recognition problem. This problem has been studied in a quite large number of references so far, and is appropriate to evaluate the classification power. In this section we empirically evaluate our method using eq.(12) as a feature extractor for the face discrimination problem. In the face discrimination problem, a dataset of faces and non-faces are cropped and resampled from original images, so that the problem is basically equivalent to the face detection.

The face detection has been intensively studied since the break-through of Viola-Jones (Viola & Jones 2004), and improved using such as SURF, AdaBoost, and Cascade (Li & Zhang 2013). Datasets of face detection has also been renewed, and it is difficult to compare the generalization power with previous studies.

We describe the experimental results on a subset of CMU+MIT dataset, which is out-of-date now, but the detector of Viola-Jones was tested on this dataset. Alvira and Rifkin (Alvira & Rifkin 2001) prepared a subset of CMU+MIT dataset for the purpose of classifier evaluation, and conducted experiments of face discrimination. We utilize their dataset for training and testing of our classifier.

In their data set each face or non-face image is cropped to a 19×19 window, and each pixel has 256 grayscale values. There are 2429 face and 4548 non-face cropped images for training, and 472 face and 23573 non-face cropped images for testing. This dataset was previously available on the CBCL webpage, but it is unavailable now.

For training the classifier we used 2429 face images, and equipped three non-face datasets; the first set consists of the first 2429 non-face images, the second set consists of 4215 non-face images, and the third set is constructed by adding the mirror images of the first set to the second set resulted in 6644 non-face images. For testing we used 470 face images out of 472 images (the first one and the last one is re-

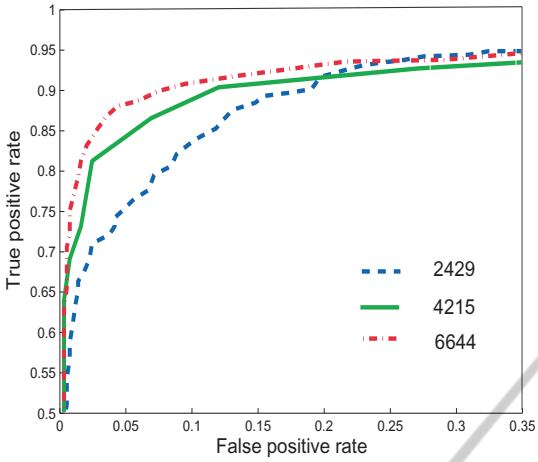

Figure 3: ROC curves for 8 training iteration of KRF corresponding to non-face datasets 2429, 4215, and 6644.

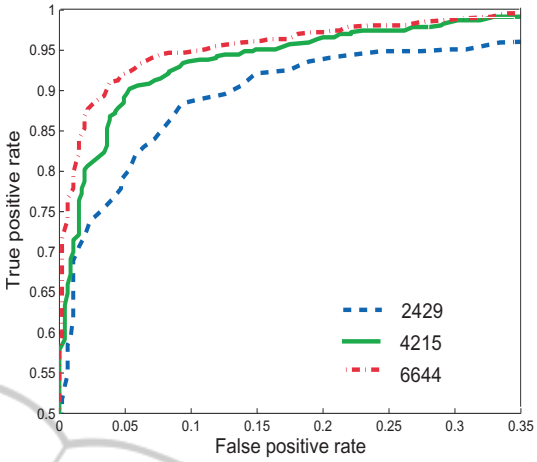

Figure 4: ROC curves for 10 training iteration of KRF corresponding to non-face datasets 2429, 4215, and 6644.

moved), and randomly chosen 470 non-face images from 23573 images.

Our approach is appearance-based, that is, the grayscale intensity of image pixels serve as the feature. Each image is transformed into the projective space in order to normalize with intensity ratio. Specifically, each pixel value is divided by the norm of the image, and multiplied by 19/2, mapping almost all pixel values into $[0, 1]$. For discrete states of KRF model, we discretize the pixel values in 9 values from 0 to 1 at intervals of 0.125.

We constructed KRF by 1200 autocorrelation kernels with the first 1200 training face data. The order of autocorrelation is set at $d = 4$. Then KRF was trained using 1600 training face data (400 data was added to the 1200 kernel data) until the absolute value of eq.(11) became less than 1/100 of the initial values.

We trained a linear SVMon the features of eq.(12) using 2429 face data and each of the above described three non-face dataset. LSVM (Mangasarian & Musicant 2001) is used for this purpose. The soft-margin parameter of LSVM is set to 0.3∼0.5 so as to discriminate almost all training data.

We show the ROC curves in Figure 3 for 8 iterations and Figure 4 for 10 iterations of KRF training. In order to investigate the effectiveness of the feature extractor with KRF (eq.(12) for 10 training iterations), we compare it with auto-correlation kernel+LSVM in Figure 5. In this figure ROC curves are shown using 1200 kernels for KRF, and 2429 face + 6644 non-face kernels for auto-correlation kernel+LSVM classifier.

From these results we see that the feature representation of KRF is essentially linearly separable. The recent tendency of face detection is based on the set of simple visual features called 'integral features'

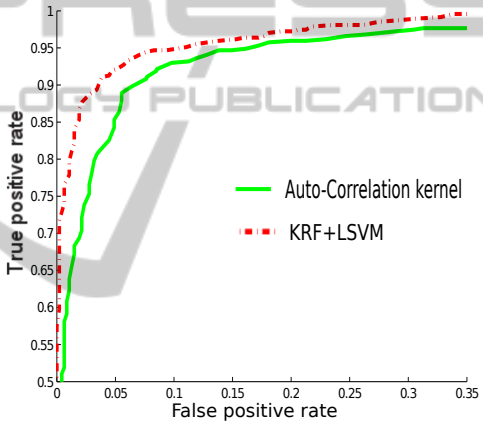

Figure 5: ROC curves for KRF+LSVM classifier with 1200 (face) kernels and auto-correlation kernel + LSVM classifier with 2429 face + 6644 non-face kernels.

proposed in (Viola & Jones 2004), choosing critical visual features with AdaBoost, and cascade allocation of discriminators. The cascade allocation aims at cut-down of detection time with discarding clearly non-face examples at the early stage of detection.

Unfortunately results with state-of-the-art methods are not available for comparison as they are based on a rich training data. However, as in (Li & Zhang 2013) CMU+MIT full test set is used for evaluation of a state-of-the-art methods, and ROC curves comparing with (Viola & Jones 2004) are presented, we pick up the coresponding values from the ROC curves to show comparison in Table 1 only for reference. From Table 1 we can see that our classifier is comparable with the state-of-the-art detectors specialized to the face detection.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Table 1: Recognition/detection rates(%) for false positives (10, 20%) on CMU+MIT test set.

Systems \ False positives	10 %	20%
KRF(10 iters)+LSVM on the subset of CMU+MIT test set	94.8	97.2
Viola & Jones 2004	92.1	93.2
SURF cascade, J.Li and Y. Zhang (picked up from ROC curves of (Li & Zhang 2013))	(94[1])	–

1 Corresponding point to 92% value of Viola & Jones

7 CONCLUSIONS

In this paper we proposed a new kernel machine called KRF+LSVM, and showed that its classification capability out performs SVM through the experments with empirical evaluation of face discrimination. We claimed that our feature extraction method can give essentially linearly separable expression. We explained it through a property of Fisher score, and the experimental results could support it. The chief advantage of KRF+SVM method lies in its simple structure similar to SVM, in that kernel features are constructed with Fisher score. We also proposed efficient computational framework of KRF and autocorrelation kernels. In what extent our model works well should be investigated further, and it is main interest of future research.

ACKNOWLEDGEMENTS

This work was supported in part by a Grant-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology, Japan (No. 24500165).

REFERENCES

Alvira M. and Rifkin R. 2001, 'An Empirical Comparison of SNoW and SVMs for Face Detection', in MIT CBCL Memos (1993 - 2004).

Bengio Y., Courville A. and Vincent P. 2012, 'Representation Learning: A Review and New Perspectives' , arXiv:1206.5538 [cs.LG], Cornell University Library [Oct 2012].

Erhan D., Bengio Y., Courville A., Manzagol P.A., Vincent P. and Bengio S., 2010, 'Why Does Unsupervised Pretraining Help Deep Learning?, Journal of Machine Learning Research 11, 625-660 .

Mangasarian O.L. and Musicant D.R. 2001, 'Lagrangian Support Vector Machines', Journal of Machine Learning Research 1, 161-177.

Horikawa Yo. 2004, 'Comparison of Support Vector Machines with Autocorrelation Kernels for Invariant Texture Classification', Proceedings of the 17th Omt. Conf. on Pat. recog. (ICPR'04), 4647-4651.

Jaakkola ,T and Haussler, D. 1998, 'Exploiting Generative Models in Discriminative Classifiers' In Advances in Neural Information Processing Systems 11, pp 487-493. MIT Press.

Lafferty J. Zhu X., and Liu Y. 2004,'Kernel conditional random fields: representation and clique selection', Proc. of the twenty-first int. conf. on Machine learning, Canada, Page: 64 .

Lafferty J., McCallum A. and Pereira F. 2004, 'Exponential Families for Conditional Random Fields', Conditional Random Fields: ACM Int. Conf. Proc. Series; Vol. 70, . 20th Conf.on Uncertainty in artificial intelligence, Banff, Canada, 2 - 9.

Li, J. , Zhang Y. 2013, 'Learning SURF Cascade for Fast and Accurate Object Detection', CVPR, 3468-3475.

Pham, M. T., Gao, Y. , Hoang,V. D. D., and T. J.Cham, T. J., 2010, 'Fast Polygonal Integration and Its Application in Extending Haar-like Features to Improve Object Detection', Proc. IEEE conf. on Comp. and Pat. Recog. (CVPR), San Francisco.

Roscher, R., 'Kernel Discriminative Random Fields for land cover classification', Pattern Recognition in Remote Sensing (PRRS), 2010 IAPR Workshop on Date of Conference, 22-22 Aug.

Viola P. A. and Jones M. J. 2004, 'Robust real-time face detection', IJCV, 57(2), 137-154.

APPENDIX

In computing the mean field of eq.(10) we need n repeated computation of autocorrelation kernel for $n-1$ dimensional variables each removing x_i for $i = 1,...,n$. We show a method that manage with one time computation of degree $d-1$ autocorrelation kernel of n dimensional variable.

For $d-1=1$, we can construct the first order symmetric polynomial of $n-1$ variables from that of n as

$$S_1(x_1,..x_{i-1},x_{i+1},...,x_n) = S_1 - x_i = S_1 - S_1^i$$

Similarly for $d-1=2$

$$S_2(x_1,..x_{i-1},x_{i+1},...,x_n) = S_2 - S_2^i - \hat{S}_1^{i-1}x_i$$

For $d-1=3$

$$S_3(x_1,..x_{i-1},x_{i+1},...,x_n) = S_3 - S_3^i - x_i\hat{S}_2^{i-1} - S_1^{i-1}x_i\hat{S}_1^i$$

In general

$$S_k(x_1,..x_{i-1},x_{i+1},...,x_n) = S_k - \sum_{h=1}^{k} \hat{S}_{k-h}^{i-1}S_h^i$$