1 Estimating Conditional Densities of Structured Outputs in RKHS

Yasemin Altun
Toyota Technological Institute at Chicago, Chicago IL 60637 USA
altun@tti-c.org

Alex Smola
Statistical Machine Learning Programme, NICTA, 0200 ACT, Australia
alex.smola@nicta.com.au

In this paper we study the problem of estimating conditional probability distributions for structured output prediction tasks in Reproducing Kernel Hilbert Spaces. More specifically, we prove decomposition results for undirected graphical models, give constructions for kernels, and show connections to Gaussian Process classification. Finally we present efficient means of solving the optimization problem and apply this to label sequence learning. Experiments on named entity recognition and pitch accent prediction tasks demonstrate the competitiveness of our approach.

1.1 Introduction

The benefits of a framework for designing flexible and powerful input representations for machine learning problems has been demonstrated by the success of kernel-based methods in binary and multiclass classification as well as regression. However, many real-world prediction problems also involve complex output spaces, with possible dependencies between multiple output variables. Markov-chain dependency structure is a prominent example of this kind and is ubiquous in natural language processing (e.g. part-of-speech tagging, shallow parsing), speech recognition (e. g. pitch accent prediction), information retrieval (e. g. named entity recognition) and computational biology (e. g. protein secondary structure prediction). More complicated dependency structures such as hierarchies and parse trees are also commonplace.

A well-known approach for solving these problems are Conditional Random Fields (CRFs), proposed by (Lafferty et al., 2001), an extension of logistic regression that takes dependencies between random variables in a graph (e. g. neighboring labels along a chain) into account. Related approaches include (Punyakanok and Roth, 2000; McCallum et al., 2000). More recently, other discriminative methods such as AdaBoost (Altun et al., 2002), perceptron learning (Collins, 2002), and Support Vector Machines (SVMs) (Altun et al., 2003; Taskar et al., 2003; Tsochantaridis et al., 2004) have been extended to learning the prediction of structured objects.

In this chapter, which is an extension of our work in (Altun et al., 2004b) and (Altun et al., 2004a) and is closely related to (Lafferty et al., 2004), we study the problem of estimating conditional probability distributions over structured outputs within a Reproducing Kernel Hilbert Space. Framing this as a special case of inverse problems, we show that maximizing entropy with respect to approximate moment matching constraints in Banach spaces leads to the maximum a posteriori estimation and exponential families. The space in which the moments are defined and approximated, specify a regularization of the conditional log-likelihood of the sample. When this space is ℓ_2, we have a Gaussian Process over the structured input-output space. Then, one can construct and learn in Reproducing Kernel Hilbert Spaces (RKHS), thereby overcome the limitations of (finite-dimensional) parametric statistical models and achieve the flexibility of implicit data representations.

Our framework preserves the main strength of CRFs, namely their rigorous probabilistic semantics, which is not the case for other discriminative methods such as max-margin approaches. There are two important advantages of a probabilistic model. First, it is very intuitive to incorporate prior knowledge within a probabilistic framework. Second, in addition to predicting the best labels, one can compute posterior label probabilities and thus derive confidence scores for predictions. This is a valuable property in particular for applications requiring a cascaded architecture of classifiers. Confidence scores can be propagated to subsequent processing stages or used to abstain on certain predictions. Another advantage over max-margin methods for structured output prediction is its consistency with infinite samples. Even though the performance of a learning method on small-sample problems do not necessarily coincide with its performance on infinite sample, asymptotic consistency analysis provides useful insights.

Performing density estimation via Gaussian Processes over structured input-output spaces faces serious tractability issues, since the space of parameters, although finite, is exponential in the size of the structured object. We prove decomposition results for Markov Random fields, which allow us to obtain an optimization problem scaling polynomially with the size of the structure. This leads to the derivation of an efficient estimation algorithm, for which we provide the details for label sequence learning. We report experimental results on pitch accent prediction and named-entity recognition.

1.2 Estimating Conditional Probability Distributions over Structured Outputs

1.2.1 General Setting

In this paper, we are interested in the prediction of structured variables. The goal is to learn a mapping $h : \mathcal{X} \rightarrow \mathcal{Y}$ from structured inputs to structured response variables, where the inputs and response variables form a dependency structure. There exists a cost function $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \Re$, where $\Delta(\mathbf{y}, \bar{\mathbf{y}})$ denotes the cost of predicting $\bar{\mathbf{y}}$ instead of $\mathbf{y}$. We restrict our attention to cost functions such that $\Delta(\mathbf{y}, \mathbf{y}) = 0$ and $\Delta(\mathbf{y}, \bar{\mathbf{y}}) > 0$ for $\mathbf{y} \neq \bar{\mathbf{y}}$. This function is generally the standard 0-1 classification error for multiclass classification problems. However, in the structured prediction problems, it can incorporate differences in the structure of $\mathbf{y}$ and $\bar{\mathbf{y}}$, such as Hamming loss of sequences or $1 - F_1$ score of parse trees.

Let us define $\mathcal{Z}$ as $\mathcal{X} \times \mathcal{Y}$. We assume that there is a fixed but unknown distribution P over $\mathcal{Z}$ according to which input/output pairs $(\mathbf{x}, \mathbf{y})$ are generated. If this distribution was known, the optimal predictor i. e. the Bayes decision rule, is given by

$$h_B(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{y} \in \mathcal{Y}} \sum_{\mathbf{y}' \neq \mathbf{y}} p(\mathbf{y}'|\mathbf{x}) \Delta(\mathbf{y}', \mathbf{y}). \tag{1.1}$$

In the special case of 0-1 loss, this is equivalent to $h_B(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{y}|\mathbf{x})$. Then, one can reduce the (structured) prediction problem to one of estimating the conditional distribution $p(\mathbf{y}|\mathbf{x})$ and performing the argmin operation as in (1.1).

1.2.2 Maximum Entropy with approximate matching

In supervised learning, we are given a sample S of ℓ input-output pairs $S = \{(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^\ell, \mathbf{y}^\ell)\}$. There exists a set of measurements relevant for the learning task, commonly referred as *moment* or *feature* functions, $\phi : \mathcal{Z} \rightarrow \mathcal{B}$, where $\mathcal{B}$ is a Banach space in the most general form of the problem. The goal is to find the conditional probability distribution p over $\mathcal{Y}|\mathcal{X}$ such that the expectation of the features with respect to $p(\mathbf{y}|\mathbf{x})$ for all $\mathbf{x} in S_{\mathbf{x}}$ $(E_{\mathbf{y} \sim p}[\phi(\mathbf{x}, \mathbf{y})|\mathbf{x}])$ matches their empirical values $(\tilde{\phi})$, $E_{\mathbf{y} \sim p}[\phi(\mathbf{x}, \mathbf{y})|\mathbf{x}] = \tilde{\phi}$, Here $S_{\mathbf{x}}$ denotes the set of ℓ inputs in S, $\{\mathbf{x}^i\}$ for $i = 1, \ldots, \ell$. The empirical values of the features are generally derived from the sample by

$$\tilde{\phi} = \frac{1}{\ell} \sum_{i=1}^{\ell} \phi(\mathbf{x}^i, \mathbf{y}^i). \tag{1.2}$$

This estimation task is an instance of a more general problem, namely the *inverse problem*, where the goal is to find x satisfying $Ax = b$. Inverse problems are known to be ill-formed and are stabilized by imposing a regularity or smoothness measure, such as the entropy of the distribution (Ruderman and Bialek, 1994). Then, the estimation problem can be formulated as finding the maximum entropy distribution

p (equivalently minimizing the relative entropy of p with respect to a constant distribution) such that it satisfies the moment matching constraints,

$$\min_p \; KL(p\|q) \text{ subject to } E_{\mathbf{y}\sim p}[\phi(\mathbf{x},\mathbf{y})|\mathbf{x}] = \tilde{\phi}.$$

In general, it is very difficult to satisfy the moment matching constraints exactly, especially when the number of features can be very large (and possibly infinite). For example, the solution may lie on the boundary of a feasibility region. Moreover, this can lead to severe overfitting of the data. In order to overcome this obstacle, the constraints can be relaxed to approximate matches such that the difference between the expected and the empirical values are small with respect to the norm of the space $\mathcal{B}$, denoted by $\|.\|_{\mathcal{B}}$. An adaptation of Theorem 8 of (Altun and Smola, 2006) gives the solution of this estimation problem and establishes the duality connection between the KL divergence minimization with approximate moment matching constraints and the maximum *a posteriori* (MAP) estimation.

Theorem 1 (Approximate KL Minimization) *Let p, q be conditional probability distributions over $\mathcal{Y}|\mathcal{X}$ and S be a sample of size ℓ. Moreover, $\phi : \mathcal{Z} \to \mathcal{B}$ is a mapping from $\mathcal{Z}$ to a Banach space $\mathcal{B}$, $\tilde{\phi} \in \mathcal{B}$ and $\mathcal{B}^*$ is the dual space of $\mathcal{B}$. Then, for any $\epsilon \geq 0$ the problem*

$$\min_p \; KL(p\|q) \textit{ subject to } \left\| E_{\mathbf{y}\sim p}[\phi(\mathbf{x},\mathbf{y})|\mathbf{x}] - \tilde{\phi} \right\|_{\mathcal{B}} \leq \epsilon$$

has a solution of the form

$$p_{\mathbf{w}}(\mathbf{y}|\mathbf{x}) = q(\mathbf{y}|\mathbf{x}) \exp\left(\langle \phi(\mathbf{x},\mathbf{y}), \mathbf{w}\rangle - g(\mathbf{w};\mathbf{x})\right) \tag{1.3}$$

where $\mathbf{w} \in \mathcal{B}^*$ and g is the log*-partition function guaranteeing p to be a probability distribution. Moreover, the optimal value of $\mathbf{w}$ is found as the solution of*

$$\min_{\mathbf{w}} \; -\left\langle \tilde{\phi}, \mathbf{w} \right\rangle + \frac{1}{\ell} \sum_{\mathbf{x}\in S_{\mathbf{x}}} g(\mathbf{w};\mathbf{x}) + \epsilon \|\mathbf{w}\|_{\mathcal{B}^*}. \tag{1.4}$$

Equivalently, for every feasible ϵ, there exists a $\Lambda \geq 0$ such that the minimum of $-\left\langle \tilde{\phi}, \mathbf{w} \right\rangle + \frac{1}{\ell}\sum_{\mathbf{x}\in S_{\mathbf{x}}} g(\mathbf{w};\mathbf{x}) + \frac{\Lambda}{2}\|\mathbf{w}\|^2_{\mathcal{B}^*}$ minimizes (1.4).

Note that the well-known connection between conditional maximum entropy optimization (MaxEnt) with exact moment matching constraints and conditional maximum likelihood estimation is a special case of Theorem 1 with $\epsilon = 0$. Thus, relaxing the constraints corresponds to a regularization in the dual problem scaled by the relaxation parameter. Since the dual (1.4) is an unconstrained optimization problem (over a possibly finite domain), it is common to solve the density estimation problem by performing the optimization in the dual space.

1.2.3 Exponential Families

As we have seen in (1.3), exponential families arise from the optimization of KL divergence with respect to (approximate) moment matching constraints. Here $\mathbf{w}$ is called the canonical or natural parameter, $\phi(\mathbf{x}, \mathbf{y})$ is the corresponding vector of sufficient statistics and $g(\mathbf{w}; \mathbf{x})$ is the log-partition function or moment generating function.

$$g(\mathbf{w}; \mathbf{x}) := \log \sum_{\mathbf{y} \in \mathcal{Y}} \exp(\langle \phi(\mathbf{x}, \mathbf{y}), \mathbf{w} \rangle) q(\mathbf{y}|\mathbf{x}) \,. \tag{1.5}$$

The log-partition function plays an important role in estimating probability distributions. In particular, it can be used to compute the moments of the distribution, see e.g. Lauritzen (1996):

Proposition 2 $g(\mathbf{w}; \mathbf{x})$ *is a convex* C^∞ *function. Moreover, the derivatives of* g *generate the corresponding moments of* ϕ

$$\partial_{\mathbf{w}} g(\mathbf{w}; \mathbf{x}) = \mathbf{E}_{\mathbf{y}|\mathbf{x} \sim p_{\mathbf{w}}}[\phi(\mathbf{x}, \mathbf{y})] \qquad \textit{Mean} \tag{1.6a}$$

$$\partial^2_{\mathbf{w}} g(\mathbf{w}; \mathbf{x}) = \mathrm{Cov}_{\mathbf{y}|\mathbf{x} \sim p_{\mathbf{w}}}[\phi(\mathbf{x}, \mathbf{y})] \qquad \textit{Covariance}\,. \tag{1.6b}$$

It is important to use exponential families with rich sufficient statistics. One can show that if $\phi(\mathbf{z})$ for $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \mathcal{Z}$ is powerful enough, exponential families become universal density estimators, in the sense that the set of all functions of the form $\langle \mathbf{w}, \phi(.) \rangle$ is dense in the set of all bounded continous functions defined on $C^0(\mathcal{Z})$. This is advantageous, as it can open the domain of nonparametric estimation to an area of statistics which so-far was restricted to parametric distributions. In Proposition 3, we show that exponential families can in fact be dense over $C^0(\mathcal{Z})$, if they are defined with respect to a universal kernel. More precisely, we restrict the Banach space $\mathcal{B}$ be a Reproducing Kernel Hilbert Space (RKHS) with the kernel function k and define a linear discriminant function $F : \mathcal{Z} \to \Re$ as

$$F(.) = \langle \phi(.), \mathbf{w} \rangle \,, \tag{1.7}$$

for $F \in \mathcal{H}$ such that

$$F(\mathbf{z}) = \langle F, k(\mathbf{z}, \cdot) \rangle_{\mathcal{H}} \tag{1.8}$$

where ϕ is the feature map induced by k. We can now represent p as a function of $\mathbf{w}$ and F interchangeably.

Proposition 3 (Dense Densities) *Let $\mathcal{Z}$ be a measurable set with respect to the Lebesgue measure. Moreover, let $\mathcal{H}$ be universal and $\mathcal{P}$ be the space of $C^0(\mathcal{Z})$ densities with $\|p\|_\infty < \infty$. Then the class of exponential family densities*

$$\mathcal{P}_{\mathcal{H}} := \{p_F | p_F(z) : F \exp\left(F(z) - g(F)\right) \text{ and } F \in \mathcal{H}\} \quad (1.9)$$
$$\text{where } g(F) := \log \int_{\mathcal{Z}} \exp\left(F(z)\right) dz$$

is dense in $\mathcal{P}$. Moreover, for any $p \in \mathcal{P}$ with $\|\log p\|_\infty \leq C$ and $\epsilon < 1$ we have

$$\|\log p - F\|_\infty \leq \epsilon \text{ implies } D(p\|p_F) \leq 2\epsilon \text{ and } \|p - p_F\|_\infty \leq 4\epsilon e^C \quad (1.10)$$

Proof We prove the second part first: Let $\epsilon > 0$. If $\|\log p\|_\infty \leq C$, we can find some $F \in \mathcal{H}$ such that $\|F - \log p\|_\infty \leq \epsilon$. By the definition of $g(F)$ it follows that

$$|g(F)| = \left|\log \sum_z \exp(F(z))dz\right| = \left|\log p(z) \int_z \exp(F(z) - \log p(z))dz\right| \leq \epsilon. \quad (1.11)$$

Since $\log p_F = F - g(F)$ have $\|\log p - \log p_F\|_\infty \leq 2\epsilon$. This immediately shows the first part of (1.10). For the second part, all we need to do is exponentiate the bound and use the fact that $\log p$ is bounded by C.

To see the general result, pick some $\delta < \epsilon/Z$ with $\epsilon < 1$, where Z is the measure of $\mathcal{Z}$. Moreover, let $p_\delta(z) := \max(p(z), \delta)$. Finally, pick F such that $\|F - \log p_\delta\| \leq \epsilon$. By the same reasoning as in (1.11) it follows that $|g(F)| \leq 2\epsilon$. Hence $\|p - p_F\| \leq 4\epsilon \|p\|_\infty$. Since ϵ was arbitrary, this proves the claim. ■

Note that a similar result can be obtained whenever $\mathcal{Z}$ is endowed with a measure ν, if we state our results for densities whose Radon-Nikodym derivatives with respect to ν satisfies the properties above. Moreover, similar results apply for the conditional densities, as long as $\log p(\mathbf{y}|\mathbf{x})$ is a well behaved function of $\mathbf{x}$ and the kernel k is universal kernel in $\mathbf{y}$ for every $\mathbf{x}$.

Many RKHS $\mathcal{H}$ are dense in $C^0(\mathcal{X})$. See Steinwart (2001) for examples and a proof. This shows that choosing a density from a suitable exponential family is not restrictive in terms of approximation quality.

1.2.4 Objective Function and Relationship to Gaussian Processes

Theorem 1 establishes that, when $\mathcal{B}$ is an RKHS, our density estimation problem is given by

$$\min_{\mathbf{w}} - \left\langle \tilde{\phi}, \mathbf{w} \right\rangle + \frac{1}{\ell} \sum_{i=1}^{\ell} g(\mathbf{w}; \mathbf{x}^i) + \frac{\Lambda}{2} \|\mathbf{w}\|_2^2 .$$

Thus, we perform a regularized maximum likelihood estimate, where a normal distribution is assumed as a prior on $\mathbf{w}$. It follows from (Williams, 1999) that a normal prior on $\mathbf{w}$ corresponds to a Gaussian Process on the collection of random variables $F(\mathbf{z})$ with zero mean and a covariance function k, where k is the kernel associated with RKHS $\mathcal{H}$ and F is defines as in (1.7). Using (1.2), (1.5), (1.7) and (1.8), we can rewrite the optimization as

$$F^* = \operatorname*{argmin}_{F \in \mathcal{H}} \frac{1}{\ell} \sum_{i=1}^{\ell} \left(-F(\mathbf{x}^i, \mathbf{y}^i) + \log \sum_{\mathbf{y} \in \mathcal{Y}} \exp(F(\mathbf{x}^i, \mathbf{y})) \right) + \frac{\Lambda}{2} \|F\|_{\mathcal{H}}^2 . \qquad (1.12)$$

Equivalently, F^* maximizes

$$p(F|S) \propto p(F) \prod_i p(\mathbf{y}^i | \mathbf{x}^i; F) \qquad (1.13)$$

where $p(F) \propto \exp(\frac{\Lambda}{2} \|F\|_{\mathcal{H}}^2)$. In Gaussian Process classification, (1.13) is approximated by a Gaussian and F^* is the mode of the Gaussian approximating $p(F|S)$ by a second order Taylor expansion of the log-posterior via Laplace approximation. Given F^*, one can approximate the curvature $\mathbf{C}$ at the mode and use this normal distribution $p(F|S) \approx \mathcal{N}(F^*, \mathbf{C})$ to obtain the predictive distribution of a new input $\mathbf{x}$ via

$$p(\mathbf{y}|S, \mathbf{x}) = \int p(\mathbf{y}|\mathbf{x}, F) p(F|S) dF.$$

Unfortunately, the Bayesian approach faces tractability problems with even moderate datasets during training. Exact inference for structured output prediction is intractable (Qi et al., 2005) and even the approximate inference is very expensive. Also motivated by the maximum entropy principle, we use the MAP estimate F^* to obtain the predictive distribution $p(\mathbf{y}|S, \mathbf{x}) \approx p(\mathbf{y}|F^*(\mathbf{x}, .))$.

1.2.5 Subspace Represention

The Representer Theorem (Kimeldorf and Wahba, 1970) guarantees that F^*, the optimizer of (1.12) is of the form

$$F^*(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\ell} \sum_{\bar{\mathbf{y}} \in \mathcal{Y}} \alpha_{(\mathbf{x}^i, \bar{\mathbf{y}})} k((\mathbf{x}, \mathbf{y}), (\mathbf{x}^i, \bar{\mathbf{y}})), \qquad (1.14)$$

with suitably chosen coefficients α. Note that there is one $\alpha_{(i, \bar{\mathbf{y}})}$ coefficient for every training example $\mathbf{x}^i$ and its possible labeling $\bar{\mathbf{y}}$, due to the fact that F is defined over $\mathcal{X} \times \mathcal{Y}$ and the log-partition function sums over all $\bar{\mathbf{y}} \in \mathcal{Y}$ for each $\mathbf{x}^i \in S$. The feature map ϕ over $\mathcal{Z}$ is induced by the kernel function k via

$$k((\mathbf{x}, \mathbf{y}), (\bar{\mathbf{x}}, \bar{\mathbf{y}})) = \langle \phi(\mathbf{x}, \mathbf{y}), \phi(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \rangle .$$

Thus, in complete analogy to Gaussian Process classification and other kernel methods, we can perform the estimation and prediction by computing the inner products between sufficient statistics without the need for evaluating $\phi(\mathbf{x}, \mathbf{y})$ explicitly.

Although the application of the Representer Theorem reduces the optimization problem from an infinite dimensional space ($\mathcal{H}$) to a finite dimensional space scaling with the size of the sample and the size of $\mathcal{Y}$ ($\Re^{\ell|\mathcal{Y}|}$), this formulation suffers from scalability issues. This is because in the structured output prediction problems, $|\mathcal{Y}|$ is very large, scaling exponentially in the size of the structure. However, if there is a decomposition of the kernel into substructures, then the dimensionality can be reduced further. We now show that such a decomposition exists when $(\mathbf{x}, \mathbf{y})$ represents a Markov Random Field (MRF).

Let G be an undirected graph, $\mathcal{C}$ be the set of maximal cliques of G, $\mathbf{z}$ be a configuration of G and z_c be the restriction of $\mathbf{z}$ on the clique $c \in \mathcal{C}$. The well-known theorem of Hammersley and Clifford (1971), Theorem **??**, states that the density over $\mathcal{Z}$ decomposes into *potential functions* defined on the maximal cliques of G. Using this theore, it is easy to show that the Gibbs form translates into a simple decomposition of the kernel functions on Markov Random Fields.

Lemma 4 (Decomposition of kernels on MRFs) *For positive probability density functions over a MRF Z on G, the kernel $k(\mathbf{z}, \bar{\mathbf{z}}) = \langle \phi(\mathbf{z}), \phi(\bar{\mathbf{z}}) \rangle$ satisfies*

$$k(\mathbf{z}, \bar{\mathbf{z}}) = \sum_{c \in \mathcal{C}} k(z_c, \bar{z}_c). \tag{1.15}$$

Proof We simply need to show that the sufficient statistics $\phi(z)$ satisfy a decomposition over the cliques. From Theorem 1 and Theorem **??**, we know that $F(\mathbf{z}) = \langle \phi(\mathbf{z}), \mathbf{w} \rangle = \sum_{c \in \mathcal{C}} \psi_c(z_c; \mathbf{w})$ for all $\mathbf{z} \in \mathcal{Z}$ and any $\mathbf{w}$. Then, we can pick an orthonormal basis of $\mathbf{w}$, say e_i and rewrite

$$\langle \phi(\mathbf{z}), e_i \rangle = \sum_{c \in \mathcal{C}} \eta_c^i(z_c)$$

for some scalar functions $\eta_c^i(z_c)$. The key point is that η_c^i depends on $\mathbf{z}$ only via its restriction on z_c. Setting $\phi_c(z_c) := (\eta_c^1(z_c), \eta_c^2(z_c), \ldots)$ allows us to compute

$$\langle \phi(\mathbf{z}), \mathbf{w} \rangle = \left\langle \phi(\mathbf{z}), \sum_i e_i \mathbf{w}_i \right\rangle = \sum_i \mathbf{w}_i \langle \phi(\mathbf{z}), e_i \rangle = \sum_{c \in \mathcal{C}} \sum_i \mathbf{w}_i \eta_c^i(z_c).$$

Rearranging terms shows that ϕ decomposes into $\phi(\mathbf{z}) = (\phi_{c_1}(z_{c_1}), \ldots, \phi_{c_n}(z_{c_n}))$. Setting

$$k(z_c, \bar{z}_c) = \langle \phi_c(z_c), \phi_c(\bar{z}_c) \rangle \tag{1.16}$$

satisfies the claim [1]. ■

Many applications involve a family of graph structures such that some subsets of cliques in these graphs share the same potential function. For instance, in

[1] Note that similar decomposition results can be achieved by selecting other basis functions. This might lead to interaction across different cliques. However, one can reduce such formulations to the decomposition presented in this lemma via rotating the basis functions.

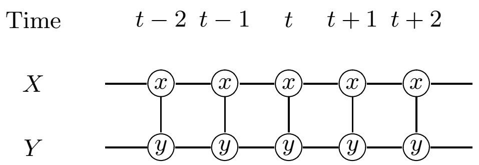

Figure 1.1 A time invariant Markov chain with three types of cliques.

the case of Markov chains that are time invariant, there exists three types of cliques, $x-x$, $x-y$ and $y-y$. Figure 1.2.5 shows these types of cliques with red,blue and green respectively, where same colored cliques share the same potential function. For such graphs, the decomposition in Lemma 4 corresponds to $k(\mathbf{z}, \bar{\mathbf{z}}) = \sum_{c \in \mathcal{C}_\mathbf{z}} \sum_{\bar{c} \in \mathcal{C}_{\bar{\mathbf{z}}}} k(z_c, \bar{z}_{\bar{c}})$, where $\mathcal{C}_\mathbf{z}$ denotes the set of maximal cliques in $\mathbf{z}$ and $k(z_c, \bar{z}_{\bar{c}}) = 0$ if c and $\bar{c}$ are of different types. In order to simplify our future notation, we define $\mathcal{W}$ as the set of all possible clique configurations for all cliques defined over the set of Markov fields and $\theta^\mathbf{z}_\omega$ as the number of times the clique configuration $\omega \in \mathcal{W}$ occurs in $\mathbf{z}$. Then,

$$k(\mathbf{z}, \bar{\mathbf{z}}) = \sum_{\omega, \bar{\omega} \in \mathcal{W}} \theta^\mathbf{z}_\omega \theta^{\bar{\mathbf{z}}}_{\bar{\omega}} k(\omega, \bar{\omega}) \tag{1.17}$$

We can now use this result to decompose the linear discriminant F given by (1.8) into a linear discriminant function $f : \mathcal{W} \to \Re$ over cliques by

$$F_f(\mathbf{z}) = \sum_{\omega \in \mathcal{W}} \theta^\mathbf{z}_\omega f(\omega), \tag{1.18}$$

$$f(\omega) = \langle f, k(\omega, .) \rangle_\mathcal{H}. \tag{1.19}$$

This gives us a new optimization problem with a subspace representation

$$f^* = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{\ell} \sum_{i=1}^{\ell} \left(-F_f(\mathbf{x}^i, \mathbf{y}^i) + \log \sum_{\mathbf{y} \in \mathcal{Y}} \exp(F_f(\mathbf{x}^i, \mathbf{y})) \right) + \frac{\Lambda}{2} \|f\|^2_\mathcal{H}, \tag{1.20}$$

$$f^*(\omega) = \sum_{\bar{\omega} \in \mathcal{W}(S_\mathbf{x})} \gamma_{\bar{\omega}} k(\omega, \bar{\omega}). \tag{1.21}$$

where $\mathcal{W}(S) \subseteq \mathcal{W}$ denotes the set of clique assignments with non-zero counts for any $(\mathbf{x}, \mathbf{y})$ for all $\mathbf{x} \in S_\mathbf{x}$ and all $\mathbf{y} \in \mathcal{Y}$. The subspace representation (1.21) holds immediately through a simple variant of the Representer Theorem Lafferty et al. (2004); Altun et al. (2006) for any *local* loss function, where locality is defined as follows:

Definition 5 (Local loss function) *A loss $\mathcal{L}$ is local if $\mathcal{L}(\mathbf{x}, \mathbf{y}, f)$ is determined by the value of f on the set $\mathcal{W}(\{\mathbf{x}\})$, i.e., for $f, g : \mathcal{W} \to \Re$ we have that if $f(p) = g(p)$ for all $p \in \mathcal{W}(\{\mathbf{x}\})$ then $\mathcal{L}(\mathbf{x}, \mathbf{y}, f) = \mathcal{L}(\mathbf{x}, \mathbf{y}, g)$.*

Note that the log-loss in (1.20) is local due to the decomposition of kernels on MRFs. This reduces the number of parameters from exponential in the size of the structure (number of vertices for MRF) to linear in the number of cliques and exponential in the number of vertices in the maximal cliques, which is generally much smaller than the total number of vertices. In this paper, we restrict our focus to such MRFs. For instance, in the Markov chain example, let the length of the chain be m and the number of possible label assignment for each vertex be n. Then, the number of parameters is reduced from n^m to $mn+n^2$. Note that since our goal is to estimate $p(\mathbf{y}|\mathbf{x})$, which is independent of $x-x$ cliques, there are no parameters for such cliques. In general, the set of cliques for which the restriction of $(\mathbf{x},\mathbf{y})$ are solely contained in $\mathbf{x}$ are irrelevant.

1.3 A Sparse Greedy Optimization

Using the representation of f^* over the subspaces of $\mathcal{Z}$, let us re-state our optimization in terms of the subspace parameters γ. Let $\mathbf{K}$ be the matrix of kernel values of clique assignments $k(\omega,\bar{\omega})$ for all $\omega,\bar{\omega}\in\mathcal{W}(S_{\mathbf{x}})$ and $\theta^{\mathbf{z}}$ be the vector of clique counts $\theta^{\mathbf{z}}_{\omega}$ in $\mathbf{z}$ for all $\omega\in\mathcal{W}(S_{\mathbf{x}})$. Then, the linear discriminant F can be written as

$$F_{\gamma}(\mathbf{x},\mathbf{y}) = \gamma^T\mathbf{K}\theta^{(\mathbf{x},\mathbf{y})}$$

and the optimization problem is given by minimizing $R(\gamma;S)$ with respect to γ where

$$R(\gamma;S) = \frac{1}{\ell}\sum_{i=1}^{\ell}\left(-\gamma^T\mathbf{K}\theta^{\mathbf{x}^i,\mathbf{y}^i} + \log\sum_{\mathbf{y}\in\mathcal{Y}}\exp(\gamma^T\mathbf{K}\theta^{\mathbf{x}^i,\mathbf{y}})\right) + \frac{\Lambda}{2}\gamma^T\mathbf{K}\gamma. \tag{1.22}$$

We now have a polynomial size convex optimization problem, whose Jacobian and Hessian is

$$\partial_{\gamma}R = \frac{1}{\ell}\sum_{i=1}^{\ell}\left(-\mathbf{K}\theta^{(\mathbf{x}^i,\mathbf{y}^i)} + \mathbf{K}\mathbf{E}_{\mathbf{y}\sim p_{\gamma}}\left[\theta^{(\mathbf{x}^i,\mathbf{y})}|\right]\right) + \Lambda\mathbf{K}\gamma \tag{1.23a}$$

$$\partial^2_{\gamma}R = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathrm{Cov}_{\mathbf{y}\sim p_{\gamma}}[\mathbf{K}\theta^{(\mathbf{x}^i,\mathbf{y})}] + \Lambda\mathbf{K}. \tag{1.23b}$$

Note that each term of the Hessian is computed by

$$\begin{aligned}\partial^2_{\gamma_{\omega},\gamma_{\bar{\omega}}}R =&\Lambda k(\omega,\bar{\omega})\\ &+1/\ell\sum_{i=1}^{\ell}\sum_{\omega',\omega''}k(\omega,\omega')k(\bar{\omega},\omega'')\left(\mathbf{E}_{\mathbf{y}}\left[\theta^{(\mathbf{x}^i,\mathbf{y})}_{\omega'}\theta^{(\mathbf{x}^i,\mathbf{y})}_{\omega''}\right] - \mathbf{E}_{\mathbf{y}}\left[\theta^{(\mathbf{x}^i,\mathbf{y})}_{\omega'}\right]\mathbf{E}_{\mathbf{y}}\left[\theta^{(\mathbf{x}^i,\mathbf{y})}_{\omega''}\right]\right).\end{aligned}$$

which involves correlation between configuration assignments of different cliques ($\mathbf{E}_{\mathbf{y}}\left[\theta_{\omega'}^{(\mathbf{x}^i,\mathbf{y})}\theta_{\omega''}^{(\mathbf{x}^i,\mathbf{y})}\right]$). Such correlations can be prohibitively expensive. For instance, in the Markov chain example, the complexity of the computation of the Hessian scales quadratically with the length of the chain. For this reason, we perform quasi-Newton optimization which simply requires 1^{st} order derivatives. The expectations can be computed using a dynamic programming algorithm in polynomial time with the size of the structure. For example, the expectations in Markov chains can be computed using the Forward-Backward algorithm whose complexity scales linearly in the length of the chain.

While optimization over the complete γ space is attractive for small data sets, the computation or the storage of $\mathbf{K}$ poses a serious problem when the data set is large. Also, classification of a new observation involves evaluating the kernel function at all the cliques in $\mathcal{W}(S_{\mathbf{x}})$, which may be more than acceptable for many applications. Hence, as in the case of standard Gaussian Process Classification, one may have to find a method for sparse solutions in terms of the γ parameters to speed up the training and prediction stages. We perform a sparse greedy subspace approximation algorithm along the lines of the method presented by Zhang (2003). In order to motivate this algorithm, we present the following lower bound on convex functions which is simply a tangent of the convex function.

Lemma 6 (Lower Bound on Convex Functions) *Let $C : \Theta \to \Re$ be a convex function on a vector space and $\theta_0 \in \Theta$. We denote by $g \in \partial_\theta C(\theta_0)$ a vector in the subdifferential of C at θ_0. Then*

$$\min_{\theta\in\Theta} C(\theta) + \|\theta\|^2 \geq C(\theta_0) + \|\theta_0\|^2 - \|\frac{g}{2} + \theta_0\|^2. \tag{1.24}$$

Proof Since C is convex, it follows that for any subdifferential $g \in \partial_\theta C(\theta_0)$ we have $C(\theta) \geq C(\theta_0) + g^\top \delta\theta$. Consequently,

$$\min_{\theta\in\Theta} C(\theta) + \|\theta\|^2 \geq \min_{\delta\theta\in\Theta} C(\theta_0) + g^\top\delta\theta + \|\theta_0 + \delta\theta\|^2. \tag{1.25}$$

The minimum is obtained for $\delta\theta = -(\frac{g}{2} + \theta_0)$, which proves the claim. ■

This bound provides a valuable selection and stopping criterion for the inclusion of subspaces during the greedy optimization process. Note in particular that $g + 2\theta_0$ is the gradient of the optimization problem in Eq. (1.24), hence we obtain a lower bound on the objective function in terms of the L_2 norm of the gradient. This means that optimization over a subspace spanned by a parameter is only useful if the gradient in the corresponding direction is large enough.

Let $\hat{\gamma}$ denote the sparse linear combination of basis vectors in $\mathcal{W}(S_{\mathbf{x}})$ and $\hat{\mathbf{K}}$ denote the matrix of the kernel function evaluated at basis vectors in $\hat{\gamma}$ and all $\mathcal{W}(S_{\mathbf{x}})$. The Sparse Greedy Subspace Approximation (SGSA) algorithm starts with an empty matrix $\hat{\mathbf{K}}$. At each iteration, it selects a training instance $\mathbf{x}^i$ and computes the gradients of the parameters associated with clique configurations $\omega \in \mathcal{W}(\{\mathbf{x}^i\})$ to select d coordinates with the largest absolute value of the gradient vector of

R over this subspace [2]. We denote those coordinates by $\mathbf{v}$. Then, R (which is defined via $\hat{\mathbf{K}}$ rather than $\mathbf{K}$ now) is optimized with respect to $\gamma_{\mathbf{v}}$ using a Quasi-Newton method. Finally, $\hat{\gamma}$ is augmented with the selected subspaces $\hat{\gamma} = [\hat{\gamma}', \gamma'_{\mathbf{v}}]'$ and $\hat{\mathbf{K}}$ is augmented with the columns associated with the selected subspaces, $\mathbf{K}e_\omega$ for each selected $\omega \in \mathbf{v}$. This process is repeated until the gradients vanish (i.e. they are smaller than a threshold value η) or some sparseness level is achieved (i. e. a maximum number p of coordinates are selected). Notice that the bottleneck of this method is the computation of the expectations of the clique assignments, $\mathbf{E}_{\mathbf{y}}\left[\theta_\omega^{(\mathbf{x},\mathbf{y})}\right]$. Therefore, once the expectations are computed, it is more efficient to include multiple coordinates rather than a single coordinate. This number, denoted by d, is a parameter of the algorithm.

We consider two alternatives for choosing the training sequence at each iteration. One method is to choose the input $\mathbf{x}$ whose set of cliques $\mathcal{W}(\{\mathbf{x}\})$ has the highest magnitude gradients. Another option is simply to select a random input from the sample.

Algorithm 1.1 Sparse Greedy Subspace Approximation (SGSA) algorithm.

```
Require: Training data $(\mathbf{x}^i, \mathbf{y}^i)_{i=1:\ell}$; Maximum number of coordinates to be selected, $p$; Number of coordinates to be selected at each iteration, $d$; Threshold value for gradients, $\eta$.
$\hat{\mathbf{K}} \leftarrow [], \hat{\gamma} \leftarrow []$
repeat
    Pick $i$:
    (1) Pick $i$ where $\omega = \arg\max_{\bar{\omega} \in \mathcal{W}(S_{\mathbf{x}})} |\partial_{\gamma_{\bar{\omega}}} R|$ and $\omega \in \mathcal{W}(\{\mathbf{x}^i\})$, or
    (2) Pick $i \in \{1, \ldots, \ell\}$ randomly
    $\mathbf{v} \leftarrow \text{argmax(d)}_{\omega \in \mathcal{W}(\{\mathbf{x}^i\})} |\partial_{\gamma_\omega} R|$ via (1.23a)
    Optimize $R$ wrt $\gamma_{\mathbf{v}}$ via dynamic programming.
    Augment $\hat{\gamma} = [\hat{\gamma}; \gamma_{\mathbf{v}}]$.
    Augment $\hat{\mathbf{K}} \leftarrow [\hat{\mathbf{K}}; \mathbf{K}e_\omega]$ for all $\omega \in \mathbf{v}$
until $\partial_\gamma R < \eta$ or $p$ coordinates selected.
```

When the input pattern is selected randomly, SGSA becomes an instance of coordinate descent or *Gauss-Seidel* method. This method is guaranteed to converge to the unique minimum asymptotically irrespective of coordinate selection sequence and the initial vector, if the optimization function is convex (Murty, 1998), which is the case for R. If we consider the convex sets of $\mathcal{W}(S_{\mathbf{x}})$, Theorem II.1 of (Zhang, 2003) shows that this algorithm has an $O(1/k)$ convergence rate where k denotes the number of iterations.

[2] In Algorithm 1.1, $\text{argmax(d)}_x\, f(x)$ selects the d number of x that maximize $f(x)$.

Theorem 7 (Zhang (2003)) *Let M_γ be an upper bound on $R''(\gamma)$. Then, after k iterations of the algorithm we have*

$$R(\hat{\gamma}^k; S) - R(\gamma^*; S) \leq 2M_\gamma/(k+2)$$

where γ^* is the true minimizer of $R(\gamma; S)$ and $\hat{\gamma}^k$ is the estimate at the k iteration.

Note in the above analysis, it is assumed that $\gamma_\omega \geq 0, \forall\omega$. This is obviously a special case of the SGSA algorithm. However, introducing the negative of all features functions enable us to generalize the non-negativity constraint and therefore apply Theorem 7.

One can establish better convergence rates if the best training sequence selection criteria (Line 4) is not prohibitively expensive. In this case, SGSA becomes an approximation of *Gauss-Southwell* method, which has been show to have linear converge rate of the form

$$R(\hat{\gamma}^{k+1}; S) - R(\gamma^*; S) \leq \left(1 - \frac{1}{\eta}\right)^k (R(\hat{\gamma}^k; S) - R(\gamma^*; S))$$

where $1 < \eta < \infty$ (Ratsch et al., 2002). Here, η depends polynomially on $|\mathcal{W}(S_{\mathbf{x}})|$. It also has dependency on M_γ. In practice, we observed that the random selection yields faster (approximate) convergence in terms of computational time. Therefore, we report experiments with this selection criteria.

1.4 Experiments: Sequence Labeling

We proceed to experiments on a specific structured prediction task, namely sequence labeling, and apply our method to two problems, pitch accent prediction and named entity recognition. We consider the chain model in (1.2.5), whose clique-structure comprises of (y_t, y_{t+1}) and (x_t, y_t) for all positions t in the sequence. Let Σ be the set of labels for each vertex, $n = |\Sigma|$ and δ be the Kronecker delta where $\delta(a, b)$ is 1 if $a = b$, and 0 otherwise. The features corresponding to the label-input cliques $\phi(x_t, y_t)$ is given by the concatenation of vectors $(\delta(y_t, \sigma)\phi(x_t))$ for all $\sigma \in \Sigma$

$$\phi(x_t, y_t) = (\delta(y_t, \sigma_1)\phi(x_t)^T, \ldots, \delta(y_t, \sigma_n)\phi(x_t)^T)^T. \tag{1.26}$$

This corresponds to the standard multiclass classification representation where the weight vector is given by the concatenation of the weight vector of each class $\mathbf{w} = (\mathbf{w}'_1, \ldots, \mathbf{w}'_n)'$. (1.26) is concatenated with a vector of 0's whose size is given by the number of features representing label-label dependencies. The features corresponding to the label-label cliques $\phi(y_t, y_{t+1})$ is given by the vector of $(\delta(y_t, \sigma)\delta(y_{t+1}, \bar{\sigma}))$ for all $\sigma, \bar{\sigma} \in \Sigma$. This vector is concatenated to a 0 vector whose size is given by the number of features representing input-label dependencies.

Then, via (1.16) the kernel function over the cliques is

$$\begin{aligned} k((x_t, y_t), (\bar{x}_{\bar{t}}, \bar{y}_{\bar{t}})) &= \delta(y_t, \bar{y}_{\bar{t}})\bar{k}(\phi(x_t), \phi(\bar{x}_{\bar{t}})), \\ k((y_t, y_{t+1}), (\bar{y}_{\bar{t}}, \bar{y}_{\bar{t}+1}) &= \delta(y_t, \bar{y}_{\bar{t}})\delta(y_{t+1}, \bar{y}_{\bar{t}+1}), \end{aligned}$$

and clearly the kernel value of different clique types is 0, as discussed in Section 1.2.5. The exponential family represented by this kernel function gives us a semi-parametric Markov Random Field. Note, when $\bar{k}$ is the linear kernel, we obtain the regularized density estimation problem of Conditional Random Fields (Lafferty et al., 2001).

The major computation in Algorithm 1.1 is the computation of R and $\partial_\gamma R$, which reduces to computing the expectations of clique configurations. For Markov chains, this is done by the Forward-Backward algorithm, using the transition and the observation matrices defined with respect to γ. The transition matrix is a $|\Sigma| \times |\Sigma|$ matrix common for all input sequences. The observation matrix of $\mathbf{x}$ is a $T \times |\Sigma|$ matrix where T is the length of $\mathbf{x}$. Let us denote the configurations of cliques of type label-input by $\mathcal{W}_{x-y} \subset \mathcal{W}(S_\mathbf{x})$ and the configurations of cliques of type label-label by $\mathcal{W}_{y-y} \subset \mathcal{W}(S_\mathbf{x})$. Furthermore for $\omega \in \mathcal{W}_{x-y}$, let ω_x and ω_y be the input and label configuration of the clique ω respectively. Then the two matrices are given by

$$T(\sigma, \bar{\sigma}) = \sum_{\omega \in \mathcal{W}_{y-y}} \delta(\omega, (\sigma, \bar{\sigma}))\gamma_\omega \tag{1.27a}$$

$$O_\mathbf{x}(t, \sigma) = \sum_{\omega \in \mathcal{W}_{x-y}} \delta(\omega_y, \sigma)\gamma_\omega \bar{k}(x_t, \omega_x) \tag{1.27b}$$

In sequence labeling, the cost function Δ is generally the Hamming loss. Then, given a new observation sequence $\mathbf{x}$, our goal is to find $\mathbf{y}^*$ via (1.1)

$$\begin{aligned} \mathbf{y}^* &= \underset{\mathbf{y}}{\operatorname{argmin}} \sum_{\mathbf{y}' \neq \mathbf{y}} p_\gamma(\mathbf{y}'|\mathbf{x}) \sum_{t=1}^{T} [\![y_t \neq y'_t]\!] \\ &= \underset{\mathbf{y}}{\operatorname{argmax}} \sum_t \sum_{\mathbf{y}' : y'_t = y_t} p_\gamma(\mathbf{y}'|\mathbf{x}). \end{aligned}$$

Thus, the best label sequence is the one with the highest marginal probability at each position, which can be found by the Forward-Backward algorithm. Note that this is different from finding the best label sequence with respect to $p(.|\mathbf{x})$ which is given by the Viterbi algorithm.

1.4.1 Pitch accent prediction

Pitch accent prediction, a sub-task of speech recognition, is detecting the words that are more prominent than others in an utterance. We model this problem as a sequence annotation problem, where $\Sigma = \{\pm 1\}$. We used Switchboard Corpus (Godfrey et al., 1992) to experimentally evaluate the described method by ex-

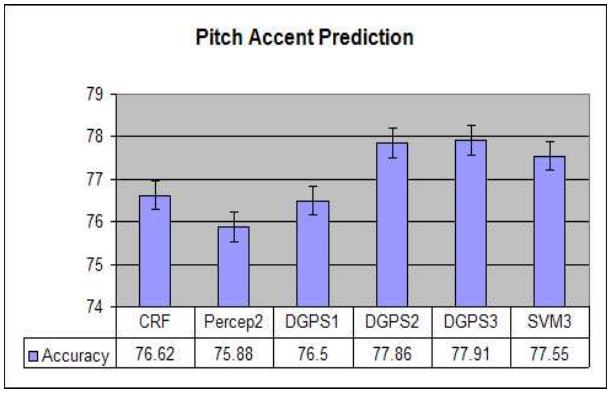

Figure 1.2 Test accuracy of Pitch Accent Prediction task.

tracting 500 sentences from this corpus and running experiments using 5-fold cross validation. Features consist of probabilistic, acoustic and textual information from the neighborhood of the label over a window of size 5 (Gregory and Altun, 2004). We chose polynomial kernel of different degrees for kernel over the inputs.

We compared the performance of CRFs and HM-SVMs(Altun et al., 2003) with the dense and sparse optimization of our approach according to their test accuracy on pitch accent prediction. When performing experiments on the dense optimization, we used polynomial kernels with different degrees (denoted with DGPS$\mathbf{X}$ in Figure 1.2 where $\mathbf{X} \in \{1, 2, 3\}$is the degree of the polynomial kernel). We used third order polynomial kernel in HM-SVMs (denoted with SVM3 in Figure 1.2)

As expected, CRFs and DGPS1 performed very similar. When 2^{nd} order features were incorporated implicitly using second degree polynomial kernel (DGPS2), the performance increases. Extracting 2^{nd} order features explicitly results in a 12 million dimensional feature space, where CRFs slow down dramatically. We observed that 3^{rd} order features do not provide significant improvement over DGPS2. HM-SVM3 performs slightly worse than DGPS2.

To investigate how the sparse optimization (denoted by SGPS) affects the performance, we report the test accuracy with respect to the sparseness of solution in Figure 1.3 using the random training sequence selection criteria where the number of parameters selected at each iteration d is 3 [3]. Sparseness is measured by the percentage of the parameters selected. The straight line is the performance of the dense optimization using second degree polynomial kernel. Using 1% of the parameters, SGPS achieves 75% accuracy (1.48% less than the accuracy of dense one). When 7.8% of the parameters are selected, the accuracy is 76.18% which is not significantly different than the performance of the dense optimization (76.48%). We observed that these parameters were related to 6.2% of the observations along

[3] The results reported here and below are obtained using a different set of features where the performance of the dense algorithm is 76.48%.

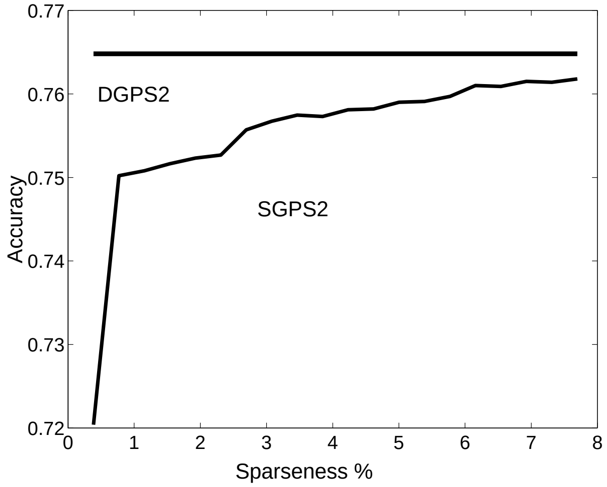

Figure 1.3 Test accuracy of Pitch Accent Prediction w.r.t. the sparseness of GPS solution.

with 1.13 label pairs on average. Thus, during inference one needs to evaluate the kernel function only at about 6% of the observations which reduces the inference time dramatically.

In order to experimentally verify how useful the predictive probabilities are as confidence scores, we forced the classifier to abstain from predicting a label when the probability of an individual label is lower than a threshold value. In Figure 1.4, we plot precision-recall values for different thresholds. We observed that the error rate decreased 8.54%, when the classifier abstained on 14.93% of the test data. The improvement on the error rate shows the validity of the probabilities generated by the classifier.

1.4.2 Named Entity Recognition

Named Entity Recognition (NER), a subtask of Information Extraction, is finding phrases containing names in a sentences. The individual labels consists of the beginning and continuation of person, location, organization and miscellaneous names and non-name. We used a Spanish newswire corpus, which was provided for the Special Session of CoNLL 2002 on NER, to select 1000 sentences (21K words). As features, we used the word and its spelling properties from a neighborhood of size 3.

The experimental setup was similar to pitch accent prediction task. We compared the performance of CRFs with and without the regularizer term (CRF-R, CRF respectively) with the dense and sparse optimizations of our approach methods. We set the sparseness parameter of SGPS to 25%, i.e. $p = 0.25|\mathcal{W}(S_{\mathbf{x}})||\Sigma|^2$, where $|\Sigma| = 9$ and $\mathcal{W}(S_{\mathbf{x}}) = 21\text{K}$ on average. The results are summarized in Table 1.1.

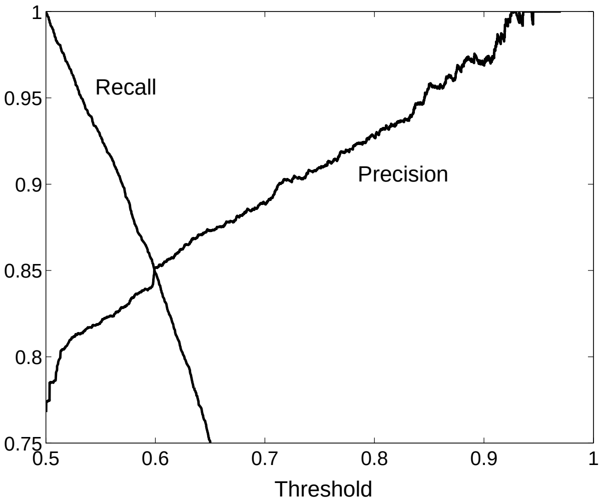

Figure 1.4 Precision-Recall curves for different threshold probabilities to abstain on Pitch Accent Prediction

	DGPS1	DGPS2	SGPS2	CRF	CRF-R
Error	4.58	4.39	4.48	4.92	4.56

Table 1.1 Test error of Named Entity Recognition task.

Qualitatively, the behavior of the different optimization methods is comparable to the pitch accent prediction task. Second degree polynomial DGPS achieved better performance than the other methods. SGPS with 25% sparseness achieves an accuracy that is only 0.1% below DGPS. We observed that 19% of the observations are selected along with 1.32 label pairs on average, which means that one needs to compute only one fifth of the gram matrix. Note, CRF without the regularization term corresponds to the maximum likelihood estimate, i. e. the estimation of the probability distribution such that the expectation of the features match exactly the empirical values. The loss of accuracy in this case shows the importance of the relaxation of the moment matching constraints.

1.5 Discussion

We presented a method for estimation conditional probability distributions over structured outputs within RKHSs. This approach is motivated through the well-established Maximum-Entropy framework. It combines the advantages of the rigorous probabilistic semantics of CRFs and overcomes the curse of dimensionality problem using kernels to construct and learn over RHKSs. The decomposition results for MRFs renders the problem tractable. Using this decomposition, we presented an efficient sparse approximate optimization algorithm. Empirical analysis showed that our approach is competitive with the state-of-the-art methods on sequence labeling.

Cost sensitivity is an important aspect in structured output prediction problem. For cost functions which decompose into cliques, such as the Hamming loss in the sequence applications, we proposed estimating the conditional distribution and performing the cost sensitive inference efficiently. Another approach is to incorporate the cost function in the optimization problem. If the cost function decomposes into the cliques, then the method presented here can be applied with minor changes. If this is not the case, the decomposition results may not hold, which may pose serious tractability problems.

Investigating the density estimation problem within the Maximum-Entropy framework points out to new directions in terms of regularizations. In particular, defining the metric of the Banach space appropriately, one can impose different regularizations for features that possess different characteristics, such as features encoding inter-label dependencies and features encoding observation-label dependencies in sequences. This is one of the topics of our future work.

References

Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *International Conference of Machine Learning*, 2003.

Y. Altun, T. Hofmann, and A.J. Smola. Exponential families for conditional random fields. In *Uncertainty in Artificial Intelligence UAI*, 2004a.

Y. Altun, T. Hofmann, and A.J. Smola. Gaussian process classification for segmenting and annotating sequences. In *International Conference on Machine Learning ICML*, 2004b.

Yasemin Altun and Alex Smola. Unifying divergence minimization and statistical inference via convex duality. In *In Nineteenth International Conference on Algorithmic Learning Theory (COLT'06)*, 2006.

Yasemin Altun, Thomas Hofmann, and Mark Johnson. Discriminative learning for label sequences via boosting. In *Proceedings of Advances in Neural Information Processing Systems (NIPS*15)*, pages 977–984, 2002.

Yasemin Altun, David McAllester, and Mikhail Belkin. Maximum margin semi-supervised learning for structured variables. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 2006.

M. Collins. Discriminative training methods for hidden markov models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2002.

James Godfrey, Ellen Holliman, and John McDaniel. SWITCHBOARD: Telephone speech corpus for research and develo pment. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 517–520, 1992.

Michelle Gregory and Yasemin Altun. Using conditional random fields to predict pitch accents in conversational speech. In *Proceedings of ACL'04:Fourty-second Annual Meeting of the Association for Computational Linguistics*, 2004.

J. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. Unpublised Manuscript, 1971.

G. S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, 41:495 – 502, 1970.

J. D. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic modeling for segmenting and labeling sequence data. In *18th International Conference on Machine Learning ICML*, 2001.

John Lafferty, Yan Liu, and Xiaojin Zhu. Kernel conditional random fields: Representation, clique selection, and semi-supervised learning. In *(ICML)*, 2004.

S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591 – 598. Morgan Kaufmann, San Francisco, CA, 2000.

Katta G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, 1998.

Vasin Punyakanok and Dan Roth. The use of classifiers in sequential inference. In *Advances in Neural Information Processing Systems (NIPS)*, pages 995–1001, 2000.

Yuan Qi, Martin Szummer, and Thomas P. Minka. Bayesian conditional random fields. In *AISTATS*, 2005.

Gunnar Ratsch, Sebastian Mika, and Manfred K. Warmuth. On the convergence of leveraging. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

D.L. Ruderman and W. Bialek. Statistics of natural images: Scaling in the woods. *Phys. Rev. Letters*, 73:814–817, 1994.

I. Steinwart. On the generalization ability of support vector machines. Technical report, University of Jena, 2001.

B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Neural Information Processing Systems 2003*, 2003.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of ICML '04: Twenty-first international conference on Machine learning*, 2004.

C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning and Inference in Graphical Models*, pages 599 – 621. MIT Press, 1999.

Tong Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Transactions on Information Theory*, 49(3):682–691, March 2003.