Semi-Supervised Transductive Speaker Identification

Oscar Täckström
Swedish Institute of Computer Science
SE-16429, Kista, Sweden
oscar@sics.se

Abstract

We present an application of transductive semi-supervised learning to the problem of speaker identification. Formulating this problem as one of transduction is the most natural choice in some scenarios, such as when annotating archived speech data. Experiments with the CHAINS corpus show that, using the basic MFCC-encoding of recorded utterances, a well known simple semi-supervised algorithm, *label spread*, can solve this problem well. With only a small number of labelled utterances, the semi-supervised algorithm drastically outperforms a state of the art supervised *support vector machine* algorithm. Although we restrict ourselves to the transductive setting in this paper, the results encourage future work on semi-supervised learning for inductive speaker identification.

Keywords: Speaker Identification, Semi-supervised Learning, Transductive Learning

1. Introduction

An ever growing body of recorded audio material, such as interviews, radio programmes, and legislative debates, can be found in archives around the world. This material carries a great potential value to broadcast companies as well as to the public and to scholars in the humanities and social sciences. However, without proper annotations, the material is not accessible, and due to its sheer size, manual annotation is in most cases an insurmountable task (Jong et al., 2008). As a first step to overcome this barrier, we have investigated the use of semi-supervised learning for automatic speaker identification, in order to facilitate annotating such material with least possible manual effort.

There are many dimensions along which recorded audio material can be annotated. Higher order dimensions of potential interest include the dimensions of gender, dialect, and emotion. In this paper we focus on the more low level task of *speaker identification*. We hope, however, that the same methods could be applicable for higher level annotations as well.

Speaker identification is a variant of speech recognition that amounts to identifying who, out of a group of speakers, made a given utterance. In the *closed set* scenario, it is a priori known that the utterance comes from a fixed set of speakers. This is contrasted with the *open set* scenario, in which the utterance could also come from some other speaker, i.e., from the complement of a fixed closed set. In this work we only consider the closed set scenario.

Most previous approaches to speaker identification have framed the problem as one of inductive learning. The aim is then to learn a classifier with optimal generalisation capability, i.e., a classifier with maximal expected performance on unseen utterances. We instead take a transductive approach, in which the aim is reduced to learning a classifier with optimal performance on a finite set of instances, to which the classifier is given access during training. Although transductive learning is limited compared to inductive learning, in some cases it might be the most natural setting. This is the case, for example, in corpus creation scenarios and with annotation of archived speech, where the set of speakers is fixed and the data is static, so that there is no need to make generalisations beyond the given sample.

2. Semi-supervised learning

Most machine learning research have been focused on *supervised learning*, in which the learner is given access to only labelled data during training, or *unsupervised learning*, in which the learner is only provided with unlabelled data. In the last decade there has been a surge of interest in *semi-supervised learning*, in which the learner is given access to both labelled and unlabelled data examples during training. The goal is to use as little labelled data as possible, since the manual labelling is often expensive to produce, by leveraging much more cheaply obtained unlabelled data. For an overview of this rapidly developing field, see Chapelle et al. (2006b).

More formally, a semi-supervised learning problem has the following setup. Let $\mathcal{X}$ be an input space; typically a metric space. Let $X_u = \{x_i | x_i \in \mathcal{X}\}_{l+1}^{u}$ be a set of unlabelled instances, drawn i.i.d. from some distribution on $\mathcal{X}$ (the choice of indexing will soon become clear). Let $\mathcal{Y}$ be an output space, and let $X_l = \{x_i | x_i \in \mathcal{X}\}_1^l$ be a set of labelled instances, with labels given by $Y = \{y_i | y_i \in \mathcal{Y}\}_1^l$, and with pairs $(X_l, Y) = \{(x_i, y_i)\}_1^l$ drawn i.i.d. from some distribution on $\mathcal{X} \times \mathcal{Y}$.

In standard supervised learning one seeks to learn a mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ from the limited training data (X_l, Y). In the case of speaker identification, this amounts to learning a classification function that maps utterances to their corresponding speaker. With semi-supervised learning, the goal is to make use of the structure implicitly provided by X_u – which does not provide any information on the mapping $\mathcal{X} \mapsto \mathcal{Y}$ – together with information on this mapping, provided by (X_l, Y). The hope is that a small set of labelled instances can be compensated for by a large set of unlabelled instances, which is much cheaper to get hold of.

A further distinction is made between *inductive* and *transductive* learning. In inductive learning the aim of the learner is to find a classifier that labels instances, drawn i.i.d. from the same underlying distribution as generated $\mathcal{X}$, with as small expected loss as possible. In a transductive learning setting, in contrast, the aim of the learner is only to find an optimal labelling $\mathcal{Y} = \{y_i\}_1^u$ of the set $X_l \cup X_u$, with performance usually only measured on $\{(x_i, y_i)\}_{l+1}^u$. Which loss function to use is determined based on the specific problem at hand; in this work performance is measured using the binary loss function.

There is an ongoing discussion in the machine learning community as to whether transductive learning is an, in principle, simpler problem than inductive learning, and thus more appropriate when out-of-sample extensions are not really required. Chapelle et al. (2006a) present different views on this issue. As discussed above, we are only concerned with the transductive setting.

The central idea underlying all approaches to semi-supervised learning is that the structure of the set $\mathcal{X}$ alone, can provide valuable information on the labelling of the instances in $\mathcal{X}$. This notion is encoded in the *clustering assumption*, which states that decision boundaries should lie in low-density regions or, equivalently, that points belonging to the same cluster are likely to belong to the same class; and the *manifold assumption*, according to which the high-dimensional instance space $\mathcal{X}$ actually lies on a low-dimensional manifold. One or both of these assumptions, together with the assumption of *local consistency*, which states that nearby points are likely to share labels, are assumed to hold by most semi-supervised algorithms (Zhou et al., 2003; Chapelle et al., 2006b).

3. Data representation and distance measures

In order to devise concrete algorithms based on the abstract formulation of the semi-supervised learning problem above, we need to choose a representation for the instances (utterances), $x_i \in \mathcal{X}$, and a measure of distances between pairs, $(x_i, x_j) \in \mathcal{X}$, of instances. In this paper we assume that instances are represented as real valued vectors in $\Re^n$ and that distances are computed by a positive semi-definite kernel function $K : \mathcal{X} \times \mathcal{X} \mapsto \Re^+$. As for the representation of the speakers, we assume that there are m different speakers, indexed such that $\mathcal{Y} = \{y_i\}_1^m$. This is a standard kernel based classifier learning scenario; for a comprehensive introduction to kernel methods in machine learning, see for example, (Shawe-Taylor and Cristianini, 2004).

3.1. Utterance encoding

The predominant encoding methods used for speaker identification are the same as those used in automatic speech recognition. This is in a way contradictory, since the goal of speech recognition is to provide as speaker independent models of content as possible, while the aim of speaker identification is to provide models agnostic to speech content. Despite this contradiction, the same models seem to work quite well for both tasks, at least under controlled conditions; but see (Grimaldi and Cummins, 2008) for a recent critique on the use of source-filter based encoding and assumptions of local stationarity in speaker identification and verification tasks. The main argument put forward in the cited paper is that this encoding is highly sensitive to speaking and channel conditions.

The most common encoding schemes for speech data are linear filter cepstral coefficients (LFCCs), Mel-scale cepstral coefficients (MFCCs), linear predictive coding coefficients (LPCs) and perceptual linear prediction coefficients (PLPs) (Holmes and Holmes, 2002). Of these MFCCs seem to be the most popular for speaker identification and verification. Since the focus of this work is on assessing the potential for applying semi-supervised learning to speaker annotation, rather than on optimal encoding, we use a standard MFCC based encoding in which each utterance is represented as a sequence of frames. Each frame is represented by a real valued vector with elements corresponding to 12 cepstral coefficients, mean energy level coefficient and the Δ approximation of the first and second order time derivatives of these coefficients. This results in 39 dimensions for each frame vector, with 100 frames being generated per second with a window size of 25 milliseconds. We used the open source HTK-toolkit, available at http://htk.eng.cam.ac.uk, to extract these features, with configuration parameters according to table 1. No additional pre-processing was performed, except for that provided by HTK by default.

Parameter	Setting
TARGETKIND	MFCC_E_D_A
TARGETRATE	100 000
WINDOWSIZE	250 000
USEHAMMING	T
PREEMCOEF	0.0
NUMCHANS	26
NUMCEPS	12
ENORMALISE	T
LOFREQ	0
HIFREQ	8000

Table 1: HTK configuration for MFCC extraction

3.2. Kernel functions

As described in the previous section, each utterance is represented as a sequence of frame vectors capturing the locally stationary spectral properties of the speech signal. In order to use these frame vectors in the learning scenario sketched above, we need to define a distance measure between pairs of utterances, i.e., between pairs of sequences of frame vectors.

The choice of an appropriate distance measure is dependent on the learning algorithm. For example, a Gaussian mixture model (GMM, briefly discussed below) does not exploit any sequential information and only makes use of frame level information – it is equivalent to a single state hidden Markov model (HMM) – and implicitly makes use of the standard Euclidian distance on $\Re^n$ in the computation of the mixture memberships for each frame.

The kernel based methods that are the focus of this work on the other hand rely on a distance measure on pairs of sequences of frames. In order for theoretical results on the convergence of these algorithms to hold, the distance measure must be a positive semi-definite kernel function. A substantial range of kernels defined on structured data, such as sequences, have been proposed; see (Gärtner, 2003) for a survey. Kernels proposed for speaker verification and identification include the computationally expensive Fisher kernel (Haussler, 1999) used by Wan and Renals (2005) and the *mean* and *max*[1] kernels employed by Mariéthoz and Bengio (2007).

We take the following simple route to the problem of handling the sequential structure of the instances. First we sum the frame vectors for each utterance and then we normalise the resulting utterance vectors to unit Euclidean norm. Any valid kernel function could then be used to compute the distance between utterance vectors, however we again keep things simple and use a linear kernel:

$$k_{\text{lin}}(x_i, x_j) = \phi(x_i) \cdot \phi(x_j),$$

where $\phi(x) = \sum_{t=1}^{T_x} x^{(t)} / \| \sum_{t=1}^{T_x} x^{(t)} \|$, $x^{(t)}$ denotes the tth frame vector of utterance x, T_x is the total number of frames in x, and $\cdot$ denotes the standard dot-product in $\Re^n$; or as a radial basis function (RBF) kernel with variance σ:

$$k_{\text{rbf}}(x_i, x_j) = \exp\left(\frac{-\|\phi(x_i) - \phi(x_j)\|^2}{2\sigma^2} \right).$$

In the case of the linear kernel, this scheme corresponds to normalising each frame before computing the pair-wise distances between all pairs of frames, which is a similar operation to that performed by the *mean* kernel when a linear kernel is used to compute distances between pairs of frames.

A further issue in semi-supervised learning is whether the clustering and manifold assumptions are plausible, given the chosen representation and distance measure. The Mel-scale cepstral coefficients are known to capture at least some aspects of human speech that are specific to the speaker, and since each speaker has a rather stable and characteristic voice, utterances should form clusters under this encoding. Furthermore, since the human vocal tract has limited degrees of freedom, utterances should indeed be well described by a manifold of lower dimension. Both assumptions should thus be considered plausible in this case.

4. Learning algorithms

The predominant framework for speaker identification and verification is based on a generative Gaussian mixture model (GMM) (Reynolds and Rose, 1995). The parameters of the model are usually fit to the data using the method of expectation maximisation (EM). For speaker identification one can then use the n-way classification function $f(x) = \operatorname{argmax}_{y_i} P(x|\theta_{y_i})P(\theta_{y_i})$, where θ_{y_i} are the parameters estimated for speaker y_i, to predict the speaker of utterance x. Recently, discriminative frameworks, most notably support vector machines (SVMs), that directly try to model $\operatorname{argmax}_{y_i} P(y_i|x)$ instead of indirectly by way of $P(x|\theta)P(\theta)$ have gained popularity for speaker identification (Wan and Renals, 2005; Mariéthoz and Bengio, 2007).

In this work we are mainly interested in the semi-supervised label spread algorithm described in the next section. For comparison we also make use of the supervised SVM algorithm. Since this is a very well known algorithm, we refer the reader to, for example, Vapnik (2000) for a description. For the experiments below we used the SVM implementation provided by the open source LIBSVM library (Chang and Lin, 2001). We performed the experiments using both the linear and RBF kernels described in the previous section.

The label spread algorithm, introduced in (Zhou et al., 2003), is a transductive semi-supervised learning algorithm based on the clustering and manifold assumptions previously discussed. The idea is to find a labelling Y_u of the set X_u such that the labelling is smooth with respect to local distances as well as with respect to the underlying structure of the data; this is referred to as *local* and *global* consistency, respectively.

Local distances are defined by means of the RBF kernel, k_{rbf}, while the global structure is encoded by a normalised version of the *affinity matrix* W, with $W_{ij} = k_{\text{rbf}}(x_i, x_j)$ for $i = 1 \ldots l + u$, $i \neq j$ and $W_{ii} = 0$. This matrix represents the edges of the graph of pair-wise weighted distances between instances in $X_l \cup X_u$, which captures the geometry induced by both labelled and unlabelled data.

The idea of the label spread algorithm is to iteratively let each instance spread information on its label to other instances. The amount of information spread is dependent on the geometry of the data, with nearby instances receiving more information than distant instances. After convergence, the labels will have spread in such a way that similar instances have the same labels and instances belonging to the same cluster – with clusters determined by the structure of the graph G – have the same labels.

The algorithm can be described as performing the following steps (Zhou et al., 2003):

1. Compute the affinity matrix W as defined above and set $t = 0$.
2. Form the normalised graph Laplacian $L = D^{-1/2}WD^{-1/2}$, with D being the diagonal degree matrix with $D_{ii} = \sum_j W_{ij}$.
3. Initialise $Y^{(0)} = (Y_1^T, \ldots, Y_l^T, 0, \ldots, 0)^T$, where Y_i is the class indicator row vector with all elements zero except for element $Y_{ij} = y_i$.
4. Iterate $Y^{(t+1)} = \alpha L Y^{(t)} + (1 - \alpha)Y^{(0)}$ until convergence to $Y^{(\infty)}$, where α is a parameter in (0,1).
5. Label point x_i according to $f(x_i) = \operatorname{argmax}_j Y_{ij}^{(\infty)}$.

Zhou et al. (2003) give a proof of convergence for the above algorithm, and they show that it has the closed form solution $Y^{(\infty)} = (I - \alpha L)^{-1} Y^{(0)}$.

The introduction of the Laplacian, L, may be easier grasped by formulating the above algorithm as the equivalent regularised minimisation problem (Zhou et al., 2003):

$$\frac{1}{2} \left(\underbrace{\sum_{i,j=1}^{l+u} W_{ij} \left\| \frac{Y_i}{\sqrt{D_{ii}}} - \frac{Y_j}{\sqrt{D_{jj}}} \right\|^2}_{\text{smoothness}} + \underbrace{\mu \sum_{i=1}^{l} \left\| Y_i - Y_i^0 \right\|^2}_{\text{fitness}} \right),$$

[1] Note that the *max* kernel is not a positive semi-definite function.

where μ is a regularisation parameter.

By construction W_{ij} is non-zero in regions where points are close, and zero or small in regions where points are far apart. The term $W_{ij} \left\| Y_i/\sqrt{D_{ii}} - Y_j/\sqrt{D_{jj}} \right\|^2$ will thus penalise large variations of the labelling function in high-density regions with respect to the manifold, in effect implementing the clustering and manifold assumptions.

5. Experiments

In order to evaluate the different approaches described above, we conducted a set of experiments in which we investigated the following:

1. The potential for using the semi-supervised learning algorithm – label spread – for transductive speaker identification as compared to a reference inductive supervised learning algorithm – the support vector machine.

2. How the performance of these learning algorithms is affected by the number of labelled instances.

3. The effect of the number of speakers on the algorithms' learning performance.

All experiments where performed on datasets created from data in the CHAINS corpus (Cummins et al., 2006).[2] This corpus contains utterances recorded under varying different speaking conditions. For these experiments we only made use of the SLOW part, which is comprised of 33 utterances each by 36 speakers. Each utterance is approximately 2-3 seconds in length.

From this corpus we generated a total of 4×7 datasets by varying the number of speakers, $m \in \{4, 8, 16, 36\}$, and the number of labelled instances from each class $l_j \in \{1, 2, 4, 7, 10, 14, 17\}$, corresponding to proportions of labelled instances according to $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$. For each dataset we ran each learning algorithm 10 times, in order to reduce the effects of random noise and to estimate the variance of the classifier performance. For each run, the labelled part of each of the datasets was picked by sampling l_j utterances from each speaker, without replacement.

Before turning to the results of the experiments on these datasets there are two caveats. First, each of the algorithms has parameters which need to be optimised, in order to get maximum performance for each dataset. However, this goes against the idea of using semi-supervised learning, in which we want to annotate as few data points as possible. We therefore cheated somewhat by running prior experiments to determine reasonable parameter values. Fortunately, the optimal parameter values were very stable in these experiments, indicating that choosing a standard setting should work well on similar datasets. Note that in the literature on semi-supervised learning, it is customary to report best parameter values in this way, though lately this has been criticised.

Second, by picking the labelled instances according to the a priori known uniform distribution over classes we are cheating as well. Since in general we cannot expect to know the exact distribution over classes, we need to randomly sample the set of labelled instances. When we are selecting a very small number of labelled instances, we run a significant risk of obtaining an erroneous estimate of the label distribution. This can be a severe problem in practice, since the algorithms in use are sensitive to this estimate. A more systematic perturbation analysis is thus necessary in order to assess the utility of these algorithms in real world scenarios.

With these caveats in mind, the results of the experiments are given in figure 1 (a-d). As indicated by these figures, the performance of the semi-supervised learning algorithm is vastly superior to the supervised algorithms when the number of labelled instances is small. Even when only one utterance is provided for each speaker, the label spread algorithm gives rather useable results. When the proportion of labelled examples is increased label spread performs on par with the support vector machine with the RBF-kernel.

Since the label spread algorithm is more computationally demanding, it does not make sense to apply it when more labelled training data is available. However, semi-supervised algorithms generally perform better when more unlabelled data is available as well. Unfortunately we were unable to investigate this issue further, due to the small size of the currently used corpus.

Although we have only presented results on speaker identification in this paper, when analysing the errors made by the semi-supervised algorithm, we noted that errors were much less common across gender and dialect borders, than within. This suggest that the same method can be used for annotating spoken data along other dimensions, such as those mentioned in the introduction, as well. This would be a particularly interesting possibility for scholars, who could select an annotation dimension of choice, manually annotate a small subset of their data along this dimension, and let the semi-supervised algorithm do the rest.

6. Conclusions

We have shown that semi-supervised learning can be successfully applied to the task of transductive speaker annotation. When the number of labelled utterances is very small this method significantly outperforms inductive support vector machines, while performing on par when the number of labelled utterances is increased. While the utility of transductive learning might be limited compared to that of inductive learning, these results should encourage further work on using semi-supervised learning transductive as well as for inductive speaker identification.

References

Chang, C. and C. Lin (2001). LIBSVM: a library for support vector machines.

Chapelle, O., B. Schölkopf, and A. Zien (2006a). A discussion of Semi-Supervised learning and transduction.

[2] CHAINS is released under a Creative Commons licence, and can be downloaded free of charge at http://chains.ucd.ie.

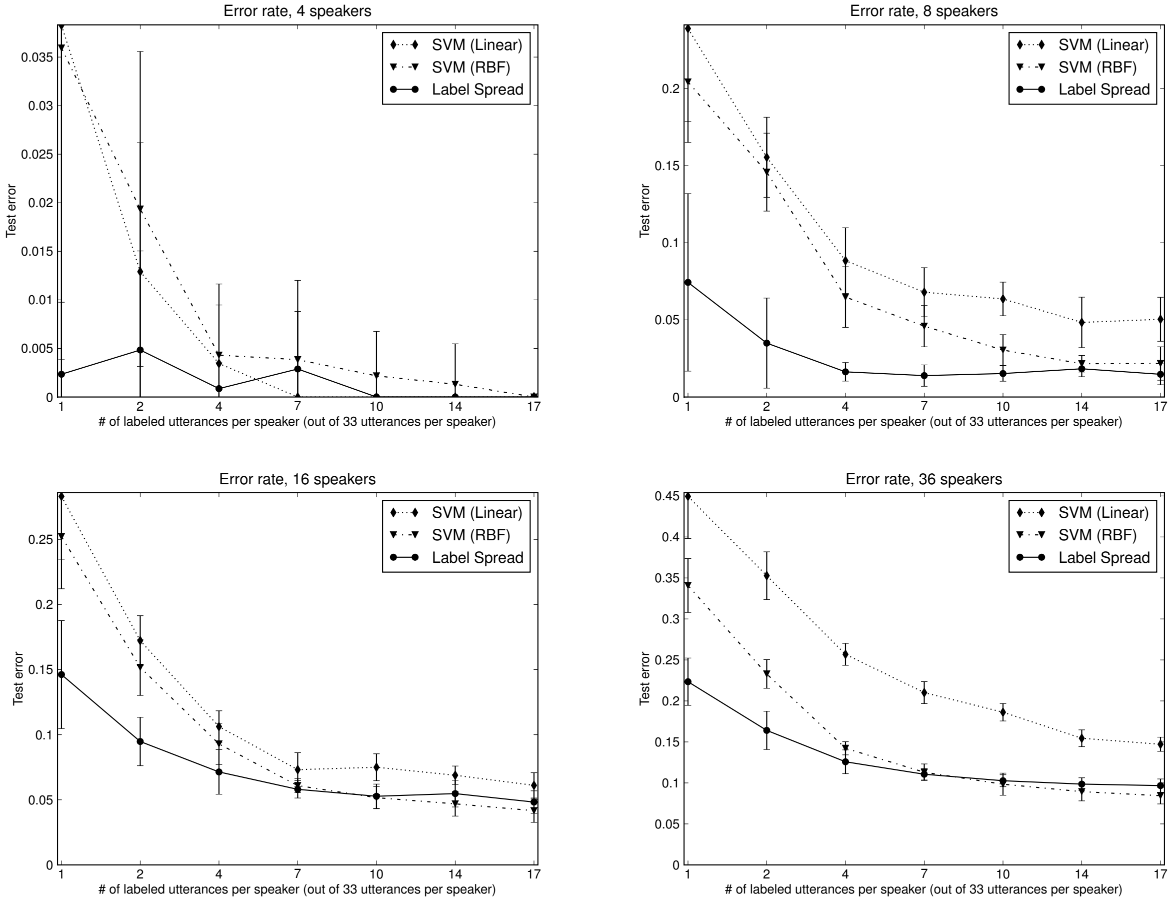

Figure 1: Top-left, top-right, bottom-left and bottom-right: the error rates for 4,8,16 and 36 speakers, respectively, on the dataset generated from the SLOW part of the CHAINS corpus. Error bars indicate standard deviations calculated after running each experiment 10 times.

In O. Chapelle, B. Schölkopf, and A. Zien (Eds.), *Semi-Supervised Learning*, Adaptive Computation and Machine Learning, pp. 473–478. The MIT Press.

Chapelle, O., B. Schölkopf, and A. Zien (Eds.) (2006b). *Semi-Supervised Learning*. Adaptive Computation and Machine Learning. The MIT Press.

Cummins, F., M. Grimaldi, T. Leonard, and J. Simko (2006). The CHAINS corpus: CHAracterizing INdividual speakers. In *Proceedings of SPECOM'06*, pp. 431–435.

Grimaldi, M. and F. Cummins (2008). Speaker identification using instantaneous frequencies. *IEEE Transactions on Audio, Speech, and Language Processing 16*(6), 1097–1111.

Gärtner, T. (2003). A survey of kernels for structured data. *ACM SIGKDD Explorer Newsletter 5*(1), 49–58.

Haussler, D. (1999). Convolution kernels on discrete structures. Technical Report UCSC-RL-99-10, UCSC.

Holmes, J. and W. Holmes (2002). *Speech Synthesis and Recognition* (2 ed.). Taylor & Francis, Inc.

Jong, F. D., D. W. Oard, W. Heeren, and R. Ordelman (2008). Access to recorded interviews: A research agenda. *Journal on Computational and Cultural Heritage 1*(1), 1–27.

Mariéthoz, J. and S. Bengio (2007). A kernel trick for sequences applied to text-independent speaker verification systems. *Pattern Recognition 40*(8), 2315–2324.

Reynolds, D. and R. Rose (1995). Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing 3*(1), 83–72.

Shawe-Taylor, J. and N. Cristianini (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. Springer Verlag.

Wan, V. and S. Renals (2005). Speaker verification using sequence discriminant support vector machines. *Speech and Audio Processing, IEEE Transactions on 13*(2), 203–210.

Zhou, D., O. Bousquet, T. Lal, J. Weston, and B. Schölkopf (2003). Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, Volume 16.