A farewell to SVM: Bayes Factor Speaker Detection in Supervector Space

Niko Brümmer

April 4, 2006

1 Introduction

We are interested in the speaker detection problem where, given just two speech segments, we have to decide:

H_1: The two speech segments were spoken by the same speaker.

H_2: The two speech segments were spoken by two different speakers.

(The methodology that we shall develop in this paper allows for the generalization where $N-1$ speech segments, all from the same speaker, are given and the question is posed whether the Nth speech segment is from the same or a different speaker. But for simplicity of exposition, we consider the two-segment case.)

Recent work in speaker detection has suggested that there exist good supervector-based strategies to detect speakers. This is a three-part strategy which can be described thus:

1. Extract a low-dimensional *feature vector* for every 10ms frame of each input speech segment. This leads to a separate *variable-length* sequence of feature vectors for each of the two speech segments. All further processing is based solely on these two sequences of feature vectors.

2. Map each of the two feature-vector sequences to a *fixed-size supervector* of very high dimension.(A supervector is just a vector, where the prefix super- is used to emphasize the distinction from feature vectors.) All further processing is based only on these two supervectors.

3. Process the two supervectors to decide between H_1 and H_2.

Examples of the supervectors extracted in step 2 include:

- Averaged polynomial expansion of feature vectors.
- Concatenated GMM means.
- MLLR transform parameters obtained when a *speech* recognizer is adapted in unsupervised mode on each segment.

All of these representations give fixed-length supervectors (of high dimension) and have been proven to contain high-quality speaker information.

In what follows, we shall be interested in step 3. How do we make the decisions in supervector space? In all of the speaker recognition literature where this three-part strategy is followed, SVM modeling is used to accomplish this. The recipe is roughly the following:

2 SVM recipe

1. Choose one of the supervectors to be the 'training' vector and train an SVM to distinguish this supervector from those of a large set of supervectors of background speakers. A linear SVM kernel is invariably used. This results in a model vector of the same dimension as the input supervectors[1].
2. Project the other supervector, denoted the 'test' vector onto the model vector (dot-product) to obtain the SVM score. This score can be thresholded to make decisions. This results in a linear (hyperplane) decision boundary in supervector space.
3. To improve performance, directions in supervector space which are considered (by some heuristic) to contain high intra-speaker variability, may be projected away before training and/or testing.

3 Bayesian solution

Although very good performance can be achieved with the SVM recipe, it remains an ad-hoc solution. The Bayesian solution is to explicitly model all sources of uncertainty and then to use probability theory to make decisions.

Recent publications and NIST evaluations have shown that the following model for inter- and intra-speaker variability (particularly in the case of GMM-based supervectors) give good results (but admittedly not in quite the same three-part strategy):

3.1 Supervector model

A supervector $\mathbf{x}$ for a given segment is an additive combination between a speaker-dependent supervector $\mathbf{s}$ and an intra-speaker nuisance vector $\mathbf{n}$:

$$\mathbf{x} = \mathbf{s} + \mathbf{n} \tag{1}$$

where we assume both speaker and nuisance vectors are normally[2] distributed:

$$\begin{aligned} \mathbf{s} &\sim \mathcal{N}(\mu, \mathbf{D}) \\ \mathbf{n} &\sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \end{aligned} \tag{2}$$

That is, we model inter-speaker variability with the covariance matrix $\mathbf{D}$ and intra-speaker variability with the covariance matrix[3] $\mathbf{C}$.

[1] and also an offset constant

[2] We can of course take steps to improve the normality of our supervectors. Recent SRI publications have shown that histogram normalization of supervector components are a good idea.

[3] In the general case, we shall allow $\mathbf{C}$ to not necessarily be of full rank and therefore to be non-invertible. This means the intra-speaker variability can be limited to a subspace.

From this model we can immediately deduce:

$$\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{C} + \mathbf{D}) \tag{3}$$

Now let the supervector $\mathbf{y}$ for the other speech segment be modeled similarly:

$$\mathbf{y} = \mathbf{s}' + \mathbf{n}' \tag{4}$$

where

$$\begin{aligned} \mathbf{s}' &\sim \mathcal{N}(\mu, \mathbf{D}) \\ \mathbf{n}' &\sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \\ \mathbf{y} &\sim \mathcal{N}(\mu, \mathbf{C} + \mathbf{D}) \end{aligned} \tag{5}$$

After some work, we can deduce the joint probability distributions to be:

- Under H_1, where $\mathbf{s} = \mathbf{s}'$ and where $\mathbf{n}$ and $\mathbf{n}'$ are independent:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \mathbf{D} + \mathbf{C} & \mathbf{D} \\ \mathbf{D} & \mathbf{D} + \mathbf{C} \end{bmatrix}), \text{ given } H_1 \tag{6}$$

- Under H_2, where $\mathbf{s}$ and $\mathbf{s}'$ are also independent:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \mathbf{D} + \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} + \mathbf{C} \end{bmatrix}), \text{ given } H_2 \tag{7}$$

From equation 6 we see the somewhat surprising result that the cross-correlation between supervectors of the same speaker is just the inter-speaker covariance:

$$\mathrm{E}\{(\mathbf{x} - \mu)(\mathbf{y} - \mu)^T\} = \mathrm{E}\{(\mathbf{y} - \mu)(\mathbf{x} - \mu)^T\} = \mathbf{D}, \text{ given } H_1 \tag{8}$$

where $\mathrm{E}\{\cdot\}$ denotes expectation and ^{T} denotes transpose. But this can be explained by noting that under the independence assumptions of this model, all cross-correlations between nuisance vectors and between nuisance and speaker vectors vanish.

It is also useful to consider the distributions of the following re-parameterization of the joint vector:

- Under H_1:

$$\begin{bmatrix} \mathbf{x} - \mathbf{y} \\ \mathbf{x} + \mathbf{y} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mathbf{0} \\ 2\mu \end{bmatrix}, \begin{bmatrix} 2\mathbf{C} & \mathbf{0} \\ \mathbf{0} & 2\mathbf{C} + 4\mathbf{D} \end{bmatrix}), \text{ given } H_1 \tag{9}$$

- Under H_2:

$$\begin{bmatrix} \mathbf{x} - \mathbf{y} \\ \mathbf{x} + \mathbf{y} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mathbf{0} \\ 2\mu \end{bmatrix}, \begin{bmatrix} 2\mathbf{C} + 2\mathbf{D} & \mathbf{0} \\ \mathbf{0} & 2\mathbf{C} + 2\mathbf{D} \end{bmatrix}), \text{ given } H_2 \tag{10}$$

3.2 Bayes factor

Using this model, in the general case, we may approach the decision problem via calculation of the Bayes factor:

$$\mathcal{B}(\mathbf{x},\mathbf{y}) = \frac{p(\mathbf{x},\mathbf{y}|H_1)}{p(\mathbf{x},\mathbf{y}|H_2)} = \frac{\mathcal{N}(\mathbf{z}|\mathbf{m},\Sigma_1)}{\mathcal{N}(\mathbf{z}|\mathbf{m},\Sigma_2)} \tag{11}$$

where

$$\begin{aligned} \mathbf{z} &= \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \\ \mathbf{m} &= \begin{bmatrix} \mu \\ \mu \end{bmatrix} \\ \Sigma_1 &= \begin{bmatrix} \mathbf{D}+\mathbf{C} & \mathbf{D} \\ \mathbf{D} & \mathbf{D}+\mathbf{C} \end{bmatrix} \\ \mathbf{\Sigma_2} &= \begin{bmatrix} \mathbf{D}+\mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}+\mathbf{C} \end{bmatrix} \end{aligned} \tag{12}$$

where we have assumed that $\mathbf{D}$ is of full rank, making the covariances invertible.

However under the special case where $\mathbf{C}$ is also of full rank, we may factor the equation so that the matrix inversions are smaller:

$$\begin{aligned} \mathcal{B}(\mathbf{x},\mathbf{y}) &= \frac{\int \mathcal{N}(\mathbf{x}|\mathbf{s},\mathbf{C})\mathcal{N}(\mathbf{y}|\mathbf{s},\mathbf{C})\mathcal{N}(\mathbf{s}|\mu,\mathbf{D})d\mathbf{s}}{(\int \mathcal{N}(\mathbf{x}|\mathbf{s},\mathbf{C})\mathcal{N}(\mathbf{s}|\mu,\mathbf{D})d\mathbf{s})(\int \mathcal{N}(\mathbf{y}|\mathbf{s},\mathbf{C})\mathcal{N}(\mathbf{s}|\mu,\mathbf{D})d\mathbf{s})} \\ &= \frac{\mathcal{N}(\mathbf{x}-\mathbf{y}|\mathbf{0},2\mathbf{C})\mathcal{N}(\frac{\mathbf{x}+\mathbf{y}}{2}|\mu,\mathbf{D}+\frac{\mathbf{C}}{2})}{\mathcal{N}(\mathbf{x}|\mu,\mathbf{C}+\mathbf{D})\mathcal{N}(\mathbf{y}|\mu,\mathbf{C}+\mathbf{D})} \end{aligned} \tag{13}$$

This equation also shows why we call $\mathcal{B}(\mathbf{x},\mathbf{y})$ a *Bayes factor* — if we consider $\mathbf{s}$ to be a speaker model, then we are effectively integrating over all possible models. With this approach there is no need to make hard decisions in order to obtain speaker models. We can directly compute the score from the two supervectors, in a symmetrical way.

3.3 Score

We let our *detection score* for a trial $(\mathbf{x},\mathbf{y})$ be the log-likelihood-ratio which is given by the log of the Bayes factor. This can be written as:

$$\mathcal{S}(\mathbf{x},\mathbf{y}) = \log \mathcal{B}(\mathbf{x},\mathbf{y}) = \mathbf{x}^T\mathbf{M}_1\mathbf{x}+\mathbf{y}^T\mathbf{M}_1\mathbf{y}+\mathbf{x}^T\mathbf{M}_2\mathbf{y}+\mathbf{x}^T\mathbf{m}_3+\mathbf{y}^T\mathbf{m}_3+m_4 \tag{14}$$

where the square matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ and the vector $\mathbf{m}_3$ and the scalar m_4 are constants which depend on the parameters $\mathbf{C}$, $\mathbf{D}$ and μ. In short, the score is a quadratic function of the joint vector $\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$.

Note that the score does *not* give a distance measure between $\mathbf{x}$ and $\mathbf{y}$. Rather, the score as a function of the joint vector has a saddle-point at $\mathbf{x} = \mathbf{y} = \mu$, from where the values rise to arbitrarily large positive, when $\mathbf{x}$ is close to $\mathbf{y}$ and both far from from μ; and from where the values fall to arbitrarily large negative, when all of $\mathbf{x}$, $\mathbf{y}$ and μ are far from each other. This has the following interpretation:

- When $\mathbf{x}$ and $\mathbf{y}$ are close to μ, we can never affirm with great certainty that they are the same speaker, not even when $\mathbf{x} = \mathbf{y}$.
- The further away from μ we move $\mathbf{x}$ and $\mathbf{y}$, the more certain we can be of either H_1 or H_2.

If this model proves to be an accurate description of what is really going on, then this offers some explanation for the "wolf/sheep/lamb/goat" phenomenon of the difference in recognizability of different speakers[4].

3.4 Regularization

The matrices $\mathbf{C}$ and $\mathbf{D}$ are huge, having numbers of elements that grow as the square of the supervector dimension. We have to make some regularization assumptions in order to work with these parameters. In particular, we shall work with *factor analysis* decompositions of these covariance matrices:

$$\mathbf{C} = \mathbf{E} + \mathbf{F}\mathbf{G}\mathbf{F}^T \tag{15}$$
$$\mathbf{D} = \mathbf{H} + \mathbf{J}\mathbf{K}\mathbf{J}^T \tag{16}$$

where $\mathbf{E}$ and $\mathbf{H}$ are d by d diagonal; $\mathbf{F}$ is d by n_F rectangular; $\mathbf{J}$ is d by n_J rectangular; $\mathbf{G}$ is n_F by n_F diagonal; $\mathbf{K}$ is n_J by n_J diagonal; d is the supervector dimension and where $n_F << d$ and $n_J << d$.

Note that in the special case where $\mathbf{E}$ and $\mathbf{H}$ are *isotropic*, that is if they are scalar multiples of the identity matrix, then the factor analysis model is called a probabilistic principal component analysis (PPCA).

3.4.1 Factor analysis model format

The matrices $\mathbf{G}$ and $\mathbf{K}$ are strictly speaking unnecessary, if $\mathbf{F}$ and $\mathbf{J}$ are unconstrained. But if we constrain $\mathbf{F}$ and $\mathbf{J}$ to be orthonormal, we need them. We can collapse the model to a simpler format by letting:

$$\mathbf{W} = \mathbf{F}\mathbf{G}^{\frac{1}{2}} \tag{17}$$
$$\mathbf{V} = \mathbf{J}\mathbf{K}^{\frac{1}{2}} \tag{18}$$

so that

$$\mathbf{C} = \mathbf{E} + \mathbf{W}\mathbf{W}^T \tag{19}$$
$$\mathbf{D} = \mathbf{H} + \mathbf{V}\mathbf{V}^T \tag{20}$$

If we start with a factor analysis model in this reduced $\mathbf{W}\mathbf{W}^T$ format, we can convert it back to the $\mathbf{F}\mathbf{G}\mathbf{F}^T$ format via an eigen-analysis[5] of the small n_F by n_F matrix $\mathbf{W}^T\mathbf{W}$. This eigen-analysis (or diagonalization) gives:

$$\mathbf{W}^T\mathbf{W} = \mathbf{R}\Lambda\mathbf{R}^T \tag{21}$$
$$\mathbf{W}\mathbf{W}^T = (\mathbf{W}\mathbf{R})(\mathbf{W}\mathbf{R})^T \tag{22}$$
$$\Lambda = (\mathbf{W}\mathbf{R})^T(\mathbf{W}\mathbf{R}) \tag{23}$$

[4]See `http://www.nist.gov/speech/publications/papersrc/icslp_98.pdf`.

[5]Fortuitously we don't need to do an eigen-analysis of the huge d by d matrux $\mathbf{W}\mathbf{W}^T$.

where $\mathbf{R}$ is a unitary[6] matrix having the normalized eigenvectors of $\mathbf{W}^T\mathbf{W}$ as columns; and where Λ is a diagonal matrix of eigenvalues. As the last equation shows, the columns of the tall d by n_F matrix ($\mathbf{WR}$) are orthogonal, but not normalized. Now if we let $\mathbf{F}$ be the orthonormal matrix of normalized columns of ($\mathbf{WR}$) and if we let $\mathbf{G} = \Lambda$, then we have the desired format conversion:

$$\mathbf{W}\mathbf{W}^T = \mathbf{F}\mathbf{G}\mathbf{F}^T \tag{24}$$

3.4.2 Linear combinations

We need to be able to form linear combinations of $\mathbf{C}$ and $\mathbf{D}$. Fortunately if we use the format $\mathbf{C} = \mathbf{E} + \mathbf{W}\mathbf{W}^T$ and $\mathbf{D} = \mathbf{H} + \mathbf{V}\mathbf{V}^T$, this is easy. We get a result that is in the *same* format:

$$\alpha\mathbf{C} + \beta\mathbf{D} = \mathbf{A} + \mathbf{U}\mathbf{U}^T \tag{25}$$

$$\mathbf{A} = \alpha\mathbf{E} + \beta\mathbf{D} \tag{26}$$

$$\mathbf{U} = \left[\, \sqrt{\alpha}\mathbf{W}\sqrt{\beta}\mathbf{V} \,\right] \tag{27}$$

where again $\mathbf{A}$ is diagonal and $\mathbf{U}$ is of low rank, d by $n_F + n_J$.

3.4.3 Inversion and determinants

This factor analysis covariance model lends itself, via the *matrix inversion lemma*, to tractable ways of calculating the determinants and inverses of the huge d by d matrices which we need to implement equation 13. In the case of $\mathbf{C} = \mathbf{E} + \mathbf{F}\mathbf{G}\mathbf{F}^T$ we have:

$$\mathbf{C}^{-1} = \mathbf{E}^{-1} - \mathbf{E}^{-1}\mathbf{F}\mathbf{L}^{-1}\mathbf{F}^T\mathbf{E}^{-1} \tag{28}$$

$$|\mathbf{C}| = |\mathbf{E}||\mathbf{G}||\mathbf{L}| \tag{29}$$

$$\mathbf{L} = \left(\mathbf{G}^{-1} + \mathbf{F}^T\mathbf{E}^{-1}\mathbf{F}\right) \tag{30}$$

where $\mathbf{E}$ is easy to work with because it is diagonal and $\mathbf{L}$ and $\mathbf{G}$ are easy to work with because they are small: n_F by n_F. Of course, all the other covariance matrices in either of the formats discussed can be treated similarly.

4 Training

Training of this speaker detection system consists of using large numbers of detection trials $(\mathbf{x}, \mathbf{y})$ to assign values to the parameters $\mathbf{C}$ and $\mathbf{D}$. As always, we may consider both generative and discriminative ways to do this training.

4.1 Generative training

At a first glance, generative training would consist of finding separate maximum-likelihood (ML) or MAP solutions under each of the hypotheses H_1 and H_2. But looking at equations 9 and 10, we see that these maximizations would not be independent. We also see that obtaining the sub-covariances $2\mathbf{C}$ and $2\mathbf{C} + 4\mathbf{D}$ under H_1 can help us separate $\mathbf{C}$ and $\mathbf{D}$ but training under H_2 cannot because the sub-covariances are identical.

[6] $\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}$

This suggests that ML or MAP training under H_1 is sufficient to determine the whole system.

In what follows, we shall consider only constrained ML solutions as opposed to MAP solutions for $\mathbf{C}$ and $\mathbf{D}$. In particular, we shall employ the factor analysis constraints of equations 15 and 16.

4.1.1 Simple solution

First, to keep things simple, we constrain $\mathbf{D}$ to be diagonal, (i.e. $n_J = 0$). This is motivated by the work of Patrick Kenny which suggests that this is a good (although perhaps not optimal) assumption. (As a further explanation for this, we can note that it seems to be a good idea not to 'remember' speaker characteristics with too many parameters, because in new data the speakers will all be different. But it is a good idea to reserve more parameters to 'remember' how intra-speaker noise (including channel effects) is structured, because presumably this structure will be repeated in future.)

It would probably be a good idea to formally derive a maximum likelihood-likelihood solution with respect to the parameters in the model. But, in the mean time, here is a quick-and-dirty calculus-shy solution:

Input: A large set of same-speaker (H_1) trials: $(\mathbf{x}_i, \mathbf{y}_i), i = 1, 2, \ldots, N$.

Step 1: Let $\mu = \frac{1}{2N} \sum_{i-1}^{N} \mathbf{x}_i + \mathbf{y}_i$.

Step 2: Let element d_{jj} of diagonal matrix $\mathbf{D}$ be:

$$d_{jj} = \frac{1}{N} \sum_{i=1}^{N} (x_{ji} - \mu_j)(y_{ji} - \mu_j) \tag{31}$$

where μ_j, x_{ji} and y_{ji} are the respective components of vectors μ, $\mathbf{x}_i$ and $\mathbf{y}_i$.

Step 3: Choose $n_F << d$ and perform a factor analysis so that:

$$\mathbf{C} = \lambda \mathbf{I} + \mathbf{F}\mathbf{G}\mathbf{F}^T \approx \Gamma = \frac{1}{2N} \sum_{i=1}^{N} (\mathbf{x} - \mathbf{y})(\mathbf{x} - \mathbf{y})^T \tag{32}$$

Where the columns of $\mathbf{F}$ are normalized. We could do the factor analysis, for example, by doing a PCA analysis[7] of Γ and then choosing λ so that $\text{trace}(\mathbf{C}) = \text{trace}(\Gamma)$. Of course, $\lambda > 0$ makes $\mathbf{C}$ invertible.

4.2 Discriminative training

Possibly, the above quick-and-dirty solution is not an optimal solution for the generative case. Most probably the optimal generative solution does not have a closed-form solution, requiring some sort of iterative optimization. Possibly the assumptions of our generative model are far from good. All of these are

[7] (For example, the MATLAB function EIGS can do this kind of PCA analysis, to find a few eigen-values and -vectors of a large matrix.)

reasons to consider also iterative discriminative training solutions. In this case, we maximize the following objective function:

$$\begin{aligned}\mathcal{O}(\mathbf{C},\mathbf{D}) =& \frac{p}{\|\mathcal{T}_1\|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}_1}\log\sigma(\mathbf{x},\mathbf{y}) \\ &+\frac{1-p}{\|\mathcal{T}_2\|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}_2}\log(1-\sigma(\mathbf{x},\mathbf{y})) \\ \sigma(\mathbf{x},\mathbf{y}) =& \sigma\big(\mathcal{S}(\mathbf{x},\mathbf{y})+\text{logit}(1-p)\big)\end{aligned} \tag{33}$$

where the sigmoid or *logistic* function $\sigma(\cdot)$ is:

$$\sigma(x) = \text{logit}^{-1}(x) = \frac{1}{1+\mathrm{e}^{-x}} = -\frac{\partial\log\sigma(-x)}{\partial x} \tag{34}$$

$$\sigma(-x) = 1-\sigma(x) = \frac{1}{1+\mathrm{e}^{x}} = \frac{\partial\log\sigma(x)}{\partial x} \tag{35}$$

where $\mathcal{T}_1$ is a training set of same-speaker trials and $\mathcal{T}_2$ is a training set of different-speaker trials.

A good approach to perform the iterative optimization, is with a conjugate-gradient algorithm. This requires analytical solution for the gradient of $\mathcal{O}(\mathbf{C},\mathbf{D})$ with respect to every parameter involved in the optimization. To start this agenda, we can express the partial derivative of the objective w.r.t. a parameter, say α as:

$$\begin{aligned}\frac{\partial\mathcal{O}}{\partial\alpha} =& \frac{p}{\|\mathcal{T}_1\|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}_1}(1-\sigma(\mathbf{x},\mathbf{y}))\frac{\partial\mathcal{S}(\mathbf{x},\mathbf{y})}{\partial\alpha} \\ &-\frac{1-p}{\|\mathcal{T}_2\|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}_2}\sigma(\mathbf{x},\mathbf{y})\frac{\partial\mathcal{S}(\mathbf{x},\mathbf{y})}{\partial\alpha}\end{aligned} \tag{36}$$

Note that this is a weighted sum of the score derivatives $\frac{\partial\mathcal{S}(\mathbf{x},\mathbf{y})}{\partial\alpha}$, where the weighting depends on the parameters that we are optimizing. The second derivative will also come in handy:

$$\begin{aligned}\frac{\partial^2\mathcal{O}}{\partial\alpha^2} =& -\frac{p}{\|\mathcal{T}_1\|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}_1}(1-\sigma(\mathbf{x},\mathbf{y}))\sigma(\mathbf{x},\mathbf{y})\left(\frac{\partial\mathcal{S}(\mathbf{x},\mathbf{y})}{\partial\alpha}\right)^2 \\ &\qquad+(1-\sigma(\mathbf{x},\mathbf{y}))\frac{\partial^2\mathcal{S}(\mathbf{x},\mathbf{y})}{\partial\alpha^2} \\ &-\frac{1-p}{\|\mathcal{T}_2\|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}_2}\sigma(\mathbf{x},\mathbf{y})(1-\sigma(\mathbf{x},\mathbf{y}))\left(\frac{\partial\mathcal{S}(\mathbf{x},\mathbf{y})}{\partial\alpha}\right)^2 \\ &\qquad+\sigma(\mathbf{x},\mathbf{y})\frac{\partial^2\mathcal{S}(\mathbf{x},\mathbf{y})}{\partial\alpha^2}\end{aligned} \tag{37}$$

To get the score derivatives, we assume an even simpler structure for our covariances matrices, which can be discriminatively 'tuned'.

4.2.1 Simple solution

We start from a generatively trained baseline. We assume the generative training has given us a model of the form:

$$\mathbf{C} = \mathbf{E} + \mathbf{F}\mathbf{G}\mathbf{F}^T \tag{38}$$

$$\mathbf{D} = \mathbf{H} + \mathbf{J}\mathbf{K}\mathbf{J}^T \tag{39}$$

We can rewrite this for convenience as

$$\mathbf{C} = \sum_{i=1}^{n_F+1} \mathrm{e}^{\gamma_i}\Gamma_i \tag{40}$$

$$\mathbf{D} = \sum_{i=n_F+2}^{n_F+n_J+2} \mathrm{e}^{\gamma_i}\Gamma_i \tag{41}$$

where

$$(\Gamma_1, \Gamma_2, \ldots, \Gamma_{n-1}) = (\mathbf{E}, \mathbf{f}_1\mathbf{f}_1^T, \mathbf{f}_2\mathbf{f}_2^T, \ldots, \mathbf{H}, \mathbf{j}_1\mathbf{j}_1^T, \ldots, \mathbf{j}_{n_J}\mathbf{j}_{n_J}^T) \tag{42}$$

and where[8] $n = n_F + n_J + 3$ and where $\mathbf{f}_i$ is the ith column of $\mathbf{F}$ and $\mathbf{j}_i$ is the ith column of $\mathbf{J}$. It is easy to choose suitable values for the weight-vector $\vec{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_n]^T$ to make the equalities (40) and (41) true, but the object of the exercise is now to tune these scalar parameters with our discriminative optimization to assume *different* values to those given by the generative solution. Note that exponentiation keeps our weights positive, so that the parameter vector $\vec{\gamma} \in \mathbb{R}^n$ can be *unconstrained*.

To form our score, we use the Gaussian supervector models given by (9) and (10):

$$\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{x} - \mathbf{y} \\ \mathbf{x} + \mathbf{y} - 2\mu \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} Q_1 & \mathbf{0} \\ \mathbf{0} & Q_2 \end{bmatrix}), \text{ given } H_1 \tag{43}$$

and

$$\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{x} - \mathbf{y} \\ \mathbf{x} + \mathbf{y} - 2\mu \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} Q_3 & \mathbf{0} \\ \mathbf{0} & Q_3 \end{bmatrix}), \text{ given } H_2 \tag{44}$$

where

$$Q_1 = \quad 2\mathbf{C} \quad = \sum_{j=1}^{n-1} a_{1j}\mathrm{e}^{\gamma_j}\Gamma_j \tag{45}$$

$$Q_2 = 2\mathbf{C} + 4\mathbf{D} = \sum_{j=1}^{n-1} a_{2j}\mathrm{e}^{\gamma_j}\Gamma_j \tag{46}$$

$$Q_3 = 2\mathbf{C} + 2\mathbf{D} = \sum_{j=1}^{n-1} a_{3j}\mathrm{e}^{\gamma_j}\Gamma_j \tag{47}$$

where the $a_{ij} \geq 0$ are chosen to make these equalities valid. Now we can write:

$$\mathcal{S}(\mathbf{x}, \mathbf{y}) = \gamma_n - \mathbf{v}^T(\mathbf{Q}_1^{-1} - \mathbf{Q}_3^{-1})\mathbf{v} + \mathbf{w}^T(\mathbf{Q}_3^{-1} - \mathbf{Q}_2^{-1})\mathbf{w} \tag{48}$$

[8] notice we have added an extra parameter γ_n, to be used later

and where we have chosen to replace the determinant-derived constant simply by the free parameter γ_n.

An interesting note is that if both $\mathbf{C}$ and $\mathbf{D}$ are of full rank, then both of the matrices forming the quadratic terms are positive definite[9]

$$\begin{aligned}
\mathbf{N} = \mathbf{Q}_1^{-1} - \mathbf{Q}_3^{-1} &= \mathbf{Q}_3^{-1}\mathbf{D}\mathbf{C}^{-1} = \mathbf{C}^{-1}\mathbf{D}\mathbf{Q}_3^{-1} && (49)\\
&= \frac{1}{2}\mathbf{C}^{-1}(\mathbf{C}^{-1} + \mathbf{D}^{-1})^{-1}\mathbf{C}^{-1} && (50)\\
&= \frac{1}{2}(\mathbf{C} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C})^{-1} && (51)\\
\mathbf{M} = \mathbf{Q}_3^{-1} - \mathbf{Q}_2^{-1} &= 2\mathbf{Q}_3^{-1}\mathbf{D}\mathbf{Q}_2^{-1} = 2\mathbf{Q}_2^{-1}\mathbf{D}\mathbf{Q}_3^{-1} && (52)\\
&= \frac{1}{2}(\mathbf{C}\mathbf{D}^{-1}\mathbf{C} + 3\mathbf{C} + 2\mathbf{D})^{-1} && (53)
\end{aligned}$$

This means the quadratic terms are non-negative. This confirms the behaviour observed above:

- When the magnitude of the difference vector increases, the score decreases.
- When the sum vector moves away from 2μ, then the score increases.

This allows us to write:

$$\mathcal{S}(x, y) = \gamma_n + \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}^T \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & -\mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \tag{54}$$

Or we can re-parameterize to $\mathbf{z} = \begin{bmatrix} \mathbf{x} - \mu \\ \mathbf{y} - \mu \end{bmatrix}$. Then we get:

$$\mathcal{S}(x, y) = \gamma_n + \mathbf{z}^T \begin{bmatrix} \mathbf{M} - \mathbf{N} & \mathbf{M} + \mathbf{N} \\ \mathbf{M} + \mathbf{N} & \mathbf{M} - \mathbf{N} \end{bmatrix} \mathbf{z} \tag{55}$$

[9]The matrix $\mathbf{C}\mathbf{D}^{-1}\mathbf{C}$ is positive definite if $\mathbf{D}$ is positive definite and if $\mathbf{C}$ is of full rank. (A matrix $\mathbf{M}$ is positive definite if and only if, for any vector $\mathbf{v} \neq \mathbf{0}$ the quadratic form $\mathbf{v}^T\mathbf{M}\mathbf{v} > 0$.