S蛋HCE KHER TECHNICKS APPLIED TO SPEEK L'TAR LINEAR PREDICTION

Jordi Solé i Casals (1), Enric Monte i Moreno (2), Christian Jutten (3), Anisse Taleb (3,4)

(1)University of Vic, Sagrada Família 7, 08500, Vic (Catalunya, Spain)
(2) Polytechnical University of Catalonia, Jordi Girona, 1-3, 08034 Barcelona (Catalunya, Spain)
(3)INPG-LIS, 46 Av. Félix Viallet, 38031, Grenoble Cedex, (France)
(4) ATRI, Curtin University of Technology, GPO BOX U 1987, Perth, Western Australia 6845 (Australia)

ABSTRACT
The prediction filters are well known models for speech signal, in communications, control and many others areas. The classical method for deriving linear prediction coding (LPC) filters is often based on the minimization of a mean square error (MSE). Consequently, second order statistics are only required, but the estimation is only optimal if the residue is independent and identically distributed (iid) Gaussian. However, if the residue is not Gaussian, the estimation is no longer optimal. If one knows the theoretical statistics, it is possible to improve the estimation by using optimal (odd value higher order) statistics. Otherwise, i.e. if the statistics is not known, one can wonder how to implementing a quasi-optimal estimation. In this paper, we derive the ML estimate of the prediction filter. Relationships with robust estimation of auto-regressive (AR) processes, with blind deconvolution and with source separation based on mutual information minimization are shown. The algorithm, based on the minimization of a high-order statistics criterion, uses on-line estimation of the residue statistics. Improvements in the experimental results with speech signals emphasize on the interest of this approach.

1. CLASSICAL LPC
The classical LPC methods are based on the minimization of a mean square error, defined as the difference between the input signal \( x(k) \) and the predicted signal \( y(k) = [w(z)]x(k-1) \), where \( w(z) \) is a \( L \)-th order causal finite impulse response filter, i.e. a filter whose entries \( w_i = 0 \) for \( i \notin \{0, 1, \ldots, L-1\} \). The block diagram of a linear predictor is shown in Fig. 1. Denoting \( E[x(k)x(k-l)] = R_{xx}(l) \), the cost function reduces to:

\[
J = E[y^2(k)] = R_{xx}(0) - 2 \sum_{n=0}^{L-1} w_n R_{xx}(n+1) + \sum_{m=0}^{L-1} \sum_{n=0}^{L-1} w_n w_m R_{xx}(m-n)
\]

(1)

This estimation can be viewed as a maximum likelihood (ML) estimate in the special case of independent and identically distributed (iid) Gaussian error: in fact, first consider only the prediction at time \( k \). Taking into account the relation \( y(k) = x(k) - e(k) \), and denoting \( p_E(.) \) the probability density function (pdf) of the residue \( e(k) \), the log-ML estimation is:

\[
\text{ArgMax}_w \left[ \sum_{i=0}^{N-1} \ln(p_E(e(k+i))) \right]
\]

(2)

Assuming that the error \( e(k) \) is a Gaussian zero mean random variable, the Maximum Likelihood estimation is:

\[
\text{ArgMin}_w \left[ \sum_{i=0}^{N-1} (e(k+i))^2 \right]
\]

(3)

As it is well known, in the Gaussian case, asymptotically, the ML is nothing but the minimum mean square error (MMSE) estimate.

2. SCORE FUNCTION METHOD
Unfortunately, if the error is not Gaussian, the MMSE estimate is no longer equal to the ML estimate. In fact, from (2), one can compute the ML equation by deriving the equation with respect to the entries \( w_j \):

\[
\sum_{i=0}^{N-1} \frac{\partial}{\partial w_j} \ln(p_E(e(k+i))) = \sum_{i=0}^{N-1} \frac{p'_E((e(k+i)))}{p_E} \frac{\partial}{\partial w_j} (e(k+i)) = - \sum_{i=0}^{N-1} \Psi_{E}(e(k+i))x(k+i - j - 1)
\]

(4)

where \( \Psi_{E}(.) \) denotes the derivative of \( \ln p_E(.) \), the so-called score function.

Consequently, asymptotically, for any error distribution, the ML estimate of \( w_j \), \( j = 0, K, L-1 \), is equivalent to the equation set.
\[ E[\psi_k(e(k)) x(k-j-1)] = 0, \quad j = 0, K, L-1. \quad (5) \]

Basically, the score function is a nonlinear function, except in the Gaussian case. In this case, \( \psi_k(e) = -e \). Then, equation (5) prove that the optimal ML estimate involves higher (than 2) order statistics, except in the Gaussian case.

### 3. LPC, DECONVOLUTION AND SOURCE SEPARATION

LPC is based on the assumption that the signal \( x(k) \) is linear, \textit{i.e.} the linear auto-regressive (AR) filtering of an iid sequence \( n(k) \):

\[ h(k)\ast x(k) = n(k) \quad (6) \]

where \( H(z) = 1 + h_1 z^{-1} + \cdots + h_K z^{-K} = Z\{h(k)\} \).

It is clear that the linear prediction of \( x(k) \) is nothing but the deconvolution of \( x(k) \), with a filter with the constrained structure of Fig. 1, and that the optimal solution should verify:

\[ 1 - z^{-1} w(z) = H(z) \quad (7) \]

as shown in Fig. 2. If \( n(k) \) is Gaussian, classical LPC and second order deconvolution are equivalent, because the optimal filter must provide Gaussian residue \( e(k) \). On the contrary, if \( n(k) \) is not Gaussian, the error residue must not be Gaussian. Equations (5) show that the optimal deconvolution as well as the optimal LPC must provide iid residue \( e(k) \) with the same pdf than \( n(k) \). Both problems involve higher order statistics, and the knowledge of the pdf or of the score function is required for choosing the optimal (high order) statistics at the ML sense.

Recently, Taleb \textit{et al.} \cite{2, 3} addressed the problem of Wiener system blind inversion using source separation methods. Of course, this approach can also be used for blind linear deconvolution. This technique will be used in our situation to obtain the coefficients of the quasi-optimal LPC filter.

![Figure 2:](image)

\textbf{Figure 2:} On the top, the convolution system \( F(z) \) and the deconvolution system \( G(z) \). The recovered signal \( y(k) \) is \( x(k) \) iff \( G(z)F(z)=1 \). On the down, the classical LPC viewed as a deconvolution problem: \( 1/H(z) \) is the unknown filter and the dashed bloc its inverse, to be estimated, in order to recover \( e(k)=n(k) \).

### 4. EXPERIMENTS

We compare the LPC gain obtained with LPC with second-order statistics (the classical method, optimal for Gaussian error pdf) using the Matlab LPC function, and the quasi-optimal algorithm (our algorithm).

As input speech signal we use the Spanish sentence “el golpe de timón fue sobrecogedor". The signal, sampled at 8KHz, is processed frame by frame, each frame of 64 samples (8 msec).

The performance criterion used to evaluate the results is the prediction gain, defined as:
\[ G_p (dB) = 10 \log \left( \frac{E[x^2(n)]}{E[y^2(n)]} \right) \]  

(8)

In almost every processed frame, we obtain higher prediction gain with our quasi-optimal algorithm than with the classic method. In the table 1 we can see the results for 30 consecutive frames, using the two methods. The mean gain for this experiment is 4.16404 %. This gain is calculated as follows:

\[ G(\%) = 100 \left( \frac{\sum_{i=1}^{30} 10 \log (G_p_{\text{PSI}}(i)) - \sum_{i=1}^{30} 10 \log (G_p_{\text{CLASSIC}}(i))}{\sum_{i=1}^{30} 10 \log (G_p_{\text{CLASSIC}}(i))} \right) \]

(9)

<table>
<thead>
<tr>
<th>Frame number</th>
<th>Classic method (dB)</th>
<th>Quasi-optimal method (dB)</th>
<th>( G_p ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>270</td>
<td>10.0848</td>
<td>10.1247</td>
<td>0.3961</td>
</tr>
<tr>
<td>271</td>
<td>11.9504</td>
<td>12.3313</td>
<td>3.1871</td>
</tr>
<tr>
<td>272</td>
<td>9.8499</td>
<td>10.3773</td>
<td>5.3546</td>
</tr>
<tr>
<td>273</td>
<td>12.1980</td>
<td>12.2023</td>
<td>0.0356</td>
</tr>
<tr>
<td>274</td>
<td>8.6503</td>
<td>9.4586</td>
<td>9.3446</td>
</tr>
<tr>
<td>275</td>
<td>10.0545</td>
<td>10.1247</td>
<td>0.6989</td>
</tr>
<tr>
<td>276</td>
<td>10.9533</td>
<td>11.1976</td>
<td>11.3606</td>
</tr>
<tr>
<td>277</td>
<td>11.1610</td>
<td>11.3146</td>
<td>1.3763</td>
</tr>
<tr>
<td>278</td>
<td>13.4018</td>
<td>14.1954</td>
<td>5.9214</td>
</tr>
<tr>
<td>279</td>
<td>14.2846</td>
<td>14.5468</td>
<td>1.8356</td>
</tr>
<tr>
<td>280</td>
<td>15.5271</td>
<td>15.5067</td>
<td>-0.1316</td>
</tr>
<tr>
<td>281</td>
<td>13.3579</td>
<td>15.8124</td>
<td>18.3746</td>
</tr>
<tr>
<td>282</td>
<td>14.5607</td>
<td>14.7157</td>
<td>1.0651</td>
</tr>
<tr>
<td>283</td>
<td>15.4494</td>
<td>16.0692</td>
<td>4.0118</td>
</tr>
<tr>
<td>284</td>
<td>15.2417</td>
<td>15.8007</td>
<td>3.6678</td>
</tr>
<tr>
<td>285</td>
<td>15.2237</td>
<td>15.8949</td>
<td>4.4092</td>
</tr>
<tr>
<td>286</td>
<td>13.8686</td>
<td>13.8917</td>
<td>0.1797</td>
</tr>
<tr>
<td>287</td>
<td>13.6749</td>
<td>13.7366</td>
<td>0.4509</td>
</tr>
<tr>
<td>288</td>
<td>11.8004</td>
<td>11.9904</td>
<td>1.6101</td>
</tr>
<tr>
<td>289</td>
<td>11.0287</td>
<td>11.0198</td>
<td>-0.0807</td>
</tr>
<tr>
<td>290</td>
<td>8.6047</td>
<td>11.5144</td>
<td>33.8161</td>
</tr>
<tr>
<td>291</td>
<td>7.8922</td>
<td>7.8855</td>
<td>-0.0856</td>
</tr>
<tr>
<td>292</td>
<td>8.0864</td>
<td>8.2752</td>
<td>2.3347</td>
</tr>
<tr>
<td>293</td>
<td>8.0439</td>
<td>8.0406</td>
<td>-0.0410</td>
</tr>
<tr>
<td>294</td>
<td>5.1554</td>
<td>5.1585</td>
<td>0.0588</td>
</tr>
<tr>
<td>295</td>
<td>7.3950</td>
<td>7.4127</td>
<td>0.2389</td>
</tr>
<tr>
<td>296</td>
<td>5.5061</td>
<td>6.2517</td>
<td>13.5424</td>
</tr>
<tr>
<td>297</td>
<td>3.1956</td>
<td>3.1929</td>
<td>-0.0826</td>
</tr>
<tr>
<td>298</td>
<td>3.9758</td>
<td>4.0476</td>
<td>1.8061</td>
</tr>
<tr>
<td>299</td>
<td>4.7339</td>
<td>4.7392</td>
<td>0.1119</td>
</tr>
</tbody>
</table>

Table1: Prediction gains (dB) for 30 frames, using the two methods. Quasi-optimal method, based on score function, gives better results than classic method in 25 of 30 frames. The mean gain, averaged over this 30 frames, is more than 4%.

In figure 3 we present the input signal corresponding to this 30 frames and prediction gain \( G_p \) obtained with the two methods. Line with starts corresponds to classic method and line with diamonds to quasi-optimal method. We observe clearly that quasi-optimal method give us higher gain than classical method.

![Figure 3](image3.png)  

Figure 3: 30 frames of the input signal, with the prediction gain obtained for classical method (line with stars) and quasi-optimal method (line with diamonds)

In figure 4 we present the histogram of this 30 frames considered in the example. We observe that this distribution corresponds to a non-Gaussian and non-symmetrical signal. Therefore, classical method give us poor results because the input signal is in fact non-Gaussian and skewed. Instead, quasi-optimal method works well because of the distribution estimation of the signal.

![Figure 4](image4.png)  

Figure 4: Histogram, compared with a Gaussian distribution, for the considered 30 frames of the input signal. We observe that the signal distribution is not centered and non-Gaussian.

In figure 4 we present the histogram of this 30 frames considered in the example. We observe that this distribution corresponds to a non-Gaussian and non-symmetrical signal. Therefore, classical method give us poor results because the input signal is in fact non-Gaussian and skewed. Instead, quasi-optimal method works well because of the distribution estimation of the signal.

To confirm this reasoning, we present the results obtained with a Gaussian part of the signal. This result corresponds to 30 consecutive frames of a silence part (low energy). In figure 5 we can see the signal and the prediction gain obtained with the two methods. Here, the two methods are almost equivalent. The
mean gain for this 30 frames, calculated with equation (9), is 1.5974 %, very poor compared with the previous example. This is due because the considered signal is more Gaussian than the previous signal. In figure 6 we can see the histogram of this signal, compared with a Gaussian signal. It is clear than in this case the distributions are very similar.

5. SUMMARY

Inspired by source separation techniques, we have presented a new algorithm for performing speech linear prediction, which gives better results than the classical LPC methods, especially for small samples.

The method is based on a criterion which requires the knowledge of error pdf, or more precisely of the score functions. Implicitly, this criterion involves higher order statistics, which can be chosen optimally with a good estimation of the score function, e.g., computed from kernel estimators of the error pdf.

Real speech signal experiments show that this method is always better than Matlab LPC function, simply based second order statistics.

Currently, practical and theoretical issues are both addressed: (i) how improve the estimation of the score function for enhancing the results and obtaining faster and better algorithms, (ii) computing the performance of the method according to the estimation error on the score function.

6. ACKNOWLEDGMENTS

This work has been in part supported by the Direcció General de Recerca de la Generalitat de Catalunya and by the CICYT Spanish research project TIC98-0683.

7. REFERENCES