Wiener kernels
Joshua I Gold

I. Expansions (see Rieke et al, A.4)

Most of us are familiar with the *Taylor expansion* of a function, $y = f(x)$:

$$y = f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2 + \ldots$$

$$= f_0 + f_1(x - x_0) + f_2(x - x_0)^2 + \ldots$$

There is an analogous expansion for a functional, which operates not on a single value but rather a sequence (like a function of time): $y(t) = F(x(t))$. This is called a Volterra expansion, which can be thought of as a "Taylor series with memory" (Franz & Scholkopf, 2003):

$$y(t) = h_0 + \int d\tau_1 h_1(\tau_1) x(t - \tau_1) + \int d\tau_1 \int d\tau_2 h_2(\tau_1, \tau_2) x(t - \tau_1) x(t - \tau_2) + \ldots$$

The functions h_i are called the Volterra "kernels". The proper choice of kernels will provide a complete description of any transformation $x(t) \rightarrow y(t)$ (Volterra, 1930). Note that the Volterra kernels for a given output are not unique: there are many *asymmetric* kernels. However, there is only one *symmetric* kernel; that is, a kernel $h_n(\ldots,\tau_i, \ldots, \tau_j, \ldots) = h_n(\ldots,\tau_j, \ldots, \tau_i, \ldots)$. Note also that in the original formulation each term $H_n[x(t)]$ is scaled by $1/n!$, which simplifies some calculations but is omitted here.

In principle, the integrals in the above equation range from $-\infty$ to ∞. However, in practice we are typically interested in "causality" – expressing the value of $y(t)$ in terms of only those values of $x(t)$ that have already occurred. In this case, y(t) is written in terms of x(t–τ), where τ is always positive:

$$y(t) = h_0 + \int_0^\infty d\tau_1 h_1(\tau_1) x(t - \tau_1) + \int_0^\infty d\tau_1 \int_0^\infty d\tau_2 h_2(\tau_1, \tau_2) x(t - \tau_1) x(t - \tau_2) + \ldots$$

One difficulty in computing this expansion is that the terms ("operators") are not orthogonal; that is, to compute one you typically need to know others. Wiener solved this problem by defining the process in terms of a given distribution on the input signals. In this case, it is possible to "orthogonalize" the operators with respect to the given input distribution. The result is operators that are sums of different order Volterra operators. In the special case in which the input is a white Gaussian distribution with zero mean, one gets the *Wiener expansion* (Wiener, 1958):

$$y(t) = \sum_{n=0}^{\infty} G_n[x(t)] = G_0 + G_1[x(t)] + G_2[x(t)] + \ldots$$

where

$$
\begin{aligned}
&G_0 = g_0\\
&G_1[x(t)] = \int_0^\infty d\tau_1 g_1(\tau_1)x(t-\tau_1)\\
&G_2[x(t)] = \int_0^\infty d\tau_1 \int_0^\infty d\tau_2 g_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2) - S_x \int_0^\infty d\tau g_2(\tau,\tau)
\end{aligned}
$$

etc.

where S_x is the power spectrum of x.

Here, the g_n's are the Wiener kernels. If the input is white noise with zero mean and variance A, it turns out that correlating the input with the output isolates the first kernel. Similarly, correlating higher powers of the (white noise) input with the output isolates higher kernels:

$$
\begin{aligned}
&g_0 = \langle y(t) \rangle\\
&g_1(\tau) = \frac{1}{S_x}\langle y(t)x(t-\tau)\rangle\\
&g_2(\tau_1,\tau_2) = \frac{1}{S_x^2}\langle y(t)x(t-\tau_1)x(t-\tau_2)\rangle
\end{aligned}
$$

etc.

II. Optimal linear (Wiener) filters for arbitrary input

The first Wiener kernel, described above, acts as a linear filter: the functional $G_1[x(t)]$ convolves the impulse response $g_1(\tau)$ with the input $x(t)$. For white noise input, the impulse response $g_1(\tau)$ is defined as the expected value of the cross-correlation between the input $x(t)$ and the output $y(t)$. One may wonder, why is this a good choice? The answer is that this minimizes the mean-squared error (MSE) between the predicted and actual output if the input is zero-mean white noise. In general, this is a good thing…

The following is from Gauss, translated by Stewart (1995), presented in Kailath et al ("State-space estimation theory: Wiener and Kalman Filtering"):

"It is by no means self-evident how much loss should be assigned to a given observation error. On the contrary, the matter depends in some part on our own judgment. Clearly, we cannot set the loss equal to the error itself; for if positive errors were taken as losses, negative errors would have to represent gains. The size of the loss is better represented by a function that is naturally positive. Since the number of such functions is infinite, it would seem that we should choose the simplest function having this property. That function is unarguably the square, and the principle proposed above results from this adoption.

"Now if someone should object that this convention has been chosen arbitrarily with no compelling necessity, I will gladly agree. In fact, the problem has a certain intrinsic vagueness about it that can only be resolved by a more or less arbitrary principle.

"Laplace has also considered the problem in a similar manner, but he adopted the absolute value of the error as the measure of this loss. Now if I am not mistaken, this convention is no less arbitrary than mine. Should an error of double size be considered as tolerable as a single error twice repeated or worse? Is it better to assign only twice as much in influence to a double error or more? The answers are not self-evident, and the problem cannot be resolved by mathematical proofs, but only by an arbitrary decision.

"Moreover, it cannot be denied that Laplace's convention violates continuity and hence resists analytic treatment, while the results that my convention leads to are distinguished by their wonderful simplicity and generality."

In the following, we derive a more general form of the optimum linear filter; that is, a filter (we'll call the impulse response $w(t)$, to emphasize that these can be thought of as "weights" on the input) that when applied to an input $x(t)$ generates a prediction or estimate $\hat{y}(t)$ that minimizes MSE between the estimated output $\hat{y}(t)$ and the actual output $y(t)$. The primary constraint is that this will work only if x(t) is wide-sense stationary; that is, the mean value and autocorrelation don't depend on the interval sampled:

$[x(t)] = k$ for all t,
$R_{xx}(t_1, t_2) = R_{xx}(t_1\text{-}t_2)$ for all t_1, t_2

Let's consider discrete-time data, so that x_k is the k^{th} input sample and y_k is the k^{th} output sample. We will assume that the Wiener filter is an FIR (finite impulse response) filter with N coefficients (here we assume that inputs x_{k-N} through x_{k-1} are used to estimate the output y_k). The estimated output $\hat{y}_k$ is defined as:

$$\hat{y}_k = \sum_{i=1}^{N} w_i x_{k-i}$$

The error is thus:

$$e_k = y_k - \hat{y}_k = y_k - \sum_{i=1}^{N} w_i x_{k-i}$$

Because the input, output, and filter values are discrete, we can use matrix notation (the following is from DSP Ch. 7):

$$\mathbf{X}_k = \begin{bmatrix} x_{k-1} \\ x_{k-2} \\ \vdots \\ x_{k-N} \end{bmatrix} \qquad \mathbf{W}_k = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$$

In this form,

$$e_k = y_k - \mathbf{W}^T\mathbf{X}_k = y_k - \mathbf{X}_k^T\mathbf{W}$$

The (instantaneous) squared error of the signal is just the square of this:

$$e_k^2 = y_k^2 - 2\mathbf{W}^T(y_k\mathbf{X}_k) + \mathbf{W}^T\mathbf{X}_k\mathbf{X}_k^T\mathbf{W}$$

The mean square error is just the expectation of this, which is:

$$\xi = E\left[e_k^2\right] = E\left[y_k^2\right] - 2\mathbf{W}^T E\left[y_k \mathbf{X}_k\right] + \mathbf{W}^T E\left[\mathbf{X}_k \mathbf{X}_k^T\right]\mathbf{W}$$

Here we can use the autocorrelation matrix $\boldsymbol{R}_{xx}$, which is defined as $E[\boldsymbol{X}_k \boldsymbol{X}_k^T]$ (where $r_{xx}[m]$ is the autocorrelation at lag m):

$$R_{\mathbf{XX}} = E\left[\mathbf{X}_k \mathbf{X}_k^T\right] = E\begin{bmatrix} x_{k-1}x_{k-1} & x_{k-1}x_{k-2} & \cdots & x_{k-1}x_{k-N} \\ x_{k-2}x_{k-1} & x_{k-2}x_{k-2} & & \\ \vdots & & \ddots & \\ x_{k-N}x_{k-1} & & & x_{k-N}x_{k-N} \end{bmatrix} = \begin{bmatrix} r_{xx}[0] & r_{xx}[1] & \cdots & r_{xx}[N-1] \\ r_{xx}[1] & r_{xx}[0] & & \\ \vdots & & \ddots & r_{xx}[1] \\ r_{xx}[N-1] & & r_{xx}[1] & r_{xx}[0] \end{bmatrix}$$

We can also use the cross-correlation matrix R_{yX} (where r_{yx}[m] is the (cross) correlation at lag m):

$$R_{y\mathbf{X}} = E\left[y_k \mathbf{X}_k^T\right] = E\begin{bmatrix} y_k x_{k-1} \\ y_k x_{k-2} \\ \vdots \\ y_k x_{k-N} \end{bmatrix} = \begin{bmatrix} r_{yx}[0] \\ r_{yx}[1] \\ \vdots \\ r_{yx}[N-1] \end{bmatrix}$$

Where

So that we have:

$$\xi = E\left[y_k^2\right] - 2\mathbf{W}^T \mathbf{R}_{y\mathbf{X}} + \mathbf{W}^T \mathbf{R}_{\mathbf{XX}} \mathbf{W}$$

To minimize MSE, we take the gradient with respect to the weight vector:

$$\nabla = E\left\{\frac{\partial e_k^2}{\partial \mathbf{W}}\right\} = E\left\{2e_k \frac{\partial e_k}{\partial \mathbf{W}}\right\} = 2E\left\{\left(y_k - \mathbf{X}_k^T \mathbf{W}\right)\frac{\partial}{\partial \mathbf{W}}\left(y_k - \mathbf{X}_k^T \mathbf{W}\right)\right\} = 2E\left\{\left(y_k - \mathbf{X}_k^T \mathbf{W}\right)\left(-\mathbf{X}_k^T\right)\right\}$$

And set to 0:

$$E\left\{\left(y_k - \mathbf{X}_k^T \mathbf{W}_{opt}\right)\mathbf{X}_k\right\} = 0$$

This is the so-called *orthogonality condition*: the error (the term in parentheses, actual minus predicted) is orthogonal to the input X. This has an intuitive appeal: if the error were correlated with the input, then some part of it would be predictable and one could use X more efficiently to make the prediction of Y. Continuing:

$$-E\left\{\mathbf{X}_k y_k\right\} + E\left\{\mathbf{X}_k \mathbf{X}_k^T\right\}\mathbf{W}_{opt} = 0$$
$$-\mathbf{R}_{y\mathbf{X}} + \mathbf{R}_{\mathbf{XX}} \mathbf{W}_{opt} = 0$$
$$\mathbf{W}_{opt} = \mathbf{R}_{\mathbf{XX}}{}^{-1} \mathbf{R}_{y\mathbf{X}}$$

This is the *Wiener-Hopf equation* in matrix form.

In MATLAB, this is very easy to compute. For input vector *xs* and output vector *ys*, computing the Wiener kernel for *N* coefficients is:

```
xx_corr = xcorr(xs,xs,N);
yx_corr = xcorr(ys,xs,N);
W = toeplitz(xx_corr(N+1:-1:1))\yx_corr(N+1:-1:1);
```