Table 3.2 Basic adaptive learning algorithms for principal component analysis (PCA).

No.	Learning Algorithm	Notes, References
1.	$\tilde{\mathbf{v}}_1(k+1) = \mathbf{v}_1(k) + \eta_1 \mathbf{v}_1^T(k)\mathbf{x}(k)\mathbf{x}(k)$ $\mathbf{v}_1(k+1) = \tilde{\mathbf{v}}_1(k+1)/\|\tilde{\mathbf{v}}_1(k+1)\|^2$	Amari (1978) [20]
2.	$\Delta\mathbf{v}_1 = \eta_1 \left[\mathbf{R_{xx}}\mathbf{v}_1 - \mathbf{v}_1\mathbf{v}_1^T\mathbf{R_{xx}}\mathbf{v}_1\right]$ $\cong \eta_1 [y_1\mathbf{x} - \|y_1\|^2\mathbf{v}_1]$	Oja (1982) [924]
3.	$\Delta\mathbf{v}_1 = \eta_1 \left[\mathbf{R_{xx}}\mathbf{v}_1(\mathbf{v}_1^T\mathbf{v}_1) - \mathbf{v}_1\mathbf{v}_1^T\mathbf{R_{xx}}\mathbf{v}_1\right]$ $\cong \eta_1 [y_1\mathbf{x}\mathbf{v}_1^T\mathbf{v}_1 - y_1^2\mathbf{v}_1]$	Chen, Amari (1998) [190]
4.	$\Delta\mathbf{v}_1 = \eta_1 \left[y_1\mathbf{x} - \|\mathbf{v}_1\|^2\mathbf{v}_1\right]$	Yuille *et al.* (1994) [1350]
5.	$\Delta\mathbf{v}_1 = \left[\dfrac{\mathbf{R_{xx}}\mathbf{v}_1}{\mathbf{v}_1^T\mathbf{R_{xx}}\mathbf{v}_1} - \mathbf{v}_1\right]$ $\cong \dfrac{\sum_{k=1}^N y_1(k)\mathbf{x}(k)}{\sum_{k=1}^N y_1^2(k)} - \mathbf{v}_1$	Roweis (1998) [1026] Tipping, Bishop (1999) [1157]
6.	$\Delta\mathbf{v}_1 = \eta_1 g(\mathbf{v}_1^T\mathbf{v}_1)\left[\mathbf{R_{xx}}\mathbf{v}_1 - \mathbf{v}_1(\mathbf{v}_1^T\mathbf{R_{xx}}\mathbf{v}_1/\mathbf{v}_1^T\mathbf{v}_1)\right]$ $g(\mathbf{v}_1^T\mathbf{v}_1) = 1$ or $\mathbf{v}_1^T\mathbf{v}_1$ or $(\mathbf{v}_1^T\mathbf{v}_1)^{-1}$	Luo *et al.* (1996), Chatterje (1999) [178]
7.	$\Delta\mathbf{v}_1 = \eta_1 \left[\mathbf{R_{xx}}\mathbf{v}_1 - \mathbf{v}_1\mathbf{v}_1^T\mathbf{R_{xx}}\mathbf{v}_1 - \mathbf{v}_1(1 - \mathbf{v}_1^T\mathbf{v}_1)\right]$ $\cong \eta_1 [y_1\mathbf{x} - \|y_1\|^2\mathbf{v}_1 - \mathbf{v}_1(1 - \mathbf{v}_1^T\mathbf{v}_1)]$	Abed-Meraim, Douglas, Hua, Chatterje (1999) [179]
8.	$\Delta\mathbf{v}_1 = \eta_1 \left[2\mathbf{R_{xx}}\mathbf{v}_1 - \mathbf{v}_1\mathbf{v}_1^T\mathbf{R_{xx}}\mathbf{v}_1 - \mathbf{R_{xx}}\mathbf{v}_1\mathbf{v}_1^T\mathbf{v}_1)\right]$ $\cong \eta_1 [2y_1\mathbf{x} - \|y_1\|^2\mathbf{v}_1 - y_1\mathbf{v}_1^T\mathbf{v}_1\mathbf{x})$	Abed-Meraim, Douglas, [3, 414] Hua (1999) [581]
9.	$\Delta\mathbf{v}_1 = \eta_1 y_1 \Psi_1(\mathbf{x} - y_1\mathbf{v}_1)$	Robust Algorithm, Cichocki - Unbehauen (1993) [282]
10.	$y_i(k) = \mathbf{v}_i^T(k)\mathbf{x}_i(k)$ $\eta_i^{-1}(k) = \gamma\eta_i^{-1}(k-1) + \|y_i(k)\|^2$ $\mathbf{v}_i(k+1) = \dfrac{y_i(k)}{\eta_i^{-1}(k)}\left[[\mathbf{x}_i(k) - y_i(k)\mathbf{v}_i(k)\right]$ $\mathbf{x}_{i+1}(k+1) = \mathbf{x}_i(k) - y_i(k)\mathbf{v}_{i*}$ $\mathbf{x}_1(k) = \mathbf{x}(k), \;\; \eta_i^{-1}(0) = \sigma_{yi}^2 = E\{\|y_i\|^2\}$	Fast RLS Algorithm, Cichocki, Kasprzak, Skarbek [282, 270] Yang (1995) [1327]

Table 8.1 Basic cost functions for ICA/BSS algorithms without prewhitening.

No.	Cost function $J(\mathbf{y}, \mathbf{W})$
1.	$-\frac{1}{2}\log\det(\mathbf{W}\mathbf{W}^T) - \sum_{i=1}^{n} E\{\log(p_i(y_i))\}$
2.	$-\frac{1}{2}\log\det(\mathbf{W}\mathbf{W}^T) - \frac{1}{1+q}\sum_{i=1}^{n} \lvert C_{1+q}(y_i)\rvert$ For $q = 3, \quad C_4(y_i) = \kappa_4(y_i) = \frac{E\{y_i^4\}}{E^2\{y_i^2\}} - 3$
3.	$-\frac{1}{2}\log\det(\mathbf{W}\mathbf{W}^T) - \frac{1}{q}\sum_{i=1}^{n} E\{\lvert y_i\rvert^q)\}$ $q = 2$ for nonstationary sources, $q > 2$ for sub-Gaussian sources, $q < 2$ for super-Gaussian sources
4.	$\frac{1}{2}[-\log\det E\{\mathbf{y}\,\mathbf{y}^T\} + \sum_{i=1}^{n}\log E\{y_i^2\}]$ Sources are assumed to be nonstationary

Analogously, we can derive a similar algorithm for the estimation of the mixing matrix $\mathbf{H} \in \mathbb{R}^{m\times n}$ as

$$\boxed{\widehat{\mathbf{H}}(l+1) = \widehat{\mathbf{H}}(l) - \eta(l)\left[\widehat{\mathbf{H}}(l) - \mathbf{C}_{1,q}(\mathbf{x},\mathbf{y})\,\mathbf{S}_{1+q}(\mathbf{y})\right]} \tag{8.57}$$

or equivalently

$$\boxed{\widehat{\mathbf{H}}(l+1) = \widehat{\mathbf{H}}(l) - \eta(l)\widehat{\mathbf{H}}(l)\left[\mathbf{I} - \mathbf{C}_{1,q}(\mathbf{y},\mathbf{y})\,\mathbf{S}_{q+1}(\mathbf{y})\right].} \tag{8.58}$$

The equivariant algorithms presented in this section have several interesting properties which can be formulated in the form of the following theorems (see S. Cruces *et al.* for more details) [322, 327, 328].

Theorem 8.1 *The local convergence of the cumulant based equivariant algorithm (8.34) is isotropic, i.e., it does not depend on the source distribution, as long as their $(1+q)$-order cumulants do not vanish.*

Table 8.2 Family of equivariant adaptive learning algorithms for ICA for complex-valued signals.

No.	Learning Algorithm	References
1.	$\Delta \mathbf{W} = \eta \Big[\mathbf{\Lambda} - \langle \mathbf{f}(\mathbf{y})\, \mathbf{g}^H(\mathbf{y}) \rangle \Big] \mathbf{W}$ $\mathbf{\Lambda}$ is a diagonal matrix with nonnegative elements λ_{ii} $\mathbf{W}(l+1) = \Big[\mathbf{I} \mp \eta\, [\mathbf{I} - \langle \mathbf{f}(\mathbf{y})\, \mathbf{g}^H(\mathbf{y}) \rangle] \Big]^{\mp 1} \mathbf{W}(l)$	Cichocki, *et al.* (1994) [286, 285, 284] Cruces *et al.* (1999) [322, 333]
2.	$\Delta \mathbf{W} = \eta \Big[\mathbf{\Lambda} - \langle \mathbf{f}(\mathbf{y})\, \mathbf{y}^H \rangle \Big] \mathbf{W}$ $\lambda_{ii} = \langle f(y_i(k)) y_i^*(k) \rangle$ or $\lambda_{ii} = 1, \ \forall i$	Amari, Cichocki, Yang (1996) [33] Amari, Chen, Cichocki (1998) [30]
3.	$\Delta \mathbf{W} = \eta \Big[\mathbf{I} - \langle \mathbf{y}\, \mathbf{y}^H \rangle - \langle \mathbf{f}(\mathbf{y})\, \mathbf{y}^H \rangle + \langle \mathbf{y}\, \mathbf{f}^H(\mathbf{y}) \rangle \Big] \mathbf{W}$	Cardoso, Laheld, (1996) [155]
4.	$\Delta \mathbf{W} = \eta \Big[\mathbf{I} - \langle \mathbf{y}\, \mathbf{y}^H \rangle - \langle \mathbf{f}(\mathbf{y})\, \mathbf{y}^H \rangle + \langle \mathbf{f}(\mathbf{y})\, \mathbf{f}^H(\mathbf{y}) \rangle \Big] \mathbf{W}$	Karhunen, Pajunen (1997) [682]
5.	$\Delta \mathbf{W} = -\eta \Big[\mathbf{W}\mathbf{W}^H \langle \mathbf{f}(\mathbf{y})\, \mathbf{y}^H \rangle - \langle \mathbf{y}\, \mathbf{f}^H(\mathbf{y}) \rangle \Big] \mathbf{W}$	Douglas (1999) [390]
6.	$\tilde{\mathbf{W}} = \mathbf{W} + \eta \Big[\mathbf{\Lambda} - \langle \mathbf{f}(\mathbf{y})\, \mathbf{y}^H \rangle \Big] \mathbf{W}, \ \ \lambda_{ii} = \langle f(y_i)\, y_i^* \rangle$ $\eta_{ii} = [\lambda_{ii} + \langle f'(y_i) \rangle]^{-1}; \quad \mathbf{W} = (\tilde{\mathbf{W}}\tilde{\mathbf{W}}^H)^{-1/2}\, \tilde{\mathbf{W}}$	Hyvärinen, Oja (1999) [598]
7.	$\Delta \mathbf{W} = \eta \Big[\mathbf{I} - \mathbf{\Lambda}^{-1} \langle \mathbf{y}\, \mathbf{y}^H \rangle \Big] \mathbf{W}$ $\lambda_{ii}(k) = \langle y_i(k)) y_i^*(k) \rangle$	Amari, Cichocki (1998) [31] Choi, Cichocki, Amari (2000) [235]
8.	$\Delta \mathbf{W} = \eta \Big[\mathbf{I} - \mathbf{C}_{1,q}(\mathbf{y}, \mathbf{y})\, \mathbf{S}_q(\mathbf{y}) \Big] \mathbf{W}$ $C_{1,q}(y_i, y_j) = Cum(y_i, \underbrace{y_j, \ldots, y_j}_{\frac{1+q}{2}-1}, \underbrace{y_j^*, y_j^*, \ldots, y_j^*}_{\frac{1+q}{2}})$	Cruces *et al.* (1999) [327]
9.	$\mathbf{W}(l+1) = \exp(\eta\, \mathbf{F}[\mathbf{y}])\, \mathbf{W}(l)$ $\mathbf{F}(\mathbf{y}) = \mathbf{\Lambda} - \langle \mathbf{y}\, \mathbf{y}^H \rangle - \langle \mathbf{f}(\mathbf{y})\, \mathbf{y}^H \rangle + \langle \mathbf{y}\, \mathbf{f}^H(\mathbf{y}) \rangle$	Nishimori, (1999) [891] Cichocki, Georgiev (2002) [293]
10.	$\Delta \mathbf{W} = \eta \mathbf{F}[\mathbf{y}] \mathbf{W}$ $f_{ij} = \Big[\lambda_{ii}\delta_{ij} - \alpha_{1i} \Big\langle y_i y_j^* \Big\rangle - \alpha_{2i} \Big\langle f(y_i) y_j^* \Big\rangle + \alpha_{3i} \Big\langle y_i f(y_j^*) \Big\rangle \Big]$	Amari, (1997) [28] L. Zhang *et al.* (2000) [1366]

Table 8.3 Typical cost functions for blind signal extraction of a group of e-sources ($1 \leq e \leq n$) with prewhitening of sensor signals, i.e., $\mathbf{A}\mathbf{A}^T = \mathbf{I}$.

No.	Cost function $J(\mathbf{y}, \mathbf{W})$	Remarks
1.	$\sum_{i=1}^{e} h(y_i) = -\sum_{i=1}^{e} E\{\log p_i(y_i)\}$ s.t. $\mathbf{W}_e \mathbf{W}_e^T = \mathbf{I}_e$	Minimization of entropy Amari [26, 34]
2.	$\frac{1}{1+q} \sum_{i=1}^{e} \lvert C_{1+q}(y_i) \rvert$ s.t. $\mathbf{W}_e \mathbf{W}_e^T = \mathbf{I}_e$ Cumulants $C_{1+q}(y_i)$ must not vanish for the extracted source signals. For $q = 3$, $C_4(y_i) = \kappa_4(y_i) = E\{y_i^4\} - 3E^2\{y_i^2\}$	Maximization of distance from Gaussianity Cruces *et al.* [329, 330]
3.	$-\frac{1}{q} \sum_{i=1}^{e} E\{\lvert y_i \rvert^q\}$ s.t. $\mathbf{W}_e \mathbf{W}_e^T = \mathbf{I}_e$	Minimization of generalized energy [235, 284]
4.	$\frac{1}{2} \sum_{i=1}^{e} E\{\log(y_i^2)\}$ s.t. $\mathbf{W}_e \mathbf{W}_e^T = \mathbf{I}_e$	Maximization of negentropy Matsuoka *et al.* [836, 837] Choi *et al.* [235]

and

$$\boxed{\widehat{\mathbf{H}}_e(l+1) = \widehat{\mathbf{H}}_e(l) - \eta(l) \left[\widehat{\mathbf{H}}_e(l) - \mathbf{C}_{1,q}(\mathbf{x}, \mathbf{y}) \, \mathbf{S}_{1+q}(\mathbf{y}) \right],} \tag{8.75}$$

where $\mathbf{R}_{\mathbf{xx}}^{+} \in \mathbb{R}^{m \times m}$ (with $m \geq n$ and $1 \leq e \leq n$) is the pseudo-inverse of the estimated covariance matrix of the observed sensor signals. Then, in order to extract the sources, we only have to alternatively iterate equations (8.74) and (8.75), which constitute the kernel of the implementation of the robust BSE algorithm. The algorithm is able to recover an arbitrary number $e < n$ of sources without the prewhitening step. However, this is done at the extra cost of having to compute the pseudo-inverse of the correlation matrix of the

Table 8.4 BSE algorithm based on cumulants without prewhitening [332].

1. Initialization: $e \le rank(\mathbf{R}_{\mathbf{xx}})$, $\eta_0 < 1$, δ, c, $\mathbf{H}_e(0)$.

   Computing: $\mathbf{R}^+_{\mathbf{x}\,\mathbf{x}}$, $\mathbf{W}_e(0) = (\mathbf{H}^H_e(0)\mathbf{R}^+_{\mathbf{x}\,\mathbf{x}}\mathbf{H}_e(0))^{-1}\mathbf{H}^H_e(0)\mathbf{R}^+_{\mathbf{x}\,\mathbf{x}}$ and $\mathbf{y} = \mathbf{W}(0)\mathbf{x}$.

2. Estimation of the optimal learning rate:

$$\eta(l) = min\left\{\frac{2\eta_0}{1+3\eta_0}, \frac{\eta_0}{1+\eta_0\|\mathbf{C}_{1,3}(\mathbf{y},\mathbf{y})\|_c}\right\}$$

3. Estimation of a column of the mixing matrix and extraction of a group of sources:

$$\begin{aligned}
\widehat{\mathbf{H}}_e(l+1) &= \widehat{\mathbf{H}}_e(l) - \eta(l)(\widehat{\mathbf{H}}_e(l) - \mathbf{C}_{1,3}(\mathbf{x},\mathbf{y})\,\mathbf{S}_4(\mathbf{y})), \\
\mathbf{W}_e(l+1) &= [\widehat{\mathbf{H}}^H_e(l+1)\,\mathbf{R}^+_{\mathbf{x}\,\mathbf{x}}\,\widehat{\mathbf{H}}_e(l+1)]^{-1}\widehat{\mathbf{H}}^H_e(l+1)\,\mathbf{R}^+_{\mathbf{x}\,\mathbf{x}}, \\
\mathbf{y}(k) &= \mathbf{W}_e(l+1)\,\mathbf{x}(k).
\end{aligned}$$

4. $n = n + 1$,
   UNTIL ($\|\mathbf{C}_{1,3}(\mathbf{y},\mathbf{y})\,\mathbf{S}_4(\mathbf{y}) - \mathbf{I}_e\|_c < \delta$) RETURN TO 2.

5. IF deflation
   STORE $\mathbf{y}$,
   $\mathbf{x} = (\mathbf{I}_n - \widehat{\mathbf{H}}_e(l)[\widehat{\mathbf{H}}_e(l)]^+)\,\mathbf{x}$,
   RETURN TO 1.
   ELSE END.

sensor signals. The advantage of this algorithm is that it is insensitive to Gaussian noise on the condition that the covariance of the noise is known or can be reliably estimated. Practical implementation of the algorithm for the complex-valued signals using fourth order cumulants is summarized in Table 8.4 [332].

8.6 RECURRENT NEURAL NETWORK APPROACH FOR NOISE CANCELLATION

8.6.1 Basic Concept and Algorithm Derivation

Assume that we have successfully estimated an unbiased estimate of the separating matrix $\mathbf{W}$ via one of the previously described approaches. Then, we can estimate a mixing matrix

Table 9.1 Relationships between instantaneous blind source separation and multichannel blind deconvolution for complex-valued signals and parameters.

Blind Source Separation	Multichannel Blind Deconvolution - Time Domain	Multichannel Blind Deconvolution - z-Transform Domain
	Mixing-Unmixing Model	
$\mathbf{x}(k) = \mathbf{H}\,\mathbf{s}(k)$	$\underline{\mathbf{x}}(k) = \underline{\mathbf{H}} * \underline{\mathbf{s}}(k)$	$\mathbf{X}(z) = \mathbf{H}(z)\mathbf{S}(z)$
$\mathbf{y}(k) = \mathbf{W}\,\mathbf{x}(k)$	$\underline{\mathbf{y}}(k) = \underline{\mathbf{W}} * \underline{\mathbf{x}}(k)$	$\mathbf{Y}(z) = \mathbf{W}(z)\mathbf{X}(z)$
$x_i(k) = \sum_{j=1}^{n} h_{ij} s_j(k)$	$\underline{x}_i(k) = \sum_{j=1}^{n} \underline{h}_{ij} * \underline{s}_j(k)$	$X_i(z) = \sum_{j=1}^{n} H_{ij}(z) S_j(z)$
$y_i(k) = \sum_{j=1}^{n} w_{ij}(l) x_j(k)$	$\underline{y}_i(k) = \sum_{j=1}^{n} \underline{w}_{ij}(l) * \underline{x}_j(k)$	$Y_i(z) = \sum_{j=1}^{n} W_{ij}(z) X_j(z)$
	Contrast Functions: $\phi(\mathbf{y}, \mathbf{W})$ or $\phi(\mathbf{y}, \underline{\mathbf{W}})$ or $\phi(\mathbf{W}(z))$	
$-\log\lvert\det(\mathbf{W})\rvert$	$-\log\lvert\det(\underline{\mathbf{W}})\rvert$	$-\frac{1}{2\pi j}\oint \log\lvert\det \mathbf{W}(z)\rvert z^{-1} dz$
$-\sum_{i=1}^{n} \log(q_i(y_i))$	$-\sum_{i=1}^{n} \log(q_i(y_i))$	$-\sum_{i=1}^{n} \log(q(y_i))$
	Natural Gradient Rules: $\Delta\mathbf{W}(l)$ or $\Delta\underline{\mathbf{W}}(l)$ or $\Delta\mathbf{W}(z)$	
$-\eta \frac{\partial\phi}{\partial\mathbf{W}} \mathbf{W}^H(l)\mathbf{W}(l)$	$-\eta \frac{\partial\phi}{\partial\underline{\mathbf{W}}} * \underline{\mathbf{W}}^H_{(-p)}(l) * \underline{\mathbf{W}}(l)$	$-\eta \frac{\partial\phi}{\partial\mathbf{W}(z)} \mathbf{W}^H(z^{-1})\mathbf{W}(z)$
$-\eta\mathbf{W}(l)\left[\frac{\partial\phi}{\partial\mathbf{W}}\right]^H \mathbf{W}(l)$	$-\eta\underline{\mathbf{W}}(l) * \left[\frac{\partial\phi}{\partial\underline{\mathbf{W}}}\right]^H * \underline{\mathbf{W}}(l)$	$-\eta\mathbf{W}(z) \quad \frac{\partial\phi}{\partial\mathbf{W}(z)} \quad ^H \mathbf{W}(z)$
	Batch Learning Algorithms: $\Delta\mathbf{W}(l)$ or $\Delta\underline{\mathbf{W}}(l)$ or $\Delta\mathbf{W}(z, l)$	
$\eta\Big[\mathbf{W}(l) - \langle \mathbf{f}[\mathbf{y}(k)]\,\mathbf{u}^H(k)\rangle\Big]$ where $\mathbf{u}(k) = \mathbf{W}^H(l)\,\mathbf{y}(k)$	$\eta\Big[\underline{\mathbf{W}}(l) - \langle \mathbf{f}[\underline{\mathbf{y}}(k)] * \underline{\mathbf{u}}^H(-k)\rangle\Big]$ where $\underline{\mathbf{u}}(k) = \underline{\mathbf{W}}^H_{(-p)}(l) * \underline{\mathbf{y}}(k)$	$\eta\Big[\mathbf{W}(z, l) - \langle Z\{\mathbf{f}[\mathbf{y}(k)]\}\mathbf{U}^H(z^{-1})\rangle\Big]$ where $\mathbf{U}(z) = \mathbf{W}^H(z^{-1}, l)\mathbf{Y}(z)$
$\eta\Big[\mathbf{W}(l) - \langle \mathbf{y}(k)\,\mathbf{v}^H(k)\rangle\Big]$ where $\mathbf{v}(k) = \mathbf{W}^H(l)\,\mathbf{g}[\mathbf{y}(k)]$	$\eta\Big[\underline{\mathbf{W}}(l) - \langle \underline{\mathbf{y}}(k) * \underline{\mathbf{v}}^H(-k)\rangle\Big]$ where $\underline{\mathbf{v}}(k) = \underline{\mathbf{W}}^H_{(-p)}(l) * \mathbf{g}[\underline{\mathbf{y}}(k)]$	$\eta\Big[\mathbf{W}(z, l) - \langle \mathbf{Y}(z)\mathbf{V}^H(z^{-1})\rangle\Big]$ where $\mathbf{V}(z) = \mathbf{W}^H(z^{-1}, l)\, Z\{\mathbf{g}[\mathbf{y}(k)]\}$