Distribution times its conjugate gives a constant times instance of the original distribution. Common distrs:

Bernoulli $\leftarrow$ Beta
Categorical $\leftarrow$ Dirichlet
Univariate normal $\leftarrow$ Normal inversa Gamma
Multivariate normal $\leftarrow$ Normal inverse Wishart

$$Pr(x) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}}\, exp\left[-0.5(x-\mu)^T\Sigma^{-1}(x-\mu)\right]$$

When doing multi-class with normal, likelihood of each class is just: $\Pr(x|w=c) = Norm_x[\mu_c, \Sigma_c]$.
EM algorithm is used to fit PDFs which are a marginalization of a joint distribution with a hidden variable. This lower bound on log likelihood is iteratively maximized:

$$\text{Lower bound} = \sum_{i=1}^{I} \int q_i(h_i) \log\left[\frac{Pr(x, h_i|\theta)}{q_i(h_i)}\right] dh_{1\dots I}$$

Factor analysis: $\Pr(x) = Norm_x[\mu, \phi\phi^T + \Sigma]$. In marginalized form, $\Pr(x|h) = Norm_x[\mu + \Phi h, \Sigma]$ and $\Pr(h) = Norm_h[0, I]$. Can mix different marginalization models to make **mixture of robust subspace models**.

Linear regression: $\Pr(w_i|x_i, \theta) = Norm_{w_i}[\phi^T x, \sigma^2]$. Solve for ϕ using least squares, solve for σ^2 by computing variance. Can use **matrix inversion lemma** in Bayesian regression to compute inverses of size $I \times I$ and $D \times D$. For **non-linear** regression, can use $arctan$ or RBFs. In RBFs, need to compute parameter λ using MAP to make sure function is not too smooth and is not discontinuous.
Use Student t-distr. to encourage **sparsity**. Dual regression expresses the solution weights as a lin. combination of data points. **Relevance Vector Machines** combine dual regression and sparsity.

For **logistic regression** use $Bern_w\left[sig[\phi^T x]\right]$. ession matrix is P.D. hence problem is convex. Use non-linear optimization like Newton's method: $\theta_{t+1} = \theta_t + \lambda(\frac{\partial^2 f}{\partial\theta^2})^{-1}\frac{\partial f}{\partial\theta}$.
For **Bayesian logistic regr.**, find MAP estimate $\hat{\phi}$, and approximate Bayesian using normal with $\mu = \hat{\phi}$ and Σ equal to negative inverse of Hessian evaluated at $\hat{\phi}$ (Laplace approximation).
Incremental fitting: Fit one function and bias first, then second function in bias, etc.

During **learning in GMs**, need to compute gradient w.r.t. $Z_t(\theta)$ which is intractable. Contrastive divergence solves this by approximating the gradient.
Can compute marginals using **forward-backward algorithm**: $\Pr(w_n|x_{1\dots N}) = f_n[w_n]b_n[w_n]$ where:

$$f_n[w_n] = \Pr(x_n|w_n)\sum_{w_{n-1}} \Pr(w_n|w_{n-1})f_{n-1}[w_{n-1}]$$

$$b_{n-1}[w_{n-1}] = \sum_{w_n} \Pr(x_n|w_n)\Pr(w_n|w_{n-1})b_n[w_n]$$

Can use **sum-product algorithm** in factor graphs (factor graphs sometimes don't have loops). In **Markov random fields**, we don't want convex losses since they result in blurring across edges - use truncated quadratic instead. Use **alpha expansion algorithm** for multi-class non-submodular grids but pairwise cost must obey norm rules.

To do **PCA** with K principal components, zero-centre the data and make matrix $X \in M_{D\times I}$, then find first K eigenvectors of XX^T. Dual PCA is a similar idea but you find first K eigenvectors of $X^T X$.

$$\textbf{Minim. objective:} \sum_{i=1}^{I} \|x_i - \Phi w_i - \tau\|^2 \tag{1}$$

For **Eucledian transform**, set $\tau = \mu_x - \Omega\mu_w$. Substitute back into (1) and form matrix system $B - \Omega A$. Do SVD $BA^T = ULV^T$, set $\Omega = VU^T$, then find τ. For **similary transform**, find Ω as before. Find scale ρ based on differences in size of rotate vectors $\|...\|^2$, find τ as before except apply scale to rotated vectors. For **affine transform**, learn Φ, τ by minimizing linear system formed from (1).
Homography includes a perspective divide, so not linear. Note that points $\tilde{x}$ and $\tilde{w}$ should be parallel after homography transform, so can exploit: $\tilde{x} \times \Phi\tilde{w} = 0$. Setup linear system, solve for ϕ using SVD and taking last column, then do non-linear optim. Need at least 4 points (8 equations) to compute.
Non-planar extrinsics: setup linear system, solve for Ω, τ using SVD, make Ω orthogonal using SVD again, resize τ, then do non-linear optim.
Planar extrinsics: Compute homography Φ, remove intrinsics using $\Phi' = \Lambda^{-1}\Phi$. Compute SVD of the first two columns, then do UI^*V^T to compute first two columns of Ω. The last column is cross product of the first two, multiply by -1 if determinant is negative. Rescale τ using difference after SVD.
Non-planar intrinsics: setup linear system, just solve directly.
Planar intrinsics: Estimate extrinsics and intrinsics iteratively.
Can infer 3D points from multiple cameras by solving linear equations.

To derive **essential matrix:** Find relationship between two points without intrinsics, take cross product with τ, then inner product with $\tilde{x}_2$, getting $\tilde{x}_2^T\tau \times \Omega\tilde{x}_1 = 0$. Hence $E = \tau_\times\Omega$. Link to **fundamental matrix** is $E = \Lambda_2^T F \Lambda_1$.
Can find **fundamental matrix** using the 8-point algorithm - detect 8 pairs, expand $x_2^T F x_1 = 0$, make linear system, solve using SVD. Need to set last singular value to 0, rescale the data beforehand. Can recover $\tau_\times$ and Ω from E using SVD of E, then, $\tau_\times = ULWU^T$ and $\Omega = UW^{-1}V^T$, where W is a special matrix.
Epipolar lines are $l_1 = \tilde{x}_2^T E$. Epipoles $\tilde{e}_1$ are the last column of V after SVD of E.
Planar rectif.: Move origin to center of image, rotate epipole to horizontal direction, move epipole to infinity.

$$\begin{aligned}
&\text{Temp. evol.:} && \Pr(w_t|x^{t-1}) = \int \Pr(w_t|w_{t-1})\Pr(w_{t-1}|x^{t-1}) \\
&\text{Meas. update:} && \Pr(w_t|x^t) = \frac{\Pr(x_t|w_t)\Pr(w_t|x^{t-1})}{\int \dots\, dw_t} \\
&\text{Kalman temp.:} && \Pr(w_t|w_{t-1}) = Norm_{w_t}[\mu_p + \Psi w_{t-1}, \Sigma_p] \\
&\text{Kalman meas.:} && \Pr(x_t|w_t) = Norm_{x_t}[\mu_m + \Phi w_t, \Sigma_m]
\end{aligned}$$