Vision as Bayesian Inference

Alan Yuille

February 8, 2020
Lecture 3

- The K-means algorithm.
- Soft-coding: mixture of Gaussians with EM.
- Mixture of Von Mises Fisher.
- Mini-epitomes. Image shifts.
Suppose we have a filter $\vec{B}$ and an input image patch $\vec{I}_p$. We want to find the best fit of the filter to the image by allowing us to transform the filter by $\vec{B} \mapsto a\vec{B} + b\vec{e}$, where $\vec{e} = (1/\sqrt{N})(1, \ldots, 1)$. This corresponds to scaling the filter by $a$ and adding a constant vector $b$. If $\vec{B}$ is a derivative filter then, by definition, $\vec{B} \cdot \vec{e} = 0$. We normalize $\vec{B}$ and $\vec{e}$ so that $\vec{B} \cdot \vec{B} = \vec{e} \cdot \vec{e} = 1$.

The goal is to find the best scaling/contrast $a$ and background $b$ to minimize the match:

$$E(a, b) = |\vec{I}_p - a\vec{B} - b\vec{e}|^2.$$

The solution $\hat{a}, \hat{b}$ are given by (take derivatives of $E$ with respect to $a$ and $b$, recalling that $\vec{B}$ and $\vec{e}$ are normalized):

$$\hat{a} = \vec{B} \cdot \vec{I}_p, \quad \hat{b} = \vec{e} \cdot \vec{I}_p.$$
Matched Filters (2)

- In this interpretation, the filter response is just the best estimate of the contrast \( a \). The estimate of the background \( b \) is just the mean value of the image. Finally, the energy \( E(\hat{a}, \hat{b}) \) is a measure of how well the filter "matches" the input image.

- The idea of a matched filter leads naturally to the idea of having a “dictionary” of filters \( \{ \vec{B}^\mu : \mu \in \Lambda \} \), where different filters \( \vec{B}^\mu \) are tuned to different types of image patches. In other words, the input image patch is encoded by the filter that best matches it. The magnitude of the dot product \( \vec{B} \cdot \vec{I} \) is less important than deciding which filter best matches the input \( \vec{I}_p \).

- Matched filters can be thought of an extreme case of sparsity. In the previous lecture an image was represented by a linear combination of basis functions whose weights were penalized by the \( L1\)-norm, \( \sum_i |\alpha_i| \). By comparison, matched filters represent an image by a single basis function. This gives an ever sparser representation of the image, but at the possible cost of a much larger image dictionary. Matched filters can be thought of as feature detectors because they respond only to very specific inputs.
K-means (1)

- One way to learn a dictionary of basis functions, for matched filters, is by using the \( K \)-means algorithm. This is a classic clustering algorithm but there are many others. As we will show, it related to mixtures of Gaussians and the EM algorithm.

- For simplicity, we will set \( a = 1 \) and \( b = 0 \) (i.e. ignore contrast and background, we will return to them later). Hence we seek a set of basis functions which minimize \( E(\{B^k\}) = \sum_{n=1}^{N} \min_k |\vec{I}_n - \vec{B}^k|^2 \).

- We can find the dictionary \( \{B^k\} \) by the K-means algorithm. This is not guaranteed to converge to a global minimum, but there are efficient methods like k++ for initialization. K-means is a clustering algorithm because it clusters data into different subgroups (one basis for each subgroup).
K-means (2)

- The input to K-means is a set of unlabeled data: \( D = \{x_1, \ldots, x_n\} \). The goal is to decompose it into disjoint classes \( w_1, \ldots, w_k \) where \( k \) is known. The basic assumption is that the data \( D \) is clustered around (unknown) mean values \( m_1, \ldots, m_k \).

- We define an association variable \( V_{ia} \). \( V_{ia} = 1 \) if datapoint \( x_i \) is associated to mean \( m_a \) and \( V_{ia} = 0 \) otherwise. We have the constraint \( \sum_a V_{ia} = 1 \) for all \( i \) (i.e. each datapoint is assigned to a single mean). This gives a decomposition of the data. \( D_a = \{i : V_{ia} = 1\} \) is the set of datapoints associated to mean \( m_a \). The set \( D = \bigcup_a D_a \) is the set of all datapoints. \( D_a \cap D_b = \phi \) for all \( a \neq b \), where \( \phi \) is the empty set.

- We define a goodness of fit:

\[
E(\{V\}, \{m\}) = \sum_{i=1}^{n} \sum_{a=1}^{k} V_{ia} (x_i - m_a)^2 = \sum_{a=1}^{k} \sum_{x \in D_a} (x - m_a)^2 \tag{1}
\]

- The goal of the k-means algorithm is to minimize \( E(\{V\}, \{m\}) \) with respect to \( \{V\} \) and \( \{m\} \). \( E(.,.) \) is a non-convex function and no known algorithm can find its global minimum. But k-means converges to a local minimum.
The k-means algorithm

1. Initialize a partition \( \{D^0_a : a = 1 \text{ to } k\} \) of the data. (I.e. randomly partition the datapoints – or use K++).

2. Compute the mean of each cluster \( D_a \), \( m_a = \frac{1}{|D_a|} \sum_{x \in D_a} x \).

3. For \( i=1 \text{ to } n \), compute \( d_a(x_i) = |x_i - m_a|^2 \). Assign \( x_i \) to cluster \( D_{a^*} \) s.t. \( a^* = \arg \min \{ d_a(x_i), \ldots, d_k(x_i) \} \).

4. Repeat steps 2 & 3 until convergence.

This will converge to a minimum of the energy function because steps 2 and 3 each decrease the energy function (or stop if the algorithm is at a local minimum). This will divide the space into disjoint regions.

K-means can be formulated in terms of the assignment variable. At step 2, \( m_a = \frac{1}{\sum_i V_{ia}} \sum_i V_{ia} x_i \). At step 3, \( V_{ia} = 1 \) if \( |x_i - m_a|^2 = \min_b |x_i - m_b|^2 \) and \( V_{ia} = 0 \) otherwise.
A "softer" version of k-means – the Expectation-Maximization (EM) algorithm. Assign datapoint $x_i$ to each cluster with probability $(P_1, \ldots, P_k)$

1. Initialize a partition of the datapoints.

2. For $j=1$ to $n$. Compute the probability that $x_j$ belongs to $\omega_a$.

$$P(\omega_a|x_j) = \frac{\exp - \frac{1}{2\sigma^2} (x_j - m_a)^2}{\sum_b \exp - \frac{1}{2\sigma^2} (x_j - m_b)^2}.$$ 

3. Compute the mean for each cluster: $m_a = \sum_j x_j P(\omega_a|x_j)$

4. Repeat steps 2 & 3 until convergence.

In this version the hard-assign variable $V_{ia}$ is replaced by a soft-assign variable $P(\omega_a|x_j)$. Observe that $\sum_a P(\omega_a|x_j) = 1$. Also observe that the softness is controlled by $\sigma^2$. In the limit, as $\sigma^2 \rightarrow 0$, the distribution $P(\omega_a|x_j)$ will become binary valued, and soft k-means will be the same as k-means.
Soft k-means can be reformulated in terms of mixtures of Gaussians and the Expectation-Maximization (EM) algorithm.

This assumes that the data is generated by a mixture of Gaussian distributions with means \( \{m\} \) and variance \( \sigma^2 I \).

\[
P(x|\{V\}, \{m\}) = \frac{1}{Z} \exp\left\{ - \sum_{ia} V_{ia} \frac{||x_i-m_a||^2}{\sigma^2} \right\}.
\]

This is equivalent to a mixture of Gaussians:

\[
P(x|V, m) = \mathcal{N}(x : \sum_a V_{ia} m_a, \sigma^2),
\]

where the variable \( V \) identifies the mixture component (i.e. \( V_{ia} = 1 \) if datapoint \( x_i \) was generated by mixture \( a \)).

We need to impose a prior \( P(\{V\}) \) on the assignment variable \( V \). It is natural to choose a uniform distribution \( P(V) = 1/Z \), where \( Z \) is the number of possible assignments of the datapoints to the means.
This gives distributions $P(x, \{V\}|\{m\}) = P(x|\{V\}, \{m\})P(\{V\})$. This form enables us to use the EM algorithm (see later lecture). EM will estimate the mean variables $\{m\}$ despite the presence of unknown/missing/latent variables $\{V\}$.

The EM algorithm can be applied to problems like this where there are quantities to be estimated but also missing/latent variables. The EM algorithm can be formulated in terms of minimizing an energy function, but this energy function is non-convex and EM can be only guaranteed to converge to a minimum of the energy function and not to a global minimum. Deriving the soft k-means algorithm by applying the EM algorithm to $P(x|V, m)$ is left as an exercise for the reader.

We can extend soft k-means in several ways. The simplest is to allow the covariances of the Gaussians to differ and to estimate them as well.

But, more generally, we can have a process $P(x, h|\theta)$ where $x$ is the observed data, $h$ is a hidden/missing/latent variable, and $\theta$ are the model parameters.
A second example arises if we require that the data has unit norm $|x_i| = 1, \forall i$ and hence lies on the unit sphere. This can be used to deal with the scaling of images. Recall $I(x) \mapsto a I(x) + b$, where $a$ is the scale (contrast) and $b$ is the background. We set $b = 0$ and normalize the images by $I(x) \mapsto \frac{I(x)}{|I(x)|}$ (so that $I(x)$ has unit norm).

The Von Mises Fisher distribution is $P(x|k, \lambda_k) = \exp\{\lambda_k m_k \cdot x\} \frac{1}{Z(\lambda_k)}$. Here $|x| = |m_k| = 1$, and $\sigma_k$ is a positive constant.

Note that this distribution is related to the Gaussian distribution (with spherical covariance). The exponent of this Gaussian is $-(x - m_k)^2 / 2\sigma^2$. If we require $|x| = |m_k| = 1$, then the exponent becomes $\frac{(x \cdot m_k - 1)^2}{\sigma^2}$. So if we identify $\lambda_k$ with $1/\sigma_k^2$ we recover Von Mises Fisher. In other words, Von Mises Fisher is the natural way to re-formulate mixtures of Gaussians for data that lies on the unit sphere.
Mini Epitomes (1)

▶ This is another way to learn a dictionary with a more complicated generative model with more hidden variables. It is motivated by the fact that images are shift-invariant (unless they are carefully aligned). Recall, see powerpoints, that we want invariance to $I(x) \mapsto aI(x - x_0) + b$, where $x_0$ is a shift.

▶ Let $\{x_i\}_{i=1}^N$ be a set of possibly overlapping patches of size $h \times w$ pixels cropped from a large collection of images.

▶ Our dictionary comprises $K$ mini-epitomes $\{\mu_k\}_{k=1}^K$ of size $H \times W$, with $H \geq h$ and $W \geq w$. The length of the vectorized patches and epitomes is then $d = h \cdot w$ and $D = H \cdot W$, respectively.

▶ We approximate each image patch $x_i$ with its best match in the dictionary by searching over the $N_p = h_p \times w_p$ (with $h_p = H - h + 1$, $w_p = W - w + 1$) distinct sub-patches of size $h \times w$ fully contained in each mini-epitome. Typical sizes we employ are $8 \times 8$ for patches and $16 \times 16$ for mini-epitomes, implying that each mini-epitome can generate $N_p = 9 \cdot 9 = 81$ patches of size $8 \times 8$. 
We model the appearance of image patches using a Gaussian mixture model (GMM). We employ a generative model in which we activate one of the image epitomes $\mu_k$ with probability $P(l_i = k) = \pi_k$, then crop an $h \times w$ sub-patch from it by selecting the position $p_i = (x_i, y_i)$ of its top-left corner uniformly at random from any of the $N_p$ valid positions.

We assume that an image patch $x_i$ is then conditionally generated from a multivariate Gaussian distribution $P(x_i | z_i, \theta) = \mathcal{N}(x_i; \alpha_i T_{p_i} \mu_{l_i} + \beta_i 1, \sigma_i^2 \Sigma_0)$. 

The label/position latent variable vector $z_i = (l_i, x_i, y_i)$ controls the Gaussian mean via $\nu_{z_i} = T_{p_i} \mu_{l_i}$. Here $T_{p_i}$ is a $d \times D$ projection matrix of zeros and ones which crops the sub-patch at position $p_i = (x_i, y_i)$ of a mini-epitome. The scalars $\alpha_i$ and $\beta_i$ determine an affine mapping on the appearance vector and account for some photometric variability and $1$ is the all-ones $d \times 1$ vector. Here $\bar{x}$ denotes the patch mean value and $\lambda$ is a small regularization constant (we use $\lambda = d$ for image values between 0 and 255).
We choose $\pi_k = 1/K$ and fix the $d \times d$ covariance matrix $\Sigma_0^{-1} = D^T D + \epsilon I$, where $D$ is the gradient operator computing the $x-$ and $y-$ derivatives of the $h \times w$ patch and $\epsilon$ is a small constant.

To match a patch $x_i$ to the dictionary, we seek the mini-epitome label and position $z_i = (l_i, x_i, y_i)$, as well as the photometric correction parameters $(\alpha_i, \beta_i)$ that maximize the probability, or equivalently minimize the squared reconstruction error (note that $D1 = 0$).

The squared reconstruction error is:

$$R^2(x_i; k, p) = \frac{1}{c_i^2} (\|D(x_i - \alpha_i T_p \mu_k)\|^2 + \lambda(\|\alpha_i\| - 1)^2),$$

where the last regularization term discourages matches between patches and mini-epitomes whose contrast widely differs.
Mini Epitomes (4)

- We can compute in closed form for each candidate match $\nu_{zi} = T_{pi} \mu_{li}$ in the dictionary the optimal $\hat{\beta}_i = \bar{x}_i - \hat{\alpha}_i \bar{\nu}_{zi}$ and $\hat{\alpha}_i = \frac{\bar{x}_i^T \bar{\nu}_{zi} \pm \lambda}{\bar{\nu}_{zi}^T \bar{\nu}_{zi} + \lambda}$, where $\bar{x}_i = Dx_i$ and $\bar{\nu}_{zi} = D \nu_{zi}$ are the whitened patches.
- The sign in the nominator is positive if $\bar{x}_i^T \bar{\nu}_{zi} \geq 0$ and negative otherwise. Having computed the best photometric correction parameters, we can evaluate the reconstruction error $R^2(x_i; k, p)$.
- In order to learn the parameters we use the EM algorithm. Given a large training set of unlabeled image patches $\{x_i\}_{i=1}^{N}$, our goal is to learn the maximum likelihood model parameters $\theta = (\{\pi_k, \mu_k\}_{k=1}^{K})$ for the epitomic GMM model. As is standard with Gaussian mixture model learning, we employ the EM algorithm and maximize the expected complete log-likelihood.
- The loglikelihood is
  \[ L(\theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{p \in \mathcal{P}} \gamma_i(k, p) \cdot \log \left( \pi_k N \left( x_i; \alpha_i T_p \mu_k + \beta_i c_i^2 \Sigma_0 \right) \right), \]
  where $\mathcal{P}$ is the set of valid positions in the epitome.
Mini Epitomes (5)

- In the E-step, we compute the assignment of each patch to the dictionary, given the current model parameter values. We use the hard assignment version of EM and set $\gamma_i(k, p) = 1$ if the $i$-th patch best matches in the $p$-th position in the $k$-th mini-epitome and 0 otherwise.

- In the M-step, we update each of the $K$ mini-epitomes $\mu_k$ by
  \[
  \left( \sum_{i, p} \gamma_i(k, p) \frac{\alpha_i^2}{c_i^2} T_p^T \Sigma_0^{-1} T_p \right) \mu_k = \sum_{i, p} \gamma_i(k, p) \frac{\alpha_i}{c_i^2} T_p^T \Sigma_0^{-1} (x_i - \bar{x}_i 1).
  \]

- See powerpoints for the results.