EN.601.783: Vision as Bayesian Inference
Homework 1

Ha Bui
hbui13@jhu.edu

Spring 2023

1 Image Representation and PCA Sparsity (40 points)

1. *What are orthogonal basis functions? How can an input image patch be expressed as a combination of orthogonal basis functions?*
Orthogonal basis functions are a set of orthogonal and linearly independent basis vectors $\{b_i(x) : i = 1, ..., N\}$, where N is the number of dimensions in the Cartesian Coordinates, such that:

$$\begin{cases} \sum_x \{b_i(x)\}^2 = 1, \\ \sum_x b_i(x) b_j(x) = 0 \text{ if } i \neq j. \end{cases} \tag{1}$$

Since the set $\{b_i(x) : i = 1, ..., N\}$ is basis functions set in the Cartesian Coordinates, any image patch $I(x)$ can be expressed as a linear combination of the set by:

$$I(x) = \sum_i^N \alpha_i b_i(x), \tag{2}$$

where $\alpha_i \in \mathbb{R}$ are the coefficients (scalar). Because the set $\{b_i(x) : i = 1, ..., N\}$ are orthogonal, we get:

$$\alpha_i = \sum_x I(x) b_i(x). \tag{3}$$

2. *Give two examples of orthogonal basis functions.*
Example (1) in 2-D, the orthogonal basis functions are: $\{(0, 1), (1, 0)\}$. Example (2) in 3-D, the orthogonal basis functions are: $\{(0, 0, 1), (0, 1, 0), (1, 0, 0)\}$

3. *Give a method for estimating a set of basis vectors given a training set of images. What form do these basis functions take if the image is shift-invariant?*
For example in JPEG Coding, given a training set of images $\{I^\mu(x) : \mu \in \Lambda\}$, we can estimate a set of basis vectors $\{b_i(x) : i = 1, ..., N\}$ by minimizing the cost function:

$$\frac{1}{|\Lambda|} \sum_{\mu \in \Lambda} \sum_x \left| I^\mu(x) - \sum_i^N \alpha_i^\mu b_i(x) \right|^2. \tag{4}$$

If images are shift-invariant, then the coefficients α_i are small. As a result, the eigenvectors of the correlation matrix $K(x, y) = \frac{1}{|\Lambda|} \sum_{\mu \in \Lambda} I^\mu(x) I^\mu(y)$ from the basis functions are sinusoids.

4. *How can we represent images in terms of a linear combination of over-complete basis functions by imposing a sparsity constraint?*
We can represent images from image patches $I(x) = \sum_i^N \alpha_i b_i(x)$ by α_i is the solution of the objective function with a sparsity constraint:

$$\sum_x \left| I(x) - \sum_i^N \alpha_i b_i(x) \right|^2 + \lambda \sum_i^N ||\alpha_i||_p. \tag{5}$$

5. *What is the miracle of sparsity? Describe L1 sparsity and show, for a simple example, how it results in a sparse representation.*
The sparsity is a regularization to help α_i close to zero, reducing the number of ways to represent a similar image $I(x)$ and over-fitting.
For L1 sparsity, i.e., $||\alpha_i||_p = |\alpha_i|$ in Equation 5, let's consider a simple example with the following objective function:

$$f(w; I) = (w - I)^2 + \lambda|w|. \tag{6}$$

We have:

$$\begin{cases} f_+(w; I) = (w - I)^2 + \lambda w \text{ if } w \geq 0, \\ f_-(w; I) = (w - I)^2 - \lambda w \text{ if } w < 0. \end{cases} \tag{7}$$

Take derivative, we get:

$$\begin{cases} \frac{df_+}{dw} = 2(w - I) + \lambda, \\ \\ \frac{df_-}{dw} = 2(w - I) - \lambda. \end{cases} \tag{8}$$

Therefore, we obtain the following optimal solutions:

$$\hat{w}(I) = \begin{cases} \frac{2I-\lambda}{2} \text{ if } I \geq \frac{\lambda}{2}, \\ 0 \text{ if } |I| \leq \frac{\lambda}{2}, \\ \frac{2I+\lambda}{2} \text{ if } I \leq \frac{-\lambda}{2}. \end{cases} \tag{9}$$

Therefore, the result shows that for a small enough $|I| \leq \frac{\lambda}{2}$, we can obtain $\hat{w}(I) = 0$, i.e., many α_i will be zero.

6. *Discuss the relative advantages of Principal Component Analysis and Sparse Coding for face recognition.*
In face recognition, if the face images are aligned (e.g., center them in the image patch), the shift-invariant will be removed. As a result, the bases will not be sinusoids and the relative advantage of PCA and Spare Coding is that we can obtain the eigenfaces.

2 Dictionaries, Mixtures of Gaussians, Mini-Epitomes, EM (30 points)

1. *What is the k-means algorithm? What are the means, the assignment variable, and k? What are its convergence properties? What are the advantages of k-means++?*
- k-means is a clustering algorithm that subdivides N datapoint into k clusters such that points are nearer to the "center" (i.e., mean) of their cluster.
- The mean m_a is the center of each cluster $D_a : a = 1, ..., k$ s.t. $m_a = \frac{1}{|D_a|} \sum_{x \in D_a} x$.
- The assignment variable V_{ia} is an indicator (binary variables) to decide to assign each data point x_i to a single mean, i.e.,

$$V_{ia} = \begin{cases} 1 \text{ if } x_i \text{ associated with } m_a, \\ 0 \text{ otw}, \end{cases} \tag{10}$$

s.t. $\forall i$

$$\sum_a V_{ia} = 1. \tag{11}$$

- k is the number of clusters, i.e., we have k cluster D_a where $a = 1, ..., k$.
- The k-means algorithms converge to a minimum of the energy function:

$$E(\{V\}, \{m\}) = \sum_{i=1}^{n} \sum_{a=1}^{k} V_{ia}(x_i - m_a)^2 = \sum_{a=1}^{k} \sum_{x \in D_a} (x - m_a)^2, \tag{12}$$

because the repeating process in k-means can be rewritten with the assignment variable V_{ia} by:

- Computing the mean: $m_a = \frac{1}{\sum_i V_{ia}} \sum_i \sum_{ia} x_i$.
- Assigning data points to the nearest cluster:

$$V_{ia} = \begin{cases} 1 \text{ if } |x_i - m_a|^2 = \min_b |x_i - m_b|^2, \\ 0 \text{ otw.} \end{cases} \tag{13}$$

- The advantage of k-means++ is it avoids the sub-optimal solution w.r.t. the k-means objective function by assigning each data point x_i to each cluster with a probability $\{p_1, ..., p_k\}$.

2. *How can k-means be used to learn a set of dictionary elements for image patches?*
We can learn a set of dictionary elements $\{\alpha_i^\mu, b_i(x)\}_{i=1}^N$ for the set of image patches $\{I^\mu(x)\}_{\mu \in \Lambda}$ with k-means by replacing variables $\{V\}, \{m\}$ in Equation 12 to $\{\alpha_i^\mu\}_{\mu \in \Lambda}, \{b_i(x)\}_{i=1}^N$, i.e., minimizing:

$$E[\alpha, b] = \frac{1}{|\Lambda|} \sum_{\mu \in \Lambda} \sum_x \left| I^\mu(x) - \sum_i^N \alpha_i^\mu b_i(x) \right|^2, \tag{14}$$

with constraints $\sum_i^N \{b_i(x)\}^2 = 1$ and only one α_i^μ is non-zero for each μ.

3. *What is a mixture of Gaussian distribution? And how does k-means relate to a mixture of Gaussian distributions?*
- The mixture of Gaussian distributions is a function that is comprised of k Gaussian clusters of the dataset, each cluster $a : 1, ..., k$ in the mixture is comprised of the following parameters: a mean m_a that defines its center and variance σ_a^2 that defines its width.
- The k-means could be represented by a mixture of Gaussian distributions by the following Equation:

$$\mathbb{P}(x | \{V\}, \{m\}) = \mathcal{N}(x : \sum_a V_{ia} m_a, \sigma_a), \tag{15}$$

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

4. *What are mini-epitomes? How do they deal with shift-invariant? What algorithm is used to learn them? How well can they represent images?*
- Mini-epitomes are additional variables in the dictionary, where each element mini-epitome $\mu_k : k = 1, ..., K$ is a rectangle of size $H \times W$, with $H \geq h$ and $W \geq w$, where h, w are height and with of image patches.
- It can deal with shift-invariant by using a complicated variant of mixtures of Gaussian:

$$\mathbb{P}(x_i | l_i, p_i) = \mathcal{N}(x_i : \alpha \mathbf{T}_{p_i} \mu_{l_i}, \sigma^2 \mathbf{I}), \tag{16}$$

where $\mathbf{T}_{p_i}$ is a projection matrix of zeros and ones which crops the sub-patch at position p_i. This more complicated variant of mixtures of Gaussian leads to building less redundant epitomic dictionaries, mitigating problems of shift-invariant.
- We can apply EM algorithm to learn the mini-epitomes by obtaining the solution of k, p from minimizing the reconstruction error:

$$R^2(x_i; k, p) = \|x_i - \alpha_i \mathbf{T}_p \mu_k\|^2. \tag{17}$$

In the E-step, we compute the assignment of each patch to the dictionary, given the current model parameter values to obtain the expected value of the log-likelihood function of μ. In the M-step, we update each of the K mini-epitomes μ_k such that maximum quantity in E-step.
- By using EM algorithm, it treats mini-epitomes μ as hidden variables to enhance parameter estimation of the mixed model. This helps represent images with less over-fitting and mitigate the shift-invariant problems of image patches.

3 Super Pixels (30 points)

1. *What is the Expectation-Maximization (EM) algorithm? How can EM be applied to learning a mixture of Gaussian distributions? Describe why the EM algorithm converges.*
- The EM algorithm is a way to estimate parameters θ of a model if some variables x can be observed, but others h are hidden, i.e., doing MLE:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\ p(x \mid \theta) = \underset{\theta}{\operatorname{argmax}}\ \sum_h p(x, h \mid \theta). \tag{18}$$

- The mixture component $P(x|\{V\},\{m\})$ in Equation 15 suggests applying EM algorithm to estimate the mean of variable $\{m\}$ if we let variable $\{V\}$ as a hidden variable, i.e., we find the solution:

$$\{m\}^* = \underset{\{m\}}{\operatorname{argmax}}\ p(x \mid \{m\}) = \underset{\{m\}}{\operatorname{argmax}}\ \sum_{\{V\}} p(x, \{V\} \mid \{m\}). \tag{19}$$

In the form of variational inference with free-energy function, the EM minimizes:

$$F(\{m\}, q) = -\log p(x \mid \{m\}) + \sum_{\{V\}} q(\{V\}) \log \frac{q(\{V\})}{p(\{V\} \mid x, \{m\})}, \tag{20}$$

equivalents to minimizing:

$$F(\{m\}, q) = \sum_{\{V\}} q(\{V\}) \log(\{V\}) - \sum_{\{V\}} q(\{V\}) \log p(\{V\}, x \mid \{m\}). \tag{21}$$

At the i^{th} iteration, for E-step, we fix $\{m\}$ and compute:

$$Q(\{m\}, \{m\}^i) = \sum_{\{V\}} q(\{V\}) \log p(\{V\}, x \mid \{m\}). \tag{22}$$

For M-step, we fix q and update:

$$\{m\}^{i+1} = \underset{\{m\}}{\operatorname{argmax}}\ Q(\{m\}, \{m\}^i). \tag{23}$$

- The EM algorithm converges because it increases the likelihood $p(x \mid \{m\})$ at each iteration, i.e.,:

$$p(x \mid \{m\}^{i+1}) \geq p(x \mid \{m\}^i). \tag{24}$$

2. *What are super-pixels? Briefly describe the SLIC algorithm. What are the advantages of representing an image in terms of super-pixels?*
- Super-pixels are the results of decomposing images D into non-overlapping sub-regions D_a s.t. $D = \bigcup_a D_a$ and $\{D_a \bigcap D_b\} = \emptyset, \forall a \neq b$, where the intensity/texture properties are roughly homogeneous within each sub-region.
- SLIC algorithm:

- Initialize K clusters with the center $C_k = [l_k, a_k, b_k, x_k, y_k]$, where $[x, y]$ is the pixel position and $[l, a, b]$ is the pixel color vector in grid positions.
- Move K clusters to lowest gradient positions (x, y):

$$G(x, y) = \|\mathbf{I}(x+1, y) - \mathbf{I}(x-1, y)\|^2 \|\mathbf{I}(x, y+1) - \mathbf{I}(x, y-1)\|^2. \tag{25}$$

- Assign each pixel to a cluster center
- Recalculate the centers as the average *labxy* vector of all pixels belonging to each cluster
- Iterative until convergence
- Fix disconnected segments

- The advantage of representing an image in terms of super-pixels is it helps simplifies the representation into a structure that should be more meaningful and easier to analyze.

4 Image Statistics and Weak Membrane Models (40 points)

1. *What is the Mumford and Shah model for image segmentation?*
- Mumford and Shah model formulated image segmentation of a domain D as the minimization of a functional:

$$E[J,B] = C\int (I(\vec{x}) - J(\vec{x}))^2 \, d\vec{x} + A\int_{D/B} \vec{\nabla} J(\vec{x}) \cdot \vec{\nabla} J(\vec{x}) d\vec{x} + B\int_B ds, \tag{26}$$

where I is an image, J is a smoothed image, $C, A \geq 0$ are constants, B is the position of boundaries that separates D into subdomain $D = \bigcup D_i$, with $D_i \bigcap D_j = 0, \forall i \neq j$ and $B = \bigcup \partial D_i$ (i.e., b specifies the positions of a one-dimensional set of points).

2. *What is convexity? What is the steepest descent algorithm? Why is convexity important for the steepest descent?*
- Convexity is a property of a function. Specifically, an energy function $E[J, I]$ is convex if $\forall \alpha \in [0, 1]$ and any J_1, J_2, we have:

$$\alpha E[J_1, I] + (1-\alpha)E[J_2, I] \geq E\left[\alpha J_1 + (1-\alpha)J_2\right]. \tag{27}$$

- The steepest descent algorithm updates J in the direction of the gradient $-\frac{\partial E}{\partial J}$.
- The convexity is important for the steepest descent because it is a criterion for the algorithm can find a unique minimum.

3. *What is the Rudin-Osher-Fatemi, or total variation, model? Why is it more practically useful than the weak membrane model? Why is it less effective than dictionary methods for denoising images?*
- Rudin-Osher-Fatmei model formulated image segmentation of a domain D as the minimization of a functional:

$$E[J,I] = \int_D \left|\vec{\nabla} J\right| d\vec{x} + \frac{\lambda}{2}\int_D (J(\vec{x}) - I(\vec{x}))^2 \, d\vec{x}. \tag{28}$$

- Compared to the weak membrane model (e.g., Mumford and Shah which is minimize a non-convex function), Rudin-Osher-Fatmei is more practically useful by minimizing a convex function. Hence applied mathematicians can develop efficient algorithms for finding its global minimum in real-time applications.
- It is less effective than dictionary methods for denoising images because it does not decompose the image into a sum of disjoint regions, leading to unable to capture of longer-range interactions when compared with dictionary methods,

4. *What is variational bounding and CCCP? How do they compare to the steepest descent? How do they guarantee that each iteration decreases the cost?*
- Variational bounding is a discrete iterative algorithm that does not require choosing a step size. It proceeds by obtaining a sequence of bounding functions $E_b(\vec{x}, \vec{x_n})$ where x_n is the current state. The bounding functions obey:

$$E_b(\vec{x}, \vec{x_n}) \geq E(\vec{x}), \forall \vec{x}, \vec{x}_n \text{ and } E_b(\vec{x}, \vec{x}_n) = E_b(\vec{x_n}). \tag{29}$$

CCCP is a special case of this approach where $E(\vec{x})$ is decomposed into a concave $E_c(\vec{x})$ and convex part $E_v(\vec{x})$ s.t. $E(\vec{x}) = E_c(\vec{x}) + E_v(\vec{x})$ with the update rule:

$$\vec{\nabla} E_v(\vec{x}_{n+1}) = -\vec{\nabla} E_c(\vec{x}_n). \tag{30}$$

- The steepest descent can also be derived as a special case by expressing $E(\vec{x}) = E(\vec{x}) + \frac{\lambda}{2}|\vec{x}|^2 - \frac{\lambda}{2}|\vec{x}|^2$. If λ is large enough, $E(\vec{x}) + \frac{\lambda}{2}|\vec{x}|^2$ will be convex and $-\frac{\lambda}{2}|\vec{x}|^2$ will be concave. Applying CCCP we can re-derive iterative steepest descent (with Δ in $\vec{x}_{n+1} = \vec{x}_n - \Delta\vec{\nabla}E(\vec{x}_n)$ depending on λ).
- These algorithms can guarantee that each iteration $\vec{x}_{n+1} = \underset{\vec{x}}{\text{argmin}}\, E_b(\vec{x}_n)$ decrease the cost $E(\vec{x})$ because the convexity of $E(\vec{x})$ and the update rule following the descent of gradient, i.e., $\vec{x}_{n+1} = \vec{x}_n - \Delta\vec{\nabla}E(\vec{x}_n)$.

5. *What forms do the histograms of derivative operators of images normally take?*
- The histograms of derivative operators $\frac{dI}{dx}$ of images $I(x, y)$ normally take a form of Laplacian distribution:

$$p(x) = \frac{1}{Z(k)} \exp\{-k|x|\}, \tag{31}$$

where $k > 0$ is a constant and $Z(k)$ is normalize of distribution function. This is because the value of derivative $\frac{dI}{dx}$ is often high at the edges while small elsewhere and the number of elsewhere positions is often much more than the number of edges positions in the image.

5 Decision Theory (40 points)

1. *What is Bayesian Decision Theory (BDT)? What is the loss function, the risk, the probabilities?*
- Bayesian Decision Theory is a framework for making optimal decisions in the presence of uncertainty. For example, in edge detection, we predict $y \in \mathcal{Y} \subset \{-1, 1\}$ to indicate if the edge is present or not from filters image input $x \in \mathcal{X}$.
- Let data point $(x, y) \overset{iid}{\sim} \mathbb{P}(X, Y)$. Since $\mathbb{P}(X, Y) = \mathbb{P}(X \mid Y)\mathbb{P}(Y)$, in the Bayesian setting, we have the prior is $\mathbb{P}(Y)$, the likelihood is $\mathbb{P}(X \mid Y)$, and the posterior is $\mathbb{P}(Y \mid X)$. Denote $\hat{Y} = \alpha(X)$, then we have the loss function is:

$$L(\alpha(X), Y), \tag{32}$$

which is the cost of making decision $\alpha(x)$ if the real decision should be y. And the risk is:

$$R(\alpha) = \sum_{x,y} p(x, y) L(\alpha(x), y). \tag{33}$$

The Bayes rule is $\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}}\, R(\alpha)$ and the Bayes risk is $\min_\alpha R(\alpha) = R(\hat{\alpha})$.

2. *Is average case risk a good idea? How can the loss function be used to prevent undesirable errors like failing to detect a baby sitting on a road.*
- Minimizing average case risk is not really a good idea. On the one hand, it leads to a conceptually attractive and often very useful theory. However, on the other hand, it needs to assume the probabilities are known.
- The baby sitting on a road detection could be cast to binary decision problem $y \in \{-1, 1\}$. To avoid the failure of this detection (i.e., the False Negatives where $y = 1$ but $\hat{y} = -1$), we can add a penalty for this incorrect (similar to False Positive case) while still choosing to pay no penalty for the correct decision. It follows that we can express the Bayes rule in terms of a log-likelihood ratio test $\log \frac{p(x|y=1)}{p(x|y=-1)} > T$, where T depends on prior $\mathbb{P}(Y)$ and the loss function $L(\alpha(X), Y)$.

3. *What are special cases of BDT if the loss function penalizes all errors equally and/or the prior probabilities are uniform? For binary classification, how do we obtain the log-likelihood ratio test from BDT?*
- If the loss function penalizes all errors equally, i.e., $L(\alpha(x), y) = K_1$ if $\alpha(x) \neq y$ and $L(\alpha(x), y) = K_2$ if $\alpha(x) = y$ (with $K_1 \geq K_2$), then the Bayes rule corresponds to the maximum a posteriori estimator $\alpha(x) = \underset{y}{\operatorname{argmax}}\, p(y|x)$, where $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$ is the posterior distribution of y conditioned on y. If, in addition, the prior is a uniform distribution, i.e., $p(y)$ is constant, then Bayes rule reduces to the maximum likelihood estimate $\alpha(x) = \underset{y}{\operatorname{argmax}}\, p(x|y)$.
- For binary classification, we can obtain the log-likelihood ratio test from BDT by firstly deriving the Bayes risk to:

$$R(\alpha) = \sum_x p(x) \sum_y L(\alpha(x), y) p(Y|x). \tag{34}$$

The divide data (x, y) into four sets:

- the true positives $\{(x, y) : \text{ s.t. } \alpha(x) = y = 1\}$
- the true negatives $\{(x, y) : \text{ s.t. } \alpha(x) = y = -1\}$
- the false positives $\{(x, y) : \text{ s.t. } \alpha(x) = 1, y = -1\}$
- the false negatives $\{(x, y) : \text{ s.t. } \alpha(x) = -1, y = 1\}$

These four cases correspond to loss function values $L(\alpha(x) = 1, y = 1) = T_p$, $L(\alpha(x) = -1, y = -1) = T_n$, $L(\alpha(x) = 1, y = -1) = F_p$, and $L(\alpha(x) = -1, y = 1) = F_n$ respectively. Then the decision rule α_T with log-likelihood ratio test from BDT reduces to:

$$\log \frac{p(x|y = 1)}{p(x|y = -1)} > \log \frac{T_n - F_p}{T_p - F_n} + \log \frac{p(y = -1)}{p(y = 1)}. \tag{35}$$

4. *How does BDT relate to machine learning? In particular, to ML algorithms for learning by minimizing the empirical risk.*
In the machine learning setting, the goal is to minimize the distance between the model and empirical data distribution. This is equivalent to doing the maximum likelihood (MLE) and the standard algorithm to do this is empirical risk minimization. We know that in BDT, if $p(y)$ is constant, then Bayes rule also reduces to the MLE $\alpha(x) = \underset{y}{\text{argmax}}\, p(x|y)$. As a result, this is equivalent to minimizing the empirical risk in machine learning.

5. *From the BDT perspective, what are the pro's and con's of Bayesian approaches (which learn the probability distributions) compared with regression-based approaches?*
Compared to regression-based which belongs to frequentist approaches, from the BDT perspective, the advantages of Bayesian approaches include:

- It provides a better uncertainty estimation by estimating all probability distributions instead of point estimation in regression-based (deterministic/frequentist).
- It is less over-fitting in the case of lack of training data by providing prior knowledge $p(y)$ to infer posterior $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$.

However, Bayesian approaches also contain disadvantages, including:

- It require prior $p(y)$ and the selection of prior $p(y)$ is non-trivial.
- It is slower by needing to make inferences and often needs sampling techniques to deal with the intractable problem in the marginal likelihood $p(x) = \int p(x|y)p(y)$ of the denominator.
- It requires putting probability distributions on $p(x|y)$ and learning $p(x|y)$ is hard in vision by the high-dimensional of x,