67750: Advanced Practical Machine Learning 2017/8

Quiz

TA: Daniel Gissin

1 Manifold Learning (25 Points)

1.1 LLE

In the last step of LLE, after estimating each data point using an affine transformation of its k nearest neighbors described by the weight matrix W, we decompose $M = (I - W)^T (I - W)$ into its eigenvectors.

1. Given the calculated W from step 2 of the algorithm, what was our original minimization objective $\Phi(Y)$, which we showed to be equivalent to minimizing $y^T M y$?
2. Show that $\mathbf{1}$ is an eigenvector of M. What is it's eigenvalue?
3. Which eigenvector of M should we take as the coordinates for the 1D case? Why?

1.2 Constrained Optimization

Let $A \in R^{m \times n}$. We wish to find the vector v on the unit sphere which, after multiplication by A, has the minimal squared L_2 norm.

1. Define the Lagrangian for the above constrained optimization problem.
2. Show that the solution v^* can be obtained by an eigendecomposition of a matrix, and show which matrix we need to decompose.
3. Which eigenvector do we need to take to minimize the squared norm? What will the minimal norm be?

2 Unsupervised Image Denoising (25 Points)

2.1 The Gauss-Markov Theorem

1. State the Gauss-Markov theorem.
2. Complete the following proof for the scalar case (given in class) and also explain why the given steps of the proof below are true:

Proof. Let $A : support(Y) \rightarrow support(X)$

$$\mathbb{E}\|x - A(y)\|^2 = \int_x \int_y p(x,y)\|x - A(y)\|^2 dydx \tag{1}$$

$$= \int_y p(y) \int_x p(x|y)\|x - A(y)\|^2 dxdy \tag{2}$$

$$\text{We require that } \forall y \ \frac{\partial}{\partial A(y)}\left[\int_x p(x|y)\|x - A(y)\|^2 dx\right] = 0 \tag{3}$$

$$\iff ?? \tag{4}$$

2.2 MLE Calculation

In class we saw that the Expectiation of the log likelihood of the Gaussian Mixture Model can be written as:

$$\mathbb{E}[LL(S, Z, \theta)] = const + \sum_{i=1}^{n}\sum_{y=1}^{k} c_{i,y} log(\pi_y N(x_i; \mu_y, \Sigma_y))$$

In the exercise, we implemented EM for the Gaussian Scale Mixture (GSM) model, and calculated the maximization step for the scaling constants $\{r_y\}_{y=1}^k$. In case you forgot, the GSM model assumes that image patches are sampled from a a mixture of k Gaussians, with the distributions $N(0, r_y^2 \Sigma)$.

Derive the maximization update for r_y^2 using MLE.

Some linear algebra identities and the MVN distribution might help get you started (A is an $n \times n$ matrix, c is a scalar):

$$N(x; \mu, \Sigma) = \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

$$det(cA) = c^n det(A)$$

$$(cA)^{-1} = c^{-1} A^{-1}$$

3 Clustering (25 Points)

3.1 k-means

1. k-means++ is a method to initialize the centroids in the k-means algorithm. How is initialization performed in k-means++? Explain the logic behind k-means++. What situations is it trying to avoid?
2. Prove that k-means converges in a finite number of steps.

3.2 Spectral Clustering

In spectral clustering we view the data as a graph with n vertices, where the edges are weighted as some function of the distance between the data points.

1. Write the Ratio-Cut objective function for $k = 2$.
2. Why do we use the Ratio-Cut objective function instead of simply looking for a minimal cut in the graph?

4 Structured Prediction (25 Points)

4.1 Markov Chains

You have successfully modeled the weather as a Markov chain and have learned the probabilities for a day being sunny, rainy or snowy, given the day before. These are the probabilities you've learned:

$$\begin{array}{lll} Pr(sunny \rightarrow sunny) = 0.7 & Pr(rainy \rightarrow sunny) = 0.3 & Pr(snowy \rightarrow sunny) = 0 \\ Pr(sunny \rightarrow rainy) = 0.3 & Pr(rainy \rightarrow rainy) = 0.5 & Pr(snowy \rightarrow rainy) = 0.7 \\ Pr(sunny \rightarrow snowy) = 0 & Pr(rainy \rightarrow snowy) = 0.2 & Pr(snowy \rightarrow snowy) = 0.3 \end{array}$$

Today was rainy.

1. What is the probability of it raining in two days?
2. What is the probability of it being rainy in the distant future?

4.2 Hidden Markov Models

HMMs are defined by the transition distribution (τ) and the emission distribution (ϵ), assuming we added a "START" state at time $t = 0$.

1. Write the definition of $V_t(i)$, the value which we calculate in the Viterbi dynamic programming algorithm.
2. Write the recursive formula for calculating $V_t(i)$, given V_{t-1}.
3. Write the value of $V_1(i)$.

Good luck

(did you write your ID on all of the pages?)