Gaussian Mixtures and Tensor Decompositions

Cyril Zhang

April 20, 2015

Today, we're going to look at an algorithm by Hsu and Kakade for learning mixtures of spherical Gaussians, using orthogonal tensor decompositions. Before covering the main result, we'll present a review of the algebra of tensors, then a gentle introduction to the method of moments and Gaussian mixture models.

1 Tensors

Tensors are the generalizations of vectors $v \in \mathbb{R}^n$ and matrices $M \in \mathbb{R}^{m \times n}$. These are respectively 1-tensors and 2-tensors, which we can represent using one- and two-dimensional arrays of real numbers. Although there are more general definitions, for our purposes, a p-th order tensor (or a p-tensor) is an object that can be represented using a p-dimensional array of real numbers. Technically, today by "tensor" we're strictly referring to **covariant Cartesian tensors**.

We can add tensors of the same order and shape, and multiply them by scalars.

1.1 Multilinear forms

Each 1-tensor can be associated with a linear functional, which takes $\mathbb{R}^n$ to $\mathbb{R}$:

$$A(u) = \sum_{i=1}^{n} A_i u_i.$$

We know this as the inner product $\langle A, u \rangle$.

Each 2-tensor can be associated with a bilinear form, which takes a pair of vectors to $\mathbb{R}$:

$$A(u, v) = \sum_{i,j} A_{ij} u_i v_j.$$

In the language of matrices, we know this as $u^T A v$.

We associate any tensor with an analogous multilinear form. For a 3-tensor, we have $A(u, v, w) = \sum_{i,j,k} A_{ijk} u_i v_j w_k$. For the rest of this introduction, we'll build an intuition for tensors of arbitrary order by considering 3-tensors.

We can also "partially" apply this multilinear form. Suppose we're working with 2-tensors. If we let the first argument of the bilinear form $A(u, v)$ range over the elements of the standard basis $\{e_1, \ldots, e_m\}$, keeping v fixed, we can assemble the m results into a vector in $\mathbb{R}^m$. In the case of 2-tensors, this corresponds to the matrix-vector product, a linear map taking $\mathbb{R}^n$ to $\mathbb{R}^m$. We'll call this $A(\cdot, v)$. In general, if we fix k arguments for a p-tensor's multilinear form, we can get a tensor of order $p - k$.

1.2 Tensor product

Now, how do we build higher-order tensors from lower-order tensors? Let's look at how we might build the standard basis for matrices from the standard basis for vectors:

$\mathbb{R}^3$ has basis $\{e_1, e_2, e_3\}$.

$\mathbb{R}^{3\times 3}$ has basis $\{e_{11} = e_1 e_1^T, e_{12} = e_1 e_2^T, \ldots, e_{33} = e_3 e_3^T\}$.

Now, let's forget that 2-tensors are matrices. We *define* $\mathbb{R}^{m\times n}$ (we prefer to write $\mathbb{R}^m \otimes \mathbb{R}^n$) as the space with basis elements $e_i \otimes e_j$, where the e_i's and e_j's are bases for $\mathbb{R}_m$ and $\mathbb{R}_n$, respectively. The tensor product of e_i and e_j is just $e_i \otimes e_j$, by definition.

Now, we define some rules that allow us to take the tensor product of arbitrary tensors:

- Distributivity: $A \otimes (B + C) = A \otimes B + A \otimes C$, $(A + B) \otimes C = A \otimes C + B \otimes C$.
- Scalar multiplication rule: $kA \otimes B = A \otimes kB := k(A \otimes B)$.
- Associativity: $A \otimes (B \otimes C) = (A \otimes B) \otimes C := A \otimes B \otimes C$. (Really, we're saying that there's a natural isomorphism between the tensor products of spaces parenthesized differently.)

Some examples:

- $\begin{bmatrix}1\\2\end{bmatrix} \otimes \begin{bmatrix}3\\4\end{bmatrix} = (e_1 + 2e_2) \otimes (3e_1 + 4e_2) = 3e_1 \otimes e_2 + 4e_1 \otimes e_2 + 6e_2 \otimes e_1 + 8e_2 \otimes e_2 = \begin{bmatrix}3 & 4\\6 & 8\end{bmatrix}$. We see here that the vector-vector case is simply the outer product uv^T.
- The tensor product of three vectors is a natural analogue of the outer product: $\begin{bmatrix}1\\2\end{bmatrix} \otimes \begin{bmatrix}1\\2\end{bmatrix} \otimes \begin{bmatrix}1\\2\end{bmatrix} = \begin{bmatrix}1 & 2\\2 & 4\end{bmatrix} \begin{bmatrix}2 & 4\\4 & 8\end{bmatrix}$.

- Say z is a vector of independent Gaussians, each with mean 0 and variance σ^2. Can we compute $\mathbb{E}[z \otimes z]$? The diagonal terms are just $\mathbb{E}[z_i^2] = \sigma^2$. The cross terms are $\mathbb{E}[z_i z_j] = \mathbb{E}[z_i]\mathbb{E}[z_j] = 0$. So, the answer is $\sigma^2 \sum e_i \otimes e_i = \sigma^2 I$. You might recognize this as a **covariance matrix**.

- What about $\mathbb{E}[z \otimes z \otimes z]$? The "diagonal" terms are $\mathbb{E}[z_i^3] = 0$, and the other terms have at least one $\mathbb{E}[z_i] = 0$. So, the answer is the zero tensor of order 3.

1.3 Tensor symmetry

From now on, we'll only concern ourselves with "square" tensors like $\mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$. In fact, we'll mostly deal with symmetric ones.

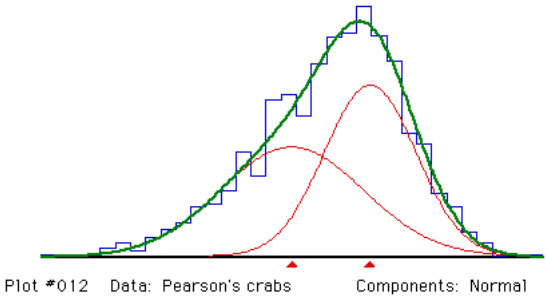
A symmetric tensor A is one for which $A_{i_1 \ldots i_p} = A_{i_{\sigma(1)} \ldots i_{\sigma(p)}}$ for any permutation σ. This is a strong condition: there are $p!$ permutations of the coordinates.

Recall that for symmetric 2-tensors, the spectral theorem gives an eigendecomposition: $A = \sum_i \lambda_i (u_i \otimes u_i)$, with eigenvalues $\{\lambda_i\}$ and an orthonormal basis of eigenvectors $\{u_i\}$. In general, a higher-order tensor does not enjoy such a clean decomposition, but we'll give an algorithm that finds one when one exists. More on that later.

2 Gaussian Mixtures

The Gaussian mixture model is a classic, very useful statistical model. In general, I have a number of multivariate Gaussian sources, with different locations and shapes, and each source has a weight. For each sample, I pick a source with probability according to its weight, then sample from that source. Given a large number of samples, you want to estimate the parameters.

Karl Pearson was the first to consider this problem, in 1894. [**?**] He had the forehead breadth to body length ratios of 1000 crabs, which were distributed like this:



Pearson's hypothesis was that he was actually looking at a mixture of two different species of crabs, whose measurements were normally distributed with different means and variances. He introduced the method of moments: he computed estimates for the first five moments $\hat{\mu} = \frac{1}{n}X_i \approx \mathbb{E}[X]$, $\hat{\sigma}^2 = \frac{1}{n}(X_i - \hat{\mu})^2 \approx \mathbb{E}[(X-\mu)^2]$, $\frac{1}{n}(X_i - \hat{\mu})^k \approx \mathbb{E}[(X-\mu)^k]$, and equated them to the theoretical moments. Then, he found a ninth-degree polynomial whose solutions corresponded to mixtures of two Gaussians that matched these moments. This narrowed it down to two solutions, and he chose the one that best matched the sixth moment. He did this entirely by hand.

Fast-forward to modern times:

- In 1999, Dasgupta found a polynomial-time algorithm to solve this in the multivariate case, as long as the Gaussians are well-separated. [**?**]
- In 2010, Kalai, Moitra, and Valiant found a polynomial-time algorithm which didn't need well-separated components, but required a number of samples exponential in the number of Gaussian sources. [**?**]
- We'll present an algorithm by Hsu and Kakade [HK13] that learns a mixture of spherical Gaussians, without a separation condition, requiring only a polynomial number of samples.

3 The Algorithm

Formally, we have k d-dimensional Gaussian sources $\{(\mu_i, \sigma_i^2)\}$, and a discrete distribution $w \in \mathbb{R}^k$, $w_i > 0$, $\sum w_i = 1$. For each of N samples, we pick $h \sim w$, then report $X = \mu_h + z \in \mathbb{R}^d$, where $z \sim \mathcal{N}(0, \sigma_h^2)$. Then, there is a polynomial-time algorithm that obtains ϵ-accurate estimates on the parameters with constant probability, with a polynomial number of samples.

For simplicity, let's assume $k = d$, and that μ_i's are linearly independent. The general case just involves taking some pseudoinverses instead of inverses. Let's also assume that all of the Gaussians have the same variance $\sigma_1 = \sigma_2 = \ldots = \sigma^2$.

3.1 Observing σ^2

Lemma 1: we can estimate σ^2.

Here, we start using the method of moments. We won't worry about the error analysis. By taking the mean $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} X_i$ of many samples, we can estimate $\mathbb{E}[X] = \mathbb{E}_h[\mu_h] = \sum_{i=1}^{d} w_i \mu_i$.

Now, we estimate $\Sigma = \mathbb{E}[(X - \hat{\mu}) \otimes (X - \hat{\mu})]$, the covariance matrix. This is equal to

$$\begin{aligned}\mathbb{E}_h[(\mu_h - \hat{\mu} + z) \otimes (\mu_h - \hat{\mu} + z)] &= \mathbb{E}_h[(\mu_h - \hat{\mu}) \otimes (\mu_h - \hat{\mu}) + z \otimes z] \\ &= \mathbb{E}_h[(\mu_h - \hat{\mu}) \otimes (\mu_h - \hat{\mu})] + \sigma^2 I \\ &= \sum_{i=1}^{k} w_i (\mu_i - \hat{\mu}) \otimes (\mu_i - \hat{\mu}) + \sigma^2 I.\end{aligned}$$

What do we know about the first term, the sum of rank-one maps? Since it's a sum of psd matrices, it's psd. And since the vectors $(\mu_i - \hat{\mu})$ are linearly dependent (they add up to 0), it can't have full rank d. Thus, it has some 0 eigenvalues; the rest are positive. Adding $\sigma^2 I$ shifts all of its eigenvalues up by σ^2.

So, the smallest eigenvalue of $\hat{\Sigma}$ is an estimate for σ^2: $\hat{\sigma}^2 := \lambda_{min}(\hat{\Sigma})$.

3.2 Observing M_2 and M_3

Lemma 2: we can estimate $M_2 = \mathbb{E}_h[\mu_h \otimes \mu_h]$ and $M_3 = \mathbb{E}_h[\mu_h \otimes \mu_h \otimes \mu_h]$.

By averaging the quantity $X \otimes X$ over many samples, we can compute an estimate for $\mathbb{E}[X \otimes X] = \mathbb{E}[(\mu_h + z) \otimes (\mu_h + z)] = \mathbb{E}_h[\mu_h \otimes \mu_h] + \sigma^2 I$. The first term is M_2, which is what we want. And we have an estimate for σ^2, so we can subtract it off: $\hat{M}_2 = \mathbb{E}[X \hat{\otimes} X] - \hat{\sigma}^2 I$.

Similarly, we can compute an estimate for $\mathbb{E}[X \otimes X \otimes X] = \mathbb{E}[(\mu_h + z)^{\otimes 3}]$. This expands to 8 terms. But recall that $\mathbb{E}[z \otimes z \otimes z] = 0$. Terms like $\mathbb{E}[z \otimes \mu_h \otimes \mu_h]$ are also 0. So, we're left with $M_3 = \mathbb{E}[\mu_h \otimes \mu_h \otimes \mu_h]$, as well as three terms that look like $\mathbb{E}[\mu_h \otimes z \otimes z]$. But this simplifies to

$$\mathbb{E}[\mu_h \otimes (\sigma^2 \sum_{i=1}^{d} e_i \otimes e_i)]$$

$$= \mu \otimes (\sigma^2 \sum_{i=1}^{d} e_i \otimes e_i).$$

Since we have estimates for both μ and σ, we have an estimate for this term. The other terms are the same, with tensor indices permuted. So, we get

$$\hat{M}_3 = \mathbb{E}[X \otimes \hat{X} \otimes X] - \hat{\sigma}^2 \sum_{i=1}^{d} (\hat{\mu} \otimes e_i \otimes e_i - e_i \otimes \hat{\mu} \otimes e_i - e_i \otimes e_i \otimes \hat{\mu}).$$

Now we have estimates for M_2 and M_3. Why do we care about them?

3.3 Learning the Mixture

Let $v_i = \sqrt{w_i} M_2^{-1/2} \mu_i$, and $V = [v_1 | \dots | v_d]$. Then, we have

$$\begin{aligned} VV^T &= \sum w_i M_2^{-1/2} \mu_i \mu_i^T M_2^{-1/2} \\ &= M_2^{-1/2} (\sum w_i \mu_i \mu_i^T) M_2^{-1/2} \\ &= M_2^{-1/2} M_2 M_2^{-1/2} \\ &= I. \end{aligned}$$

So we know that V is an orthogonal matrix; that means the v_i's form an orthonormal basis for $\mathbb{R}^d$. This suggests the following strategy: if we can find an estimate for some matrix of the form $M = VDV^T$, then M is diagonalizable. If the entries in D (the eigenvalues of M) are distinct, then the eigendecomposition is unique up to permutation and sign, allowing us to recover the v_i's. Then, using our estimate for M_2, we can transform these into the μ_i's.

Let's state V in terms of our parameters. Let $A = [\mu_1 | \dots | \mu_d]$, and $W = \text{diag}(w)$. Then, $V = M_2^{-1/2} A W^{1/2}$, so $VDV^T = M_2^{-1/2} A W^{1/2} D W^{1/2} A^T M_2^{-1/2} = M_2^{-1/2} A W D A^T M_2^{-1/2}$. We already have an estimate for $M_2^{-1/2}$, so it would be great if we could observe some matrix of the form $AWDA^T$.

Turns out we can use M_3 to find such a matrix. Let's pick a random vector η, and plug it into one of the arguments of the trilinear form $M_3(\cdot, \cdot, \cdot)$, collapsing it down to a 2-tensor

$$\begin{aligned} \mathbb{E}[\eta^T \mu_h (\mu_h \otimes \mu_h)] &= \sum w_i \eta_T \mu_i (\mu_i \otimes \mu_i) \\ &= AW \text{diag}(\eta^T \mu_1, \dots, \eta^T \mu_d) A^T. \end{aligned}$$

So, this gives us a matrix of the desired form. Also note that with probability 1, the entries of this diagonal matrix are distinct.

So, popping the stack:

- Compute $\hat{\mu}$, $\hat{\sigma}^2$, $\hat{M}_2$, $\hat{M}_3$.
- Pick a random η, and compute the matrix $E = M_3(\eta, \cdot, \cdot)$.
- Compute an estimate for $M_2^{-1/2} E M_2^{-1/2} = VDV^T$. Diagonalize to find V and D. Now we have eigenpairs $(\pm v_i, \eta^T \mu_i)$. Apply $M_2^{1/2}$ to obtain $(\pm \sqrt{w_i} \mu_i, \eta^T \mu_i)$.
- Recover the signs, then weights: $\frac{\eta^T \mu_i}{\eta^T (\pm \sqrt{w_i} \mu_i)} = \pm \frac{1}{\sqrt{w_i}}$, which must be positive. Use these to recover the μ_i's from the $\pm \sqrt{w_i} \mu_i$'s we computed. Done.

3.4 Reflection

That was cool, but the use of M_3 was underwhelming. All we needed it for was to produce a matrix that was simultaneously diagonalizable with M_2, with distinct eigenvalues. Turns out that we can further exploit the structure of the tensor.

There is a 3-rd order tensor T whose trilinear form $T(u, v, w)$ is $M_3(M_2^{-1/2}u, M_2^{-1/2}v, M_2^{-1/2}w)$. You can verify that it's just

$$\begin{aligned} T &= \sum w_i (M_2^{-1/2}\mu_i)^{\otimes 3} \\ &= \sum w_i (\frac{1}{\sqrt{w_i}} v_i)^{\otimes 3} \\ &= \sum \frac{1}{\sqrt{w_i}} v_i^{\otimes 3}. \end{aligned}$$

You can explicitly compute its entries using a matrix multiplication-like rule, but we won't worry about that. If you're worried about consistency, just let u, v, w range through all triples of standard basis vectors, and that allows you to read off the matrix entries.

T is a very special symmetric 3-tensor: it has an orthogonal decomposition. It's a sum of rank-one terms of the form $\lambda(u \otimes u \otimes u)$, where the v_i's are orthogonal. It looks just like the spectral decomposition guaranteed for symmetric 2-tensors. Given T, if we can find this decomposition directly, then we can recover the weights and v_i's, from which we can obtain the μ_i's.

4 Tensor Power Method

This comes from a result by Anandkumar, Ge, Hsu, Kakade, and Telgarsky [AGHKT12], with a cool simplification suggested by Sushant.

Let's quickly review the matrix power method, in tensorial terms. Say you have a symmetric matrix M.

- Start with a random unit vector θ_0.
- Repeat many times: $\theta_{t+1} \leftarrow M(\cdot, \theta_t)$; $\theta_{t+1} \leftarrow \theta_{t+1}/|\theta_{t+1}|$.

$M\theta/\theta$ converges to the largest-magnitude eigenvalue, with eigenvector θ. To find the entire decomposition, repeat the process, working in subspaces orthogonal to the eigenvectors you've found.

The tensor power method is almost identical. Suppose you have a symmetric tensor $T = \sum \frac{1}{\sqrt{w_i}} v_i \otimes v_i \otimes v_i$ that admits an orthogonal decomposition.

- Start with a random unit vector θ_0.

- Repeat many times: $\theta_{t+1} \leftarrow T(\cdot, \theta_t, \theta_t)$; $\theta_{t+1} \leftarrow \theta_{t+1}/|\theta_{t+1}|$. Note that unlike in the matrix case, the map $\theta \mapsto T(\cdot, \theta, \theta)$ is not linear.

4.1 Gist of convergence analysis

Write θ_0 in the spectral basis: $\theta_0 = \sum \alpha_i v_i$, where $\alpha_i = v_i^T \theta_0$. We can omit the normalization step (we only need to apply it once at the end), and see how the coefficients of θ_T relate to those of θ_0. We can show

$$T(\cdot, \theta_1, \theta_1) = \sum \frac{1}{\sqrt{w_i}} \alpha_i^2 v_i.$$

Similarly, if c_i is the coefficient of v_i for some θ_t, the corresponding coefficient in θ_{t+1} is $\frac{c_i^2}{\sqrt{w_i}}$. By induction, after T iterations, the coefficient of v_i becomes $\sqrt{w_i}(\frac{\alpha_i}{\sqrt{w_i}})^{2^T}$. Thus, the eigenvector with the largest $\frac{\alpha_i}{\sqrt{w_i}} = \frac{\eta^T v_i}{\sqrt{w_i}}$ will dominate rapidly (doubly exponentially, compared to exponentially as in the matrix power method).

The tensor power method is also unlike the matrix power method in that any of the (pairwise orthogonal) eigenvectors can be fixed points of the map $\theta \mapsto T(\cdot, \theta, \theta)$. Thus, finding distinct eigenvectors is easier. To get an eigenvector you haven't gotten before, simply pick θ_0 to be orthogonal to the vectors you've found so far. (This fix is due to Sushant.) This decomposition algorithm turns out to be much more numerically stable than finding a matrix SVD.

References

[AGHKT12] Anima Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. *Tensor decompositions for learning latent variable models.* `http://arxiv.org/abs/1210.7559`.

[Das99] Sanjoy Dasgupta. *Learning Mixtures of Gaussians.* `http://cseweb.ucsd.edu/~dasgupta/papers/mog.pdf`.

[HK13] Daniel Hsu and Sham Kakade. *Learning mixtures of spherical Gaussians: moment methods and spectral decompositions.* `http://arxiv.org/abs/1206.5766`.

[KMV12] Adam Kalai, Ankur Moitra, and Gregory Valiant. *Disentangling Gaussians.* `http://research.microsoft.com/en-us/um/people/adum/publications/2012-disentangling-gaussians.pdf`.

[Pea1894] Karl Pearson. Contributions to the mathematical theory of evolution. *Phil. Trans. Roy. Soc. London,* A 185, 71-110. `https://archive.org/details/philtrans02543681`.