Trabajo Final Estimación Bayesiana
Introducción a la Estadística Bayesiana Variacional y utilización de Variational Autoencoders (VAE) como herramienta dentro del campo para resolver problemas aplicados.

Saire - Yudcovsky

Noviembre 2024

1 Introduction

En primer lugar, se dará un marco teórico sobre la estadística bayesiana variacional, y cómo funciona a grandes rangos la inferencia variacional (y sus aplicaciones).

Luego, dentro de esta rama de estudios, se mostrará lo que es un VAE, más conocido como variational autoencoder, y se presentarán dos papers de investigadores relacionados con la facultad que lo usaron como herramienta para resolver problemas aplicados.

2 ¿Qué es la Inferencia Variacional?

Sea D un conjunto de datos, cuyo tamaño es n. Se puede organizar a D en una matriz de diseño $X \in R^n$ y un vector de variables $z = z_{1:d}$. Ahora bien, se apunta a hacer un análisis de datos sobre X y z . En estadística bayesiana, dados un prior p(z) y los datos X = D fijos, se quiere conocer la distribución del posterior. Por Bayes, se computa el posterior como:

$$P(z \mid X = D) = \frac{P(X = D \mid z)P(z)}{P(D)} = \frac{P(X = D \mid z)P(z)}{\int_z P(x, z)dz}$$

Ahora bien, para computar la "marginal" de los datos o más conocida como la constante normalizadora del posterior, $P(D)$, se la desarrolla de la siguiente manera:

$$P(D) = \int_z P(x, z)dz = \int_{z_0} \cdots \int_{z_d-1} P(x, z)dz_0 \cdots dz_{d-1}$$

Esta integral resulta intratable a partir de un número alto de dimensiones y a veces ni siquiera se puede encontrar una fórmula cerrada, por lo que es directamente imposible de calcular.

¿Qué se puede hacer entonces para resolver este problema? Esto es de lo que se ocupa esta área de la disciplina. Lo que se quiere hacer, entonces, es encontrar una "distribución sustituta" q(z) $\approx P(z \mid X = D)$, cuya distribución es parte de alguna familia conocida. Vale aclarar que, en general, el punto de todo esto es que el posterior no pertenece a una familia conocida. La idea sería que q(z) sea casi tan buena como el posterior original, conservando las características principales del mismo. Esto suele convertirse, en matemática, en un Problema de Optimización. Se lo puede plantear de la siguiente forma:

Se busca $q^*(z) = \text{argmin}_{q(z)\in Q}(M)$ donde

- Q es una familia de distribuciones conocida de la misma dimensión que los datos.
- M es una métrica que mide qué tan cerca están el posterior y la distribución conocida.

Con este planteo, podemos inferir que la pregunta que tiene que responder la inferencia variacional son cómo tomar Q y cuál va a ser M, para poder lograr encontrar unos parámetros para esa distribución conocida lo suficientemente buenos para parecerse al posterior.

2.1 Solución a no tener la marginal: ELBO

Una vez planteado el problema de optimización, la tarea se reduce a encontrar una buena métrica que mida qué tan cerca están estas dos distribuciones entre sí. La propuesta que hacen en la bibliografía es utilizar la métrica KL (Kullback-Leibler Divergence), que mide la cercanía de las dos distribuciones y tiene parentezco con la cross entropy (una medida de la información que provee cierta expresión al sistema). Va a tener sentido querer minimizar este medida entre $q(z)$ y $P(z \mid X = D)$ para que sean los más parecidas posibles.
La divergencia KL queda definida como:

$$KL(q(z) \mid\mid P(z \mid D) = E_{z \approx q(z)}[log(\frac{q(z)}{P(z \mid D)})] = \int_{z_0} \cdots \int_{z_d - 1} q(z) log(\frac{q(z)}{P(z \mid D)}) dz_0 \cdots dz_{d-1}$$

Sin embargo, se sabe que esta expresión es intratable y no se dispone de la marginal. Entonces, se reacomoda la expresión de la siguiente manera:

$$\int_z q(z) log(\frac{q(z)}{P(z \mid D)}) dz = \int_z q(z) log(\frac{q(z)P(D)}{P(z, D)}) dz$$

$$= \int_z q(z) log(\frac{q(z)}{P(z, D)}) dz + \int_z q(z) log(P(D)) dz$$

Pero ambos sumandos son por definición una esperanza, y la marginal es un numero, entonces:

$$= E_{z \approx q(z)}[log(\frac{q(z)}{P(z, D)})] + E_{z \approx q(z)} log(P(D))$$

$$= -E_{z \approx q(z)}[log(\frac{P(z, D)}{q(z)}] + log(P(D))$$

Si llamamos $\mathscr{L}(q)$ al primer término, y "evidencia" al segundo, tenemos que:

$$KL = -\mathscr{L}(q) + evidence$$

Pero KL es positiva , y como el logaritmo de una probabilidad es negativo (ya que cualquier número entre 0 y 1 tiene logaritmo negativo) y está fijo, sabemos que $\mathscr{L}(q) \leq 0$ y menor a la evidencia.
Por lo tanto, está L caligráfica será más conocida como ELBO (evidence lower bound). Entonces nuestro problema se reduce a hacer la siguiente observación: $ELBO = log(P(D) \iff KL(q(z) \mid\mid P(z \mid D) = 0$, pero este valor generalmente no se alcanza. Intentaremos que sea lo más cercano a cero posible. Entonces, la tarea de la Inferencia Variacional será maximizar el ELBO.

Otra forma de ver esto es recordando la desigualdad de Jensen aplicada a distribuciones de probabilidad. Cuando f es cóncava, se cumple que $f(E[X]) \geq E[f(X)]$. Usamos la desigualdad de Jensen en el logaritmo de la probabilidad de las observaciones,

$$\log p(x) = \log \sum_z p(x, z)$$

$$= \log \sum_z \frac{p(x, z)}{q(z)} q(z)$$

$$= \log E_q \left[\frac{p(x, Z)}{q(Z)} \right] \geq E_q[\log p(x, Z)] - E_q[\log q(Z)].$$

Este es el ELBO. (Nota: Este es el mismo límite inferior que se usa en la derivación del algoritmo de Expectation-Maximization).

2.2 Inferencia variacional de campo medio - Mean Field Approach -

Una vez planteado el problema y una forma de computar la distancia entre dos distribuciones, se busca que sea lo más chica posible. Se dijo de este sustituto $q(z) \in Q$, ¿pero quién es Q?
Para responder esa pregunta, se recurre a "campo medio", que es la técnica de aproximación que se utiliza frecuentemente en esta disciplina para simplificar la conjunta, asumiendo independencia en las variables. El método se realiza de la siguiente forma:

- Se asume que las familia Q que contiene a q(z) está compuesta por $z_1, ..., z_m$ independientes, por tanto, se puede factorizar la conjunta:

$$q(z_1, \ldots, z_m) = \prod_{j=1}^{m} q(z_j). \tag{17}$$

  Entonces, si se asume independencia entre las componentes del vector z, la conjunta es factorizable.

- Típicamente, la familia elegida puede no contener al verdadero posterior ya que algunas variables pueden ser dependientes entre sí. Por ejemplo, más adelante en el trabajo se nombrarán el modelo de mezclas gaussianas, en donde todas las probabilidades que tienen cierto dato que pertenece a uno de los k clusters dependen de los otros datos. Por tanto, cada variable z_i que será "la cantidad de cada feature que va en cada cluster", no es independiente de las demás.

- Entonces, la idea sería optimizar el ELBO para esta distribución factorizada.

- A continuación se describe un método iterativo, que se llama "coordinate ascent inference", que brevemente consiste en fijar N-1 variables de esta distribución e inferir la restante.

2.2.1 Derivación de la expresión para el paso iterativo

Se supone $z \in R^3$.

$$q(z) = q_0(z_0)q_1(z_1)q_2(z_2) = \arg\max_{q_0,q_1,q_2} (\mathscr{L}(q(z))$$

Fijando z_1 y z_2:

$$\mathscr{L}(q) = \mathscr{L}(q_0) = \int_{z_0} q_0(E_{1,2}[logP] - logq_0)dz_0 + \text{constante}$$

Derivando e igualando a cero:

$$log[q_j(z_j)] = E_{i\neq j}[log(p(z, X = D]$$

Luego, se reescribe la función objetivo en términos de $q(z_k)$:

$$\mathcal{L}_k = \int q(z_k)E_{-k}\left[\log p(z_k|z_{-k}, x)\right]\, dz_k - \int q(z_k)\log q(z_k)\, dz_k.$$

Se deriva con respecto a $q(z_k)$:

$$\frac{d\mathcal{L}_j}{dq(z_k)} = E_{-k}\left[\log p(z_k|z_{-k}, x)\right] - \log q(z_k) - 1 = 0.$$

Esto (junto con multiplicadores de Lagrange) conduce a la actualización de ascenso de coordenadas para $q(z_k)$:

$$q^*(z_k) \propto \exp\left\{E_{-k}\left[\log p(z_k|Z_{-k}, x)\right]\right\}.$$

Sin embargo, el denominador de la posterior no depende de z_j, por lo que:

$$q^*(z_k) \propto \exp\left\{E_{-k}\left[\log p(z_k, Z_{-k}, x)\right]\right\}.$$

2.2.2 Descripción algorítmica

En el método iterativo queda por tomar la esperanza de todas las demás variables excepto la j-ésima (que están fijas). Esto significa que, en cada iteración, se actualiza uno de los factores (por ejemplo, el parámetro o conjunto de parámetros en cuestión) para maximizar nuestra función objetivo que ya dijimos que es maximizar la ELBO. Este proceso se repite para cada factor hasta que la ELBO converge, logrando una buena aproximación de la posterior con algún criterio de parada.

3 Variational Autoencoders

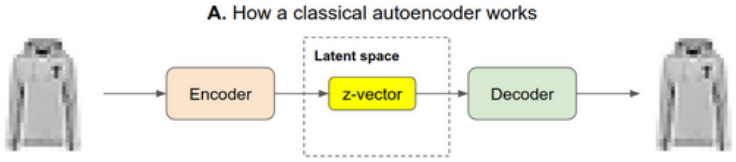
En el marco teórico de inferencia variacional, se puede construir un método de aprendizaje profundo basado en alguna familia específica. En vez de buscar una familia Q, se asume normalidad para construir la arquitectura de la red. Estos metodos se enfocan en disminuir la distancia entre la distribución original y la parametrización sustituta.

3.1 ¿Qué es un Autoencoder?

Un autoencoder es un tipo de estructura utilizado en deep learning para aprender features de tipos de datos a través de una muestra de los mismos, donde se tienen capas de nodos de input y output, y capas ocultas (imagen A). Las capas se unen a través de aristas que van a tener pesos, que primero son random y luego se van mejorando a través del algoritmo de Backpropagation.

La idea se asemeja al perceptrón, que es un modelo de red neuronal artificial donde cada una de las dendritas (aristas entrantes) van hacia una neurona para después de la activación (función no lineal), disparar un potencial que transmite información a la siguiente capa. Las redes neuronales actuales son Multi-Layer-Perceptron (MLP) que son redes de varias capas donde cada nodo (excepto el último) se comunica con todas las aristas de la capa posterior.

- Capa encoder: se van haciendo convoluciones para reducir la dimensión del input obteniendo diferentes features en cada capa. Luego, se hacen transformaciones lineales con funciones de activación no lineales.
- Capa latente: se conservan las features que se consideraron más importantes de cada uno de los datos. Se va corrigiendo a través del cómputo de las funciones de pérdida, volviéndose un problema de optimización de la diferencia entre el espacio input y el latente.
- Capa decoder: convertimos la información importante nuevamente a las dimensiones originales con modificaciones de utilidad específica al caso de uso. Por ejemplo, si es un problema de clasificación o clustering, se determina su clase o su clúster respectivamente.

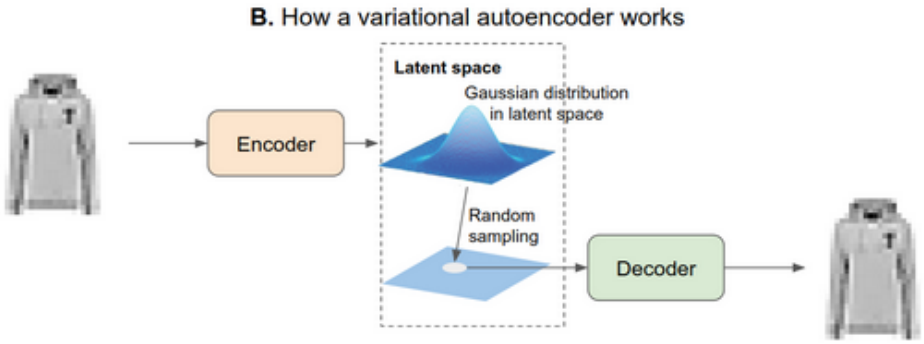


3.2 Ahora, ¿qué es un VAE?

Un Variational Autoencoder es una arquitectura de red neuronal que utiliza la teoría de Inferencia Variacional trabajada arriba. El VAE conecta una red neural de codificación (encoder) y una de decodificación (decoder) mediante un espacio latente probabilístico.

Aunque inicialmente diseñado para aprendizaje no supervisado, el VAE ha demostrado efectividad en tareas de aprendizaje semi-supervisado y supervisado.

- Capa encoder: esta capa permanece sin modificaciones con respecto al autoencoder originial, con el agregado de que en lugar de usar la distribución compleja que tienen los daots, vamos a mapear cada dato a ciertos parámetros para una distribución conocida. Se hacen operaciones explicar que es igual que en todos los autoencoders pero hay una capa que incluye dos parámetros de probabilidad (por ejemplo, si usáramos una normal, serían mu y sigma), hacemos la transformación afín del ϵ y después podemos meter backpropagation.
- Capa latente: típicamente representado por una distribución gaussiana multivariada. Las features que importan son: μ y σ (Σ). Se parametriza una normal con las mismas y se samplea.
- Capa decoder: luego del sampleo, se realiza el upsampling como el decoder original. Se aumenta la dimension hasta que coincida con el input. Por lo tanto, mapea desde el espacio latente al espacio de entrada, siguiendo una distribución que en general omite ruido en la práctica. Ambos componentes se entrenan conjuntamente reparametrizando como mencionamos antes.



3.2.1 Solución a la no derivabilidad de la capa Aleatoria

Supongamos que se va aproximar la función complicada original con una distribución $\mathcal{N}(\mu, \sigma)$. En este contexto, se agregan a las capas del encoder una capa aleatoria de donde se samplean los valores de μ y σ, que posteriormente se combinan para representar el espacio latente. Por lo tanto, se introducen nodos f_μ y f_σ, que corresponden a las distribuciones de las cuales se extraen las medias y desviaciones estándar necesarias para generar las muestras latentes.

El principal desafío de esta construcción radica en la necesidad de aplicar el algoritmo de backpropagation a través de toda la red neuronal. Para que esto sea posible, tanto los valores de las capas como los pesos de la red deben ser diferenciables, ya que la regla de la cadena se utiliza para propagar los gradientes desde la salida hasta las capas iniciales, ajustando los pesos en consecuencia. Sin embargo, la operación de samplear de una distribución aleatoria (como la normal) introduce una discontinuidad debido a su naturaleza no determinista, lo que rompe la diferenciabilidad necesaria para computar los gradientes.

Aunque la función de densidad de una distribución normal estándar $\mathcal{N}(0, 1)$, con forma

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}},$$

es completamente diferenciable, el proceso de sampleo no lo es, ya que implica tomar un valor aleatorio en lugar de una función determinista. Esto impide la propagación del gradiente directamente a través de la capa aleatoria. Por este motivo, se emplea la *reparametrización* para resolver este problema, reformulando el sampleo como una operación diferenciable. En el caso de la distribución normal, esto se logra con:

$$z = \mu + \sigma \cdot \epsilon, \quad \text{donde } \epsilon \sim \mathcal{N}(0, 1).$$

De esta manera, el sampleo aleatorio se desvincula de μ y σ, permitiendo que las operaciones en el espacio latente sean diferenciables y el algoritmo de backpropagation funcione correctamente

3.2.2 Forma genérica de función de Pérdida en el VAE

Para entender la eficacia del modelo planteado, primero hay que definir una función de pérdida diferenciable para actualizar los pesos de la red mediante retropropagación.

La optimización del modelo se basa en minimizar dos términos: **Error de recontrucción + Distancia KL**. Se puede reescribir la definición del ELBO que estaba más arriba como:

$$\mathcal{L}_{\theta,\phi}(x) = \ln p_\theta(x) - D_{KL}(q_\phi(\cdot|x) \| p_\theta(\cdot|x)).$$

que es exactamente lo que se quiere.

Ahora bien, sería ideal maximizar la ELBO, debido a que el logarimo de una probabilidad es negativo y el segundo término es positivo pero multiplicado por -1. Entonces:

$$\theta^*, \phi^* = \underset{\theta,\phi}{\operatorname{argmax}} \mathcal{L}_{\theta,\phi}(x),$$

es equivalente a maximizar $\ln p_\theta(x)$ y minimizar $D_{KL}(q_\phi(z|x) \| p_\theta(z|x))$. Esto implica maximizar la verosimilitud de los datos observados y reducir la diferencia entre el modelo aproximado $q_\phi(z|x)$ y el posterior exacto $p_\theta(z|x)$.

Reescribiendo la ELBO en una forma más conveniente:

$$\mathcal{L}_{\theta,\phi}(x) = E_{z\sim q_\phi(\cdot|x)}\left[\ln p_\theta(x|z)\right] - D_{KL}(q_\phi(\cdot|x)\|p_\theta(\cdot)).$$

Si se asume que $x \sim \mathcal{N}(D_\theta(z), I)$, la distribución condicional $p_\theta(x|z)$ tiene la forma:

$$\ln p_\theta(x|z) \propto -\frac{1}{2}\|x - D_\theta(z)\|_2^2.$$

Asimismo, si $q_\phi(z|x) \sim \mathcal{N}(E_\phi(x), \sigma_\phi(x)^2 I)$ y $p_\theta(z) \sim \mathcal{N}(0, I)$, se puede derivar:

$$\mathcal{L}_{\theta,\phi}(x) = -\frac{1}{2}E_{z\sim q_\phi(\cdot|x)}\left[\|x - D_\theta(z)\|_2^2\right] - \frac{1}{2}\left(N\sigma_\phi(x)^2 + \|E_\phi(x)\|_2^2 - 2N\ln\sigma_\phi(x)\right) + \text{Constante},$$

donde N es la dimensión de z.

4 Aplicaciones prácticas

El artículo "Disentangling Variational Autoencoders" investiga cómo "desenredar" el espacio latente mediante tres modelos de VAE entrenados en un conjunto de 60,000 imágenes de dígitos manuscritos. El objetivo del desenredamiento es mejorar la independencia y la interpretabilidad del espacio latente, permitiendo un control más preciso sobre las características de los datos generados. Un espacio latente enredado, por el contrario, mezcla múltiples factores en una misma dimensión, dificultando el control independiente de los atributos al generar nuevas muestras.

Se analiza el equilibrio entre la calidad de reconstrucción y el nivel de desenredamiento. Logran alinear tres dimensiones latentes con propiedades visuales claras: peso de línea, inclinación y ancho de los dígitos. Los resultados indican que aumentar la contribución de la divergencia Kullback-Leibler y usar etiquetas de clase mejora el desenredamiento del espacio latente.

5 Comparación con otros modelos de Deep Learning

Ahora bien, los VAE no son los únicos que permiten realizar inferencia bayesiana en los modelos. El método más utilizado para samplear de una distribución es Markov-Chain Monte Carlo, en lugar del método lineal que uso el VAE para samplear en el encoder previo al espacio latente. Podemos comparar ambos métodos, y analizar en qué caso conviene usar cada uno.

En primer lugar, el VAE es es más eficiente computacionalmente, debido a que utiliza optimización basada en gradientes (backpropagation), en lugar de realizar múltiples simulaciones como en MCMC. Esto hace que converja a algo cercano al resultado mucho más rápido, especialmente en problemas de alta dimensión o con grandes conjuntos de datos. Por dicho motivo, esta arquitectura es escalable, y como suele trabajar en dimensiones grandes, se suele usar para reducción de la dimensionalidad. Y obviamente, como se trató en todo el trabajo, es bueno aproximando distribuciones que no tienen una forma conocida. En cambio los métodos MCMC son más lentos porque requieren generar cadenas largas para realizar una buena exploración para samplear de la distribución, por lo que cuesta escalar estos algoritmos. Además, como se vio en el entregable 2, puede tardar en converger, y sobre todo puede que converja a un mínimo/máximo local y que sea difícil saberlo. Por otro lado, los métodos MCMC son exactos en la teoría aunque requieran mucho tiempo para converger a ese resultado teórico, y el VAE no se sabe si converge en algún momento aunque sea rápido al principio. Aparte, estos métodos no dependen de tener un posterior bien definido, sino que utilizan una función g que se va adaptando al posterior original f sin saber exactamente la forma, mientras que en el VAE alguna familia $q \in Q$ se busca y cuesta cierto conocimiento del campo encontrarla, además de que sugerir alguna distribución específica puede acarrear un sesgo, que de la otra manera no está.

En general, los VAE son más prácticos para problemas de aprendizaje profundo y generación de datos, mientras que MCMC es más utilizado en análisis bayesianos en contextos donde la calidad de la precisión y la robustez son más importantes que la velocidad o escalabilidad. Algunas sugerencias que aparecen en los foros es que si el enfoque es bayesiano y se está trabajando con deep learning, se puede hacer una mezcla de ambos métodos, por ejemplo se podría usar un VAE para inicializar el espacio latente y luego refinar con MCMC en una etapa posterior.

6 Bibliografía

1. Notas de Princeton: `https://www.cs.princeton.edu/courses/archive/fall11/cos597C/lectures/variational-inference-i.pdf`
2. Definición VAE: `https://en.wikipedia.org/wiki/Variational_autoencoder`
3. Links de Youtube de consulta:
   - Arxiv Insights: https://www.youtube.com/watch?v=9zKuYvjFFS8t=1s
   - 3Blue-1Brown: https://www.youtube.com/watch?v=aircAruvnKkt=1s
   - Umar Jamil (un indú que explica clarísimo, obvio): https://www.youtube.com/watch?v=iwEzwTTalbgt=155s
4. Papers con Aplicaciones de Interés:
   - VesselVAE: Recursive Variational Autoencoders for 3D Blood Vessel Synthesis: `https://arxiv.org/pdf/2307.03592`
   - Disentangling Variational Autoencoders: https://arxiv.org/pdf/2211.07700