Gaußsches Mischmodell (GMM)

Autor und Herausgeber

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Zusammenfassung

Ein Gaußsches Mischmodell (GMM) ist ein Wahrscheinlichkeitsmodell zur Erzeugung von Datenpunkte mit numerischen Merkmalsvektoren. Jeder Datenpunkt wird erzeugt, indem zuerst ein latenter Cluster-Index gezogen und dann der Merkmalsvektor aus der multivariate Normalverteilung dieses Clusters gezogen wird. Die resultierende Randverteilung ist eine gewichtete Summe von multivariate Normalverteilungen. Die Modellparameter werden so gewählt, dass die likelihood function eines Datensatzs maximiert wird. Das resultierende maximum likelihood-Optimierungsproblem ist eine Instanz von empirische Risikominimierung (ERM) unter Verwendung einer spezifischen Wahl für die Verlustfunktion. Dieses Optimierungsproblem ist nicht-konvex und wird typischerweise mit dem EM-Algorithmus (EM)-Algorithmus gelöst. Dieser Algorithmus besteht aus einer fixed-point iteration, die das empirical risk niemals erhöht.

Definition

Ein Foto eines Berges wird in kleine quadratische Patches geschnitten, und jeder Patch wird durch zwei Zahlen beschrieben: wie grün und wie blau er im Durchschnitt ist (Abb. 1). Vegetation, Berg und Dunst sowie heller Himmel bilden drei Cluster solcher Patches, die Datenpunkte sind, charakterisiert durch einen Merkmalsvektor $\mathbf{x} \in \mathbb{R}^d$. Ein GMM ist ein Wahrscheinlichkeitsmodell zur Erzeugung solcher Datenpunkte (Bishop, 2006; Murphy, 2012).

Ein GMM nimmt an, dass jeder Datenpunkt erzeugt wird, indem zuerst ein latenter Cluster-Index $I \in \{1, \ldots, k\}$ gemäß den Cluster-Wahrscheinlichkeiten gezogen wird

$$\mathbb{P}(I = c) = p_c, \qquad \sum_{c=1}^{k} p_c = 1.$$

Bei gegebener Wahl $I = c$ wird der Merkmalsvektor $\mathbf{x}$ aus einer multivariate Normalverteilung $\mathbb{P}^{(c)} = \mathcal{N}\left(\boldsymbol{\mu}^{(c)}, \mathbf{C}^{(c)}\right)$ gezogen, mit Mittelwert $\boldsymbol{\mu}^{(c)}$ und Kovarianzmatrix $\mathbf{C}^{(c)}$. Die resultierende Randverteilung von $\mathbf{x}$ ist daher eine gewichtete Summe von multivariate Normalverteilungen (Abb. 1), d.h.,

$$\mathbb{P} = \sum_{c=1}^{k} p_c \mathcal{N}\left(\boldsymbol{\mu}^{(c)}, \mathbf{C}^{(c)}\right).$$

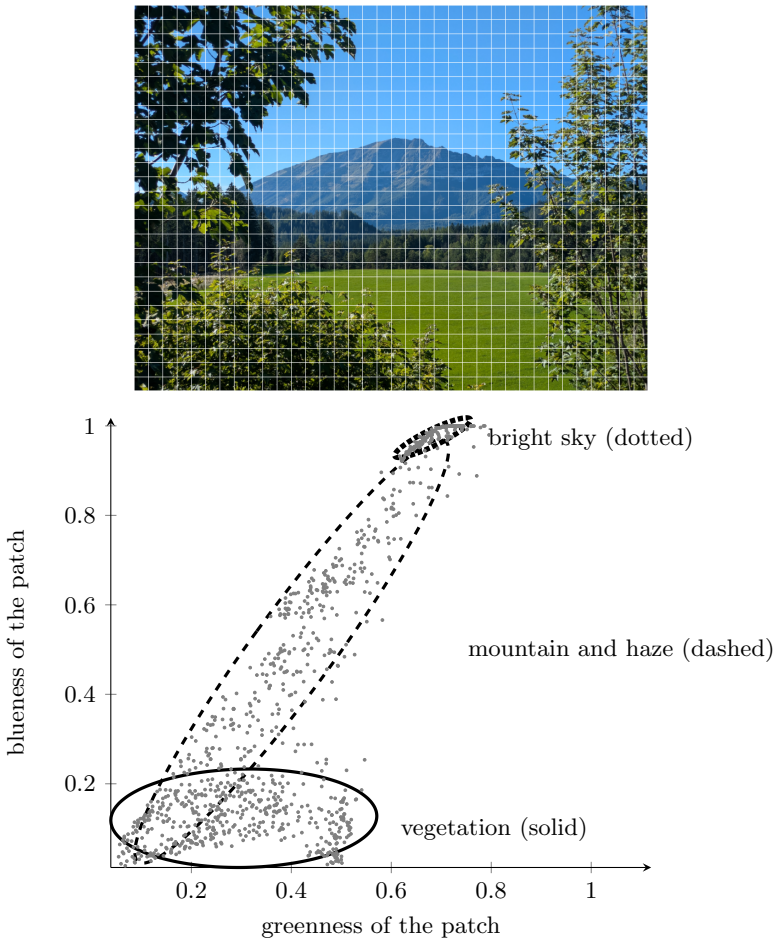


Abbildung 1: Oben: das Foto des Ötscher-Massivs mit dem darauf gezeichneten Patch-Gitter; jeder Patch ist ein Datenpunkt. Unten: die 972 Patches, jeder beschrieben durch seine durchschnittliche Grünheit und seine durchschnittliche Bläue, beide auf einer Skala von 0 bis 1. Die drei beschrifteten Kurven, Vegetation (durchgezogene Kurve), Berg und Dunst (gestrichelte Kurve) und heller Himmel (gepunktete Kurve), sind Konturen der multivariate Normalverteilungen eines GMM, sodass jede Kovarianzmatrix als Ellipse erscheint, verlängert, weil Grünheit und Bläue innerhalb jedes Clusters korreliert sind. Daten generiert von `pythondemos/gmm.py`

Ein GMM wird durch die Cluster-spezifische Wahrscheinlichkeit $p_c = \mathbb{P}(I = c)$, den Mittelwert $\boldsymbol{\mu}^{(c)}$ und die Kovarianzmatrix $\mathbf{C}^{(c)}$ für $c = 1, \ldots, k$ parametrisiert. Diese Modellparameter können aus einem Datensatz $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$ mittels maximum likelihood gelernt werden. Das resultierende Optimierungsproblem hat keine geschlossene Lösung, und die Standardmetho-

de zu seiner annähernden Lösung ist der EM-Algorithmus (EM)-Algorithmus (Dempster et al., 1977), der zwischen zwei Schritten wechselt. Der E-Schritt berechnet für jeden Datenpunkt $\mathbf{x}^{(r)}$ und jeden Cluster-Index c die A-posteriori-Verteilung unter den aktuellen Modellparameter,

$$u_c^{(r)} := \mathbb{P}\left(I = c \mid \mathbf{x}^{(r)}\right) = \frac{p_c \,\mathcal{N}\left(\boldsymbol{\mu}^{(c)}, \mathbf{C}^{(c)}\right)\left(\mathbf{x}^{(r)}\right)}{\sum_{c'=1}^{k} p_{c'} \,\mathcal{N}\left(\boldsymbol{\mu}^{(c')}, \mathbf{C}^{(c')}\right)\left(\mathbf{x}^{(r)}\right)}, \tag{1}$$

und der M-Schritt passt die Modellparameter als Durchschnitte neu an, gewichtet durch diese A-posteriori-Verteilungen,

$$\begin{aligned} p_c &:= \frac{1}{m}\sum_{r=1}^{m} u_c^{(r)}, \qquad \boldsymbol{\mu}^{(c)} := \frac{\sum_{r=1}^{m} u_c^{(r)}\,\mathbf{x}^{(r)}}{\sum_{r=1}^{m} u_c^{(r)}}, \\ \mathbf{C}^{(c)} &:= \frac{\sum_{r=1}^{m} u_c^{(r)}\,(\mathbf{x}^{(r)} - \boldsymbol{\mu}^{(c)})(\mathbf{x}^{(r)} - \boldsymbol{\mu}^{(c)})^\top}{\sum_{r=1}^{m} u_c^{(r)}} \quad \text{für } c = 1, \ldots, k. \end{aligned} \tag{2}$$

Die M-Schritt-Formeln (2) sind keine geschlossene Lösung des maximum likelihood-Optimierungsproblems, da die A-posteriori-Verteilungen auf ihrer rechten Seite von den Modellparameter über den E-Schritt (1) abhängen (Bishop, 2006, Sect. 9.2.2). Die beiden Schritte werden wiederholt, bis sich die Modellparameter nicht mehr ändern (Bishop, 2006, Sect. 9.2.2).

Das maximum likelihood-Optimierungsproblem für die Modellparameter eines GMM ist eine Instanz von empirische Risikominimierung (ERM) mit einer bestimmten Wahl für die Verlustfunktion. Es wird angenommen, dass jede Kovarianzmatrix $\mathbf{C}^{(c)}$ invertible ist, sodass das GMM eine Wahrscheinlichkeitsdichte auf $\mathbb{R}^d$ besitzt. Der Verlust, der durch einen Merkmalsvektor $\mathbf{x}$ entsteht, ist der negative Logarithmus dieser Wahrscheinlichkeitsdichte bei $\mathbf{x}$,

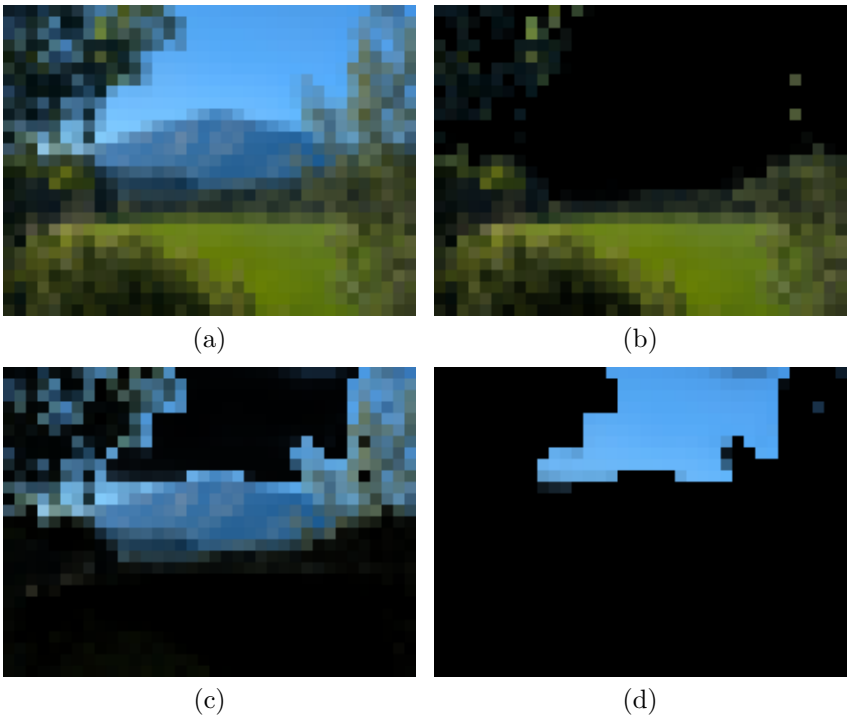
$$L\left(\mathbf{x}, \mathbb{P}\right) = -\ln \sum_{c=1}^{k} p_c \mathcal{N}\left(\boldsymbol{\mu}^{(c)}, \mathbf{C}^{(c)}\right)(\mathbf{x}).$$

Das empirical risk eines GMM wird dann als der durchschnittliche Verlust über das gesamte Datensatz erhalten. Es zu minimieren maximiert die likelihood function, da der Logarithmus monoton steigend ist und das Vorzeichen umgedreht wird. Das Anpassen eines GMM ist daher ERM über den Hypothesenraum von Mischungen aus k multivariate Normalverteilungen.

Die Verlustfunktion ist eine nicht-konvexe Funktion der Modellparameter (Murphy, 2012, Sect. 11.3.2). Der EM-Algorithmus könnte daher zu einem lokalen minimum des empirical risk konvergieren (Bishop, 2006, Sect. 9.2.2), daher wird er typischerweise mehrmals mit verschiedenen Initialisierungen ausgeführt, wobei der Lauf mit dem kleinsten empirical risk beibehalten wird (Murphy, 2012, Sect. 11.3.2). Eine übliche Konstruktion der initialen Modellparameter führt k-Means zuerst aus und beginnt mit dessen Clusterzentren (Bishop, 2006, Sect. 9.2.2).

Unabhängig von seiner Initialisierung ist der EM-Algorithmus, wie k-Means, eine fixed-point iteration. Man fasse die Modellparameter eines GMM in einem

Vektor $\mathbf{w}$ zusammen. Eine Iteration, der E-Schritt (1) gefolgt vom M-Schritt (2), wendet einen Operator $\mathcal{F}$ auf die aktuellen Modellparameter an, sodass die Methode die fixed-point iteration $\mathbf{w}^{(t+1)} = \mathcal{F}\big(\mathbf{w}^{(t)}\big)$ für $t = 0, 1, \ldots$ ist. Die Fixpunkte von $\mathcal{F}$ sind Modellparameter, bei denen der Gradient des empirical risk verschwindet, d.h. die stationären Punkte des empirical risk (siehe zero-gradient condition) (Wu, 1983, Sect. 2.1). Convergence folgt aus monotonem Abstieg und nicht aus einer contractive operator-Eigenschaft: Das Anwenden von $\mathcal{F}$ erhöht das empirical risk niemals (Dempster et al., 1977, Thm. 1), sodass die Sequenz der empirical risk-Werte nicht-steigend ist. Sie konvergiert, wenn das empirical risk nach unten beschränkt ist. Diese Schranke ist für ein GMM nicht automatisch: Eine Komponente, deren Mittelwert mit einem Datenpunkt übereinstimmt und deren Varianz gegen Null schrumpft, treibt das empirical risk gegen $-\infty$ (Bishop, 2006, Sect. 9.2.1). Die Schranke gilt, sobald die Eigenwerte jeder Kovarianzmatrix $\mathbf{C}^{(c)}$ mindestens $\delta > 0$ betragen, da dann die Wahrscheinlichkeitsdichte des GMM nirgends $(2\pi\delta)^{-d/2}$ überschreitet. Unter den Bedingungen von Wu (1983), Thm. 2, die diese Schranke implizieren, konvergiert die Sequenz der empirical risk-Werte und jeder Häufungspunkt der Iterierten $\mathbf{w}^{(t)}$ ist ein Fixpunkt. Auf den Patches von Abb. 1 erhöhte sich das empirical risk in keiner der 300 Iterationen von EM. Trainiert durch EM auf den 972 Patches von Abb. 1, trennt ein Dreikomponenten-GMM Vegetation, Berg und Dunst sowie hellen Himmel. Jeder Patch, den es der blauesten Komponente zuordnet, liegt in der oberen Hälfte des Fotos. Die gelernten Modellparameter definieren ein soft clustering: Die A-posteriori-Verteilung $\mathbb{P}\left(I = c \mid \mathbf{x}\right)$ ist die Wahrscheinlichkeit, dass ein Datenpunkt zum Cluster c gehört, und sie quantifiziert den Zugehörigkeitsgrad zum Cluster c. Im Gegensatz dazu ordnen hartes Clustering-Methoden jeden Datenpunkt genau einem Cluster zu. Abb. 2 zeigt diese Zugehörigkeitsgrade auf dem Foto selbst.



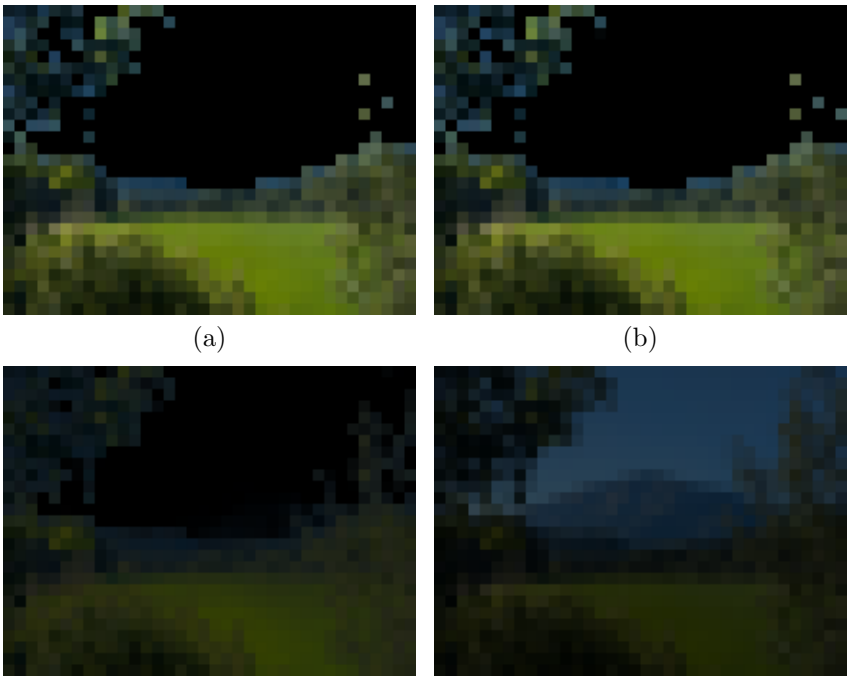
(a) (b) (c) (d)

Abbildung 2: (a) Das Foto aus Abb. 1 in Patch-Auflösung, jedes Patch ein Datenpunkt. Dasselbe Foto dreimal, jedes Patch abgedunkelt durch seine A-posteriori-Verteilung für ein Cluster: Vegetation (b), Berg und Dunst (c) und heller Himmel (d). Ein Patch, das nur zu einem Cluster gehört, ist in einer Kopie hell und in den beiden anderen schwarz, während ein geteiltes Patch in mehreren Kopien dunkel ist. Da sich die drei Wahrscheinlichkeiten eines Patches zu eins summieren, ergeben die drei Kopien zusammen wieder (a). Daten generiert von `pythondemos/gmm.py`

Der EM-Algorithmus zum Lernen der Modellparameter eines GMM reduziert sich unter zusätzlichen Einschränkungen auf den Lloyd's algorithm für k-Means. Man fixiere jede Cluster-Wahrscheinlichkeit auf $1/k$ und jede Kovarianzmatrix auf $\sigma^2\mathbf{I}$, ein Vielfaches der Einheits-Matrix mit einem gemeinsamen σ^2. Die A-posteriori-Verteilung hängt dann von $\mathbf{x}$ nur über ihre euklidische Abstände zu den Mittelwerte ab, und wenn σ^2 gegen 0 schrumpft, konzentriert sie sich auf die nächste. Im Grenzwert ($\sigma^2 \to 0$) wird der E-Schritt (1) zu einer Zuordnung jedes Datenpunkts zu seinem nächsten Mittelwert, und der M-Schritt (2) zentriert die Mittelwerte auf den ihnen zugewiesenen Datenpunkte neu (Bishop, 2006, Sect. 9.3.2, Eq. (9.42)). Die beiden Schritte sind dann eine Iteration des Lloyd's algorithm für k-Means.

Abb. 3 zeigt den Effekt des gemeinsamen σ^2 auf den Zugehörigkeitsgrad zum Vegetations-Cluster. Mit $\sigma^2 = 10^{-4}$ erhält jeder Patch einen Zugehörigkeitsgrad

von 0 oder 1, und die Graduierung stimmt mit der k-Means-Zuordnung auf mehr als 99% der Patches überein. Mit $\sigma^2 = 2 \cdot 10^{-2}$ erhalten 63% der Patches einen Zugehörigkeitsgrad zwischen 0.2 und 0.8. Auf den Patches von Abb. 1 kollabieren mit $\sigma^2 = 1$ die drei Mittelwerte auf den Durchschnitt aller Patches und jeder Patch erhält den gleichen Zugehörigkeitsgrad 1/3. Das Lernen der Kovarianzmatrizen und der Cluster-Wahrscheinlichkeiten anstatt sie zu fixieren, lässt die Graduierung sich an die Daten anpassen, wobei k-Means jeden Datenpunkt einem Cluster zuordnet.



(a) (b) (c) (d)

Abbildung 3: Das Foto in Patch-Auflösung, jedes Patch abgedunkelt durch seinen Zugehörigkeitsgrad zum Vegetations-Cluster unter dem eingeschränkten Gaußschen Mischmodell mit Cluster Wahrscheinlichkeiten $1/k$ und Kovarianzmatrizen $\sigma^2\mathbf{I}$. (a) Die k-Means-Zuordnung: Ein Patch ist hell, wenn es dem Vegetations-Cluster zugewiesen ist, und sonst schwarz. (b) $\sigma^2 = 10^{-4}$: Die Abstufung ist wieder 0 oder 1 und stimmt mit (a) überein. (c) $\sigma^2 = 2 \cdot 10^{-2}$: Die Wiese wird am höchsten eingestuft, das Laub und der Horizont dazwischen, der Himmel am niedrigsten. (d) $\sigma^2 = 1$: Jedes Patch wird mit 1/3 bewertet. Daten generiert von `pythondemos/gmm.py`

Siehe auch: Wahrscheinlichkeitsmodell, multivariate Normalverteilung, Clustering, EM, fixed-point iteration, maximum likelihood, soft clustering, Zugehörigkeitsgrad, hartes Clustering, k-Means, Lloyd's algorithm.

Querverweise

Literatur

1. Bishop (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media, New York, NY, USA.

2. Murphy (2012). Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge, MA, USA.

3. Wu (1983). On the convergence properties of the EM algorithm. Ann. Statist., 11(1), pp. 95–103.

4. Dempster et al. (1977). Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc.: Ser. B (Methodological), 39(1), pp. 1–22.