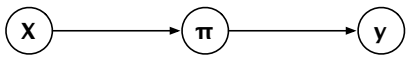
Gibbs sampling

Sebastian Pado

October 30, 2012

1 Bayessche Vorhersage

- Seien X die Trainingdaten, y ein Testdatenpunkt, π die Parameter des Modells
- Uns interessiert $P(y|X)$, wobei wir über das Modell marginalisieren möchten (d.h. wir möchten uns nicht festlegen)
- Annahme: Inferenzprozess funkioniert wie ein Bayes-Netz. Daten generieren Modell, Modell generiert Vorhersage:



  - Faktorisierung a la Bayes-Netz: $P(X, y, \pi) = P(y|\pi)P(\pi|X)P(X)$
  - $P(X)$ interessiert uns nicht, die Daten sind gegeben: $P(y, \pi|X) = P(y|\pi)P(\pi|X)$
  - Dann Marginalisierung über π:

$$P(y|X) = \int P(y|\pi)P(\pi|X)d\pi \tag{1}$$

- Interpretation dieser Formel:
  - Term 1, $P(y|\pi)$: Vorhersage für y durch das Modell
  - Term 2, $P(\pi|X)$: "Gewichtung" dieser Vorhersage je nach der Wahrscheinlichkeit dieses Modells

2 Sampling

- Problem mit der Formel (1): Integral ist schwer zu berechnen
  - Insbesondere ist die "Gewichtung" $P(\pi|X)$ oft ein sehr komplexer Term. Können wir verhindern, $P(\pi|X)$ explizit oft berechnen zu müssen?
- Idee: Wir nähern (1) durch *Sampling* an
  - Q: Warum ist Sampling einfacher als berechnen?

  - A: Für Sampling reicht es, wenn wir eine Funktion berechnen, die *proportional* zur echten Wahrscheinlichkeitsverteilung ist. Sie muss aber nicht normalisiert sein (= zu 1 summieren). Das ist oft eine deutliche Vereinfachung der Aufgabe (siehe unten!).

- Algorithmus:
  - Ziehe Werte für π_i aus $P(\pi|X)$
  - Berechne jeweils $P_i = P(y|\pi_i)$
  - Es gilt:

$$E[P(y|X)] = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} P_i \approx \frac{1}{T} \sum_{i=1}^{T} P_i \tag{2}$$

    * Wo ist die "Gewichtung" $P(\pi|X)$ hinverschwunden?
    * Antwort: In der Auswahl der Werte für π. Das Ziehen aus $P(\pi|X)$ sorgt automatisch dafür, dass wir uns präferiert in "relevanten" (hochgewichteten) Abschnitten der Funktion aufhalten
  - "Monte Carlo": bei Sampling ist Zufall im Spiel (das erste MC von MCMC)

- Wenn wir π zufällig ziehen, landen wir oft an "unwahrscheinlichen" Stellen (d.h. Werte von π, an denen $P(\pi|X)$ niedrig ist). Das Problem wird schlimmer, je hochdimensionaler π ist.

- Wir suchen nach einem Verfahren, das Samples π_i mit einer hohen Wahrscheinlichkeit $P(\pi_i|X)$ produziert

- Idee: *Markov Chain*-Verfahren: Berechne das nächste Sample jeweils ausgehend vom vorherigen Sample (das zweite MC in MCMC)
  - Versuche, dadurch in Richtung höherer Wahrscheinlichkeit zu wandern
  - Parallele zu EM, wo die Reestimationsschritte auch zu wahrscheinlicheren Parametern führen
  - EM kann aber in lokalen Maxima hängen bleiben.

3 Gibbs Sampling: Das Prinzip

- Markov-Chain Monte Carlo Sampling-Methode

- Anwendbar, wenn Modell mehr als zwei Variablen hat (also quasi immer)
  - Formal: Variablenvektor $\vec{z} = (z_1, \ldots, z_k)$
  - Samples bekommen ein Superskript, das den Sample-Index (d.h. den Durchgang) anzeigt: z_5^4 ist zB das vierte Sample der 5. Variable

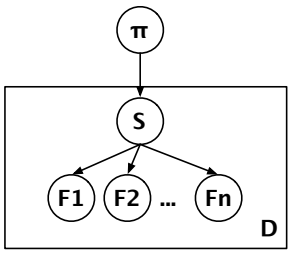
- Intuition: Resample die Variablen eines Samples immer sequentiell
  - Verwende zur Reestimation von z_k^{t+1} (d.h. im Durchgang $t+1$) alle anderen Variablen $z_{i \neq k}$.
  - Verwende dabei das "aktuelle" Sample $t+1$ für alle bereits geresampelten Variablen, also $z_{i<k}^{t+1}$.
  - Verwende das letzte Sample für die noch nicht geresampelten Variablen, also $z_{i>k}^{t}$.

  Als Pseudocode:

  for $t = 1 \rightarrow T$ **do**
    for $i = 1 \rightarrow k$ **do**
      $z_i^{t+1} \sim P(z_i | z_1^{t+1}, \ldots, z_{i-1}^{t+1}, z_{i+1}^{t}, \ldots, z_k^{t})$
    end for
  end for

- Beispiel: 3 Variablen
  - Initialisiere z_1^0, z_2^0, z_3^0 zufällig
  - Sample $z_1^1 \sim P(z_1 | z_2^0, z_3^0)$
  - Sample $z_2^1 \sim P(z_2 | z_1^1, z_3^0)$
  - Sample $z_3^1 \sim P(z_3 | z_1^1, z_2^1)$
- NB. Gibbs Sampling erfordert die Berechnung von *bedingten Wahrscheinlichkeiten* für alle Variablen gegeben die jeweils anderen Variablen

4 Gibbs Sampling: Ein ausgearbeitetes Beispiel



- Idee: Dokumente gehören zu zwei (unbekannten) Klassen (zB gute und schlechte Benutzermeinungen zu Produkten: Sentiment classes)
  - Jedes Dokument hat ein Etikett S
  - Jedes Dokument hat n binäre Features F (an oder aus)
  - Das Modell soll also Dokumente nach Featurähnlichkeit in zwei Klassen clustern.
- Knoten im Modell:

- π: Verhältnis der beiden Klassen (Wahrscheinlichkeit für Klasse 0)
- S: Label für Dokument (0 oder 1). $P(S|\pi) = Bernoulli(\pi)$
- F: Binäre Features (0 oder 1). $P(F_i|S) = Bernoulli(\pi_{i,S})$. Dh. für jede Kombination aus einer Klasse und einem Feature gibt es einen Parameter, der beschreibt, wie wahrscheinlich es ist, dass F_i aktiviert wird für die Klasse S.
- Das Modell hat 1 + 2*n Parameter. 1 Parameter ist π (Klassenprior). 2*n Parameter für alle Kombinationen aus Klassen und Features.

- Gemeinsame Wahrscheinlichkeit:

$$P(\pi, \vec{S}, \mathcal{F}) = P(\pi)P(\vec{S}|\pi)P(\mathcal{F}|\vec{S}) \tag{3}$$

  - wobei $\vec{S}$ = Vektor von Labels, und $\mathcal{F}$ = Matrix von Features
  - $\mathcal{F}$ ist bekannt (und fix).
  - Was wir also sampeln müssen, sind $\vec{F}$ (entsprechend den Parametern $\pi_{i,j}$) und π.
    - Schritt 1: Sample $\vec{S}^{t+1}$ aus $P(\vec{S}|\pi^t, \mathcal{F})$
      Aktualisiere $\pi_{i,j} = P(F|S)$ (durch Zählen)
    - Schritt 2: Sample π^{t+1} aus $P(\pi|\vec{S}^{t+1}, \mathcal{F})$
    - Schritt 3: gehe zu Schritt 1
  - Dazu brauchen wir die bedingen Wahrscheinlichkeiten für diese beiden Variablen

- $P(\vec{S}|\pi, \mathcal{F}) = \prod P(S|\pi, \vec{F})$. Jedes Dokument wird unabhängig von den anderen gelabelt: wir können für jedes Dokument und seine Features $\vec{F}$ einzeln berechnen, was die Wahrscheinlichkeit für die Labels 0 und 1 ist.

$$P(S|\pi, \vec{F}) = \frac{P(S, \pi, \vec{F})}{P(\pi, \vec{F})} \tag{4}$$

$$= \frac{P(\pi)P(S|\pi)P(\vec{F}|S)}{\sum_{S'} P(\pi)P(S'|\pi)P(\vec{F}|S')} \tag{5}$$

$$\propto P(\pi)P(S|\pi)P(\vec{F}|S)^1 \tag{6}$$

$$\propto P(S|\pi) \prod_i P(F_i|S) \tag{7}$$

[1] Wir dürfen den Nennen wegwerfen, weil wir nur sampeln: wie oben diskutiert, genügt es dafür, propertional zur Original-Verteilung zu bleiben. D.h. bei Verteilungen können wir in der Regel alle Normalisierungskonstanten weglassen, was die Terme stark vereinfacht.

- Für S=0: $P(0|\pi, \vec{F}) \propto \pi \prod_i P(F_i|0) = a$
- Für S=1: $P(1|\pi, \vec{F}) \propto (1-\pi) \prod_i P(F_i|1) = b$
- Konkretes Sampling: Ziehe x uniform aus $[0; a+b]$. Wenn $x < a$, S=0. Wenn $x \geq a$, S=1.

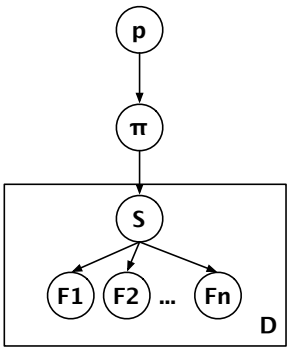
- $P(\pi = x|\vec{S}, \mathcal{F})$ – Sampling für π.

$$
\begin{aligned}
P(\pi = x|\vec{S}, \mathcal{F}) &= P(\pi = x|\vec{S}) && \text{Unabh.} && (8)\\
&= \frac{P(\vec{S}|\pi = x)P(\pi = x)}{P(\vec{S})} && \text{Bayes Rule} && (9)\\
&\propto P(\vec{S}|\pi = x) && \text{Sampling} && (10)\\
&\propto x^{C_0}(1-x)^{C_1} && && (11)
\end{aligned}
$$

  - C_0 und C_1 sind die Anzahl der Dokumente mit Label 0 und 1 sind.
  - Konkretes Sampling aus dieser binomialen Verteilung: zB in R mit `rbinom`

5 Gibbs Sampling mit Priors

- Minimale Erweiterung des Beispiels mit einem Prior über π
  - Beta-Verteilung: $\pi \sim Beta(\alpha, \beta)$
  - Parameter α, β bestimmen Grössenverhältniss der beiden Klassen:
  - $\alpha = 2, \beta = 2$: schwacher Prior für gleich grosse Klassen
  - $\alpha = 20, \beta = 80$: starker Prior für Grössenverhältnis 20/80.



- Einziger Unterschied: $P(\pi)$ jetzt konditioniert auf Prior p. (S ist unabhängig von P, weil durch π getrennt.) Die Sampling-Formel für π ändert sich also (ich ersetze

direkt p durch α, β):

$$P(\pi = x|\vec{S}, \mathcal{F}, \alpha, \beta) = P(\pi = x|\vec{S}, \alpha, \beta) \tag{12}$$

$$= \frac{P(\vec{S}|\pi = x)P(\pi = x|\alpha, \beta)}{P(\vec{S})} \tag{13}$$

$$\propto P(\vec{S}|\pi = x)P(\pi = x|\alpha, \beta) \tag{14}$$

$$\propto x^{C_0}(1-x)^{C_1}x^{\alpha-1}(1-x)^{\beta-1} \tag{15}$$

$$= x^{C_0+\alpha-1}(1-x)^{C_1+\alpha-1} \tag{16}$$

- Merke: α erhöht die Anzahl der gesehenen Instanzen von Klasse 0, β die Anzahl der gesehenen Instanzen von Klasse 1 $\rightarrow$ Priors als "Pseudocounts"

6 Literatur

- P. Resnik and E. Hardisty."Gibbs Sampling for the Uninitiated". UMIACS-TR-2010-04, 2010. `http://www.umiacs.umd.edu/~resnik/pubs/gibbs.pdf`
- B. Walsh. "MCMC and Gibbs Sampling". Lecture Notes for EEB 581. 2004. `http://web.mit.edu/~wingated/www/introductions/mcmc-gibbs-intro.pdf`
- T. Pedersen, R. Bruce. "Knowledge-Lean Word Sense Disambiguation". Proc. AAAI. 1998. `http://reference.kfupm.edu.sa/content/k/n/knowledge_lean_word_sense_disambiguation_76473.pdf`