Silvio Moioli, www.moioli.net

Aggirare semplici CAPTCHA integrando programmi di riconoscimento della scrittura con un algoritmo bayesiano – Silvio Moioli

Prefazione

Molti siti Internet, per evitare l'utilizzo automatico dei loro servizi, si servono delle immagini CAPTCHA per provare l'umanità dell'utilizzatore. Tra gli altri problemi di questi sistemi, specialmente se implementati senza i dovuti accorgimenti, c'è la possibilità di interpretare le immagini da essi prodotte con generici programmi di riconoscimento della scrittura (OCR), rendendoli completamente inutili.

In questo articolo si espone un metodo generico per utilizzare più programmi di riconoscimento della scrittura (OCR) simultaneamente e combinarne i risultati per ottenere interpretazioni migliori rispetto a quelle ottenibili utilizzando ogni programma individualmente.

Questo metodo è stato utilizzato con successo per aggirare i CAPTCHA di più siti Internet dal programma MoioSMS, scritto dall'autore di quest'articolo ed utilizzato da migliaia di utenti ogni giorno.

Definizioni

Nel seguito dell'articolo, si utilizzeranno i seguenti termini con i significati precisati:

- *immagine*: un'immagine CAPTCHA contenente una stringa di caratteri variamente deformati;
- *oracolo*: la stringa effettivamente corrispondente ad una certa *immagine*;
- *metodo*: programma o combinazione di programmi che data un'*immagine* fornisce una stringa corrispondente (non necessariamente uguale all'*oracolo*);
- *interpretazione*: la stringa fornita da un *metodo*;
- *probabilità di corretta interpretazione* di un carattere: probabilità che un metodo interpreti correttamente un carattere di un'*immagine*, ossia la probabilità che tale carattere sia uguale nell'*interpretazione* e nell'*oracolo*.

Il problema

E' possibile tentare di aggirare alcuni CAPTCHA abbastanza semplici utilizzando diversi programmi OCR esistenti. Ognuno di essi, non essendo progettato e realizzato specificatamente per questo scopo, avrà probabilmente dei punti di forza e debolezza nell'interpretare le diverse *immagini* (ad esempio, un *metodo* potrebbe riconoscere particolarmente bene alcune lettere e uno diverso altre). L'esperienza pratica conferma che tali *metodi*, non essendo concepiti per questo scopo, riescono solo in parte ad interpretare i caratteri delle *immagini* ma una combinazione ben pensata delle loro *interpretazioni* può funzionare anche con probabilità elevate. Progetti come perl_smssend.pl usano esattamente questa tecnica: applicano un buon numero di *metodi* ad una stessa *immagine*, poi un algoritmo in qualche modo combina le diverse *interpretazioni* ottenendone una particolarmente buona.

Il problema, a questo punto, si sposta sulla stesura dell'algoritmo per l'integrazione dei risultati che è particolarmente complesso se si usano molti metodi e deve essere riscritto per ogni CAPTCHA. Gli autori di perl_smssend.pl, a questo punto, hanno proceduto in maniera empirica.

L'idea

Un'idea alternativa, presentata in questo articolo, è la seguente, ed è articolata in due fasi:

1. fase di apprendimento:
    1. ottenere un insieme adeguato di *immagini* di prova, di cui sono noti gli *oracoli*;
    2. applicare un insieme di *metodi* su ogni *immagine* di prova e memorizzare le *interpretazioni* risultanti;
    3. calcolare una stima della *probabilità di corretta interpretazione* per ogni *metodo* grazie al teorema di Bayes, in pratica considerando il numero di casi in cui tale *metodo* è riuscito a fornire *interpretazioni* corrette;
2. fase di utilizzo:
    1. data una nuova *immagine*, applicare tutti i *metodi* ed ottenere le relative *interpretazioni*;
    2. per ogni carattere, scegliere l'*interpretazione* più probabile (ossia quella con la massima *probabilità* calcolata al punto 1.3).

In questo modo, se l'insieme di immagini di prova iniziale è adeguato, le probabilità del punto 1.3 saranno buone stime delle probabilità reali e le interpretazioni del punto 2.2 dovrebbero avere alte probabilità di essere corrette.

La teoria

Questo paragrafo illustra senza pretese di perfetta correttezza e rigore matematico le basi teoriche del metodo per il calcolo della probabilità di corretta interpretazione introdotta nel paragrafo precedente. Si suppongono note le nozioni di base sulla Teoria delle Probabilità.

Si può provare abbastanza semplicemente, che dati due eventi A e B, la probabilità che avvenga A dato che si è già verificato B si può esprimere con la formula:

$$P(A|B)=\frac{P(A\cap B)}{P(B)}$$

ossia la probabilità che avvenga A dato B è pari al rapporto tra la probabilità che avvengano entrambi e la probabilità di B. Altrettanto semplicemente, si dimostra anche la seguente formula:

$$P(A|B)=P(B|A)\frac{P(A)}{P(B)}$$

nota come il teorema di Bayes. Tramite questa relazione, è possibile calcolare la probabilità condizionata di un evento a partire dalla probabilità condizionata “duale” e le probabilità dei due eventi singoli.

Nel nostro caso, nella fase di utilizzo, si ottiene un certo numero (poniamo n) di *interpretazioni* dall'applicazione dei *metodi* all'*immagine*. Il problema è che, per ognuno dei (poniamo m) caratteri

nell'*immagine* si vorrebbe scegliere quale *interpretazione* utilizzare.

L'idea è: si sceglie il carattere dall'*interpretazione* che più probabilmente è corretta.

Chiamando A l'evento "il risultato del *metodo* è corretto" e B_k "il *metodo* ha letto il carattere k", interessa quindi la probabilità $P(A|B_k)$ ossia la probabilità che l'*interpretazione* del carattere sia corretta dato che è stato letto il carattere k. Si tratta quindi di scegliere il *metodo* con $P(A|B_k)$ massima.

Resta da capire come si calcola $P(A|B_k)$, o meglio come si può stimare.

Qui viene in aiuto il teorema di Bayes, per cui:

$$P(A|B_k)=P(B_k|A)\frac{P(A)}{P(B_k)}$$

Le probabilità $P(B_k|A)$, $P(A)$ e $P(B_k)$ si possono stimare dalle interpretazioni della fase di apprendimento nel seguente modo:

$$P(B_k|A)=\frac{\text{\#caratteri k letti correttamente}}{\text{\# caratteri letti correttamente}}$$

$$P(A)=\frac{\text{\#caratteri letti correttamente}}{\text{\# caratteri letti}}$$

$$P(B_k)=\frac{\text{\#caratteri k letti}}{\text{\# caratteri letti}}$$

In definitiva:

$$P(A|B_k)=\frac{\text{\#caratteri k letti correttamente}}{\text{\# caratteri letti correttamente}}\frac{\text{\#caratteri letti correttamente}}{\text{\# caratteri letti}}\frac{\text{\#caratteri letti}}{\text{\# caratteri k letti}}$$

Ossia:

$$P(A|B_k)=\frac{\text{\#caratteri k letti correttamente}}{\text{\# caratteri k letti}}\quad .$$

Se il campione di *immagini* è rappresentativo e sufficientemente numeroso, le stime di cui sopra saranno vicine ai valori "veri" delle rispettive probabilità e sarà pertanto significativo utilizzare il numero $P(A|B_k)$ per scegliere l'*interpretazione* "migliore".

L'implementazione

Il programma MoioSMS contiene una implementazione delle idee presentate in questo articolo, e l'utilizzo pratico pare confermare la validità del procedimento. Sono presentati qui di seguito i risultati di uno studio comparato delle prestazioni di due versioni del programma perl_smssend.pl contro MoioSMS, su due set di immagini CAPTCHA del sito www.190.it con versioni diverse dei due programmi OCR utilizzati.

Configurazione A: gocr 0.39 ocrad 0.13

	perl_smssend	perl_smssend (migliorato)	MoioSMS
Set 1	45%	50%	64%
Set 2	51%	57%	74%

Configurazione B: gocr 0.40 ocrad 0.15

	perl_smssend	perl_smssend (migliorato)	MoioSMS
Set 1	48%	61%	73%
Set 2	51%	66%	74%

La configurazione A è un PC con Ubuntu 6.06, gocr 0.39 e ocrad 0.13; la configurazione B è un Apple MacBook Pro con Mac OSX 10.4.8, gocr 0.40 e ocrad 0.15. Le differenze nei risultati sono dovute alle diverse versioni degli ocr utilizzati. Il set di immagini 1 contiene 45 immagini, il set 2 138 immagini (diverse).

Critica alla metodologia ed ai risultati

Vantaggi

1. Probabilmente questo algoritmo è più facilmente implementabile rispetto ad uno empirico;
2. Teoricamente, a parte un nuovo apprendimento, il sistema dovrebbe continuare a funzionare anche con CAPTCHA diversi;
3. Aggiungere OCR o in generale *metodi* diversi è molto facile;
4. La teoria garantisce che al crescere in dimensioni dell'insieme di immagini campione si ottiene una interpretazione finale migliore;
5. La pratica dimostra che il teorema di Bayes funziona in tanti diversi ambiti (es. filtri antispam);

Svantaggi

1. E' comunque necessario spendere tempo per trovare il migliore insieme di *metodi* da dare in pasto all'algoritmo bayesiano;
2. E' necessario spendere tempo per ottenere un buon insieme di immagini per la fase di apprendimento;
3. Il processo di apprendimento può essere lungo, specialmente se il campione di immagini è grande;
4. Alcune ipotesi teoriche implicite nella descrizione di cui sopra possono non essere completamente verificate per tutti i CAPTCHA. E' rilevante tra le altre la supposizione che le probabilità di corretta interpretazione non dipenda dalla posizione del carattere nella stringa;
5. E' probabilmente comunque possibile, fissato un certo insieme di immagini campione, trovare un algoritmo migliore, anche se non si avrebbero garanzie di buon funzionamento al di fuori di quell'insieme.

Silvio Moioli, www.moioli.net

Ringraziamenti

Si ringraziano vivissimamente gli autori del progetto perl_smssend.pl che per primi hanno implementato un sistema funzionante per l'interpretazione di CAPTCHA mediante OCR ed hanno avuto la gentilezza di pubblicare e spiegare il loro approccio.

Si ringraziano inoltre tutti gli utenti della mailing list del progetto MoioSMS che hanno in qualche modo contribuito alla crescita del progetto ed alla stesura di questo documento.

Licenza

L'autore

Silvio Moioli, studente di Ingegneria Informatica all'Università degli Studi di Bergamo, si occupa di programmazione e sistemi OpenSource. Mantiene un piccolo sito personale all'indirizzo http://www.moioli.net, dove sono ospitati alcuni dei progetti aperti a cui lavora.