# Lecture 15

# Rounding SDPs for CSPs[*]

Recall the canonical SDP relaxation for an instance $\mathcal{C}$ of CSP($\Gamma$):

$$\max \sum_{C=(R,S)\in\mathcal{C}} w_C \mathop{\mathbf{Pr}}_{L\sim\lambda_C}[L(S) \text{ satisfies } R],$$

subject to the following conditions:

- for all $C$, $\lambda_C$ is a probability distribution on assignments $S \to D$;
- $(I_v[\ell])_{v\in V,\ell\in D}$ are joint random variables (which can be thought of as vectors), called "pseudoindicators," that satisfy:

  1. [optional] for all $C \in \mathcal{C}$, for all $v \in C$, and for all $\ell \in D$,
  $$\mathbf{E}\big[I_v[\ell]\big] = \mathop{\mathbf{Pr}}_{L\sim\lambda_C}[L(v) = \ell];$$
  2. for all $C \in \mathcal{C}$, for all $v \in C$, for all $\ell \in D$, for all $v' \in V$, and for all $\ell' \in D$,
  $$\mathbf{E}\big[I_v[\ell]\cdot I_{v'}[\ell']\big] = \mathop{\mathbf{Pr}}_{L\sim\lambda_C}[L(v) = \ell \text{ and } L(v') = \ell'].$$

Also recall the following equivalent perspectives on the canonical SDP relaxation for a CSP:

- pseudoindicator random variables satisfying conditions 1 and 2 above;
- pseudoindicator random variables satisfying condition 2 above;
- a vector solution satisfying conditions 1 and 2 above (when viewed as a collection of pseudoindicator random variables);
- jointly Gaussian pseudoindicators satisfying conditions 1 and 2 above.

[*]Lecturer: Ryan O'Donnell. Scribe: Brian Kell.

## 15.1 Jointly Gaussian pseudoindicators

For this section, let $N = |V| \cdot |D|$. [Is this what $N$ is supposed to mean, or is $N$ arbitrary?]

**Definition 15.1.** Random variables $Z_1$, $Z_2$, …, $Z_N$ are said to be *jointly Gaussian* if

$$\begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_N \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{bmatrix} + L \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_M \end{bmatrix},$$

where $L$ is an $N \times M$ matrix and $X_1$, $X_2$, …, $X_M$ are mutually independent standard normal random variables (that is, normal random variables with mean 0 and standard deviation 1). The *covariance matrix* $\Sigma$ is the matrix $(\Sigma_{ij})_{i=1}^{N}{}_{j=1}^{N}$, where $\Sigma_{ij} = \mathbf{Cov}[Z_i, Z_j]$.

**Fact 15.2.** *The covariance matrix is given by $\Sigma = LL^{\top}$.*

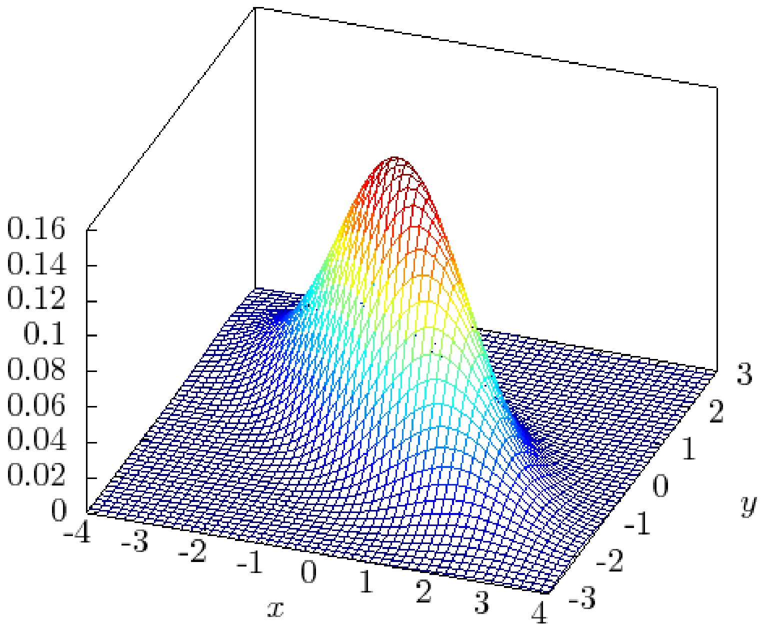

Figure 15.1: Density of joint Gaussian random variables $X$ and $Y$ with $\mathbf{Var}(X) = 2$, $\mathbf{Var}(Y) = 1$, and $\mathbf{Cov}(X, Y) = -1$. [From PlanetMath.org]

**Proposition 15.3.** *We can assume (computationally efficiently) that the pseudoindicators are jointly Gaussian. In other words, given a vector solution satisfying conditions 1 and 2, we can efficiently produce jointly Gaussian pseudoindicators satisfying conditions 1 and 2.*

*Proof.* We start with a vector solution $\vec{y}_{v,\ell} \in \mathbb{R}^N$ satisfying conditions 1 and 2. Let $Z_1$, …, $Z_N$ be mutually independent standard normal random variables.

As a first attempt, we define a collection of jointly distributed random variables $(G_v[\ell])_{v,\ell}$ as follows: To get one draw from (all of) the $G_v[\ell]$, pick $i \in [N]$ uniformly at random and output $G_v[\ell] = Z_i(\vec{y}_{v,\ell})_i$. Then the $G_v[\ell]$ are jointly Gaussian. [Why?] Moreover, they satisfy condition 2:

$$\mathop{\mathbf{E}}_{Z,i}[G_v[\ell]G_{v'}[\ell']] = \mathop{\mathbf{E}}_{Z}\left[\sum_i \frac{1}{N} Z_i(\vec{y}_{v,\ell})_i \cdot Z_i(\vec{y}_{v',\ell'})_i\right] = \frac{1}{N}\sum_i (\vec{y}_{v,\ell})_i(\vec{y}_{v',\ell'})_i \cdot \overset{1}{\cancel{\mathbf{E}[Z_i^2]}} =: \langle\langle \vec{y}_{v,\ell}, \vec{y}_{v',\ell'}\rangle\rangle.$$

However, when we check to see that they satisfy condition 1, we find that we have

$$\mathop{\mathbf{E}}_{Z,i}[G_v[\ell]] = \mathop{\mathbf{E}}_{Z}\left[\sum_i \frac{1}{N} Z_i(\vec{y}_{v,\ell})_i\right] = \frac{1}{N}\sum_i (\vec{y}_{v,\ell})_i \cdot \overset{0}{\cancel{\mathbf{E}[Z_i]}} = 0.$$

To fix this problem, we instead output

$$G_v[\ell] = \mathop{\mathrm{avg}}_j(\vec{y}_{v,\ell})_j + Z_i\big((\vec{y}_{v,\ell})_i - \mathop{\mathrm{avg}}_j(\vec{y}_{v,\ell})_j\big).$$

This definition of $G_v[\ell]$ satisfies both conditions 1 and 2. [Why? And why are these new $G_v[\ell]$ jointly Gaussian?] □

## 15.2 Goemans–Williamson for Max-Cut, redux

The Goemans–Williamson algorithm for Max-Cut can be viewed as follows:

1. Solve the SDP and get jointly Gaussian pseudoindicators $(G_v[0], G_v[1])_{v\in V}$.
2. Draw once from them to obtain numbers $(g_v[0], g_v[1])_{v\in V}$.
3. Output the assignment $F(v) = \mathrm{argmax}_{\ell\in\{0,1\}}\{g_v[\ell]\}$.

[Why can GW be viewed this way?]
The analysis of this algorithm uses the following fact, proved by Sheppard in 1899 [She99] [Is this the right reference?]: If $Z$ and $Z'$ are standard normal random variables and $\mathbf{E}[ZZ'] = \rho$, then

$$\mathbf{Pr}[\mathrm{sgn}(Z) = \mathrm{sgn}(Z')] = \frac{1}{2} - \frac{1}{\pi}\cos^{-1}\rho.$$

## 15.3 The Unique Games Conjecture

**Definition 15.4.** The problem *Unique-Games$_q$*, abbreviated *$UG_q$*, is a 2-CSP over the domain $D = \{0, 1, 2, \ldots, q-1\}$, for which

$$\Gamma = \left\{\, R : D^2 \to \{0,1\} \;\middle|\; \exists \text{ permutation } \pi \text{ of } D \text{ s.t. } R(a,b) = 1 \text{ iff } \pi(a) = b \,\right\};$$

in other words, the only allowable constraints in an instance of $UG_q$ are bijective constraints of arity 2.

**Example 15.5.** Max-Cut is a subproblem of $UG_2$. In an instance of Max-Cut, every edge $(i,j)$ in the graph has a corresponding constraint $x_i \neq x_j$, that is, the allowable assignments are $\big\{ \{x_i = 0, x_j = 1\}, \{x_i = 1, x_j = 0\} \big\}$. If the graph is bipartite, then all of these constraints are simultaneously satisfiable; otherwise, the objective of Max-Cut is to maximize the fraction of constraints that are satisfied.

**Remark 15.6.** For $UG_2$, the greedy algorithm (local propagation) is a $(1,1)$-approximation algorithm. This is like 2-coloring a bipartite graph (2-coloring is a subproblem of $UG_2$). In an instance of $UG_2$, the domain is $D = \{0,1\}$, and all constraints are of the form $x_i = x_j$ or of the form $x_i \neq x_j$. If it is possible to simultaneously satisfy all of these constraints (that is, if $\mathrm{Opt} = 1$), then we can easily find a satisfying assignment by arbitrarily choosing a starting variable and arbitrarily assigning it either 0 or 1; from then on, the value of any unassigned variable that shares a constraint with a variable that has already been assigned is determined by the constraint. The arbitrary choices we made at the beginning are not important, because the "complement" assignment of any satisfying assignment is also a satisfying assignment.

**Conjecture 15.7. Unique Games Conjecture** [Kho02]. For every $\epsilon > 0$ there exists a value of $q$ such that $(1/2,\, 1-\epsilon)$-approximating $UG_q$ is NP-hard. [Any $\alpha \in (0,1)$ can be substituted for 1/2 here.]

**Remark 15.8.** The Unique Games Conjecture implies many other optimal inapproximability results.

On the other hand, we can get a $(1/2,\, 1-\epsilon)$-approximation by solving an SDP of size $\exp(qn^{\epsilon^{\Theta(1)}})$ [ABS10], [BRS11].

We can $\big(1 - O(\sqrt{\log q}\sqrt{\epsilon}),\, 1-\epsilon\big)$-approximate $UG_q$ by SDP rounding [CMM06]. We can do the following:

1. Solve the canonical SDP relaxation to get a collection of jointly Gaussian pseudoindicators $(G_v[\ell])_{v\in V, \ell\in\{0,\dots,q-1\}}$.

2. Draw once from them to obtain numbers $(g_v[0], g_v[1], \dots, g_v[q-1])_{v\in V}$.

3. Output the assignment $F(v) = \mathrm{argmax}_\ell\{g_v[\ell]\}$.

Note that this is a randomized algorithm.

**"Theorem" 15.9.** If $\mathrm{SDPOpt}(\mathcal{C}) \geq 1-\epsilon$, then

$$\mathop{\mathbf{E}}_{F}[\mathrm{Val}_{\mathcal{C}}(F)] \geq 1 - O(\sqrt{\log q}\sqrt{\epsilon}).$$

The proof of this "theorem" is left as an exercise (and could probably be published as a paper). It should be noted that CMM did something slightly different.

## 15.4 $(\alpha, \beta)$-decision algorithms

Recall that an algorithm is said to $(\alpha, \beta)$-approximate CSP($\Gamma$) if whenever $\mathrm{Opt}(\mathcal{C}) \geq \beta$ the algorithm produces a solution with value at least $\alpha$. [For a proposed rounding algorithm, we often prove a theorem that looks something like: If $\mathrm{SDPOpt}(\mathcal{C}) \geq \beta$, then our rounding algorithm produces a solution with value at least $\alpha$. In particular, this kind of theorem shows that there exists a solution with value at least $\alpha$.]

**Definition 15.10.** An algorithm $(\alpha, \beta)$-*decides* CSP($\Gamma$) if:

- on instances $\mathcal{C}$ with $\mathrm{Opt}(\mathcal{C}) \geq \beta$, the algorithm outputs YES;
- on instances $\mathcal{C}$ with $\mathrm{Opt}(\mathcal{C}) < \alpha$, the algorithm outputs NO.

Note that $(\alpha, \beta)$-deciding CSP($\Gamma$) is strictly easier than $(\alpha, \beta)$-approximating it. If we have an algorithm to $(\alpha, \beta)$-approximate CSP($\Gamma$), then we can $(\alpha, \beta)$-decide it by doing the following:

> Given an instance $\mathcal{C}$, run the $(\alpha, \beta)$-approximation algorithm on $\mathcal{C}$ to get a solution $\mathcal{S}$. If $\mathrm{Val}(\mathcal{S}) \geq \alpha$, output YES; if $\mathrm{Val}(\mathcal{S}) < \alpha$, output NO.

This is an $(\alpha, \beta)$-decision algorithm: If $\mathrm{Opt}(\mathcal{C}) \geq \beta$, then $\mathrm{Val}(\mathcal{S}) \geq \alpha$, so we output YES. If $\mathrm{Opt}(\mathcal{C}) < \alpha$, then $\mathrm{Val}(\mathcal{S}) \leq \mathrm{Opt}(\mathcal{C}) < \alpha$, so we output NO. On the other hand, there is no way to make an $(\alpha, \beta)$-approximation algorithm from an $(\alpha, \beta)$-decision algorithm, because if $\mathrm{Opt}(\mathcal{C}) \geq \beta$ then the $(\alpha, \beta)$-decision algorithm will just say YES and will give no useful information.

A potentially good $(\alpha, \beta)$-decision algorithm for CSP($\Gamma$) is the following:

($\star$) Given an instance $\mathcal{C}$, solve the canonical SDP relaxation. If $\mathrm{SDPOpt} \geq \beta$, output YES; if $\mathrm{SDPOpt} < \beta$, output NO.

Certainly if $\mathrm{Opt}(\mathcal{C}) \geq \beta$, then $\mathrm{SDPOpt}(\mathcal{C}) \geq \mathrm{Opt}(\mathcal{C}) \geq \beta$, so this algorithm outputs YES. If we additionally know that whenever $\mathrm{Opt}(\mathcal{C}) < \alpha$ we have $\mathrm{SDPOpt}(\mathcal{C}) < \beta$, then this algorithm will correctly output NO whenever $\mathrm{Opt}(\mathcal{C}) < \alpha$; however, if there is an instance $\mathcal{C}$ with $\mathrm{Opt}(\mathcal{C}) < \alpha$ but $\mathrm{SDPOpt}(\mathcal{C}) \geq \beta$, then this algorithm will incorrectly output YES for that instance.

**Definition 15.11.** An instance $\mathcal{C}$ of CSP($\Gamma$) is an $(\alpha, \beta)$-*SDP-gap instance* if $\mathrm{SDPOpt}(\mathcal{C}) \geq \beta$ but $\mathrm{Opt}(\mathcal{C}) < \alpha$.

As observed above, the existence of such an instance is a barrier to the SDP algorithm ($\star$) being an $(\alpha, \beta)$-decision algorithm. In other words, there exists an $(\alpha, \beta)$-SDP-gap instance if and only if the algorithm ($\star$) is not an $(\alpha, \beta)$-decision algorithm.

**Definition 15.12.**

$$\mathrm{SDPGap}_\Gamma(\beta) = \inf\{\, \mathrm{Opt}(\mathcal{C}) \mid \mathcal{C} \text{ is an instance of } \mathrm{CSP}(\Gamma) \text{ with } \mathrm{SDPOpt}(\mathcal{C}) \geq \beta \,\}.$$

**Remark 15.13.** Tautologically, for every $\beta$, the SDP algorithm $(\star)$ is an $\big(\mathrm{SDPGap}_\Gamma(\beta),\, \beta\big)$-decision algorithm.

**Theorem 15.14.** *[Rag08], [RS09] [Are these the right references?] Assume the Unique Games Conjecture. Then for every $\epsilon > 0$, $\big(\mathrm{SDPGap}_\Gamma(\beta-\epsilon)+\epsilon,\, \beta-\epsilon\big)$-deciding $\mathrm{CSP}(\Gamma)$ is NP-hard.*

Essentially, doing better than the algorithm $(\star)$ is NP-hard. Also, in particular, we see that $\big(\mathrm{SDPGap}_\Gamma(\beta-\epsilon)+\epsilon,\, \beta-\epsilon\big)$-approximating is NP-hard. However, the following theorem says that this is the exact threshold, because $\big(\mathrm{SDPGap}_\Gamma(\beta-\epsilon)-\epsilon,\, \beta\big)$-approximating is in P.

**Theorem 15.15.** *[Rag08], [RS09] [Are these the right references?] For every constraint satisfaction problem $\mathrm{CSP}(\Gamma)$, for every $\epsilon > 0$, and for every $\beta$, there exists an SDP rounding algorithm which $\big(\mathrm{SDPGap}_\Gamma(\beta-\epsilon)-\epsilon,\, \beta\big)$-approximates the CSP and is $\mathrm{poly}(n)$-time. [However, it is $\exp(\exp(\mathrm{poly}(kq/\epsilon)))$-time, where $k$ is the maximum arity of the relations in the constraints and $q$ is the size of the domain.]*

*Proof.* [This needs to be cleaned up and more fully explained.] An outline of the algorithm:

1. Given $\mathcal{C}$ with $\mathrm{Opt}(\mathcal{C}) \geq \beta$, find an optimal SDP solution $\mathcal{S}$.

2. Drop an $\epsilon$-fraction of the constraints to get $\mathcal{C}'$, and get an SDP vector solution $\mathcal{S}'$ for $\mathcal{C}'$ with

$$\mathrm{SDPVal}_{\mathcal{C}'}(\mathcal{S}') \geq \mathrm{SDPVal}(\mathcal{S}) - O(\epsilon) = \mathrm{SDPOpt}(\mathcal{C}) - O(\epsilon) \geq \mathrm{Opt}(\mathcal{C}) - O(\epsilon) \geq \beta - O(\epsilon),$$

   with vectors in $\mathbb{R}^d$, where $d = \mathrm{poly}(kq/\epsilon)$.

3. Losing another $O(\epsilon)$ on $\mathrm{SDPVal}(\mathcal{S}')$, discretize the coordinates of these vectors to $\mathrm{poly}(\epsilon/kq)$. (From now on, $\mathcal{S}'$ will refer to these discretized vectors.) Now for every $v \in V$, there are only $\exp(\mathrm{poly}(kq/\epsilon))$ many possible vectors $\vec{y}_{v,\ell} \in \mathcal{S}'$.

4. For any $v, v' \in \mathcal{C}'$ with identical vectors $(\vec{y}_{v,\ell})_{\ell\in D}$, identify them, forming $\mathcal{C}''$, on $\exp(\mathrm{poly}(kq/\epsilon))$ many variables. [There is a potential problem here: Is $\mathcal{C}''$ really an instance of $\mathrm{CSP}(\Gamma)$? There is a fix for this problem.]

5. Solve $\mathcal{C}''$ in time $\exp(\exp(\mathrm{poly}(kq/\epsilon)))$ to get $F'' : V'' \to D$ with $\mathrm{Val}_{\mathcal{C}''}(F'') = \mathrm{Opt}(\mathcal{C}'')$. Unfold this to $F' : V \to D$ for $\mathcal{C}'$. Then $\mathrm{Val}_{\mathcal{C}'}(F') = \mathrm{Opt}(\mathcal{C}'')$, so $\mathrm{Val}_{\mathcal{C}}(F') \geq \mathrm{Opt}(\mathcal{C}'') - O(\epsilon)$. Hence

$$\mathrm{SDPOpt}(\mathcal{C}'') \geq \mathrm{SDPVal}_{\mathcal{C}''}(\mathcal{S}') = \mathrm{SDPVal}_{\mathcal{C}'}(\mathcal{S}') \geq \mathrm{SDPVal}_{\mathcal{C}}(\mathcal{S}) - O(\epsilon) \geq \cdots \geq \beta - \epsilon.$$

   Therefore, by definition, $\mathrm{Opt}(\mathcal{C}'') \geq \mathrm{SDPGap}_\Gamma(\beta - \epsilon)$. □

# Bibliography

[ABS10] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for Unique Games and related problems. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 563–572, 2010. 15.3

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, 2011. 15.3

[CMM06] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for Unique Games. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 205–214, 2006. 15.3

[Kho02] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775. ACM Press, 2002. 15.7

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 245–254, 2008. 15.14, 15.15

[RS09] P. Raghavendra and D. Steurer. Integrality gaps for strong SDP relaxations of Unique Games. In *Foundations of Computer Science, 2009. FOCS '09. 50th Annual IEEE Symposium on*, pages 575–585, oct. 2009. 15.14, 15.15

[She99] W. F. Sheppard. On the application of the theory of error to cases of normal distribution and normal correlation. *Royal Society of London Philosophical Transactions Series A*, 192:101–167, 1899. 15.2